ESSLLI 2012 Student Session Proceedings

Edited by

Rasmus K. Rendsvig
University of Copenhagen

Sophia Katrenko
Utrecht University

Preface

The 24th European Summer School in Logic, Language, and Information (ESSLLI) took place on August 6-17, Opole, Poland, under the auspices of FoLLI. This volume contains the proceedings of the 17th Student Session, an annual event which takes place within ESSLLI. It contains 21 papers addressing a wide range of topics in logic, language, and computation. These papers were selected from a large number of excellent submissions on the basis of reports by expert reviewers in the relevant areas.

Great thanks are due to the area experts and to the two previous chairs, Marija Slavkovik (2010) and Daniel Lassiter (2011), for their advice and support, to Kristian K. Olesen for help with matters LaTeX, and to Janusz Czelakowski, Urszula Wybraniec-Skardowska and the rest of the ESSLLI 2012 Organizing Committee. Special recognition goes to Springer for their continuing support of the Student Session. Most importantly, thanks to all those who submitted papers, and to the many reviewers who generously gave their time to provide thoughtful feedback on each submission.

Rasmus K. Rendsvig
Chair, ESSLLI 2012 Student Session
Copenhagen, August 2012

Organization

Chair:	Rasmus K. Rendsvig
Co-Chairs:	Niels Beuck (Language and Computation) Margot Colinet (Logic and Language) Maxim Haddad (Logic and Computation) Anders Johannsen (Language and Computation) Dominik Klein (Logic and Computation) Matthijs Westera (Logic and Language)
Area Experts:	Ann Copestake (Language and Computation) Friederike Moltmann (Logic and Language) Eric Pacuit (Logic and Computation)

Table of Contents

Aligning Geospatial Ontologies Logically

Heshan Du

The University of Nottingham, UK

Abstract. Information sharing and updates have become increasingly important in the rapidly changing world. However, owing to the distributed and decentralized nature of information collection and storage, it is not easy to use information from different sources synergistically. Ontology plays as an important role in establishing formal descriptions of a domain of discourse. In geographic information science, the rapid developments of crowd-sourced geospatial databases challenge and also bring opportunities to the current geospatial information development framework. In this paper, a new semi-automatic method is proposed to align disparate geospatial ontologies, based on description logic and domain experts' knowledge.

1 Introduction

Information sharing and updates have become increasingly important in the rapidly changing world, with a large amount of disparate information available. However, owing to the distributed and decentralized nature of information development, it is not easy to fully capture the information content in different sources. A same expression can have different meanings in different context, and different expressions may refer to the same meaning. Such issues are quite common when disparate and related information sources exchange their data.

Ontology plays an important role in information sharing. Ontology, originated in the work of Aristotle, is a branch of philosophy which studies the existence of entities [41]. In computer science, ontology is an explicit formal specification of a shared conceptualization [14]. Compared to its origin, computer science ontology is not only about existence, but also about meaning, and making meanings as clear as possible [41]. Ontologies are often employed as important means for establishing explicit formal vocabulary shared among applications.

Description logics are a family of formalisms for representing the knowledge of an application domain [2]. They firstly define the relevant concepts (terminology), and then use these concepts to specify properties of objects in the domain [2]. Compared to traditional first-order logic, description logics provide tools for the manipulations of compound and complex concepts [7]. Description logics support inference patterns, including classification of concepts and individuals, which are often used by humans to structure and understand the world [2]. When dealing with ontology issues, description logics can be used as the logical underpinning.

The rapid developments in geographic information science emphasize the importance of geospatial information sharing. Spatial Data Infrastructures (SDI) refers to an institutional and technical foundation of policies, standards and procedures that enable organizations at multiple levels and scales to discover, evaluate, use and maintain

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see http://ceur-ws.org/, 2012, pp. 1–12.

geospatial data [28]. Over the last few years, the current top-down approach to SDI has been challenged by the rapid pace of technological development [17]. There is a need to address the separation of national and international SDI from crowd-sourced geospatial databases [1]. Relying on volunteers for data collection, crowd-sourced data is less expensive than authenticated data. In addition, although typically not as complete in its coverage or as consistent in its geometric or metadata quality as authenticated data, crowd-sourced data may provide a rich source of complementary information with the benefit of often more recent and frequent updates than that of authenticated data [18]. It is desirable to use authenticated and crowd-sourced geospatial data synergistically.

Compared to other ontologies, geospatial ontologies have several special properties. Firstly, many words within geospatial ontologies are often more widely used in daily life, and there is less consensus about their definitions. For example, the word 'creek' can refer to a river in Australia, while cannot in the US. The word 'field' has different meanings (e.g. a branch of knowledge, a piece of land, etc.) for different people in different contexts. There are no precise formal definitions which can tell 'river' and 'stream', 'lake' and 'pond' apart. In addition, geospatial ontologies often do not have a huge number of classes as biomedical or bioinformatics ontologies do, but may have many instances, referring to real world objects, whose locations, at least in theory, are verifiable. With respect to these properties and the underspecification of geospatial ontologies, no fully automated system can ensure the correctness and completeness of generated mappings. Therefore, experts are inevitably needed to make decisions, for example on the correctness of correspondences, based on their domain knowledge, which is often implicit in the individual ontologies. This research will finally lead to the minimisation of the human intervention.

This project aims to explore logic-based approaches to aligning disparate and related geospatial ontologies to obtain harmonized and maximized information content. Ontologies are disparate if they are created independently. Ontologies are related if they contain more than one common concept. Aligning means establishing relations between the vocabularies of two ontologies [9]. The desired output will be a collection of verified such relations, for query answering over multiple ontologies. If ontologies cannot be aligned logically, a deficiency report will be produced explaining reasons and providing suggestions about further actions to take in order to align them. In this paper, we propose a new semi-automatic method to align disparate geospatial ontologies, based on description logic and domain experts' knowledge. It is assumed that original information within ontologies is believed all the time as premises, whilst generated information, including disjointness axioms and mappings, is believed by default as assumptions, which may be retracted later. Differing from other existing methods, generated disjointness axioms are seen as assumptions, which are retractable during the overall aligning process. Disparate ontologies are aligned by finding a coherent and consistent assumption set with respect to them. Based on this main idea, algorithms are designed to align ontologies at the terminology level and the instance level. With respect to the special properties of geospatial ontologies, an algorithm is designed for refining correspondences between geospatial individuals taking their geometries and semantics into account.

The rest of the paper is organized as follows. The related work is summarized in Section 2. Section 3 introduces the geospatial ontologies we use, and explains some results generated by a state-of-the-art system called S-Match [12]. Our method is discussed in Section 4. Finally, it provides conclusions in Section 5.

2 Related Work

Ontology matching is the task of finding correspondences between entities from different ontologies [11]. A correspondence represents a semantic relation, such as inclusion or equivalence. A mapping is defined as a set of these correspondences [25]. Many ontology matching methods and systems have been proposed and developed in recent years [11] [35], based on shared upper ontologies, if available, or using other kinds of information, such as lexical and structural information, user input, external resources and prior matches [30]. The existing methods can be classified into three broad categories [4]. *Syntactic methods* rely on a syntactic analysis of the linguistic expressions to generate mappings. Though these methods are direct and effective, semantic relations between entities cannot be captured. *Pragmatic methods* infer the semantic relations between concepts from associated instance data. Though they work well when the instance data is representative and overlapping, this kind of methods use a strong form of induction, thus lack correctness and completeness. *Conceptual methods* compare the lexical representation of concepts to compute mappings. Socially negotiated meanings (e.g. dictionaries) are often used when generating relations, making the problem very complicated.

Many methods are hybrid, combining different approaches and making use of structural, lexical, domain or instance-based knowledge. Most of them apply heuristics-based or machine learning techniques to various characteristics of ontology. However, mappings generated by these methods often contain logical contradictions. Some systems, such as CtxMatch [5] and its extension S-Match [12], and more recently, Automated Semantic Mapping of Ontologies with Validation (ASMOV) [20], Knowledge Organization System Implicit Mapping (KOSIMap) [33], logiC-based ONtology inTEgratioN Tool using MAPpings (ContentMap) [22], LogMap [21], and Combinational Optimization for Data Integration (CODI) [29], seem to be exceptions, since they employ logical reasoning for either mapping generation or verification. Due to limited space, not all of them are discussed in detail.

CtxMatch and its extension S-Match are early logic-based attempts for ontology matching. While most of the other methods compute linguistic or structural similarities, CtxMatch shifts the semantic alignment problem to deducing relations between logical formulae [5, 6]. Relevant lexical (concepts denoted by words), world (relations between concepts) and structural knowledge is represented as logical formulae, and logical reasoning is employed to infer different kinds of semantic relations [34]. WordNet [27], an external resource, is employed to provide both lexical and world knowledge. The S-Match system re-implements and extends CtxMatch. Taking two tree-like structures (e.g. hierarchies) as input, it computes the strongest semantic relations between every pair of concepts [12]. The semantic matching has two main steps. Firstly, at element level, relations between labels are calculated, and then, at structure level, it generates relations between concepts, whose semantics are constructed based on the semantics of labels [36]. The structure level matching task is converted into propositional validity problems, and the standard DPLL-based SAT solver [3] is employed to check the unsatisfiability of propositional formulae [13]. However, S-Match only uses information in the tree-like structures extracted from ontologies, which is insufficient to guarantee the overall coherence of ontologies after applying the mapping relations. More recently, some matching tools have been developed, involving semantic verification into the alignment process.

LogMap [21] is a logic-based and scalable ontology matching tool. It addresses the challenges when dealing with large-scale bio-medical ontologies with tens (even hundreds) of thousands of classes. It employs lexical and structural methods to compute an initial set of mapping relations as the starting point for further discovery of mapping relations. The core of LogMap is an iterative process which alternates repair and discovery steps. In the repair step, unsatisfiable classes will be detected using propositional Horn representation and satisfiability checking, and be repaired using a greedy diagnosis algorithm. However, the propositional Horn satisfiability checking is sound but incomplete, and the underlying semantics is restricted to propositional logic, and thus cannot guarantee the coherence of the mapping between more expressive ontologies. In the discovery step, new mapping relations will be generated based on the similarity between classes which are semantically related to matched classes. ISUB [37] is employed to compute the similarity scores. Mapping relations which are newly discovered are active, and only active mapping relations can be eliminated in the repair step, whilst mapping relations found in earlier iterations are seen as established or valid. In other words, each mapping relation will be checked once, against the available information at that time, which, however, cannot guarantee its correctness when new information is discovered later.

Combinational Optimization for Data Integration (CODI) [29] is a probabilistic logical alignment system. It is based on Markov logic [10], which combines first-order logic with undirected probabilistic graphical models. As the main advantages over other existing matching approaches, Markov logic can combine hard logical axioms and soft uncertain formulae for potential correspondences. Cardinality constraints, coherence constraints and stability constraints are formalized using logical axioms and similarity measures. The matching problem is transformed to a maximum-a-posteriori optimization problem subject to these constraints. The GUROBI optimizer [15] is employed to solve the optimization problems. According to Noessner and Niepert [29], CODI reduces incoherence during the alignment process for the first time, compared to all other existing methods repairing alignments afterwards. CODI is based on the rationale of finding the most likely mapping by maximizing the sum of similarity-weighted probabilities for potential correspondences. It can be argued that during the optimization process, some valid correspondences can be thrown away. In addition, the input coherence constraints will influence the resulting mapping, however, in practice, many ontologies are underspecified, within which valid disjointness axioms are not always available.

In addition, there is some recent work on debugging and repairing ontologies and mappings in ontology networks [24] [32] [40] [23], which is still at an early stage. However, all of them use disjointness axioms as premises, rather than assumptions, and none of the matching systems discussed above have addressed the special properties of geospatial information fully. Several ontology-driven methods have been developed for integrating geospatial terminologies. Most of them are based on similarity measures or a predefined top-level ontology, and logical reasoning is only employed when formal ontologies commit to the same top-level ontology [8]. However, when ontologies are developed independently, the common top-level ontology is not always available. Additionally, there exist some other methods, such as [38] and [19], following the pragmatic approach to link geospatial schemas or ontologies, inferring the terminology correspondences from the instances correspondences. As discussed above, relying on a very strong form of induction from particular to the universal, this approach will lead to the lack of correctness and completeness [4]. Therefore, more research is required to fill in the gap, exploring logic-based approaches to aligning disparate geospatial ontologies.

3 Disparate Geospatial Ontologies

The Ordnance Survey of Great Britain (OSGB) ontology [16] and the OpenStreetMap (OSM) controlled vocabularies [31] are selected to undertake initial research. OSGB and OSM are representatives of authenticated and crowd-sourced geospatial information sources respectively. OSGB is the national topographic mapping agency of Great Britain. It has built ontologies for Hydrology and for Buildings and Places [16]. OSM is a collaborative project aimed to create a free editable map of the world [31]. It employs the bottom-up approach, relying on volunteers to collect the data. Currently, OSM does not have a standard ontology, but maintains a collection of commonly used tags for main map features [31]. An OSM feature ontology is generated automatically from the existing classification of main features. Both ontologies are written in the OWL 2 Web Ontology Language [39]. The OSGB Buildings and Places ontology has 692 classes and 1230 logical axioms, and its DL expressivity is $ALCHOIQ$. There are 663 classes and 677 logical axioms in the OSM ontology, whose DL expressivity is AL. Both ontologies, containing *no* disjointness axioms, are coherent.

To understand the ontologies more deeply, S-Match is employed to generate relations between concepts from them. To distinguish concepts from different ontologies, let us label each concept with the abbreviated name of the ontology it belongs to, such as $OSGB : School$ and $OSM : School$. All the labelled concepts will be treated as belonging to one super ontology. A relation then can be represented as an axiom within which all the concepts are labelled, like $OSGB : School \sqsupseteq OSM : School$.

Some of the mapping axioms generated by S-Match seem reasonable, such as $OSGB : Roof \equiv OSM : Roof$, $OSGB : Service \sqsupseteq OSM : Service$, and $OSGB : Accommodation \sqsupseteq OSM : Accommodation$. However, there are also some problematic relations. For example, $OSGB : Thing \equiv OSM : Nothing$ is derived, because the string ‘Thing’ is considered to be close enough by the string-based matcher [1] for stating that it is equivalent to the string ‘Nothing’. The relation $OSGB : Person \sqsupseteq OSM : GuestHouse$ is generated because, ‘GuestHouse’ is split to ‘Guest’ and ‘House’, and ‘House’ is treated as a person’s name, referring to a particular individual of ‘Person’. The relation $OSGB : Person \sqsupseteq OSM : Dentist$ seems reasonable, just looking at it alone. However, $OSM : Dentist \sqsubseteq OSM : Healthcare$, and $OSM : Healthcare \sqsubseteq OSM : Amenity$. The $OSM : Dentist$ is used for tagging a place where a dentist practice or surgery is located, rather than referring to a person who is a dentist. In this case, S-Match seems too restrictive to deal with the informal use of terms and their variable meanings in crowd-sourced databases, such as OSM.

4 Method

Ontology alignment has attracted the attention of people working in several research fields, such as linguistics, philosophy, psychology and computer science. To fully understand and solve the problem, relying on only one approach is inadequate. Different approaches may play different important roles and solve different aspects of the problem effectively. This project focuses on the logic-based approach, and explores what logic can do and how far logic can go when aligning disparate geospatial ontologies.

[1] When matching labels of concepts, S-Match employs string-based, sense-based and gloss-based matchers [13].

To represent and reason with two ontologies O^i and O^j, where i, j are their names, as well as the matching relations between them, as if they all belong to one super ontology $O^i \cup O^j$, we label all atomic concepts and roles in each ontology by the name of the ontology. The ontology O^i is the set $\{\varphi^i : \varphi \in O^i\}$, where φ denotes a logical formula. A logical formula φ^i is labelled inductively as follows.

- $A^i = i : A$, for atomic concept A;
- $R^i = i : R$, for atomic role R;
- $(\neg B)^i = \neg B^i$;
- $(B \sqcap C)^i = B^i \sqcap C^i$;
- $(B \sqcup C)^i = B^i \sqcup C^i$;
- $\{o\}^i = \{o^i\}$, for nominal $\{o\}$, individual name o;
- $(\forall R.B)^i = \forall R^i.B^i$;
- $(\exists R.B)^i = \exists R^i.B^i$;
- $(\geq n\ R.C)^i = \geq n\ R^i.C^i$
- $(\leq n\ R.C)^i = \leq n\ R^i.C^i$
- $(= n\ R.C)^i == n\ R^i.C^i$
- $(R^-)^i = (R^i)^-$;
- $(R^+)^i = (R^i)^+$;
- $(B \sqsubseteq C)^i = B^i \sqsubseteq C^i$
- $(S \sqsubseteq T)^i = S^i \sqsubseteq T^i$

where B, C denote concept descriptions, S, T denote roles. Similarly, we label all individual names in each ontology by the ontology name:

- $a^i = i : a$;
- $(C(a))^i = C^i(a^i)$
- $(R(a, b))^i = R^i(a^i, b^i)$

where a, b denote individual names.

A terminology mapping is a set of correspondences between classes from different ontologies. A terminology correspondence is represented as one of the two basic forms:

$$B^i \sqsubseteq C^j \quad (1)$$

$$B^i \sqsupseteq C^j \quad (2)$$

where B, C denote class descriptions. The relation (1) states that the class B from the ontology i is more specific than or equivalent to the class C from the ontology j. The relation (2) states that the class B from the ontology i is more general than or equivalent to the class C from the ontology j. The equivalence relation (3) holds if and only if (1) and (2) both hold.

$$B^i \equiv C^j \quad (3)$$

It states that the concept B from the ontology i and the concept C from the ontology j are equivalent.

A disjointness axiom states that two or more classes are pairwise disjoint, having no common element. For example, *Person* and *Place* are disjoint. The disjointness axioms in ontologies play an important role in debugging ontology mappings. However, within the original geospatial ontologies, disjointness axioms are not always available or sufficient. Adding disjointness axioms manually, especially for large ontologies, is time-consuming and error-prone. Many existing systems employ more automatic approaches, either assuming the disjointness of siblings (e.g. KOSIMap [33]), or employing machine learning techniques to detect disjointness (e.g. [26]). After the disjointness axioms are generated by whatever means, all existing ontology matching or debugging methods, to the best of our knowledge, use them as premises, believing their validity through the overall following process, though the input disjointness axioms can be insufficient or too restrictive. Differing from these methods, we use generated disjointness axioms as assumptions, rather than premises, and ensure the assumption set is coherent. A disjointness assumption is represented as:

$$B \sqsubseteq \neg C \tag{4}$$

where B, C denote class descriptions either from ontology i or ontology j. We follow the terminology from [26], and adapt them to this context.

Definition 1 (Coherence). *An ontology O is coherent if there is no class C such that $O \models C \sqsubseteq \bot$. Otherwise, it is incoherent.*

Definition 2 (Coherence of an Assumption Set). *An assumption set A_s is incoherent with respect to an ontology O, if $O \cup A_s$ is incoherent, but O is coherent. Otherwise, it is coherent with respect to an ontology O.*

Definition 3 (Minimal Incoherent Assumption Set). *Given a set of assumptions A_s, a set $C \subseteq A_s$ is a minimal incoherent assumption set (MIA) iff C is incoherent and each $C^{'} \subset C$ is coherent.*

A minimal incoherent assumption set can be fixed by removing any axiom from it. When a MIA contains more than one element, one needs to decide which axiom to remove. Most of the existing methods remove the one either with the lowest confidence value or which is the least relevant. However, it can be argued that there is no consensus with respect to the measure of the degree of confidence or relevance. In addition, the confidence values or the relevance degrees might be unavailable, difficult to compute or compare. In such cases, it seems sensible to allow domain experts to make such decisions.

When aligning ontologies using a terminology mapping, Definition 2 is extended from one ontology O to two ontologies O_1 and O_2, given that the union of two ontologies $O_1 \cup O_2$ is an ontology. Based on these definitions, *Algorithm 1* is designed as follows [2]. An assumption in a minimal incoherent assumption set can be a disjointness axiom or a terminology correspondence axiom. The set of minimal incoherent assumption sets will be visualized clearly (Line 7). Domain experts are consulted to decide which assumption(s) to retract (Line 8). A repair action can be retracting or adding an assumption axiom. Users are allowed to take several repair actions at one time.

ALGORITHM 1: Terminology Level Alignment
Input: O_1, O_2: *coherent* ontologies
D_s: a disjointness assumption set
M_{st}: a terminology mapping between O_1 and O_2
Output: A_s: a coherent assumption set with respect to $O_1 \cup O_2$
1. $O := O_1 \cup O_2$
2. $A_s := D_s \cup M_{st}$
3. **assert** O is *coherent*

[2] In an algorithm, lines marked with $*$ may require manual intervention.

$O := O \cup A_s$
while O is *incoherent* **do**
    $S_{mia} := MIA(O)$
    $visualization(S_{mia})$
    $repair(O, S_{mia})$
    $update(A_s)$
end while
return A_s

Following the algorithm above, even if the problematic relations generated by S-Match are introduced, they can be retracted, given the disjointness assumptions such as $OSGB : Person \sqsubseteq \neg OSM : Accommodation$ and $OSGB : Person \sqsubseteq \neg OSM : Amenity$.

An instance level mapping is a set of individual correspondences. An individual correspondence is represented in one of the following forms:

$$(a^i, b^j) \in sameAs \tag{5}$$

$$(a^i, b^j) \in partOf \tag{6}$$

where a, b denote individual names. The relation (4) states that the individual name a from the ontology i and the individual name b from the ontology j refer to the same object. The relation (5) states that the individual name a from the ontology i refers to an object which is a part of the object the individual name b from the ontology j refers to.

When working with geospatial instances, *Algorithm 2* is designed to refine the initial instance level mapping using geometry, lexical and cardinality properties.

ALGORITHM 2: Refining GeoInstance Mapping
Input: M_{sa}: an initial instance level mapping for geospatial individuals
Output: M_{sa}: the refined input M_{sa}
1. **for each** individual correspondence m in M_{sa} **do**
2. $a_1 := m.individual_1,\ a_2 := m.individual_2$
3.   **if** $a_1.geometry, a_2.geometry$ are not *matched* **then**
4. $remove(M_{sa}, m)$
5.   **else if** $a_1.lexicons, a_2.lexicons$ are not *matched* **then**
6. $remove(M_{sa}, m)$
7.   **end if**
8. **end for**
9. **for each** individual b appearing more than once in M_{sa} **do**
10. $M_{s_b} := allCorrespondencesInvolving(b)$
11. $repair(M_{sa}, M_{s_b})$
12. **end for**
13. **return** M_{sa}

Given a set of correspondences linking geospatial individuals from different ontologies, *Algorithm 2* applies three main constraints, these are, geometry, lexical and cardinality. Firstly, a correspondence is invalid if the geometries of the linked individuals are not matched (Line 3-4). For example, when the geometries are both polygons, if they are spatially disjoint, they cannot be matched. Secondly, a correspondence is invalid if the lexicons, i.e. meaningful labels indicating identity, cannot be matched (Line 5-6).

The lexical matching required should be robust enough to tolerate partial differences in labelling. For example, a full name and its abbreviation should be matched. Thirdly, if an individual is involved in several different 'sameAs' correspondences, then these correspondences need to be repaired (Line 9-12), for example, by changing the relation from 'sameAs' to 'partOf'. The three constraints complement each other to cope with the following possibilities. Different geospatial individuals may share the same label or the same location in an ontology. In addition, the same geospatial individual may be represented as a whole in one ontology, whilst as several parts of it in the other.

The algorithm for aligning instances is generated by extending the assumption set to include instance correspondences (output of *Algorithm 2*) and changing coherence checking to consistency checking in *Algorithm 1*. Similarly, domain experts are consulted to make decisions to repair inconsistencies. Consider the example below.

Example 1. $OSGB : 1000002308476718$ refers to a $OSGB : HealthCentre$ labelled as 'SNEINTON HEALTH CENTRE'. $OSM : 62134030$ refers to a $OSM : Clinic$ labelled also as 'SNEINTON HEALTH CENTRE'. Their geometries are very similar. However, the existence of the following assumptions can lead to inconsistency.

$$(OSGB : 1000002308476718, OSM : 62134030) \in sameAs \tag{7}$$

$$OSGB : Clinic \equiv OSM : Clinic \tag{8}$$

$$OSGB : Clinic \sqsubseteq \neg OSGB : HealthCentre \tag{9}$$

Domain experts are consulted to decide which assumption(s) to retract. To keep the individual correspondence (7), it is reasonable to retract (9). This differs from all other methods, which use (9) as a premise, and therefore will either remove (7) or (8), though both are reasonable.

This method has been implemented as a system. Its performance is being evaluated, compared to other existing systems, such as S-Match, CODI and LogMap.

5 Conclusion

To facilitate the geospatial information sharing and updates, it is important to harmonize disparate and related geospatial ontologies. This paper discusses problems involved, and presents a new logic-based method to deal with them. Future work includes employing a truth maintenance system to track logical dependencies and qualitative spatial reasoning to check topological consistency of geospatial data.

References

1. Anand, S., Morley, J., Jiang, W., Du, H., Jackson, M.J., Hart, G.: When worlds collide: Combining Ordnance Survey and OpenStreetMap data. In: Association for Geographic Information (AGI) GeoCommunity '10 Conference (2010)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook. Cambridge University Press (2007)
3. Berre, D.L., Parrain, A.: The Sat4j Library, release 2.2. Journal on Satisfiability, Boolean Modeling and Computation 7(2-3), 59–64 (2010)
4. Bouquet, P.: Contexts and ontologies in schema matching. In: Context and Ontology Representation and Reasoning (2007), `http://ceur-ws.org/Vol-298/paper2.pdf`

5. Bouquet, P., Serafini, L., Zanobini, S.: Semantic Coordination: A New Approach and an Application. In: International Semantic Web Conference. pp. 130–145 (2003), `http://disi.unitn.it/~bouquet/papers/ISWC2003-CtxMatch.pdf`
6. Bouquet, P., Serafini, L., Zanobini, S.: Peer-to-Peer Semantic Coordination. Web Semantics: Science, Services and Agents on the World Wide Web 2(1) (2004), `http://www.websemanticsjournal.org/index.php/ps/article/view/54`
7. Brachman, R.J., Levesque, H.J.: Knowledge Representation and Reasoning. The Morgan Kaufmann Series in Artificial Intelligence, Morgan Kaufmann (2004), `http://books.google.co.uk/books?id=OuPtLaA5QjoC`
8. Buccella, A., Cechich, A., Fillottrani, P.: Ontology-Driven Geographic Information Integration: A Survey of Current Approaches. Computers and Geosciences 35(4), 710 – 723 (2009), `http://www.sciencedirect.com/science/article/pii/S0098300408002021`
9. Choi, N., Song, I.Y., Han, H.: A survey on ontology mapping. SIGMOD Record 35, 34–41 (September 2006), `http://doi.acm.org/10.1145/1168092.1168097`
10. Domingos, P., Lowd, D., Kok, S., Poon, H., Richardson, M., Singla, P.: Just Add Weights: Markov Logic for the Semantic Web. In: Uncertainty Reasoning for the Semantic Web I, ISWC International Workshops, URSW 2005-2007, Revised Selected and Invited Papers. pp. 1–25 (2008)
11. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer-Verlag, Heidelberg (DE) (2007), `http://www.springerlink.com/content/978-3-540-49611-3#section=273178&page=1`
12. Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-Match: an Algorithm and an Implementation of Semantic Matching. In: European Semantic Web Conference (ESWS). pp. 61–75 (2004)
13. Giunchiglia, F., Yatskevich, M., Shvaiko, P.: Semantic Matching: Algorithms and Implementation. Journal on Data Semantics IX 9, 1–38 (2007)
14. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition 5, 199–220 (1993), `http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.101.7493`
15. Gurobi Optimization Inc.: Gurobi Optimizer Reference Manual. http://www.gurobi.com (2012), `http://www.gurobi.com`
16. Hart, G., Dolbear, C., Kovacs, K., Guy, A.: Ordnance Survey Ontologies. http://www.ordnancesurvey.co.uk/oswebsite/ontology (2008)
17. Jackson, M.J.: The Impact of Open Data, Open Source Software and Open Standards on the Evolution of National SDI's. In: Third Open Source GIS Conference (21-22 June 2011), `http://uiwapmds01.nottingham.ac.uk/qcsplace/ondemand/events11/a5a1e446858f4867b77716111a/player.HTM`
18. Jackson, M.J., Rahemtulla, H., Morley, J.: The Synergistic Use of Authenticated and Crowd-Sourced Data for Emergency Response. In: 2nd International Workshop on Validation of Geo-Information Products for Crisis Management (VALgEO). Ispra, Italy (11-13 October 2010), `http://globesec.jrc.ec.europa.eu/workshops/valgeo-2010/proceedings`
19. Jain, P., Hitzler, P., Sheth, A.P., Verma, K., Yeh, P.Z.: Ontology Alignment for Linked Open Data. In: International Semantic Web Conference (1). pp. 402–417 (2010)
20. Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R.: ASMOV: results for OAEI 2010. In: the 5th International Workshop on Ontology Matching (OM-2010) (2010)
21. Jiménez-Ruiz, E., Grau, B.C.: LogMap: Logic-Based and Scalable Ontology Matching. In: International Semantic Web Conference (1). pp. 273–288 (2011)

22. Jiménez-Ruiz, E., Grau, B.C., Horrocks, I., Llavori, R.B.: Ontology Integration Using Mappings: Towards Getting the Right Logical Consequences. In: The 6th European Semantic Web Conference (ESWC). pp. 173–187 (2009)
23. Lambrix, P., Liu, Q.: Debugging is-a Structure in Networked Taxonomies. In: the 4th International Workshop on Semantic Web Applications and Tools for the Life Sciences. pp. 58–65. SWAT4LS '11, ACM, New York, NY, USA (2011), http://doi.acm.org/10.1145/2166896.2166914,http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/l/Lambrix:Patrick.html
24. Meilicke, C., Stuckenschmidt, H.: An Efficient Method for Computing Alignment Diagnoses. In: Third International Conference on Web Reasoning and Rule Systems. pp. 182–196 (2009)
25. Meilicke, C., Stuckenschmidt, H., Tamilin, A.: Reasoning Support for Mapping Revision. Journal of Logic and Computation 19(5), 807–829 (2008), http://disi.unitn.it/~p2p/RelatedWork/Matching/Meilicke08reasoning.pdf
26. Meilicke, C., Völker, J., Stuckenschmidt, H.: Learning Disjointness for Debugging Mappings between Lightweight Ontologies. In: Proceedings of the 16th international conference on Knowledge Engineering: Practice and Patterns. pp. 93–108. EKAW '08, Springer-Verlag, Berlin, Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-87696-0_11
27. Miller, G.A.: WordNet: a lexical database for English. Communications of the ACM 38, 39–41 (November 1995), http://dl.acm.org/citation.cfm?doid=219717.219748
28. Nebert, D.D.: Developing Spatial Data Infrastructures: The SDI Cookbook. Global Spatial Data Infrastructure Association (GSDI) (2004)
29. Niepert, M., Meilicke, C., Stuckenschmidt, H.: A Probabilistic-Logical Framework for Ontology Matching. In: American Association for Artificial Intelligence (2010)
30. Noy, N., Stuckenschmidt, H.: Ontology alignment: An annotated bibliography. In: Semantic Interoperability and Integration. Schloss Dagstuhl (2005), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.93.2574
31. OpenStreetMap: The Free Wiki World Map. http://www.openstreetmap.org (2012)
32. Qi, G., Ji, Q., Haase, P.: A Conflict-Based Operator for Mapping Revision: Theory and Implementation. In: Proceedings of the 8th International Semantic Web Conference. pp. 521–536. ISWC '09, Springer-Verlag, Berlin, Heidelberg (2009), http://dx.doi.org/10.1007/978-3-642-04930-9_33,http://www.springerlink.com/content/0v25w11500867103/fulltext.pdf
33. Reul, Q., Pan, J.Z.: KOSIMap: Use of Description Logic Reasoning to Align Heterogeneous Ontologies. In: the 23rd International Workshop on Description Logics (DL 2010) (2010)
34. Serafini, L., Zanobini, S., Sceffer, S., Bouquet, P.: Matching Hierarchical Classifications with Attributes. In: European Semantic Web Conference (ESWC). pp. 4–18 (2006)
35. Shvaiko, P., Euzenat, J.: Ontology Matching: State of the Art and Future Challenges. IEEE Transactions on Knowledge and Data Engineering (2012)
36. Shvaiko, P., Giunchiglia, F., Yatskevich, M.: Semantic Matching with S-Match, vol. Part 2, pp. 183–202 (2010)
37. Stoilos, G., Stamou, G.B., Kollias, S.D.: A String Metric for Ontology Alignment. In: International Semantic Web Conference. pp. 624–637 (2005)
38. Volz, S., Walter, V.: Linking Different Geospatial Databases by Explicit Relations. In: Proceedings of the XX th International Society for Photogramme-

try and Remote Sensing (ISPRS) Congress, Comm. IV. pp. 152–157 (2004), http://www.isprs.org/proceedings/XXXV/congress/comm4/papers/332.pdf
39. W3C: OWL 2 Web Ontology Language. http://www.w3.org/TR/owl2-overview (2009)
40. Wang, P., Xu, B.: Debugging Ontology Mappings: A Static Approach. Computers and Artificial Intelligence 27(1), 21–36 (2008), http://disi.unitn.it/~p2p/RelatedWork/Matching/MappingDebug_CAI.pdf
41. Welty, C.A.: Ontology research. AI Magazine 24(3), 11–12 (2003), http://www.aaai.org/ojs/index.php/aimagazine/article/view/1714

Doing Argumentation using Theories in Graph Normal Form

Sjur Dyrkolbotn

Department of Informatics
University of Bergen, Norway
sjur.dyrkolbotn@ii.uib.no

Abstract. We explore some links between abstract argumentation, logic and kernels in digraphs. Viewing argumentation frameworks as propositional theories in graph normal form, we observe that the stable semantics for argumentation can be given equivalently in terms of satisfaction and logical consequence in classical logic. We go on to show that the complete semantics can be formulated using Łukasiewicz three-valued logic.

1 Introduction

Abstract argumentation in the style of Dung [11], has gained much popularity in the AI-community, see [3] for an overview. Dung introduced *argumentation frameworks*, networks of abstract arguments that are not assumed to have any particular internal structure. All that arguments can do is attack one another, and extension based semantics for argumentation identify sets of arguments that are in some sense successful. Given a set A of arguments, some requirements are intuitively natural to stipulate. If, for instance, $a, b \in A$ and one of a and b attacks the other, then it seems problematic to accept A as successful - A effectively undermines itself. There are several other more or less natural constraints one might consider, some of which cannot be mutually satisfied, and this has given rise to several different semantics for argumentation frameworks, each able to capture some, but not all, intuitive requirements, see e.g. [2] for an overview and comparison of various approaches. In this paper we introduce *argumentation theories*, a representation of argumentation frameworks as propositional theories in graph normal form, a novel normal form for theories in propositional logic [4], closely connected to directed graphs.

In section 2 we give the necessary definitions from argumentation theory and we observe that stable sets in argumentation frameworks, satisfying assignments to their representations as theories, and kernels in the directed graphs obtained by reversing their attacks, are all one and the same. In section 3 we work with the representation of frameworks as theories, showing that the definition of a *complete extension* can be given equivalently in terms of satisfaction in Łukasiewicz three-valued logic Ł3.

2 Preliminaries

An *argumentation framework*, framework for short, is a finite digraph, $\mathsf{F} = \langle \mathcal{A}, \mathcal{R} \rangle$, with $\mathcal{A}$ a set of vertices, called *arguments*, and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ a set of directed edges, called

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see http://ceur-ws.org/, 2012, pp. 13–22.

the *attack relation.* For $\langle a, a' \rangle \in \mathcal{R}$ we say that the argument a *attacks* the argument a'. We use the notation $\mathcal{R}^+(x) = \{y \mid \langle x, y \rangle \in \mathcal{R}\}$ and $\mathcal{R}^-(y) = \{x \mid \langle x, y \rangle \in \mathcal{R}\}$. This notation extends point-wise to sets, e.g. $\mathcal{R}^-(X) = \bigcup_{x \in X} \mathcal{R}^-(x)$. We use the convention that $\mathcal{R}^+(\emptyset) = \mathcal{R}^-(\emptyset) = \emptyset$.

The most well-known semantics for argumentation, first introduced in [11] and [6] (semi-stable semantics), are given in the following definition.

Definition 21 *Given any argumentation framework* $\mathsf{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ *and a subset* $A \subseteq \mathcal{A}$*, we define* $\mathcal{D}(A) = \{x \in \mathcal{A} \mid \mathcal{R}^-(x) \subseteq \mathcal{R}^+(A)\}$*, the set of vertices* defended *by* A*. We say that*

- A *is* conflict-free *if* $\mathcal{R}^+(A) \subseteq \mathcal{A} \setminus A$*, i.e. if there are no two arguments in* A *that attack each other.*
- A *is* admissible *if it is conflict free and* $A \subseteq \mathcal{D}(A)$*. The set of all admissible sets in* F *is denoted* $a(\mathsf{F})$*.*
- A *is* complete *if it is conflict free and* $A = \mathcal{D}(A)$*. The set of all complete sets in* F *is denoted* $c(\mathsf{F})$*.*
- A *is the* grounded *set if it is complete and there is no complete set* $B \subseteq \mathcal{A}$ *such that* $B \subset A$*, it is denoted* $g(\mathsf{F})$*.*
- A *is* preferred *if it is admissible and not strictly contained in any admissible set. The set of all preferred sets in* F *is denoted* $p(\mathsf{F})$*.*
- A *is* stable *if* $\mathcal{R}^+(A) = \mathcal{A} \setminus A$*. The set of all stable sets in* F *is denoted* $s(\mathsf{F})$
- A *is* semi-stable *if it is admissible and there is no admissible set* B *such that* $A \cup \mathcal{R}^+(A) \subset B \cup \mathcal{R}^+(B)$*. The set of all semi-stable sets in* F *is denoted by* $ss(\mathsf{F})$*.*

For any $\mathcal{S} \in \{a, c, g, p, s, ss\}$, one also says that $A \in \mathcal{S}(\mathsf{F})$ is an *extension* (of the type prescribed by $\mathcal{S}$). Given a framework F and an argument $a \in \mathcal{A}$, we say that a is *credulously* accepted with respect to some $\mathcal{S} \in \{a, c, g, p, s, ss\}$ just in case there is some set $A \in \mathcal{S}(\mathsf{F})$ such that $a \in A$. If $a \in A$ for *every* $A \in \mathcal{S}(\mathsf{F})$, we say that A is *skeptically* accepted with respect to $\mathcal{S}$. If an argument is neither credulously nor skeptically accepted with respect to a semantics, it is *rejected.*[1] Intuitively, if an argument is credulously accepted, then it is involved in some line of argument that is successful; it is potentially useful, and should be considered further. If an argument is skeptically accepted, it is involved in all successful lines of arguments; it is beyond reproach, and arguing against it should be considered useless.

Notice that it follows from Definition 21 that the empty set is admissible and that all stable sets are semi-stable, all semi-stable sets are preferred, all preferred sets are complete and all complete sets are admissible. Also, it is not hard to see that the grounded extension is contained in every complete set of arguments. In Figure 1, we give two argumentation frameworks, F and F', that serve as examples. In the framework F, every argument is attacked by some argument, and from this it follows that we have $g(\mathsf{F}) = \emptyset$, i.e., the grounded extension is the empty set. The non-empty conflict-free sets are the singletons $\{a\}, \{b\}$ and $\{c\}$, but we observe that a does not defend himself against the attack he receives from c (since there is no attack (a, c)), and that c does not defend himself against the attack he receives from b. So the only possible non-empty admissible set is $\{b\}$. It is indeed admissible; b is attacked only by a and he defends himself, attacking a in return. In fact, since b also attacks c, the set $\{b\}$ is the unique stable set of this framework. It follows that $s(\mathsf{F}) = p(\mathsf{F}) = ss(\mathsf{F}) = \{\{b\}\}$

[1] Arguments that are credulously but not skeptically accepted are typically called *defensible*

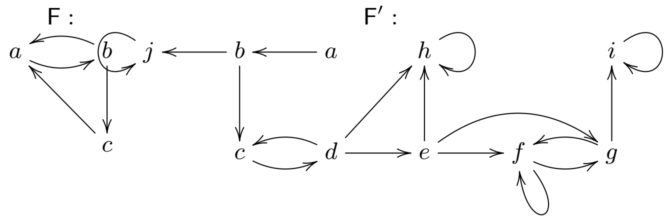


Fig. 1. Two argumentation frameworks

and $a(\mathsf{F}) = c(\mathsf{F}) = \{\emptyset, \{b\}\}$. For a more subtle example, consider F'. The first thing to notice here is that we have an unattacked argument a, so the grounded extension is non-empty. In fact, the framework is such that all semantics from Definition 21 behave differently. It might look a bit unruly, but there are many self-attacking arguments that can be ruled out immediately (since they are not in any conflict-free sets), and it is easy to verify that the extensions of F' under the different semantics are the following:

$$g(\mathsf{F}') = \{a\}, s(\mathsf{F}') = \emptyset, ss(\mathsf{F}') = \{\{a, d, g\}\}$$
$$a(\mathsf{F}') = \{\emptyset, \{a\}, \{a, c\}, \{a, c, e\}, \{d\}, \{a, d\}, \{a, d, g\}, \{d, g\}\}$$
$$p(\mathsf{F}') = \{\{a, d, g\}, \{a, c, e\}\}, c(\mathsf{F}') = \{\{a\}, \{a, d, g\}, \{a, c, e\}, \{a, d\}\}$$

Notice that $s(\mathsf{F}') = \emptyset$ - no stable set exists in the framework. That stable sets are not guaranteed to exist is a major objection against what is otherwise a fairly conclusive notion of success - an internally consistent set of arguments that defeats all others. Other semantics for argumentation can - to quite some extent - be seen as attempts at arriving at a reasonable notion of success that is weak enough so that interesting extensions always exists, while still strong enough so that it is adequate for applications and have interesting theoretical properties. As we will see later, this is intimately related to the problem of *inconsistency handling* in logic, with stable sets not existing in a framework precisely when the corresponding propositional theory is not classically consistent.

Argumentation has close links to established concepts in both graph theory and logic. We start by briefly accounting for the link with what is known as kernels in the theory of directed graphs. Given a directed graph (digraph) $\mathsf{D} = \langle D, N \rangle$ with $N \subseteq D \times D$, a set $K \subseteq D$ is a *kernel* in D if

$$N^{-}(K) = D \setminus K$$

Kernels were introduced by Von Neumann and Morgenstern in [18] in the context of cooperative games, but have since attracted quite a bit of interest from graph-theorists, see [5] for a recent overview. The connection to argumentation should be apparent. If we let $\overleftarrow{\mathsf{D}}$ denote the digraph obtained by reversing all edges in D, then it is not hard to see that a kernel in D is a stable set in $\overleftarrow{\mathsf{D}}$ and vice versa. In kernel theory, one also considers *local kernels* [17], which are sets $L \subseteq D$ such that

$$N^{+}(L) \subseteq N^{-}(L) \subseteq D \setminus L$$

It is easy to verify that a local kernel in D is an admissible set in $\overleftarrow{\mathsf{D}}$ and vice versa. These two observations seem valuable to argumentation, since in graph theory, several interesting results and techniques have been found, especially concerning the question of finding structural conditions that ensure the *existence* of kernels, see e.g. [9, 10, 15]. Still, as far as we are aware, the connection to argumentation has never been systematically explored.[2]

In this paper, we will describe some connections between argumentation and logic, and we will not explore the link with kernels any further. What is important for us, is that digraphs inspire a normal form for propositional theories where an assignment is satisfying iff it gives rise to a kernel in the corresponding digraph, see [4, 19] for two recent papers that explore this link. We remark that the observation we make here is in some sense implicit already in the original paper by Dung [11], who connects argumentation to logic programming. Still, the direct representation of argumentation frameworks as propositional theories is more straightforward and also seems useful, as suggested by the results we obtain in Section 3, and the fact that they can be established so easily.

A propositional formula, ϕ, is said to be in *graph normal form* iff $\phi = x \leftrightarrow \bigwedge_{y \in X} \neg y$ for propositional letters $\{x\} \cup X$. In [4] it is shown that this is indeed a normal form for propositional logic, every propositional theory has an equisatisfiable one containing only formulas of this form.[3] The correspondence with digraphs is detailed in [4, 19]. Instead of reversing edges in order to connect it to argumentation, we give our own formulation that can be applied directly. Given a framework F, we define the *argumentation theory* TF as follows:

$$\mathsf{TF} = \{x \leftrightarrow \bigwedge_{y \in \mathcal{R}^-(x)} \neg y \mid x \in \mathcal{A}\} \tag{1}$$

For instance, the argumentation theory for the framework F depicted in Figure 1, consists of the following equivalences:

$$a \leftrightarrow \neg b \wedge \neg c,\ b \leftrightarrow \neg a,\ c \leftrightarrow \neg b$$

We observe that the only satisfying assignment in classical logic is $\delta : \mathcal{A} \to \{\mathbf{0}, \mathbf{1}\}$ such that $\delta(a) = \delta(c) = \mathbf{0}$ and $\delta(b) = \mathbf{1}$, corresponding to the fact that the only successful line of argument under stable semantics is the set containing the argument b, defeating a and c.

Argumentation theories are written using graph normal form, and we can also move in the opposite direction. The construction seems to be of independent interest, so we will present it here. It provides, in particular, a simple way in which results and techniques from argumentation can be applied to logic. Solving SAT, for instance, becomes the search for a stable set, while the other semantics for argumentation could provide new and useful ways in which to extract information from inconsistent theories.

For a positive natural number n, let $[n] = \{1, \ldots, n\}$. Then, for any theory, $\mathsf{T} = \{x_1 \leftrightarrow \bigwedge_{x \in X_1} \neg x, \ldots, x_n \leftrightarrow \bigwedge_{x \in X_n} \neg x\}$, we define an argumentation framework $\mathsf{F}_T = \langle \mathsf{F}_T, \mathcal{R}_T \rangle$ as follows

$$\begin{array}{l} \mathcal{A}_T = \bigcup_{i \in [n]} (\{x_i\} \cup X_i \cup \{\bar{x} \mid x \in X_i \wedge \forall i \in [n] : x \neq x_i\}) \\ \mathcal{R}_T = (\bigcup_{i \in [n]} \{(x, x_i) \mid x \in X_i\}) \cup \{(x, \bar{x}), (\bar{x}, x) \mid x \in \mathcal{A}_T\} \end{array} \tag{2}$$

[2]The connection has been noted, for instance in [8], where the authors remark that some of their results concerning symmetric argumentation frameworks (all attacks are mutual) follow from basic results in kernel theory.

[3]Equisatisfiable means that for every satisfying assignment to one there is a satisfying assignment to the other, i.e., the assignments are not necessarily the same (the addition of new propositional letters must be permitted)

We introduce a fresh argument $\bar{x}$ for every propositional letter x that does not occur to the left of any equivalence. The reason is that we do not want to force acceptance of x when viewed as an argument. Rather, x should be open for both acceptance and rejection, depending on the rest of the theory. This is achieved by adding the symmetric attack $\{(x, \bar{x}), (\bar{x}, x)\}$.

Let $I = \{\bar{x} \mid x \in \mathcal{A}\}$ denote all the arguments from F_T that do not correspond to propositional letters used in T. Given a function $\alpha : X \to Y$, let $\alpha|_Z$ denote its restriction to domain $Z \subseteq X$. Also, given $\delta : X \to \{\mathbf{0}, \mathbf{1}\}$, we let $\bar{\delta}$ denote the boolean evaluation of formulas induced by δ according to classical logic.

Given a stable set $E \subseteq \mathsf{F}_T$, consider $\delta_E : \mathcal{A} \to \{\mathbf{0}, \mathbf{1}\}$ defined by $\delta_E(x) = \mathbf{1}$ iff $x \in E$ (meaning that $\delta_E(x) = \mathbf{0}$ for all $x \in \mathcal{A}_T \setminus E$). It is not hard to show that $\delta_E|_{\mathcal{A}_T \setminus I}$ is a satisfying assignment for T in classical logic, i.e., that $\bar{\delta}_E(\phi) = \mathbf{1}$ for all $\phi \in T$. Similarly, if we are given a satisfying assignment $\delta : \mathcal{A}_T \setminus I \to \{\mathbf{0}, \mathbf{1}\}$, we obtain the stable set $E_\delta = \{x \in \mathcal{A}_T \mid \delta(x) = \mathbf{1}\} \cup \{\bar{x} \in I \mid \delta(x) = \mathbf{0}\}$.

The constructions given in this section are completely analogous to the ones given in [4] in order to establish equivalence between theories in graph normal form and kernels in digraphs. Therefore, we will simply summarize, without a formal proof, one obvious consequence for argumentation. We let $\models$ denote the satisfaction relation in classical propositional logic, and we let $\perp$ denote contradiction in classical logic.

Theorem 24 *Given an argumentation framework* F*, and an argument* $a \in \mathcal{A}$

(1) F *admits a stable extension iff* $\mathsf{TF} \not\models \perp$ *(i.e. iff* TF *is satisfiable)*
(2) $a \in \mathcal{A}$ *is credulously accepted with respect to stable semantics iff* $\{a\} \cup \mathsf{TF} \not\models \perp$
(3) $a \in \mathcal{A}$ *is skeptically accepted with respect to stable semantics iff* $\mathsf{TF} \models a$

When we succeed in giving semantics for argumentation in terms of logical consequence and consistency, we no longer need to consider just atomic arguments and statements about them specified in some informal or semi-formal way. Rather, one can now use logic, and form a propositional formula which can then be evaluated against the theory corresponding to the framework. For instance, given a framework F with $a \in \mathcal{A}$, we have that a is not credulously accepted with respect to stable semantics iff TF is a model of $\neg a$, i.e., iff we have $\mathsf{TF} \models \neg a$. For another example, consider how one may write simply $a \to b$ to indicate that every successful line of argument which involves accepting a must also involve accepting b. More generally, when we succeed in providing a formulation in terms of a known logic, we can use reasoning systems developed for this logic both to address standard notions from argumentation such as credulous and skeptical acceptance, and also to check how arbitrary propositional formulas fare with respect to an argumentation theory; formulas that can now be seen as statements about interaction among arguments in the underlying argumentation framework. Conversely, we may use techniques developed in argumentation theory to analyze logical theories in graph normal form - something that might prove particularly useful for algorithmic problems, since the digraph structure of argumentation frameworks should prove particularly useful in this regard. We remark that interesting results have already been obtained which exploits digraph-properties for tackling algorithmic problems concerning the semantic properties of argumentation frameworks, see e.g., [12, 13].

It seems, in light of this, that the search for nice logical accounts of argumentation is a highly worthwhile direction of research. In fact, it has recently been taken up by logicians coming at argumentation from a more theoretical, less application-oriented, angle, see e.g., [7, 16]. Conceptually, we differ from these approaches in that we rely on the link with known concepts in digraph theory and the representation of frameworks as propositional theories, rather than on the use of modal logic.

3 Argumentation and Łukasiewicz logic Ł3

In this section we will characterize the complete extensions logically. This necessitates a move away from classical logic. In Caminada and Gabbay [7], a characterization is provided using modal logic and a complicated modal version of Equation 1. In this section we show that complete extensions can be characterized much more simply using Łukasiewicz three valued logic Ł3. This observation is also made in [14], but there it is formalized only with respect to local kernels and a new logic for reasoning about paradoxes. The details and consequences are not worked out in the context of argumentation.

Here, we directly link Ł3 to argumentation by showing that any satisfying assignment to TF gives rise to a complete extension in F and vice versa. The argument we give proceeds by showing that an assignment is satisfying iff it is what is known in argumentation theory as a *complete Caminada labeling.* While not technically challenging, this result seems nice, since it implies that skeptical and credulous acceptance of arguments with respect to the complete semantics reduces to logical consequence and satisfiability in Ł3. We remark, in particular, that Łukasiewicz three-valued logic admits a nice proof theory, see e.g., [1]. We also remark that since the grounded extension is complete and also contained in every complete extension, characterizing skeptical acceptance with respect to complete semantics amounts to completely characterizing the grounded extension.

Given a propositional language over connectives $\{\neg, \wedge, \rightarrow\}$ and propositional alphabet $\mathcal{A}$, the semantics of logic Ł3 is defined as follows.

Definition 31 *Given an assignment $\delta : \mathcal{A} \rightarrow \{0, 1/2, 1\}$ its extension, $\bar{\delta}$, is defined inductively on complexity of formulas.*

- $\bar{\delta}(a) = \delta(a)$ *for all* $a \in \mathcal{A}$
- $\bar{\delta}(\neg\phi) = 1 - \bar{\delta}(\phi)$
- $\bar{\delta}(\phi \rightarrow \psi) = min\{1, (1 - \bar{\alpha}(\phi)) + \bar{\alpha}(\phi)\}$
- $\bar{\delta}(\phi \wedge \psi) = min\{\bar{\alpha}(\phi), \bar{\alpha}(\psi)\}$

The consequence relation of Łukasiewicz logic is $\models_{\mathrm{L}} \subseteq 2^{\mathcal{L}} \times \mathcal{L}$, defined such that $\Phi \models_{\mathrm{L}} \phi$ iff for all $\delta : \mathcal{A} \rightarrow \{0, 1/2, 1\}$, we have that $\bar{\delta}(\phi) = \mathbf{1}$ whenever $\bar{\delta}(\psi) = \mathbf{1}$ for all $\psi \in \Phi$,

Notice that $\phi \rightarrow \psi$ obtains semantic value 1 ("true") under some assignment just in case ψ does not receive a lower semantic value than ϕ. Defining $\phi \leftrightarrow \psi = (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$ as usual, one also notices that $\phi \leftrightarrow \psi$ obtains the value 1 just in case ϕ and ψ obtain the same semantic value.

Next we define the complete Caminada labellings, not using the original definition, but the equivalence stated and proven in [7, Proposition 1, p. 6-7]

Definition 32 *A function $\delta : \mathcal{A} \rightarrow \{0, 1/2, 1\}$ is a complete Caminada labeling iff for all $x \in \mathcal{A}$ we have:*

- *If* $\delta(x) = 1$ *then for all* $y \in \mathcal{R}^{-}(x) : \delta(y) = 0$

– *If* $\delta(x) = 0$ *then there is* $y \in \mathcal{R}^-(x) : \delta(y) = 1$
– *If* $\delta(x) = 1/2$ *then there is* $y \in \mathcal{R}^-(x)$ *such that* $\delta(y) = 1/2$ *and there is no* $z \in \mathcal{R}^-(x)$ *such that* $\delta(z) = 1$

In [7, Theorem 2], the authors prove that if $\delta : \mathcal{A} \to \{0, 1/2, 1\}$ is a complete Caminada labeling then $\delta^1 = \{x \in \mathcal{A} \mid \delta(x) = 1\}$ is a complete extension for F. They also show that if $E \subseteq \mathcal{A}$ is a complete extension, then there is a corresponding complete Caminada labeling $\delta_E : \mathcal{A} \to \{0, 1/2, 1\}$, defined as follows:

– $\delta_E(x) = 1$ for all $x \in E$
– $\delta_E(x) = 0$ for all $x \in \mathcal{R}^+(E)$
– $\delta_E(x) = 1/2$ for all $x \in \mathcal{A} \setminus (E \cup \mathcal{R}^+(E))$

In the following, we prove that a complete Caminada labeling can be equivalently defined as a satisfying assignment for TF in the logic Ł3. This shows that the results from [7] have direct logical content, and that we do not need to introduce modal logic to characterize complete extensions logically.

Theorem 33 *Given an argumentation framework* F*, we have that* $\delta : \mathcal{A} \to \{0, 1/2, 1\}$ *is a complete Caminada labeling iff* $\bar{\delta}(\phi) = 1$ *for all* $\phi \in \mathsf{TF}$.

Proof. $\Rightarrow$) Assume that $\delta : \mathcal{A} \to \{0, 1/2, 1\}$ is a complete Caminada labeling and consider an arbitrary $\phi = x \leftrightarrow \bigwedge_{y \in \mathcal{R}^-(x)} \neg y \in \mathsf{TF}$. We need to show that $\bar{\delta}(x \leftrightarrow \bigwedge_{y \in \mathcal{R}^-(x)} \neg y) = 1$. There are three cases:

– $\delta(x) = 1$. Since δ is a complete Caminada labeling we have, for all $y \in \mathcal{R}^-(x)$, $\delta(y) = 0$. It follows that $\bar{\delta}(\neg y) = 1$ for all $y \in \mathcal{R}^-(x)$. Since all conjuncts are true, we conclude that $\bar{\delta}(\bigwedge_{y \in \mathcal{R}^-(x)} \neg y) = 1 = \delta(x) = \bar{\delta}(x)$. So $\bar{\delta}(\phi) = 1$ as desired.
– $\delta(x) = 0$. Since δ is a complete Caminada labeling it follows that there is some $y \in \mathcal{R}^-(x)$ such that $\delta(y) = 1$. Then $\bar{\delta}(\neg y) = 0$, so it follows that $\bar{\delta}(\bigwedge_{y \in \mathcal{R}^-(x)} \neg y) = 0 = \delta(x) = \bar{\delta}(x)$. So $\bar{\delta}(\phi) = 1$ as desired.
– $\delta(x) = 1/2$. Since δ is a complete Caminada labeling, there is some $y \in \mathcal{R}^-(x)$ such that $\delta(y) = 1/2$. It also follows that there is no $z \in \mathcal{R}^-(x)$ such that $\delta(z) = 1$. From this we conclude that $1/2 = min\{\bar{\delta}(y) \mid y \in \mathcal{R}^-(x)\}$ which means $\bar{\delta}(\bigwedge_{y \in \mathcal{R}^-(x)} \neg y) = 1/2 = \delta(x) = \bar{\delta}(x)$. So $\bar{\delta}(\phi) = 1$ as desired.

$\Leftarrow$) Assume that $\delta : \mathcal{A} \to \{0, 1/2, 1\}$ is a satisfying assignment for TF, i.e. that $\bar{\delta}(\phi) = 1$ for all $\phi \in \mathsf{TF}$. Consider arbitrary $\phi = x \leftrightarrow \bigwedge_{y \in \mathcal{R}^-(x)} \neg y \in \mathsf{TF}$. Again there are three cases:

– $\delta(x) = 1$. Since $\bar{\delta}(\phi) = 1$, we have $\bar{\delta}(\bigwedge_{y \in \mathcal{R}^-(x)} \neg y) = 1$. It follows that $\bar{\delta}(\neg y) = 1$ and therefore $\delta(y) = 0$ for all $y \in \mathcal{R}^-(x)$. So the criterion of Definition 32 is met in this case.
– $\delta(x) = 0$. Since $\bar{\delta}(\phi) = 1$, we have $\bar{\delta}(\bigwedge_{y \in \mathcal{R}^-(x)} \neg y) = 0$. It follows that $\bar{\delta}(\neg y) = 0$, and therefore $\delta(y) = 1$ for some $y \in \mathcal{R}^-(x)$. So the criterion of Definition 32 is met.
– $\delta(x) = 1/2$. Since $\bar{\delta}(\phi) = 1$, we have $\bar{\delta}(\bigwedge_{y \in \mathcal{R}^-(x)} \neg y) = 1/2$. This means that $1/2 = min\{\bar{\delta}(\neg y) \mid y \in \mathcal{R}^-(x)\}$. So there must be some $y \in \mathcal{R}^-(x)$ such that $\bar{\delta}(\neg y) = 1/2$, which means $\delta(y) = 1/2$. Also, it follows that there is no $z \in \mathcal{R}^-(x)$ such that $\bar{\delta}(\neg z) = 0$. So there is no $z \in \mathcal{R}^-(x)$ such that $\delta(z) = 1$, meaning that the criterion of Definition 32 is met in this case as well.

□

We conclude by stating the following corollary, which sums up the immediate consequences for argumentation. We let $\perp$ denote contradiction in Ł3.

Corollary 34 *Given an argumentation framework* F. *An argument* $a \in \mathcal{A}$ *is skeptically accepted with respect to complete semantics iff* $\mathsf{TF} \models_{\text{Ł}} a$ *and credulously accepted iff* $\mathsf{TF} \cup \{a\} \not\models_{\text{Ł}} \perp$.

Proof. For the first claim, remember that $a \in \mathcal{A}$ is said to be skeptically accepted with respect to complete semantics iff for all complete extensions $E \subseteq \mathcal{A}$ we have $a \in E$ iff $\delta(a) = 1$ for all complete Caminada labellings $\delta : \mathcal{A} \to \{0, 1/2, 1\}$. By Theorem 33 this is the same as saying that $\delta(a) = 1$ for all δ such that $\overline{\delta}(\phi) = 1$ for all $\phi \in \mathsf{TF}$. By Definition 31, this is the same as $\mathsf{TF} \models_{L} a$. The second claim follows similarly. □

As already noted, the logical approach means that we can form complex statements to express various claims about arguments and their interaction in the framework. For the case of complete semantics, we may write, for instance, $a \leftrightarrow \neg a$ to indicate that the argument a can be neither defeated nor accepted. It is not hard to see that this formula is true in a model TF iff neither a nor any of its attackers is credulously accepted with respect to the complete semantics. So it does capture the intended meaning, and we believe that the formula is a beautiful representation of an argument having malfunctioned, becoming instead a paradox: an argument such that accepting it is logically equivalent with defeating it!

Using a logical language to talk about argumentation provides clarity, but also sheds light on subtleties that we might not otherwise come to fully appreciate. As an example, consider again the framework F' from Figure 1. With some thought, we see that in order for both h and i to be defeated, it is necessary to use d to defeat h (since using e will defeat g - the only attacker of i except i itself). But if we try the formula $(\neg h \wedge \neg i) \to d$ and check if it follows logically from TF' in Ł3, we find that it does not. The explanation for this is that implication in Ł3 treats an argument that is neither accepted nor defeated as closer to truth than an argument that is defeated. Since there is a complete set ({a,c,e}) such that d is defeated while i is neither accepted nor defeated, it follows from this that a countermodel to the implication can be found. Still, it *is* possible to express a claim that correctly describes the state of affairs that obtains. In fact, after some thought, it is seen that the claim we stated informally - that you must accept d to defeat both h and i - while true, actually serves to misrepresent the situation at hand. For what we actually have is something stronger, namely that in order for h and i to obtain *any* of the two classical values (true/false or, if you like, defeated/accepted), we have to accept d. This we can express by the implication $(\neg(h \leftrightarrow \neg h) \wedge \neg(i \leftrightarrow \neg i)) \to d$, and now we observe that this implication does indeed follow logically from TF'. Also, we obtain the further logical consequence $(\neg(h \leftrightarrow \neg h) \wedge \neg(i \leftrightarrow \neg i)) \to (\neg h \wedge \neg i)$ which is also *stronger* than the original intuition that we had. Thus, what seemed at first sight a shortcoming of the logical approach really suggested a more precise analysis of the situation, one that we might not as easily have arrived at without a logical formulation.

4 Conclusion

In this paper we have observed how argumentation frameworks can be viewed as propositional theories in graph normal form. We have shown that this makes it possible to capture the stable and complete semantics using classical and three-valued Łukasiewicz logic respectively. Moreover, we have argued that argumentation theories provide a nice way in which to talk about argumentation in a logical language. It is much more straightforward than the modal approach from [7], and exploring it further seems worthwhile. For a possible first step in future work, we remark that both the preferred and semi-stable sets seem to involve notions of maximal consistency that it should be possible, and interesting, to express in terms of logic.

References

1. Arnon Avron. Natural 3Valued Logics - Characterization and Proof Theory. *Journal of Symbolic Logic*, 56:276–294, 1991.
2. Pietro Baroni and Massimiliano Giacomin. On principle-based evaluation of extension-based argumentation semantics. *Artificial Intelligence*, 171(10–15):675–700, 2007.
3. T.J.M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10–15):619–641, 2007.
4. Marc Bezem, Clemens Grabmayer, and Michał Walicki. Expressive power of digraph solvability. *Annals of Pure and Applied Logic*, 2011. [to appear].
5. Endre Boros and Vladimir Gurvich. Perfect graphs, kernels and cooperative games. *Discrete Mathematics*, 306:2336–2354, 2006.
6. Martin Caminada. Semi-stable semantics. In *Proceedings of the 2006 conference on Computational Models of Argument: Proceedings of COMMA 2006*, pages 121–130. IOS Press, 2006.
7. Martin W. A. Caminada and Dov M. Gabbay. A logical account of formal argumentation. *Studia Logica*, 93(2-3):109–145, 2009.
8. Sylvie Coste-marquis, Caroline Devred, and Pierre Marquis. Symmetric argumentation frameworks. In *Proc. 8th European Conf. on Symbolic and Quantitative Approaches to Reasoning With Uncertainty (ECSQARU), volume 3571 of LNAI*, pages 317–328. Springer-Verlag, 2005.
9. Pierre Duchet. Graphes noyau-parfaits, II. *Annals of Discrete Mathematics*, 9:93–101, 1980.
10. Pierre Duchet and Henry Meyniel. Une généralisation du théorème de Richardson sur l'existence de noyaux dans les graphes orientés. *Discrete Mathematics*, 43(1):21–27, 1983.
11. Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.
12. Paul E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artif. Intell.*, 171(10-15):701–729, 2007.
13. Wolfgang Dvorák, Reinhard Pichler, and Stefan Woltran. Towards fixed-parameter tractable algorithms for abstract argumentation. *Artif. Intell.*, 186:1–37, 2012.
14. Sjur Dyrkolbotn and Michał Walicki. Propositional discourse logic. *(submitted)*, 2011. www.ii.uib.no/ michal/graph-paradox.pdf.
15. Hortensia Galeana-Sánchez and Victor Neumann-Lara. On kernels and semikernels of digraphs. *Discrete Mathematics*, 48(1):67–76, 1984.

16. Davide Grossi. On the logic of argumentation theory. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*, AAMAS '10, pages 409–416, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.
17. Victor Neumann-Lara. Seminúcleos de una digráfica. Technical report, Anales del Instituto de Matemáticas II, Universidad Nacional Autónoma México, 1971.
18. John von Neumann and Oscar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944 (1947).
19. Michał Walicki and Sjur Dyrkolbotn. Finding kernels or solving SAT. *Journal of Discrete Algorithms*, 10:146–164, 2012.

The Expressive Power of Swap Logic

Raul Fervari

FaMAF, Universidad Nacional de Córdoba, Argentina
fervari@famaf.unc.edu.ar

Abstract. Modal logics are appropriate to describe properties of graphs. But usually these are *static* properties. We investigate dynamic modal operators that can change the model during evaluation. We define the logic $\mathcal{SL}$ by extending the basic modal language with the $\overleftarrow{\Diamond}$ modality, which is a diamond operator that has the ability to invert pairs of related elements in the domain while traversing an edge of the accessibility relation. We will investigate the expressive power of $\mathcal{SL}$, define a suitable notion of bisimulation and compare $\mathcal{SL}$ with other dynamic logics.

Keywords: Modal logic, dynamic logics, hybrid logic, bisimulation, expressivity.

1 Introduction

There are many notions in language and science that have a *modal* character, e.g. the classical notion of necessity and possibility. Modal logics [5, 6] are logics designed to deal with these notions. In general, they are adequate to describe certain patterns of behaviour of the real world. For this reason, modal logics are not just useful in mathematics or computer science; they are used in philosophy, linguistics, artificial intelligence and game theory, to name a few.

Intuitively, modal logics extend the classical logical systems with operators that represent the modal character of some situation. In particular, the basic modal logic ($\mathcal{BML}$) is an extension of propositional logic, with a new operator. We now define formally the syntax and semantics of $\mathcal{BML}$.

Definition 1 (Syntax). *Let* PROP *be an infinite, countable set of propositional symbols. The set* FORM *of* $\mathcal{BML}$ *formulas over* PROP *is defined as:*

$$\mathsf{FORM} ::= p \mid \neg\varphi \mid \varphi \wedge \psi \mid \Diamond\varphi,$$

where $p \in \mathsf{PROP}$ *and* $\varphi, \psi \in \mathsf{FORM}$*. We use* $\Box\varphi$ *as a shorthand for* $\neg\Diamond\neg\varphi$*, while* $\bot$*,* $\top$ *and* $\varphi \vee \psi$ *are defined as usual.*

Definition 2 (Semantics). *A model* $\mathcal{M}$ *is a triple* $\mathcal{M} = \langle W, R, V\rangle$*, where* W *is a non-empty set;* $R \subseteq W \times W$ *is the accessibility relation; and* $V : \mathsf{PROP} \to \mathcal{P}(W)$ *is a valuation. Let* w *be a state in* $\mathcal{M}$*, the pair* $(\mathcal{M}, w)$ *is called a pointed model; we will usually drop parenthesis and write* $\mathcal{M}, w$*. Given a pointed model* $\mathcal{M}, w$ *and a formula* φ *we say that* $\mathcal{M}, w$ *satisfies* φ ($\mathcal{M}, w \models \varphi$) *when*

$$\begin{array}{lll}
\mathcal{M}, w \models p & \textit{iff} & w \in V(p) \\
\mathcal{M}, w \models \neg\varphi & \textit{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi \wedge \psi & \textit{iff} & \mathcal{M}, w \models \varphi \textit{ and } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \Diamond\varphi & \textit{iff} & \textit{for some } v \in W \textit{ s.t. } (w, v) \in R, \mathcal{M}, v \models \varphi.
\end{array}$$

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see http://ceur-ws.org/, 2012, pp. 23–31.

φ is satisfiable if for some pointed model $\mathcal{M}, w$ we have $\mathcal{M}, w \models \varphi$.

As shown in Definition 2, modal logics describe characteristics of relational structures. But these are *static* characteristics of the structure, i.e. properties never change after the application of certain operations. If we want to describe *dynamic aspects* of a given situation, e.g. how the relations between a set of elements *evolve* through time or through the application of certain operations, the use of modal logics (or actually, any logic with classical semantics) becomes less clear. We can always resort to modeling the whole space of possible evolutions of the system as a graph, but this soon becomes unwieldy. It would be more elegant to use truly dynamic modal logics with operators that can mimic the changes that structure will undergo. This is not a new idea, and a clear example of this kind of logics is the *sabotage logic* $\mathcal{SML}$ introduced by van Benthem in [4].

Consider the following *sabotage game*. It is played on a graph with two players, Runner and Blocker. Runner can move on the graph from node to accessible node, starting from a designated point, and with the goal of reaching a given final point. He should move one edge at a time. Blocker, on the other hand, can delete one edge from the graph, every time it is his turn. Of course, Runner wins if he manages to move from the origin to the final point in the graph, while Blocker wins otherwise. van Benthem proposes transforming the sabotage game into a modal logic; this idea has been studied in several other works [8, 11]. In particular, they defined the operator of sabotage as:

$$\mathcal{M}, w \models \diamond\!\!\!\cdot\,\varphi \quad \text{iff} \quad \text{there is a pair } (v, u) \text{ of } \mathcal{M} \text{ such that } \mathcal{M}_{\{(v,u)\}}, w \models \varphi,$$

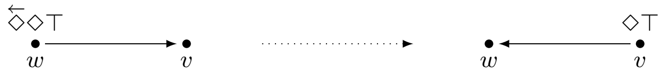
where $\mathcal{M}_E = \langle W, R \setminus E, V\rangle$, with $E \subseteq R$.

It is clear that the $\diamond\!\!\!\cdot$ operator *changes* the model in which a formula is evaluated. As van Benthem puts it, $\diamond\!\!\!\cdot$ is an "external" modality that takes evaluation to another model, obtained from the current one by deleting some transition. It has been proved that solving the sabotage game is PSpace-hard [4], while the model checking problem of the associated modal logic is PSpace-complete and the satisfiability problem is undecidable [8, 9]. It has been investigated in these articles that the logic fails to have two nice model theoretical properties: *the finite model property* (if a formula is satisfiable, then it is satisfiable in a finite model) and *the tree model property* (every satisfiable formula is satisfied in the root of a tree-like model).

Another family of model changing logics is memory logics [1, 3, 10]. The semantics of these languages is specified on models that come equipped with a set of states called the *memory*. The simplest memory logic includes a modality ⓡ that *stores* the current point of evaluation into memory, and a modality ⓚ that verifies whether the current state of evaluation has been memorized. The memory can be seen as a special proposition symbol whose extension grows whenever the ⓡ modality is used. In contrast with sabotage logics, the basic memory logic *expands* the model with an ever increasing set of memorized elements. The general properties of memory logics are similar to those of sabotage logics: a PSpace-complete model checking problem, an undecidable satisfiability problem, and failure of both the finite model and the tree model properties.

In this article, we will investigate a model changing operator that in the general case doesn't shrink nor expand the model. Instead, it has the ability to swap the direction of a traversed arrow. The $\overleftarrow{\Diamond}$ operator is a $\Diamond$ operator — to be true at a state w it requires the existence of an accessible state v where evaluation will continue— but it changes the accessibility relation during evaluation —the pair (w, v) is deleted, and the pair (v, w) is added to the accessibility relation.

A picture will help understand the dynamics of $\overleftarrow{\Diamond}$. The formula $\overleftarrow{\Diamond}\Diamond\top$ is true in a model with two related states:



As we can see in the picture, evaluation starts at state w with the arrow pointing from w to v, but after evaluating the $\overleftarrow{\Diamond}$ operator, it continues at state v with the arrow now pointing from v to w. In this article, we will study the expressive power of $\mathcal{SL}$ and will compare it with $\mathcal{BML}$ and some dynamic logics.

2 Swap Logic

Now we introduce syntax and semantics for $\mathcal{SL}$. We will define first some notation that will help us describe models with swapped accessibility relations.

Definition 3. *Let R and S be two binary relations over a set W. We define the relation $R^S = (R \setminus S^{-1}) \cup S$. When S is a singleton $\{(v,w)\}$ we write R^{vw} instead of $R^{\{(v,w)\}}$.*

Intuitively R^S stands for R with some edges swapped around (S contains the edges in their final position). The following property is easy to verify:

Proposition 1. *Let $R, S, S' \subseteq W^2$ be binary relations over an arbitrary set W, then $(R^S)^{S'} = R^{S^{S'}}$.*

We extend $\mathcal{BML}$ with a new operator $\overleftarrow{\Diamond}$. For $(v,w) \in R^{-1}$, let $\mathcal{M}^{vw} = \langle W, R^{vw}, V\rangle$. Similarly, for $S \subseteq R^{-1}$, $\mathcal{M}^S = \langle W, R^S, V\rangle$. Then we define the semantics of the new operator as follows:

$$\mathcal{M}, w \models \overleftarrow{\Diamond}\varphi \text{ iff for some } v \in W \text{ s.t. } (w,v) \in R,\ \mathcal{M}^{vw}, v \models \varphi.$$

The semantic condition for $\overleftarrow{\Diamond}$ looks quite innocent but, as we will see in the following example, it is actually very expressive.

Example 1. Define $\Box^0\varphi$ as φ, $\Box^{n+1}\varphi$ as $\Box\Box^n\varphi$, and let $\Box^{(n)}\varphi$ be a shorthand for $\bigwedge_{1\leq i\leq n}\Box^i\varphi$. The formula $\varphi = p \wedge \Box^{(3)}\neg p \wedge \overleftarrow{\Diamond}\Diamond\Diamond p$ is true at a state w in a model, only if w has a reflexive successor. Notice that no equivalent formula exists in the basic modal language (satisfiable formulas in the basic modal language can always be satisfied at the root of a tree model).

Let us analyse the formula in detail. Suppose we evaluate φ at some state w of an arbitrary model. The 'static' part of the formula $p \wedge \Box^{(3)}\neg p$ makes sure that p is true in w and that no p state is reachable within a three steps neighbourhood of w (in particular, the evaluation point cannot be reflexive). Now, the 'dynamic' part of the formula $\overleftarrow{\Diamond}\Diamond\Diamond p$ will do the rest. Because $\overleftarrow{\Diamond}\Diamond\Diamond p$ is true at w, there should be an R-successor v where $\Diamond\Diamond p$ holds once the accessibility relation has been updated to R^{vw}. Now, v has to reach a p-state in exactly two R^{vw}-steps to satisfy $\Diamond\Diamond p$. But the only p state sufficiently close for this to happen is w which is reachable in one step. As w is not reflexive, v has to be reflexive so that we can linger at v for one loop and reach p in the correct number of states.

Example 1 shows that the tree model property fails for $\mathcal{SL}$. It should be a warning about the expressivity of $\mathcal{SL}$, which is certainly above that of the $\mathcal{BML}$.

3 Bisimulation and Expressive Power

In most modal logics, bisimulations are binary relations linking elements of the domains that have the same atomic information, and preserving the relational structure of the model [6]. This will not suffice for $\mathcal{SL}$ where we also need to capture the dynamic behaviour of the $\overleftarrow{\Diamond}$ operator. The proper notion of $\mathcal{SL}$-bisimulations links states together with the context of potentially swapped edges.

Definition 4 (Swap Bisimulations). *Given models $\mathcal{M} = \langle W, R, V\rangle$ and $\mathcal{M}' = \langle W', R', V'\rangle$, together with points $w \in W$ and $w' \in W'$ we say that they are bisimilar and write $\mathcal{M}, w \leftrightarrows \mathcal{M}', w'$ if there is a relation $Z \subseteq (W \times \mathcal{P}(W^2)) \times (W' \times \mathcal{P}(W'^2))$ such that $(w, R)Z(w', R')$ satisfying the following conditions.*
Whenever $(w, S)Z(w', S')$ then

(Atomic Harmony) *for all $p \in \mathsf{PROP}$, $\mathcal{M}, w \models p$ iff $\mathcal{M}', w' \models p$;*
(Zig) *If wSv, there is $v' \in W'$ s.t. $w'S'v'$ and $(v, S)Z(v', S')$;*
(Zag) *If $w'S'v'$, there is $v \in W$ s.t. wSv and $(v, S)Z(v', S')$;*
(S-Zig) *If wSv, there is $v' \in W'$ s.t. $w'S'v'$ and $(v, S^{vw})Z(v', S'^{v'w'})$;*
(S-Zag) *If $w'S'v'$, there is $v \in W$ s.t. wSv and $(v, S^{vw})Z(v', S'^{v'w'})$.*

Theorem 1 (Invariance for Swap Logic.). *Let $\mathcal{M} = \langle W, R, V\rangle$ and $\mathcal{M}' = \langle W', R', V'\rangle$ two models, $w \in W$, $w' \in W'$, $S \subseteq R^{-1}$ and $S' \subseteq R'^{-1}$. If $\mathcal{M}, w \leftrightarrows \mathcal{M}', w'$ then for any formula $\varphi \in \mathcal{SL}$, $\mathcal{M}^S, w \models \varphi$ iff $\mathcal{M}'^{S'}, w' \models \varphi$.*

Proof. The proof is by structural induction on $\mathcal{SL}$ formulas. The base case holds by Atomic Harmony, and the $\wedge$ and $\neg$ cases are trivial.

[$\Diamond\varphi$ **case:**] Let $\mathcal{M} = \langle W, R, V\rangle$ and $\mathcal{M}' = \langle W', R', V'\rangle$. Suppose $\mathcal{M}, w \models \Diamond\varphi$. Then there is v in W s.t. wRv and $\mathcal{M}, v \models \varphi$. Since Z is a bisimulation, by (zig) we have $v' \in W'$ s.t. $w'R'v'$ and $(v, R)Z(v', R')$. By inductive hypothesis, $\mathcal{M}', v' \models \varphi$ and by definition $\mathcal{M}', w' \models \Diamond\varphi$. For the other direction use (zag).

[$\overleftarrow{\Diamond}\varphi$ **case:**] For the left to the right direction suppose $\mathcal{M}, w \models \overleftarrow{\Diamond}\varphi$. Then there is $v \in W$ s.t. wRv and $\mathcal{M}^{vw}, v \models \varphi$. Because Z is a bisimulation, by (S-zig) we have $v' \in W'$ s.t. $w'R'v'$ and $(v, R^{vw})Z(v', R'^{v'w'})$. By inductive hypothesis, $\mathcal{M}'^{v'w'}, v' \models \varphi$ and by definition $\mathcal{M}', w' \models \overleftarrow{\Diamond}\varphi$. For the other direction use (S-zag). □

Example 2. The two models in row A of Table 1 are $\mathcal{SL}$-bisimilar. The simplest way to check this is to recast the notion of $\mathcal{SL}$-bisimulation as an Ehrenfeucht-Fraïssé game as the one used for $\mathcal{BML}$, but where Spoiler can also swap arrows when moving from a node to an adjacent node. It is clear that Duplicator has a winning strategy.

Example 3. There is no $\mathcal{SL}$-bisimulation between the models in row B of Table 1. Indeed the formula $\overleftarrow{\Diamond}\Diamond\Box\bot$ is satisfied in $\mathcal{M}_2, w'$ and not in $\mathcal{M}_1, w$. Notice that the models are $\mathcal{BML}$-bisimilar.

We are now ready to investigate the expressive power of $\mathcal{SL}$.

Definition 5 ($\mathcal{L} \leq \mathcal{L}'$). *We say that $\mathcal{L}'$ is* at least as expressive as *$\mathcal{L}$ (notation $\mathcal{L} \leq \mathcal{L}'$) if there is a function Tr between formulas of $\mathcal{L}$ and $\mathcal{L}'$ such that for every model $\mathcal{M}$ and every formula φ of $\mathcal{L}$ we have that $\mathcal{M} \models_{\mathcal{L}} \varphi$ iff $\mathcal{M} \models_{\mathcal{L}'} \mathsf{Tr}(\varphi)$. $\mathcal{M}$ is seen as a model of $\mathcal{L}$ on the left and as a model of $\mathcal{L}'$ on the right, and we use in each case the appropriate semantic relation $\models_{\mathcal{L}}$ or $\models_{\mathcal{L}'}$ as required.*

We say that $\mathcal{L}'$ is strictly more expressive *than $\mathcal{L}$ (notation $\mathcal{L} < \mathcal{L}'$) if $\mathcal{L} \leq \mathcal{L}'$ but not $\mathcal{L}' \leq \mathcal{L}$. And we say that $\mathcal{L}$ and $\mathcal{L}'$ are* uncomparable *(notation $\mathcal{L} \neq \mathcal{L}'$) if $\mathcal{L} \not\leq \mathcal{L}'$ and $\mathcal{L}' \not\leq \mathcal{L}$.*

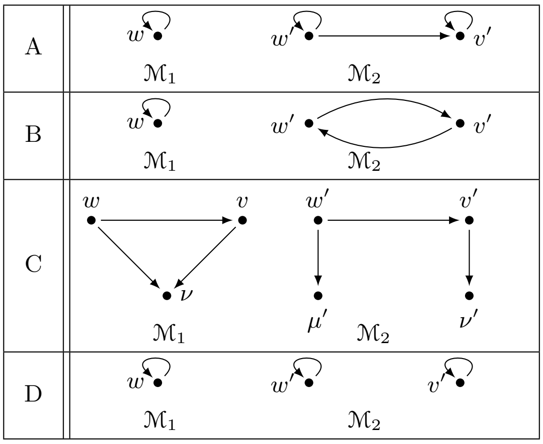


Table 1. Table of comparison between models.

The definition above requires that both logics evaluate their formulas over the same class of models. However, we can abuse of this detail when operators of some logic do not interact with the part of the model that is different of the other one. For example, memory logics operate over models that contains an aditional set to store visited states (a memory), and in our results we will evaluate formulas with an initial empty memory. When we evaluate $\mathcal{SL}$-formulas over memory logics models with an empty memory, we can as well forget about that memory and treat them as standard Kripke models. The same occurs with hybrid models (where there are nominals and $\downarrow$ operator in the language).

Our first result is fairly straightforward as it builds upon Example 3: $\mathcal{BML}$ is strictly less expressive than $\mathcal{SL}$.

Theorem 2. $\mathcal{BML} < \mathcal{SL}$.

Proof. We have to provide a translation from $\mathcal{BML}$ formulas to $\mathcal{SL}$. This is trivial as $\mathcal{BML}$ is a fragment of $\mathcal{SL}$. To prove $\mathcal{SL} \not\leq \mathcal{BML}$ consider the models in row B of Table 1. They are bisimilar for $\mathcal{BML}$ but, as we already mentioned, the $\mathcal{SL}$ formula $\overleftarrow{\Diamond}\Diamond\Box\bot$ distinguishes them. $\Box$

Now we will compare $\mathcal{SL}$ with the hybrid logic $\mathcal{H}(:,\downarrow)$, whose operator $\downarrow$ is a dynamic operator.

Definition 6 (Hybrid Logic $\mathcal{H}(:,\downarrow)$). *Let the signature $\langle \mathsf{PROP}, \mathsf{NOM}\rangle$ be given, with $\mathsf{NOM} \subseteq \mathsf{PROP}$. We extend $\mathcal{BML}$ with two new operators $n{:}\varphi$ and $\downarrow n.\varphi$, where $p \in \mathsf{PROP}$, $n \in \mathsf{NOM}$ and $\varphi, \psi \in \mathsf{FORM}$.*

A hybrid model $\mathcal{M}$ is a triple $\langle W, R, V\rangle$ as usual, but $V : \mathsf{PROP} \to \mathcal{P}(W)$ is such that $V(p)$ is a singleton if $p \in \mathsf{NOM}$. Let w be a state in $\mathcal{M}$, the semantics of the new operators is defined as:

$$\begin{array}{l} \langle W,R,V\rangle, w \models n{:}\varphi \;\; \textit{iff}\; \langle W,R,V\rangle, v \models \varphi \textit{ where } V(n) = \{v\} \\ \langle W,R,V\rangle, w \models \downarrow n.\varphi \;\textit{iff}\; \langle W,R,V_n^w\rangle, w \models \varphi, \end{array}$$

where V_n^w is defined by $V_n^w(n) = \{w\}$ and $V_n^w(m) = V(m)$ when $m \neq n$.

Formulas like $\downarrow n.\varphi$, can be intuitively read as "after naming the current state n, φ holds". For instance, $\downarrow n.\Diamond n$ means "the current state is reflexive", and $\Box\downarrow n.\Diamond n$ means "all accessible states are reflexive"

Theorem 3. $\mathcal{SL} < \mathcal{H}(:, \downarrow)$.

Proof. We will define an adequate translation from $\mathcal{SL}$ to $\mathcal{H}(:, \downarrow)$. As models of $\mathcal{H}(:, \downarrow)$ have a fixed frame, we will encode the configuration of the model (i.e., the current state of swapped links) at the syntactic level. To do so, we will take advantage of the binder $\downarrow$ and nominals to name the pair of points of the model where a swap should occur.

Let F be a non-empty set of pairs of nominals. Intuitively, this set will play a similar role than the set S of Definition 3. However, it contains purely syntactic information in the form of pair of nominals, as opposed to a direct description of swapped edge of the model. Moreover, it stores swapped links in the order of their former state. That is, F (as in "forbidden") represents the set of links that *cannot* be taken any more. We use F to keep track of the configuration of the model in which a subformula must be satisfied. In fact, we will make heavy use of nominals to exactly determine if there are links that we should not take, when translating modal logic connectors.

The complexity of the translation lies in the operators $\Diamond$ and $\overleftarrow{\Diamond}$. The translation of $\Diamond\varphi$ subformulas considers two cases:

– either the diamond is satisfied using a successor by a link that either has not been swapped, or is a swapped reflexive link. To know which accessible worlds in the hybrid model should not be accessible in the swap model, we first have to determine where we are. That is, for every forbidden pair $(x, y) \in F$, decide if x is true or not at the current point. Then for all true x, we enforce that either $\neg y$ is true at the destination point, or x is true (for the reflexive case). Of course, the translation of φ has to be true at that destination point.
– or the diamond is satisfied by taking a swapped edge (y, x). In this case the current point of evaluation should be y and we should continue evaluation at x.

The cases for $\overleftarrow{\Diamond}$ subformulas are similar, but we should record that the used link is now swapped:

– either $\overleftarrow{\Diamond}$ is satisfied by traversing a new edge, in which case we name it (x, y) and we add it to the set F.
– or $\overleftarrow{\Diamond}$ is satisfied by taking an already swapped edge $(x, y) \in F$, in which case the current point of evaluation is y, we should continue at x and remove (x, y) from F.

Formally then, the equivalence-preserving translation from $\mathcal{SL}$ to $\mathcal{H}(:, \downarrow)$ is defined as follows:

Definition 7. *Let $F \subseteq \mathsf{NOM} \times \mathsf{NOM}$. Define $(\ )'_F$ from formulas of $\mathcal{SL}$ to formulas of $\mathcal{H}(:, \downarrow)$ as*

Let $F \subseteq \mathsf{NOM} \times \mathsf{NOM}$. Define $(\ )'_F$ from formulas of $\mathcal{SL}$ to formulas of $\mathcal{H}(:, \downarrow)$ as

$$
\begin{array}{lcll}
(p)'_F & = & p & \\
(\neg\varphi)'_F & = & \neg(\varphi)'_F & \\
(\psi \wedge \varphi)'_F & = & (\psi)'_F \wedge (\varphi)'_F & \\
(\Diamond\varphi)'_F & = & \bigwedge_{c\in\mathcal{P}(F)} & (\bigwedge_{xy\in F}(\neg)_{xy\notin c}x) \ \to \Diamond((\bigwedge_{xy\in c}\neg y \vee x) \wedge (\varphi)'_F) \\
 & & \vee \bigvee_{xy\in F} & y \wedge x{:}(\varphi)'_F \\
(\overleftarrow{\Diamond}\varphi)'_F & = & \bigwedge_{c\in\mathcal{P}(F)} & (\bigwedge_{xy\in F}(\neg)_{xy\notin c}x) \ \to \downarrow i\Diamond(\downarrow j(\bigwedge_{xy\in c}\neg y \vee x) \wedge (\varphi)'_{F\cup ij}) \\
 & & \vee \bigvee_{xy\in F} & y \wedge x{:}(\varphi)'_{F\setminus xy}
\end{array}
$$

where i and j are nominals that do not appear in F.

We introduced two $\mathcal{SL}$-bisimilar models in Example 2. Finally the formula $\downarrow x.\Diamond\neg x$ distinguishes them, being true in $\mathcal{M}_2, w'$ and false in $\mathcal{M}_1, w$. □

Summing up, $\mathcal{SL}$ is strictly in between $\mathcal{BML}$ and $\mathcal{H}(:, \downarrow)$. Let us now compare $\mathcal{SL}$ with a family of dynamic modal logics called *memory logics*. Memory logics [1, 10] are modal logics with the ability to *store* the current state of evaluation into a set, and to consult whether the current state of evaluation belongs to this set. This set is also called the memory.

Definition 8 (Memory Logics). *The set* FORM *of formulas of* $\mathcal{ML}(\textcircled{r}, \textcircled{k})$ *over* PROP *is defined as in Definition 1 but adding a new zero-ary operator* $\textcircled{k}$ *and a new unary operator* $\textcircled{r}\varphi$*. The same holds for* $\mathcal{ML}(\langle\langle r\rangle\rangle, \textcircled{k})$*, but adding* $\langle\langle r\rangle\rangle\varphi$ *and deleting the classical* $\Diamond$*.*

A model $\mathcal{M} = \langle W, R, V, S\rangle$ *is an extension of an* $\mathcal{SL}$ *model with a memory* $S \subseteq W$*. Let* w *be a state in* $\mathcal{M}$*, we define satisfiability as:*

$$\begin{array}{ll} \langle W, R, V, S\rangle, w \models \textcircled{k} & \textit{iff } w \in S \\ \langle W, R, V, S\rangle, w \models \textcircled{r}\varphi & \textit{iff } \langle W, R, V, S \cup \{w\}\rangle, w \models \varphi \\ \langle W, R, V, S\rangle, w \models \langle\langle r\rangle\rangle\varphi & \textit{iff } \langle W, R, V, S\rangle, w \models \textcircled{r}\Diamond\varphi. \end{array}$$

A formula φ *of* $\mathcal{ML}(\textcircled{r}, \textcircled{k})$ *or* $\mathcal{ML}(\langle\langle r\rangle\rangle, \textcircled{k})$ *is* satisfiable *if there exists a model* $\langle W, R, V, \emptyset\rangle$ *such that* $\langle W, R, V, \emptyset\rangle, w \models \varphi$*.*

In the definition of satisfiability, the empty initial memory ensures that no point of the model satisfies the unary predicate $\textcircled{k}$ unless a formula $\textcircled{r}\varphi$ or $\langle\langle r\rangle\rangle\varphi$ has previously been evaluated there. The memory logic $\mathcal{ML}(\langle\langle r\rangle\rangle, \textcircled{k})$ does not have the $\Diamond$ operator, and its expressive power is strictly weaker than $\mathcal{ML}(\textcircled{r}, \textcircled{k})$ [3, 10]. We now show that the expressive power of $\mathcal{SL}$ is uncomparable with both $\mathcal{ML}(\textcircled{r}, \textcircled{k})$ and $\mathcal{ML}(\langle\langle r\rangle\rangle, \textcircled{k})$.

Theorem 4. $\mathcal{SL} \not\geq \mathcal{ML}(\langle\langle r\rangle\rangle, \textcircled{k})$.

Proof. As we mentioned, no $\mathcal{SL}$ formula distinguishes the models in row A of Table 1, but $\langle\langle r\rangle\rangle\neg\textcircled{k}$ is satisfiable in $\mathcal{M}_2, w'$ but not in $\mathcal{M}_1, w$. □

Theorem 5. $\mathcal{ML}(\textcircled{r}, \textcircled{k}) \not\geq \mathcal{SL}$.

Proof. The models in row C of Table 1 are bisimilar in $\mathcal{ML}(\textcircled{r}, \textcircled{k})$. Indeed they are $\mathcal{BML}$ bisimilar and acyclic, hence $\textcircled{k}$ is always false after taking an accessibility relation.

The formula $\overleftarrow{\Diamond}\overleftarrow{\Diamond}\Diamond\Diamond\Diamond\Box\bot$ is satisfiable in $\mathcal{M}_2, w'$ but not in $\mathcal{M}_1, w$. □

Corollary 1. *The expressive powers of* $\mathcal{ML}(\textcircled{r}, \textcircled{k})$ *and* $\mathcal{SL}$ *are uncomparable. The same holds for* $\mathcal{ML}(\langle\langle r\rangle\rangle, \textcircled{k})$ *and* $\mathcal{SL}$*.*

Proof. From Theorems 4 and 5, given that $\mathcal{ML}(\langle\langle r\rangle\rangle, \textcircled{k}) < \mathcal{ML}(\textcircled{r}, \textcircled{k})$ [3].□

We now compare the expressive powers of $\mathcal{ML}(\textcircled{r}, \textcircled{k})$ and $\mathcal{SL}$ with the sabotage modal logic $\mathcal{SML}$.

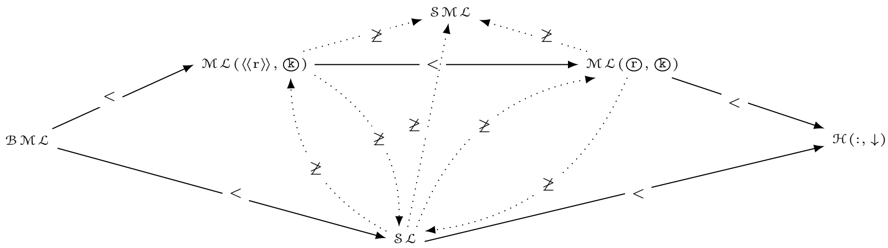
Theorem 6. $\mathcal{SL} \not\geq \mathcal{SML}$.

Proof. Consider again models in Example 2, that are $\mathcal{SL}$-bisimilar. The formula $\Diamond\!\!\!\!-\,\Diamond\top$ distinguishes them (it is satisfiable in $\mathcal{M}_2, w'$ but not in $\mathcal{M}_1, w$). □

Theorem 7. $\mathcal{ML}(\textcircled{r}, \textcircled{k}) \not\geq \mathcal{SML}$.

Proof. Models in row D of Table 1 are bisimilar in $\mathcal{ML}(\textcircled{r}, \textcircled{k})$, but the formula $\overleftarrow{\Diamond}\Diamond\top$ is satisfiable in $\mathcal{M}_2, w'$ but not in $\mathcal{M}_1, w$. ☐

The following picture sums up the results of this section (relationships between $\mathcal{ML}(\textcircled{r}, \textcircled{k})$, $\mathcal{ML}(\langle\langle r\rangle\rangle, \textcircled{k})$ and $\mathcal{H}(:, \downarrow)$ have been extracted from [3]).



It remains an open question the other directions of the comparison between $\mathcal{SML}$ and the other logics, because no notion of bisimulation for $\mathcal{SML}$ has been provided yet.

4 Conclusions

In this paper we have extended the basic modal language with the $\overleftarrow{\Diamond}$ operator, which is a diamond operator that has the ability to invert pairs of related elements in the domain while traversing an edge of the accessibility relation. Other dynamic languages that can modify the model have been investigated in the literature (e.g., sabotage logics [4, 8], memory logics [3, 10], hybrid logics [2, 7]), and we have discussed the relation between these languages and $\mathcal{SL}$. In particular, we have introduced an adequate notion of bisimulation for $\mathcal{SL}$ and used it to show that the expressive power of $\mathcal{SL}$ lies strictly in between the expressive powers of $\mathcal{BML}$ and the hybrid logic $\mathcal{H}(:, \downarrow)$, while it is uncomparable with the expressive powers of the memory logics $\mathcal{ML}(\textcircled{r}, \textcircled{k})$ and $\mathcal{ML}(\langle\langle r\rangle\rangle, \textcircled{k})$.

Many theoretical aspects of $\mathcal{SL}$ remain to be investigated. For example, it would be interesting to obtain an axiomatic characterization which is sound and complete. The task is probably non trivial, as the logic fails to be closed under uniform substitution. A proper axiomatization will require an adequate definition of when a formula is free to substitute another formula in an axiom. More generally, it is a challenge to study the properties of other dynamic operators beside $\overleftarrow{\Diamond}$. For example, two operators which would be closer to those investigated by sabotage logics would be one that access an arbitrary element in the model (similar to a global modality) and makes it a successor of the current point of evaluation, complemented with one that access a successor of the current point of evaluation and deletes the edge between the two. Investigating further examples of this kind of operator will let us have a clearer picture of the gains and losses of working with logics which can dynamically modify the model.

Acknowledgments:

This work was partially supported by grants ANPCyT-PICT-2008-306, ANPCyT-PIC-2010-688, the FP7-PEOPLE-2011-IRSES Project “Mobility between Europe and Argentina applying Logics to Systems” (MEALS) and the Laboratoire Internationale Associé “INFINIS”.

References

1. Areces, C.: Hybrid logics: The old and the new. In: Arrazola, X., Larrazabal, J. (eds.) Proc. of LogKCA-07. pp. 15–29. San Sebastian, Spain (2007)
2. Areces, C., ten Cate, B.: Hybrid logics. In: Blackburn, P., Wolter, F., van Benthem, J. (eds.) Handbook of Modal Logics, pp. 821–868. Elsevier (2006)
3. Areces, C., Figueira, D., Figueira, S., Mera, S.: The expressive power of memory logics. The Review of Symbolic Logic 4(2), 290–318 (2011)
4. van Benthem, J.: An essay on sabotage and obstruction. In: Mechanizing Mathematical Reasoning. pp. 268–276 (2005)
5. Blackburn, P., van Benthem, J.: Modal logic: A semantic perspective. In: Handbook of Modal Logic. Elsevier North-Holland (2006)
6. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic, Cambridge Tracts in Theoretical Comp. Scie., vol. 53. Cambridge University Press (2001)
7. Blackburn, P., Seligman, J.: Hybrid languages. Journal of Logic, Language and Information 4, 251–272 (1995)
8. Löding, C., Rohde, P.: Model checking and satisfiability for sabotage modal logic. In: Pandya, P., Radhakrishnan, J. (eds.) FSTTCS. LNCS, vol. 2914, pp. 302–313. Springer (2003)
9. Löding, C., Rohde, P.: Solving the sabotage game is PSPACE-hard. In: Mathematical Foundations of Computer Science 2003, Lecture Notes in Computer Science, vol. 2747, pp. 531–540. Springer, Berlin (2003)
10. Mera, S.: Modal Memory Logics. Ph.D. thesis, Univ. de Buenos Aires and UFR STMIA - Ecole Doctorale IAEM Lorraine Dép. de Form. Doct. en Informat. (2009)
11. Rohde, P.: On games and logics over dynamically changing structures. Ph.D. thesis, RWTH Aachen (2006)

Reasoning on Procedural Programs using Description Logics with Concrete Domains

Ronald de Haan

Technische Universität Wien

Abstract. Existing approaches to assigning semantics to procedural programming languages do not easily allow automatic reasoning over programs. We assign a model theoretic semantics to programs of a simple procedural language, by encoding them into description logics with concrete domains. This allows us to flexibly express several reasoning problems over procedural programs, and to solve them efficiently using existing reasoning algorithms, for certain fragments of the programming language. Furthermore, it allows us to explore for what further fragments of the programming language reasoning problems are decidable.

1 Introduction

There has been much research investigating the formal semantics of (procedural) programs. This has resulted in several different approaches (see for instance [5, 6]). Such approaches include defining several notions of semantics for programs (e.g. operational semantics, denotational semantics, axiomatic semantics) and investigating the relations between these different notions. These approaches to investigating the semantics of programming languages are very useful for proving the correctness of compilers, for instance. However, it is very difficult to automatically solve reasoning problems on procedural programs (e.g. deciding whether two programs are equivalent) using such kinds of semantics. Also, there is no unified approach to express and automatically solve various different reasoning problems, based on these existing notions of semantics of programming languages.

We propose an approach that solves these problems. We assign a model theoretic semantics to procedural programs. In particular, we will encode procedural programs as description logic knowledge bases. Description logics are widely used formalisms to reason about large and complex knowledge bases [1], and there are many efficient reasoners available for description logics. We will show that by means of this encoding of programs into description logic knowledge bases, we can express reasoning problems over programs in the description logic language, and in this way reduce such reasoning problems to description logic reasoning. This allows us to leverage the performance of existing reasoning algorithms for our reasoning problems. Furthermore, we illustrate how this approach can be helpful in identifying fragments of programming languages for which these reasoning problems are decidable, and analyzing the computational complexity of such reasoning.

The paper is structured as follows. We begin with briefly repeating how the syntax and the semantics of the prototypical description logic $\mathcal{ALC}$ can be extended with concrete domains, resulting in the description logic $\mathcal{ALC}(\mathcal{D})$, before defining a simple

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see `http://ceur-ws.org/`, 2012, pp. 32–42.

representative procedural programming language *While*. Then, we show how programs of this programming language can be encoded into the description logic $\mathcal{ALC}(\mathcal{D})$ in a semantically faithful way. Also, we illustrate how this allows us to encode reasoning problems over *While* programs into $\mathcal{ALC}(\mathcal{D})$ reasoning problems. Finally, we will discuss the benefit of this method, in combination with suggestions for further research.

2 Preliminaries

A concrete domain $\mathcal{D}$ is a pair $(\Delta_{\mathcal{D}}, \Phi_{\mathcal{D}})$, where $\Delta_{\mathcal{D}}$ is a set and $\Phi_{\mathcal{D}}$ a set of predicate names. Each predicate name $P \in \Phi_{\mathcal{D}}$ is associated with an arity n and an n-ary predicate $P^{\mathcal{D}} \subseteq \Delta_{\mathcal{D}}^n$. A predicate conjunction of the form

$$c = \bigwedge_{i \leq k} (x_0^{(i)}, \dots, x_{n_i}^{(i)}) : P_i,$$

where P_i is an n_i-ary predicate, for all $i \leq k$, and the $x_j^{(i)}$ are variables, is *satisfiable* iff there exists a function δ mapping the variables in c to elements of $\Delta_{\mathcal{D}}$ such that $(\delta(x_0^{(i)}), \dots, \delta(x_{n_i}^{(i)})) \in P_i^{\mathcal{D}}$, for all $i \leq k$. A concrete domain is called *admissible* iff its set of predicate names is closed under negation and contains a name $\top_{\mathcal{D}}$ for $\Delta_{\mathcal{D}}$, and the satisfiability problem for finite conjunctions of predicates is decidable.

An example of an admissible concrete domain is $N = (\mathbb{N}, \Phi_N)$, where Φ_N contains unary predicates $\top_N$ and ρ_n, and binary predicates ρ, for $\rho \in \{=, \neq, <, \leq, >, \geq\}$, and ternary predicates $+, -, \star, \backslash$, and the required negations of predicates. All predicates are given the usual interpretation (here ρ_n holds for a value x iff $x \; \rho \; n$ holds).

We will use the description logic $\mathcal{ALC}(\mathcal{D})$ [2], which is the extension of the prototypical description logic $\mathcal{ALC}$ [7] with concrete domains. For an overview of description logics with concrete domains, see [3].

We get the logic $\mathcal{ALC}(\mathcal{D})$, for a given concrete domain, by augmenting $\mathcal{ALC}$ with *abstract features* (roles interpreted as functional relations), *concrete features* (interpreted as a partial function from the logical domain into the concrete domain), and a new concept constructor that allows to describe constraints on concrete values using predicates from the concrete domain. More concretely, we can construct concepts $\exists u_1, \dots, u_k.P$ and $u\uparrow$, for $u, u_1, \dots, u_k$ paths and $P \in \Psi^{\mathcal{D}}$ a k-ary predicate. A path is a sequence $f_1 \dots f_n g$, where $f_1, \dots, f_n$ (for $n \geq 0$) are abstract features and g is a concrete feature.

Paths are interpreted as (partial) functions from the logical domain into the concrete domain, by taking the composition of the interpretation of their components. Concepts $\exists u_1, \dots, u_k.P$ are interpreted as the set of objects that are in the domain of the interpretation of all u_i, such that the resulting concrete objects satisfy the predicate P. Concepts $u\uparrow$ are interpreted as those objects that are not in the domain of the interpretation of u. For a complete, formal definition of the syntax and semantics of $\mathcal{ALC}(\mathcal{D})$, see [2, 3], for instance.

3 The Programming Language *While*

3.1 Syntax

We define the syntax of the simple representative procedural programming language *While* (defined and used for similar purposes in [5, 6]) with the following grammar (we use right-associative bracketing). Let $\mathcal{X}$ be a countably infinite set of variables. We let n range over values in $\mathbb{N}$[1], x over $\mathcal{X}$, a over expressions of category **AExp**, b over expressions of category **BExp**, and p over expressions of category **Prog**.

$$\begin{aligned}
a &::= n \mid x \mid a + a \mid a \star a \mid a - a\\
b &::= \top \mid \bot \mid a = a \mid a \leq a \mid \neg b \mid b\ \wedge b\\
p &::= x := a \mid \mathit{skip} \mid p; p \mid \textbf{if } b \textbf{ then } p \textbf{ else } p \mid \textbf{while } b \textbf{ do } p
\end{aligned}$$

We consider programs as expressions of category **Prog**. We denote the set of variables occurring in a program p with $Var(p)$, the set of subterms of p of category **BExp** with $Bool(p)$, and the set of subterms of p of category **AExp** with $Arith(p)$. Furthermore, with $cl(p)$ we denote the smallest set of programs such that:

- $cl(p)$ is closed under subterms, i.e., if $p_1 \in cl(p)$, $p_2 \in Sub(p)$ and $p_2 \in \mathbf{Prog}$, then $p_2 \in cl(p)$; and
- if **while** b **do** $p_1 \in cl(p)$, then also p_1; **while** b **do** $p_1 \in cl(p) \in cl(p)$.

3.2 Operational Semantics

Given a finite subset of variables $X \subseteq \mathcal{X}$, we define the set of states over X, denoted with $State_X$, as the set of total mappings $\mu : X \to \mathbb{N}$.

We define the function $\mathcal{B}^X$ that interprets expressions of category **BExp** as a function from $State_X$ to $\mathbb{B}$.

$$\begin{aligned}
\mathcal{B}^X(s, a_1\ \sigma\ a_2) &= \mathcal{A}^X(s, a_1)\ \sigma\ \mathcal{A}^X(s, a_2) \text{ for } \sigma \in \{=, \leq\}\\
\mathcal{B}^X(s, \neg b) &= \neg \mathcal{B}^X(s, b)\\
\mathcal{B}^X(s, b_1 \wedge b_2) &= \mathcal{B}^X(s, b_1) \wedge \mathcal{B}^X(s, b_2)
\end{aligned}$$

We define the function $\mathcal{A}^X$ that interprets expressions of category **AExp** as a function from $State_X$ to $\mathbb{N}$.

$$\begin{array}{ll}
\mathcal{A}^X(s, n) = n & \text{for } n \in \mathbb{N}\\
\mathcal{A}^X(s, x) = state(x) & \text{for } x \in X\\
\mathcal{A}^X(s, a_1\ \rho\ a_2) = \mathcal{A}^X(s, a_1)\ \rho\ \mathcal{A}^X(s, a_2) & \text{for } \rho \in \{+, \star\}\\
\mathcal{A}^X(s, a_1 - a_2) = \mathcal{A}^X(s, a_1) - \mathcal{A}^X(s, a_2) & \text{if } \mathcal{A}^X(s, a_1) - \mathcal{A}^X(s, a_2) \geq 0\\
\mathcal{A}^X(s, a_1 - a_2) = 0 & \text{if } \mathcal{A}^X(s, a_1) - \mathcal{A}^X(s, a_2) < 0
\end{array}$$

For $s \in State_X$, $x \in X$ and $n \in \mathbb{N}$, we define $s[x \mapsto n](y) = n$, if $x = y$, and $s[x \mapsto n](y) = s(y)$ if $x \neq y$.

For a program p and a set X such that $var(p) \subseteq X \subseteq \mathcal{X}$, we define the operational semantics as follows. We consider the transition system $(\Gamma, T, \Rightarrow)$, where $\Gamma = \{(q, s) \mid q \in cl(p), s \in State_X\}$, $T = State_X$, and $\Rightarrow\ \subseteq \Gamma \times (\Gamma \cup T)$.

We define the relation $\Rightarrow$ as the smallest relation such that for each $s \in State_X$, for each $a \in \mathbf{AExp}$, and for each $b \in \mathbf{BExp}$

- we have $(skip, s) \Rightarrow s$;
- we have $(x := a, s) \Rightarrow s[x \mapsto \mathcal{A}^X(a, s)]$;
- $(p_1, s) \Rightarrow (p_1', s')$ implies $(p_1; p_2, s) \Rightarrow (p_1'; p_2, s')$;
- $(p_1, s) \Rightarrow s'$ implies $(p_1; p_2, s) \Rightarrow (p_2, s')$;

[1] In this paper, we restrict ourselves to natural numbers, but the approach can be extended straightforwardly to other concrete domains.

– we have (**if** b **then** p_1 **else** $p_2, s) \Rightarrow (p_1, s)$, if $\mathcal{P}^X(b, s) = \top$;
– we have (**if** b **then** p_1 **else** $p_2, s) \Rightarrow (p_2, s)$, if $\mathcal{P}^X(b, s) = \bot$; and
– we have (**while** b **do** $p, s) \Rightarrow$ (**if** b **then** (p; **while** b **do** p) **else** $skip, s$).

Note that $\Rightarrow$ is deterministic, i.e., for any s, t, t', if $s \Rightarrow t$ and $s \Rightarrow t'$, then $t = t'$. We say that p terminates on s with outcome t if $(p, s) \Rightarrow^* t$ for $t \in T$. We say that p does not terminate on s if there is no $t \in T$ such that $(p, s) \Rightarrow^* t$. We take notice of the fact that if p does not terminate on s, then there is an infinite sequence $(p, s) \Rightarrow (p', s') \Rightarrow \ldots$ starting from (p, s).

3.3 Normal Form

In order to simplify our encoding of *While* programs into the description logic $\mathcal{ALC}(\mathcal{D})$ later, we define a notion of normal forms for *While* programs. Consider the following substitutions, preserving the operational semantics of programs, for a fresh variable x, for $\rho \in \{+, \star, -\}$, and $\pi \in \{=, \leq\}$:

$$\begin{array}{rll}
\varphi[x := a_1\ \rho\ a_2] \rightsquigarrow x_1 := a_1\ ;\ \varphi[x := x_1\ \rho\ a_2] & & \text{if } a_1 \notin \mathcal{X} \\
\varphi[x := a_1\ \rho\ a_2] \rightsquigarrow x_2 := a_2\ ;\ \varphi[x := a_1\ \rho\ x_2] & & \text{if } a_2 \notin \mathcal{X} \\
\varphi[a_1\ \pi\ a_2] \rightsquigarrow x_1 := a_1\ ;\ \varphi[x_1\ \pi\ a_2] & & \text{if } a_1 \notin \mathcal{X} \\
\varphi[a_1\ \pi\ a_2] \rightsquigarrow x_2 := a_2\ ;\ \varphi[a_1\ \pi\ x_2] & & \text{if } a_2 \notin \mathcal{X} \\
p \rightsquigarrow p; skip & & \text{if } p \text{ not of the form } q; skip
\end{array}$$

Using the transformations on programs given by the above substitutions, we can transform any program p to an operationally equivalent program p' such that the following holds:

– each subexpression of p' of category **AExp** is either of the form $x\ \rho\ y$, for $x, y \in \mathcal{X}$ and $\rho \in \{+, \star, -\}$, or of the form n for $n \in \mathbb{N}$;
– for each subexpression of p' of category **BExp** of the form $t\ \rho\ s$, for $\rho \in \{=, \leq\}$, holds $t, s \in \mathcal{X}$; and
– either $p' = skip$, or p' is of the form $e; skip$.

We will say that programs that satisfy this particular condition are in normal form.

4 Encoding Programs into $\mathcal{ALC}(\mathcal{D})$

We model programs in the language *While* using description logic and its model-theoretic semantics. The concrete (i.e. numerical) values in the programming language correspond to concrete values in the description logic. States are represented by objects, and programs are represented by concepts. We represent the execution of programs on states by a (functional) role $\mathsf{nextState}$. In particular, for a given program p with $Var(p) = \{x_1, \ldots, x_n\}$, we denote states $s \in State_{Var(p)}$ with objects that have concrete features $\mathsf{valueOf}_{x_i}$.

An execution of a program p will then be modelled by means of a $\mathsf{nextState}$ sequence of objects. The objects in this sequence represent the states occurring in the particular execution of the program. Whenever in this execution of the program there occurs a state s such that the execution is continued from this state s with the intermediate program p', the object corresponding to s is an element of the concept $C_{p'}$. Also, whenever this execution terminates, the final object in this sequence is an element of the concept C_{skip}.

4.1 Encoding

Take an arbitrary program p, i.e., an expression of category **Prog**. W.l.o.g., we assume p is in normal form. We define an $\mathcal{ALC}(\mathcal{D})$ TBox $\mathcal{T}^p$ as follows. We use concept names C_q for each $q \in cl(p)$, and concept names D_b for each $b \in Bool(p)$.

For each variable $x \in Var(p)$, we create a concrete feature $\mathsf{valueOf}_x$, and we require

$$\top \sqsubseteq \neg\mathsf{valueOf}_x\uparrow \tag{1}$$

We let $\mathsf{nextState}$ be an abstract feature and for C_{skip} we require:

$$C_{skip} \sqsubseteq \neg\exists\mathsf{nextState}.\top \tag{2}$$

For each $b \in Bool(p)$, we require the following, where x_1, x_2 range over X, and b_1, b_2 range over $Bool(p)$:

$$D_{x_1=x_2} \equiv \exists(\mathsf{valueOf}_{x_1})(\mathsf{valueOf}_{x_2}).= \tag{3}$$
$$D_{x_1\leq x_2} \equiv \exists(\mathsf{valueOf}_{x_1})(\mathsf{valueOf}_{x_2}).\leq \tag{4}$$
$$D_{\neg b_1} \equiv \neg D_{b_1} \tag{5}$$
$$D_{b_1\wedge b_2} \equiv D_{b_1} \sqcap D_{b_2} \tag{6}$$

Furthermore, we let $D_\top$ denote $\top$ and $D_\bot$ denote $\bot$. Then, for each $q \in cl(p)$ of the form $p_1; p_2$ we require the following for C_q, where p_1, p_2, q_1, q_2 range over $cl(p)$, x, y_1, y_2 range over X, a ranges over $Arith(p)$,

$$C_{(x:=a);p_2} \sqsubseteq \exists\mathsf{nextState}.\top \sqcap \exists\mathsf{nextState}.C_{p_2} \tag{7}$$
$$C_{skip;p_2} \sqsubseteq C_{p_2} \tag{8}$$
$$C_{(x:=n);p_2} \sqsubseteq \exists(\mathsf{nextState}\ \mathsf{valueOf}_x).=_n \tag{9}$$
$$C_{(x:=y);p_2} \sqsubseteq \exists(\mathsf{nextState}\ \mathsf{valueOf}_x)(\mathsf{valueOf}_y).= \tag{10}$$
$$C_{(x:=y_1+y_2);p_2} \sqsubseteq \exists(\mathsf{nextState}\ \mathsf{valueOf}_x)(\mathsf{valueOf}_{y_1})(\mathsf{valueOf}_{y_2}).+ \tag{11}$$
$$\begin{aligned} C_{(x:=y_1-y_2);p_2} \sqsubseteq\ & (\neg\exists(\mathsf{valueOf}_{y_2})(\mathsf{valueOf}_{y_1}).\leq \sqcup \\ & \exists(\mathsf{valueOf}_{y_1})(\mathsf{nextState}\ \mathsf{valueOf}_x)(\mathsf{valueOf}_{y_2}).+) \sqcap \\ & (\neg\exists(\mathsf{valueOf}_{y_2})(\mathsf{valueOf}_{y_1}).> \sqcup \\ & \exists(\mathsf{nextState}\ \mathsf{valueOf}_x).=_0) \end{aligned} \tag{12}$$
$$C_{(x:=y_1\star y_2);p_2} \sqsubseteq \exists(\mathsf{nextState}\ \mathsf{valueOf}_x)(\mathsf{valueOf}_{y_1})(\mathsf{valueOf}_{y_2}).\star \tag{13}$$
$$C_{(x:=a);p_2} \sqsubseteq \exists(\mathsf{valueOf}_y)(\mathsf{nextState}\ \mathsf{valueOf}_y).= \quad \text{for } y \neq x \tag{14}$$
$$C_{(\mathbf{if}\ b\ \mathbf{then}\ q_1\ \mathbf{else}\ q_2);p_2} \sqsubseteq (\neg D_b \sqcup C_{q_1;p_2}) \sqcap (D_b \sqcup C_{q_2;p_2}) \tag{15}$$
$$C_{(\mathbf{while}\ b\ \mathbf{do}\ q);p_2} \sqsubseteq (\neg D_b \sqcup C_{q;(\mathbf{while}\ b\ \mathbf{do}\ q);p_2}) \sqcap (D_b \sqcup C_{p_2}) \tag{16}$$

Notice that, in general, the TBox $\mathcal{T}^p$ is not acyclic, since Axioms (7), (8), (15) and (16) can together induce a cycle.

4.2 Correctness

In order to use the above encoding of a program p into an $\mathcal{ALC}(\mathcal{D})$ TBox $\mathcal{T}^p$ to reduce reasoning problems over programs into $\mathcal{ALC}(\mathcal{D})$ reasoning, we show the following correspondence between the operational semantics of p and the model theoretic semantics of $\mathcal{T}^p$.

Lemma 1. *For any program p, any X such that $Var(p) \subseteq X \subseteq \mathcal{X}$, any state $s \in State_X$, any $b \in Bool(p)$, and for any model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{T}^p$, we have that $d \in \Delta^{\mathcal{I}}$ and $(d, s(x_i)) \in \mathsf{valueOf}^{\mathcal{I}}_{x_i}$ for all $1 \leq i \leq n$ implies that $d \in C^{\mathcal{I}}_b$ iff $\mathcal{B}^X(b, s) = \top$.*

Proof (sketch). By induction on the structure of b. All cases follow directly from the fact that Axioms (3)-(6) hold.

Theorem 1. *For any program p, any X such that $Var(p) \subseteq X \subseteq \mathcal{X}$, any state $s \in State_X$ such that p terminates on s with outcome t, and for any model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{T}^p$ we have that $d \in C^{\mathcal{I}}_p$ and $(d, s(x_i)) \in \mathsf{valueOf}^{\mathcal{I}}_{x_i}$ for all $1 \leq i \leq n$ implies that $e \in C^{\mathcal{I}}_{skip}$ and $(e, t(x_i)) \in \mathsf{valueOf}^{\mathcal{I}}_{x_i}$ for all $1 \leq i \leq n$, for some $e \in \Delta^{\mathcal{I}}$.*

Proof. By induction on the length of the $\Rightarrow$-derivation $(p, s) \Rightarrow^k t$. Assume $d \in C^{\mathcal{I}}_p$ and $(d, s(x_i)) \in \mathsf{valueOf}^{\mathcal{I}}_{x_i}$ for all $1 \leq i \leq n$, for some $d \in \Delta^{\mathcal{I}}$. The base case $k = 0$ holds vacuously. In the case for $k = 1$, we know $p = skip$, since p is in normal form. Therefore, we know $s = t$, and thus $e = d$ witnesses the implication.

In the inductive case, we distinguish several cases. Case $p = skip; q$. We know $(p, s) \Rightarrow (q, s) \Rightarrow^{k-1} t$. Since $\mathcal{I}$ satisfies $\mathcal{T}^p$, by Axiom (8), we know $d \in C^{\mathcal{I}}_q$. The result now follows directly by the induction hypothesis.

Case $p = (x := a); q$. We know $(p, s) \Rightarrow (q, s') \Rightarrow^{k-1} t$, and $s' = s[x \mapsto \mathcal{A}^X(a, s)]$. Since $\mathcal{I}$ satisfies $\mathcal{T}^p$, by Axioms (7), (9)-(13) and (14), we know there must exist a $d' \in C^{\mathcal{I}}_q$ such that $(d', s'(x_i)) \in \mathsf{valueOf}^{\mathcal{I}}_{x_i}$ for all $1 \leq i \leq n$. Then by the induction hypothesis, the result follows directly.

Case $p = (\textbf{if}\, b\, \textbf{then}\, p_1\, \textbf{else}\, p_2); q$. Assume $\mathcal{B}^X(b, s) = \top$. Then $(p, s) \Rightarrow (p_1; q, s) \Rightarrow^{k-1} t$. By Lemma 1, we know $d \in D^{\mathcal{I}}_b$. Then, by the fact that Axiom (15) holds, we know $d \in C^{\mathcal{I}}_{p_1;q}$. The result now follows directly by the induction hypothesis. For $\mathcal{B}^X(b, s) = \bot$ an analogous argument holds.

Case $p = (\textbf{while}\ b\ \textbf{do}\ p_1); q$. Assume $\mathcal{B}^X(b, s) = \top$. Then $(p, s) \Rightarrow^2 (p_1; p, s) \Rightarrow^{k-2} t$. By Lemma 1, we know $d \in D^{\mathcal{I}}_b$. Then, by the fact that Axiom (16) holds, we know $d \in C_{p_1;p}$. The result now follows directly by the induction hypothesis.

If, however, in the same case holds $\mathcal{B}^X(b, s) = \bot$, then $(p, s) \Rightarrow^3 (q, s) \Rightarrow^{k-3} t$. By Lemma 1, we know $d \notin D^{\mathcal{I}}_b$. By the fact that Axiom (16) holds, we know $d \in C^{\mathcal{I}}_q$. The result now follows directly by the induction hypothesis.

Theorem 2. *For any program p, any X such that $Var(p) \subseteq X \subseteq \mathcal{X}$ any state $\{x_1 \mapsto c_1, \ldots, x_n \mapsto c_n\} = s \in State_X$ such that p does not terminate on s, there exists a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{T}^p$ such that for some $d \in \Delta^{\mathcal{I}}$ we have $d \in C^{\mathcal{I}}_p$, $(d, c_i) \in \mathsf{valueOf}^{\mathcal{I}}_{x_i}$, for all $1 \leq i \leq n$, and $C^{\mathcal{I}}_{skip} = \emptyset$.*

Proof. Since p does not terminate on s, we know there exists an infinite $\Rightarrow$-sequence $(p_i, s_i) \Rightarrow (p_{i+1}, s_{i+1})$, for $i \in \mathbb{N}$, where $(p_1, s_1) = (p, s)$. Consider the following interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}} = \{(p_i, s_i) \mid i \in \mathbb{N}\}$. For $q \in cl(p)$, we let $C^{\mathcal{I}}_q = \{(p_i, s_i) \mid p_i = q\}$. For $b \in Bool(p)$, we let $D^{\mathcal{I}}_b = \{(p_i, s_i) \mid i \in \mathbb{N}, \mathcal{P}^X(b, s) = \top\}$. For each $x \in X$, we let $\mathsf{valueOf}^{\mathcal{I}}_x = \{((p_i, s_i), s_i(x)) \mid i \in \mathbb{N}\}$. We let $\mathsf{nextState}^{\mathcal{I}} = \{((p_i, s_i), (p_{i+1}, s_{i+1})) \mid i \in \mathbb{N}\}$.

The definition of $\mathcal{I}$ implies that $C^{\mathcal{I}}_{skip} = \emptyset$. Assume $(p_k, s_k) \in C^{\mathcal{I}}_{skip}$. Then $p_k = skip$, and thus $(p_k, s_k) \Rightarrow s_k$, which contradicts our assumption of non-termination.

Clearly, $\mathcal{I}$ satisfies Axiom (1). Since $C^{\mathcal{I}}_{skip} = \emptyset$, $\mathcal{I}$ also satisfies Axiom (2). It is easy to verify, that by the definition of $D^{\mathcal{I}}_b$ we get that $\mathcal{I}$ satisfies Axioms (3)-(6).

To see that $\mathcal{I}$ satisfies Axioms (7)-(16), we take an arbitrary object (p_j, s_j) in the interpretation an arbitrary class $C^{\mathcal{I}}_q$, and we distinguish several cases. Consider

$p_j = skip; q$. Then by the constraints on $\Rightarrow$, $p_{j+1} = q$ and $s_{j+1} = s_j$. This witnesses that the subsumption in Axiom (8) holds.

Consider $p_j = (x := a); q$. Then by the constraints on $\Rightarrow$, $p_{j+1} = q$ and $s_{j+1} = s_j[x \mapsto \mathcal{A}^X(a, s_j)]$. By definition of $\mathcal{I}$, we know $((p_j, s_j), (p_{j+1}, s_{j+1})) \in \mathsf{nextState}^{\mathcal{I}}$. It is now easy to verify that the subsumptions in Axioms (7), (9)-(13) and (14) are satisfied.

Consider $p_j = (\textbf{if } b \textbf{ then } p'_1 \textbf{ else } p'_2); q$. Assume $\mathcal{B}^X(b, s_j) = \top$. Then $(p_j, s_j) \in D_b^{\mathcal{I}}$. Also, by the constraints on $\Rightarrow$, $p_{j+1} = p'_1; q$ and $s_{j+1} = s_j$. It is easy to verify that, in this case, the subsumption in Axiom (15) holds. The case for $\mathcal{B}^X(b, s_j) = \bot$ is completely analogous.

Consider $p_j = (\textbf{while } b \textbf{ do } p'); q$. If $\mathcal{B}^X(b, s_j) = \top$, then $(p_j, s_j) \in D_b^{\mathcal{I}}$ and $p_{j+1} = p'; p_j$ and $s_{j+1} = s_j$. If $\mathcal{B}^X(b, s_j) = \bot$, then $(p_j, s_j) \notin D_b^{\mathcal{I}}$ and $p_{j+1} = skip; q$ and $s_{j+1} = s_j$. It is easy to verify that, in either case, the subsumption in Axiom (16) holds.

Theorem 3. *For any program p, any X such that $Var(p) \subseteq X \subseteq \mathcal{X}$ any state $\{x_1 \mapsto c_1, \ldots, x_n \mapsto c_n\} = s \in State_X$ such that p terminate on s, there exists a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\mathcal{T}^p$ such that for some $d \in \Delta^{\mathcal{I}}$ we have $d \in C_p^{\mathcal{I}}$, $(d, c_i) \in \mathsf{valueOf}_{x_i}^{\mathcal{I}}$, for all $1 \leq i \leq n$.*

Proof (Sketch). We know there exists $(p, s) \Rightarrow^k (p', s') \Rightarrow t$. Analogously to the proof of Theorem 2, we can construct a model $\mathcal{I}$ from the sequence $(p, s) \Rightarrow^k (p', s')$. By similar arguments to those in the proof of Theorem 2 it follows that $\mathcal{I} \models \mathcal{T}^p$. Then, $(p, s) \in C_p^{\mathcal{I}}$ witnesses the further constraints on $\mathcal{I}$.

Note that the syntax and operational semantics of *While* can be adapted to various concrete domains with varying operators. The encoding into $\mathcal{ALC}(\mathcal{D})$ can be adapted correspondingly, and a corresponding correlation between the operational semantics and the model theoretic semantics can be proven.

4.3 Example

To illustrate the method described above, of encoding *While* programs into $\mathcal{ALC}(\mathcal{D})$ TBoxes, we will consider an example. Let p_0 be the following program in normal form, that computes the factorial of the value stored in variable x and outputs this in variable y. Note that variable z is simply used to refer to the constant value 1.

$$p_0 = \quad (y := 1; z := 1;\ \textbf{while } x > z \textbf{ do } (y := y \star x;\ x := x - z); skip)$$

Furthermore, we will use the following abbreviations to refer to subprograms of p_0.

$$\begin{aligned}
p_1 &= (z := 1;\ \textbf{while } x > z \textbf{ do } (y := y \star x;\ x := x - z); skip) \\
p_2 &= (\textbf{while } x > z \textbf{ do } (y := y \star x;\ x := x - z); skip) \\
p_3 &= (y := y \star x;\ x := x - z) \\
p_4 &= (x := x - z)
\end{aligned}$$

We can now construct the $\mathcal{ALC}(\mathcal{D})$ TBox $\mathcal{T}^{p_0}$, in the fashion described above.

$$
\begin{array}{rrcll}
\mathcal{T}^{p_0} = \{ & \top & \sqsubseteq & \neg\mathsf{valueOf}_x\uparrow, & \\
& \top & \sqsubseteq & \neg\mathsf{valueOf}_y\uparrow, & \\
& \top & \sqsubseteq & \neg\mathsf{valueOf}_z\uparrow, & \\
& C_{skip} & \sqsubseteq & \neg\exists\mathsf{nextState}.\top, & \\
& D_{x<z} & \equiv & \exists(\mathsf{valueOf}_x)(\mathsf{valueOf}_z).<, & \\
& C_{(y:=1);p_1} & \sqsubseteq & \exists\mathsf{nextState}.\top \sqcap \exists\mathsf{nextState}.C_{p_1}, & \\
& C_{(z:=1);p_2} & \sqsubseteq & \exists\mathsf{nextState}.\top \sqcap \exists\mathsf{nextState}.C_{p_2}, & \\
& C_{(y:=1);p_1} & \sqsubseteq & \exists(\mathsf{nextState}\ \mathsf{valueOf}_y).=_1, & \\
& C_{(y:=1);p_1} & \sqsubseteq & \exists(\mathsf{valueOf}_x)(\mathsf{nextState}\ \mathsf{valueOf}_x).=, & \\
& C_{(y:=1);p_1} & \sqsubseteq & \exists(\mathsf{valueOf}_z)(\mathsf{nextState}\ \mathsf{valueOf}_z).=, & \\
& C_{(z:=1);p_2} & \sqsubseteq & \exists(\mathsf{nextState}\ \mathsf{valueOf}_z).=_1, & \\
& C_{(z:=1);p_2} & \sqsubseteq & \exists(\mathsf{valueOf}_x)(\mathsf{nextState}\ \mathsf{valueOf}_x).=, & \\
& C_{(z:=1);p_2} & \sqsubseteq & \exists(\mathsf{valueOf}_y)(\mathsf{nextState}\ \mathsf{valueOf}_y).=, & \\
& C_{p_2} & \sqsubseteq & (\neg D_{x<z} \sqcup C_{p_3;p_2}) \sqcap (D_{x<z} \sqcup C_{skip}), & (\dagger) \\
& C_{(y:=y\star x);p_4;p_2} & \sqsubseteq & \exists\mathsf{nextState}.\top \sqcap \exists\mathsf{nextState}.C_{p_4;p_2}, & (\dagger) \\
& C_{(x:=x-z);p_2} & \sqsubseteq & \exists\mathsf{nextState}.\top \sqcap \exists\mathsf{nextState}.C_{p_2}, & (\dagger) \\
& C_{(y:=y\star x);p_4;p_2} & \sqsubseteq & \exists(\mathsf{nextState}\ \mathsf{valueOf}_y)(\mathsf{valueOf}_y)(\mathsf{valueOf}_x).\star, & \\
& C_{(y:=y\star x);p_4;p_2} & \sqsubseteq & \exists(\mathsf{valueOf}_x)(\mathsf{nextState}\ \mathsf{valueOf}_x).=, & \\
& C_{(y:=y\star x);p_4;p_2} & \sqsubseteq & \exists(\mathsf{valueOf}_z)(\mathsf{nextState}\ \mathsf{valueOf}_z).=, & \\
& C_{(x:=x-z);p_2} & \sqsubseteq & (\neg\exists(\mathsf{valueOf}_z)(\mathsf{valueOf}_x).\leq \sqcup & \\
& & & \exists(\mathsf{valueOf}_x)(\mathsf{nextState}\ \mathsf{valueOf}_x)(\mathsf{valueOf}_z).+) \sqcap & \\
& & & (\neg\exists(\mathsf{valueOf}_z)(\mathsf{valueOf}_x).> \sqcup & \\
& & & \exists(\mathsf{nextState}\ \mathsf{valueOf}_x).=_0), & \\
& C_{(x:=x-z);p_2} & \sqsubseteq & \exists(\mathsf{valueOf}_y)(\mathsf{nextState}\ \mathsf{valueOf}_y).=, & \\
& C_{(x:=x-z);p_2} & \sqsubseteq & \exists(\mathsf{valueOf}_z)(\mathsf{nextState}\ \mathsf{valueOf}_z).= & \}
\end{array}
$$

Note that $\mathcal{T}^{p_0}$ is not acyclic, since there are inclusion axioms (a) with C_{p_2} as lhs and $C_{p_3;p_2}$ in the rhs, (b) with $C_{p_3;p_2}$ as lhs and $C_{p_4;p_2}$ in the rhs, and (c) with $C_{p_4;p_2}$ as lhs and C_{p_2} in the rhs. These axioms are marked with the symbol †.

It is straightforward to construct models of the TBox $\mathcal{T}^{p_0}$. One can simply take an execution of the program p_0, and transform this execution into a model of $\mathcal{T}^{p_0}$ according to the intuition behind the encoding described above.

5 Reasoning Problems

Theorems 1, 2 and 3 allow us to use the encoding of *While* programs into $\mathcal{ALC}(\mathcal{D})$ to reduce several reasoning problems over *While* programs to reasoning problems over $\mathcal{ALC}(\mathcal{D})$. For instance, termination of a program p reduces to unsatisfiability of the ABox $\mathcal{A}^p = \{o_1 : C_p\}$ with respect to the TBox $\mathcal{T}^p \cup \{C_{skip} \sqsubseteq \bot\}$.

Also, we are able to encode abduction problems over *While* problems in the description logic $\mathcal{ALCO}(\mathcal{D})$, which is $\mathcal{ALC}(\mathcal{D})$ extended with nominals. The question what input states for a program p could have led to the (partial) output state s, for $dom(s) \subseteq Var(p)$, reduces to finding models for $\mathcal{A}^p = \{i : C_p, o : C_{skip}\} \cup \{(o, s(x)) : \mathsf{valueOf}_x \mid x \in dom(s)\}$ and $\mathcal{T}^p$, where C_{skip} is required to be a nominal concept.

Another example is checking whether two (terminating) programs p_1 and p_2 are equivalent. Without loss of generality, we can assume $Var(p_1) = Var(p_2)$. This equivalence check can be reduced to the problem of $\mathcal{ALCO}(\mathcal{D})$ unsatisfiability of the ABox $\mathcal{A}^{p_1,p_2} = \{o : C_{p_1}, o : C_{p_2}, s : C_{test}\}$ with respect to the TBox $\mathcal{S}^{p_1} \cup \mathcal{S}^{p_2} \cup \mathcal{T}^{eq}$, where $\mathcal{S}^{p_i}$ is

$\mathcal{T}^{p_i}$ with C_{skip} replaced by C^i_{skip}, and $\mathcal{T}^{eq} = \{C_{test} \equiv (\exists.(\mathsf{res}_1\ \mathsf{valueOf}_{x_1})(\mathsf{res}_2\ \mathsf{valueOf}_{x_1}).\neq \sqcup \cdots \sqcup \exists.(\mathsf{res}_1\ \mathsf{valueOf}_{x_n})\ (\mathsf{res}_2\ \mathsf{valueOf}_{x_n}).\neq)\ \sqcap\ \exists \mathsf{res}_1.C^1_{skip}\ \sqcap\ \exists \mathsf{res}_2.C^2_{skip}\}$, for res_1, res_2 abstract features, and C^1_{skip}, C^2_{skip} and C_{test} nominal concepts.

Naturally, this approach allows us to encode more intricate reasoning problems over *While* programs into description logic reasoning problems. Description logic offers us a very flexible formalism to express a variety of reasoning problems over *While* programs.

5.1 Decidability

A bit of care has to be taken with this powerful and general approach. The problems we consider generally balance on the bounds of decidability.

A concrete domain $\mathcal{D}$ is called arithmetic if its values contain the natural numbers, and it contains predicates for equality, equality with zero and incrementation. Unfortunately, satisfiability of $\mathcal{ALC}(D)$ concepts for arithmetic concrete domains D with respect to general TBoxes is undecidable [4]. So, for many cases, our approach doesn't directly result in a decision procedure.

This is no surprise, however. We know that we cannot decide equivalence of *While* programs in general. For instance for the concrete domain $\mathbb{Z}$ (with addition and equality) we can easily encode the undecidable problem of whether a given Diophantine equation $\Phi(x_1, \ldots, x_n) = 0$ has an integer solution (the subject of Hilbert's Tenth Problem) as a reasoning problem over *While* programs. Similarly, it can be proven that equivalence of *While* programs with concrete domain $\mathbb{N}$ is undecidable.

Nevertheless, several decidable fragments of the *While* language can be obtained by either restricting the concrete domains or by forbidding statements of the form (**while** b **do** p). By forbidding these **while**-statements, we end up with acyclic TBoxes. We know that reasoning with respect to acyclic TBoxes is decidable for $\mathcal{ALC}(\mathcal{D})$. Restricting the concrete domain (and keeping **while**-statements), does not change the fact that we are dealing with general TBoxes. In this case, the concrete domain needs to be restricted quite severely, to get decidability.

5.2 Examples

We can use the example from Section 4.3 to illustrate how to encode several reasoning problems over *While* programs into $\mathcal{ALC}(\mathcal{D})$ reasoning problems. For instance, we can encode the problem of checking whether p_0 is terminating as the unsatisfiability problem of the ABox $\{o_1 : C_{p_0}\}$ with respect to the TBox $\mathcal{T}^p \cup \{C_{skip} \sqsubseteq \bot\}$.

Using the expressivity of the description logic $\mathcal{ALC}(\mathcal{D})$, we can in fact express a variety of different semantic properties to be checked automatically. For instance, in this fashion, we can express the problem whether p_0 does not terminate with an output value for y that is less than or equal to 20 for all input values for x that are greater than or equal to 4. This problem can be reduced to the unsatisfiability of the ABox $\{o_1 : C_{p_0} \sqcap \exists(\mathsf{hasValue}_x).\geq_4\}$ with respect to the TBox $\mathcal{T}^p \cup \{C_{skip} \sqsubseteq \exists(\mathsf{hasValue}_y).\leq_{20}\}$.

These examples of expressing reasoning problems in the $\mathcal{ALC}(\mathcal{D})$ language illustrate the flexibility we get in expressing different reasoning problems. We can reduce the decision of any semantic property of *While* programs that is expressible using the modelling of *While* programs in the $\mathcal{ALC}(\mathcal{D})$ language to reasoning on $\mathcal{ALC}(\mathcal{D})$.

6 Further Research

The results presented in this paper are only the beginning of a larger inquiry investigating the possibilities and bounds of approaching automated reasoning over (procedural) programming languages by means of assigning model-theoretic semantics to programs. We suggest a number of directions for further research needed to get a better understanding of the topic.

Similar encodings of programming languages into the description logic $\mathcal{ALC}(\mathcal{D})$ could also be devised also for other procedural programming languages, as well as for declarative programming languages. It needs to be investigated to what extent this method is extendable to such other languages. Suitable programming languages for which this could be investigated as a next step include extensions of the programming language *While* with language constructs for nondeterminism or parallellism, procedural programming languages that are based on **goto**-statements rather than **while**-statements, and simple, representative functional and logic programming languages that operate on similar domains (e.g. numerical values). Once the method presented in this paper has been applied to such languages, it will also be possible to investigate to what extent reasoning problems on multiple programs of different programming languages (e.g. the equivalence problem of a *While* program and a program of a **goto**-based language) can be encoded in this framework.

Another important direction for further research is identifying larger fragments of the programming language *While* for which reasoning problems on programs such as the ones we considered in this paper are decidable. By encoding programs into description logics, we get a conceptually simpler setting to investigate such questions. Programming languages can have a variety of different constructs behaving in various ways. Reasoning over such diverse structures can get rather messy and complex. On the other hand, practically all reasoning problems for description logics can be reduced to finding models for description logic knowledge bases (i.e. reduced to the satisfiability problem). The problem of finding such models is conceptually simple, and the semantic definition of the description logic language guides the search for models. Such a conceptually simpler setting might make it easier to identify decidable fragments. In addition to this conceptual simplification of the problem, we could use results and techniques from the field of description logic when investigating the decidability of such fragments. For those fragments of the programming languages that lead to decidable reasoning problems, we can investigate the computational complexity of these reasoning problems.

7 Conclusions

We assigned a model theoretic semantics to programs of the simple procedural language *While*, by encoding them into the description logic $\mathcal{ALC}(\mathcal{D})$. This allowed us to express a variety of reasoning problems over *While* programs using the expressivity of the description logic $\mathcal{ALC}(\mathcal{D})$. Furthermore, for a number of restricted fragments of the programming language *While*, this directly results in a method of solving such reasoning problems using existing (description logic) algorithms. Furthermore, this encoding leads to a new approach of exploring what fragments of the programming language allow for decidable reasoning over programs. Further research includes a further characterization of fragments of the language that allow decidable reasoning, and extending this approach to different procedural (and declarative) programming languages.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press (2003)
2. Baader, F., Hanschke, P.: A scheme for integrating concrete domains into concept languages. In: Proceedings of the 12th international joint conference on Artificial intelligence - Volume 1. pp. 452–457. IJCAI'91, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1991)
3. Lutz, C.: Description logics with concrete domains—a survey. In: Advances in Modal Logics Volume 4. King's College Publications (2003)
4. Lutz, C.: NExpTime-complete description logics with concrete domains. ACM Trans. Comput. Logic 5(4), 669–705 (Oct 2004)
5. Nielson, F., Nielson, H.R., Hankin, C.: Principles of Program Analysis. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1999)
6. Nielson, H.R., Nielson, F.: Semantics with Applications: A Formal Introduction. John Wiley & Sons, Inc., New York, NY, USA (1992)
7. Schmidt-Schaubß, M., Smolka, G.: Attributive concept descriptions with complements. Artif. Intell. 48(1), 1–26 (Feb 1991)

Small Steps in Heuristics for the Russian Cards Problem Protocols

Ignacio Hernández-Antón

Dept. Philosophy, Logic and Philosophy of Science
University of Seville (Spain)
iha(at)us.es

Abstract. This work[1] presents a couple of algorithmic techniques applied to the Russian Cards Problem. This problem represent an idealized scenario where Dynamic Epistemic Logic [4, 5, 8] plays an important role in secure communications analysis. This logic is in a *lower layer* below the protocol desing tasks acting as a specification and verification formal tool. This work focusses not on the logical aspects but rather on the protocol design/searching problem. It is important to have present the epistemic formal notions of that logic to fully understand the epistemic functions developed here. Secure protocols found in this card game scenario may be a good starting point for developing some aspects towards unconditional information security. Hill-Climbing and Genetic Algorithms are studied as searching techniques aimed to find optimal announcements that can be part of a secure communication protocol. Some problems as the dimension and complexity of the search space are pointed out.

Keywords: Evolutionary computing, genetic algorithm, epistemic protocols, information security.

1 Introduction

We present an algorithmic approach to search unconditionally secure protocols of communicating agents within the well-known Russian Cards Problem scenario. In public/private key approaches as AES (Advanced Encription Standard), RSA (Rivest, Shamir, Adleman technique), DSA (Digital Signature Algorithm) or ECC (Elliptic Curve Cryptography), secret information if safeguarded because of the high complexity of computational operations to decrypt the message, for instance, RSA uses the IFD, the Integer Factorization Problem [10, 11]. Instead of the computational hardness for security assurance, we focus on an information-based approach to protocol design in this work. We model the communicating agents as cards players and the communicating secret is the ownership of the cards in the game. In this scenario it is possible to define good protocols regardless of the computational complexity of cryptographic techniques [4, 5, 7].

[1]This work shares some content with [9] but add the Hill-Climber section and some conlusion remarks important to understand the nature of the search and the solution space.

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see http://ceur-ws.org/, 2012, pp. 43–53.

We just study the security aspects of communication in order to avoid eavesdropping; other models of attacker are intentionally ommited here, we are aware of the importance of considering them for future works in order to gain robustness though. We study how to guarantee the privacy of the message which should only be shared by those principal agents we legitimated. This will occur regardless of other agents listening passively to the information passed. There is a logical approach where this problem is formalized using dynamic epistemic logic [4, 5].

We employ genetic algorithms to search for card deal protocols [12, 16, 17]. Genetic algorithms are a bio-inspired family of computational techniques which have natural evolution as a model for encoding some critical aspects of solutions as chromosomes-like data structures [3, 15]. An initial population is transformed by genetically inspired operations in order to produce new generations. The most important ones are **selection** (of the fittest), **crossing-over**, and **mutation**, the latter ones add variety to the search producing the exploration of the solution space. We use JAVA to specify the russian cards problem and a genetic engine called *jgap* to search for protocols. For further details on the genetic engine see `http://jgap.sourceforge.net/`.

2 The Russian Cards Problem

> *From a pack of seven known cards (for instance 0-6) two players (a, b) each draw three cards and a third player (c) gets the remaining card. How can the two first players (those with three cards) openly (publicly) inform each other about their cards without cyphering the messages and without the third player learning from any of their cards who holds it?*

Although this presentation of the problem has 7 cards in the stack and the deal distribution is 3.3.1 (agent a draws three cards, b draws three and c draws just one), one may consider other scenarios, e.g., a 10-cards stack with a 4.4.2 deal. To become familiar with a basic game scenario, let us call agents a, b and c. The cards are named 0, ..., 6. Deals distributions (size) are noted as integer strings, for instance 3.3.1. Legitimated principals are a and b while c is the eavesdropper. We suppose the actual deal is 012.345.6 w.l.o.g. Communication is done by truthful and public announcements, see [13, 14]. A public announcement for an agent a is a set of a's possible set of hands (we use a simplified notation to denote set of hands, e.g., $\{012, 125, 156\}$ instead of $\{\{0, 1, 2\}, \{1, 2, 5\}, \{1, 5, 6\}\}$).

A secure announcement in this scenario should keep c ignorant throughout the entire communication and guarantee common knowledge of this agent's ignorance. According to that approach, a good protocol comprises an announcement sequence that verifies that:

Informativeness 1: Principals a and b know each other cards.
Informativeness 2: It is common knowledge, at least for the principals, that they do know each other's cards.
Security 1: The intruder, c, remains ignorant always.
Security 2: It is common knowledge for all agents that the intruder remains ignorant.
Knowledge-based: Protocol steps are modelled as public announcements.

The reason we split the informativeness and security requirements into two parts can be found in [5]. A protocol is then a finite sequence of instructions determining sequences of announcements. Each agent chooses an announcement conditional on that agent's knowledge. The protocol is assumed to be common knowledge among all agents following Kerckhoff's principle.

One knowledge-based protocol that consitutes a solution for the riddle is as follows. Suppose that the actual deal of cards is that agent a has $\{0, 1, 2\}$, b has $\{3, 4, 5\}$ and c has $\{6\}$.

- a says: My hand is one of $\{012, 046, 136, 145, 235\}$.
- Then, b says: c's card is 6.

After this, it is common knowledge to the three agents that a knows the hand of b, that b knows the hand of a, and that c is ignorant of the ownership of any card not held by itself. For further details on the notion of common knowledge see [6].

We can also see these two sequences as the execution of a knowledge-based protocol. Given a's hand of cards, there is a (non-deterministic) way to produce her announcement, to which b responds by announcing c's card. The protocol is knowledge-based, because the agents initially only know their own hand of cards, and have public knowledge of the deck of cards and how many cards each agent has drawn from the pack. It can be viewed as an *unconditionally secure* protocol, as c cannot learn any of the cards of a and b, no matter their computational resources. The security is therefore not conditional on the high complexity of some computation.

3 Russian Hill-Climbers

Let's begin with a simple metaheuristic technique called Hill-Climbing. It is a very simple algorithm that give us the oportunity to study the primitive behaviour of the problem representation and the nature of the search space. The Genetic Algorithms can be partially viewed as a sophistication of hill-climbing as optimization technique. The latter constitutes a point of reference because, being simpler, it can give clues about the need of using more sophisticated (hence, more resource-consuming) algorithms. Hill-Climbing is related to gradient ascent, but it does not require you to know the strength of the gradient or even its direction: you just iteratively test new candidate solutions in the region of your current candidate and adopt the new ones if they are better. This enables you to climb up the hill until you reach a local optimum. To have a first flavor about how this technique would work on the russian card problem, we implemented a specific Hill-Climber (see Algorithm 1) with the next parameters:

– 100 executions over 3.3.1.
– 100000 evaluations allowed

Algorithm 1 Russian Hill-Climber 1 (RHC_1)

$ANN \leftarrow$ All-hand-in initialization
repeat
    $ANN' \leftarrow BFM_1(ANN)$
    if $Quality(ANN') > Quality(ANN)$ **then**
        $ANN \leftarrow ANN'$
    end if
until 100000 generations or good protocol found
return $\boldsymbol{x}$

Algorithm 2 Single Bit Flip Mutator (BFM_1)

$h \leftarrow Random(PossibleHandsForANN)$
if $h \in ANN$ **then**
    $ANN' \leftarrow$ Remove h from ANN
else
    $ANN' \leftarrow$ Include h in ANN
end if
return ANN'

The initialization chosen for RHC_1 produces an announcement with all hands in and a single Bit Flip Mutator BFP_1 described in Algorithm 2. A random hand from all possible hand to be included into the announcement is selected to proceed as the `if` conditional states removing it if present or including it otherwise.

After 100 executions for 3.3.1 we obtained the next results: the average number of protocol evaluations (in red) was 39288.16 with a minimum of 1716 and a maximum of 130321. Find attached the whole experiment graphics in Figure 1, see in the horizontal axis the excution number and in the vertical one the generation where the RHC_1 found a global optimal solution.

4 Modelling cards protocols with Genetic Algorithms

The final objective of modelling cards protocols with genetic algorithms is to find protocols for card deal sizes where an analytic solution is lacking. We reinvestigate protocols for 3.3.1 with genetic algorithms in order to study the behaviour of the relationship between the problem representation and the fitness function used. This way we will be able to polish them for more complex scenarios. The use of genetic algorithms requires to satisfy two conditions: possible solutions can be represented by chromosomes and an evaluation function can be defined in order to assign a value to each chromosome. Regarding this encoding representation we observe that the set of possible hands of an agent can be arranged in lexicographic order, e.g., for Russian

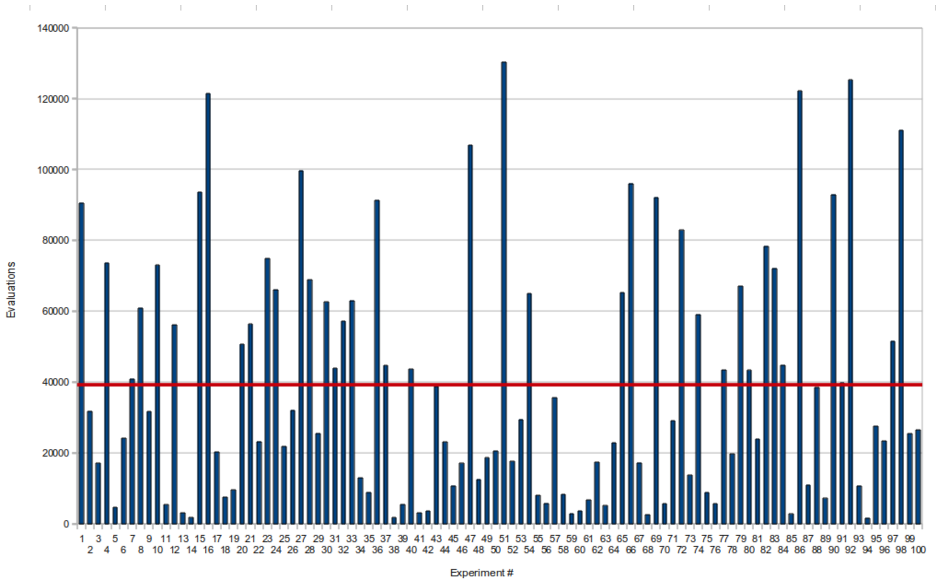
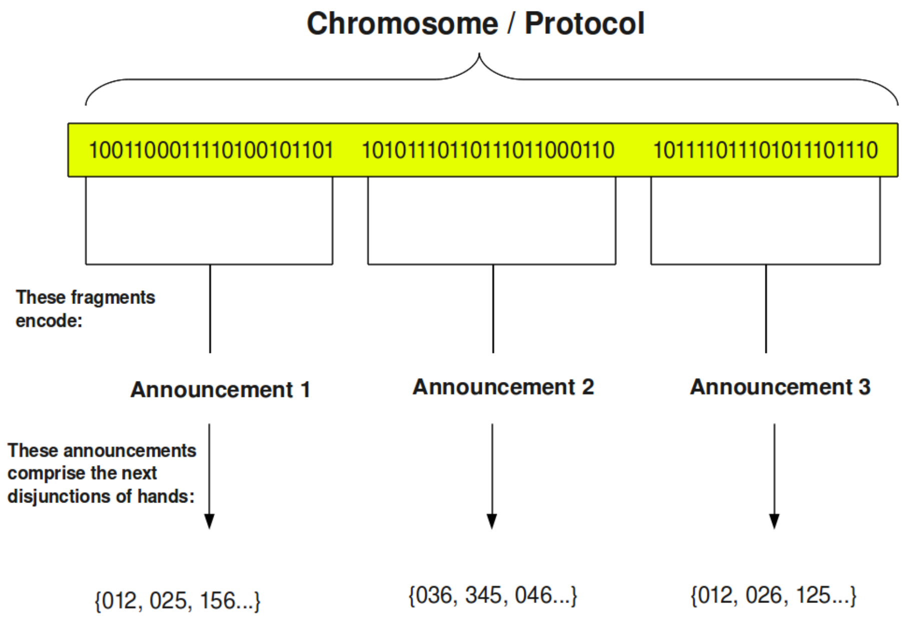


Fig. 1. 100 runs of RHC_1 over 3.3.1; BFM_1

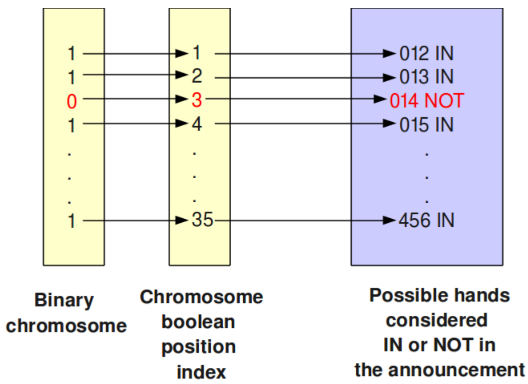
Cards, the 35 hands are listed as $012, 013, \ldots, 456$. Then we assign a binary gene to each possible hand. An announcement is then represented by a 35-bitstring. To illustrate a mapping like this see Figure 2 (a), we can encode several announcements into one chromosome as in Figure 2 (b) representing this way an composite protocol. In the 3.3.1 case as it is proved that there is a minimal protocol where just one announcement is needed, we restrict the encoding representation to just agent a announcement, but if several announcements are required, this representation is flexible enough to be adapted to it.

Algorithm 3 Genetic algorithm schema

$t \leftarrow 0$
Population initialization $P(0) : \{p_1(0), \ldots p_k(0)\}$
Evaluation of $P(0)$: $\Phi(p_1(0)), \ldots, \Phi(p_k(0))$
while $\theta(P(t) \neq \text{true})$ **do**
    Recombination $P'(t) \leftarrow r(P(t))$
    Mutation $P'(t) \leftarrow m(P'(t))$
    Evaluation $P'(t) : \Phi(p'_1(0)), \ldots, \Phi(p'_m(0))$
    Selection $P(t+1) \leftarrow s(P'(t))$
end while
return $P(t+s)$



(a) Mapping announcements



(b) Allele decoding

Fig. 2. Chromosome general structure

To evolve the population a fitness function assigns to each announcement a value. As the protocols are knowledge-based, this function needs some epistemic aspects to be considered (implemented). As we are looking for a two-step protocol [2], we only need one announcement. The fitness function evaluates possible announcements. Note that if D is the set of cards, these announcements are elements of $\mathcal{P}(\mathcal{P}(D))$ with certain properties.

The first epistemic function is $compCardsGivenAnnounce(Ann, Hand)$ which returns, for a given announcement Ann, the set of possible hands that an agent having

$Hand$ considers for the agent making the announcement:

$$compCardsGivenAnnounce : \mathcal{P}(\mathcal{P}(D)) \times \mathcal{P}(D) \mapsto \mathcal{P}(\mathcal{P}(D))$$
$$compCardsGivenAnnounce(Ann, Hand) = \{h \in Ann \mid h \cap Hand = \emptyset\}$$

Now we define $whatAgentLearnsFromAnnounce(Ann, Hand)$ that returns the set of cards that an agent having $Hand$ learns from the agent. The function is defined as:

$$whatAgentLearnsFromAnnounce : \mathcal{P}(\mathcal{P}(D)) \times \mathcal{P}(D) \mapsto \mathcal{P}(D)$$
$$whatAgentLearnsFromAnnounce(Ann, Hand) = \bigcap compCardsGivenAnnounce(Ann, Hand)$$

As an example, consider that, in the 3.3.1 setting with deal 012.345.6, a announces that her hand is one of $\{012, 016, 234\}$. Then, a's compatible hands for b are $\{012, 016\}$ and b learns that a has $\{0, 1\}$ in this case. However an agent may learn cards not only from the agent making the announcement, but also from the remaining one. In our example, as c's hand is $\{6\}$, c can also apply the previous function to learn that a holds card 2. But not that a's compatible hands for c are $\{012, 234\}$, so c learns that b holds cards 5. So c learns two cards, one from a and the other from b. The following function calculates this:

$$howManyAgentLearnsFromDeal : \mathcal{P}(\mathcal{P}(D)) \times \mathcal{P}(D) \mapsto \mathbb{N}$$
$$howManyAgentLearnsFromDeal(Ann, Hand) = |whatAgentLearnsFromAnnounce(Ann, Hand)| + |D| - |Hand| - |\bigcup compCardsGivenAnnounce(Ann, Hand)|$$

In the fitness function we have to consider not only one deal (as that in our example) but all possible deals, in order to ensure that the announcement is unconditionally secure. The following function calculates, for a given announcement Ann and an agent having x cards, the minimum number of cards that the agent learns from the deal, by considering all possible agent's hands consistent with Ann:

$$minLearn : \mathcal{P}(\mathcal{P}(D)) \times \mathbb{N} \mapsto \mathbb{N}$$
$$minLearn(Ann, x) = min(\{howManyAgentLearnsFromDeal(Ann, H) \mid H \subseteq D, |H| = x, compCardsGivenAnnounce(Ann, H) \neq \emptyset\})$$

In the same way, we can define a function $maxLearn$ to calculate the maximum number of cards that an agent may learn. Then, the fitness function, given that the size of the deal is $n.m.k$, is defined as:

$$fitness : \mathcal{P}(\mathcal{P}(D)) \times \mathbb{N} \times \mathbb{N} \mapsto \mathbb{Z}$$
$$fitness(Ann, wI, wS) = wI \cdot minLearn(Ann, m) - wS \cdot maxLearn(Ann, k)$$

The two values wI and wS are two natural numbers which measure the relevance of b's knowledge and c's ignorance, respectively. Obviously, the maximum value of the fitness function is obtained when b learns $n+k$ cards (all a's and c's cards) and c learns nothing. Then, the value of the fitness function is $wI(n + k)$. The minimum value is $-wS(n + m)$.

In the Java implementation we work with *jgap*'s chromosomes, that are sequences of binary values. Prior to apply the fitness function we need to decode the binary sequence into an announcement, as explained above. As the fitness function is only allowed to return non-negative values, $S(n + m)$ is added to every result of our previous $fitness$ function.

5 Weighing informativeness and ignorance

It seems reasonable to suppose that there is a correspondence between weighing informativeness and security, on the one hand, and the efficiency in time of the search, on the other hand. In this section we demonstrate by statistical analysis that this is not the case.

Weighing the fitness function with wI and wS we can influence the search, prioritizing one aspect or the other. If we consider informativeness more important than security, the algorithm lets survise to the next generation announcements where c could learn some cards. But if we focus on security and wish to avoid that situation, we will prioritize the security weight in order to devaluate the announcement where c can learn.

Apart from the number of experiments, the size of the populations, and the number of generations set in the algorithm, the different weight combinations also allow us to create different scenarios and configurations. Those will be useful to collect data for statistical analysis. We wish to determine if there is a combination of weights that makes the search go faster.

The results of the algorithm search are stored and a statistic data analyzer goes through them in order to find relevant information. Figure 3 shows an 3.3.1-case sample summary of the output of this analysis.

```
Best weighing MEAN preformance: 7,1
minTime sPR TimeOfSearching 465,
announcement: [[0, 2, 5], [0, 4, 6], [1, 3, 4], [1, 5, 6], [2, 3, 6]],
wInformativeness: 9,
wSecurity: 9,
Deal: 3.3.1
maxTime sPR TimeOfSearching 48875,
announcement: [[0, 1, 2], [0, 3, 4], [0, 3, 6], [1, 2, 6], [1, 3, 5],
[1, 3, 6], [2, 3, 4], [2, 3, 5], [4, 5, 6]],
wInformativeness: 6,
wSecurity: 7,
Deal: 3.3.1
(max-min) range time sPR 48410.0
Whole experiment mean timeOfSearching: 4031.433
Data analyzer timing: 915
```

Fig. 3. Stats analyzer output for 30-runs experiment $wI, wS = [1..10]$

The first line of Figure 3 represents the weights that have the best mean search time. Those weights could be good candidates to weigh other card deals (than 3.3.1), in order to investigate if there is a relation between weighing and search time. Expression `minTime sPR TimeOfSearching 465` denotes the lowest time, in millis, (the fastest protocol found) in the experiment that is associated to the announcement `[[0, 2, 5], [0, 4, 6], [1, 3, 4], [1, 5, 6], [2, 3, 6]]` that has the weight $wI =$ `wInformativeness: 9`, $wS =$ `wSecurity: 9`. We also see similar parameters for the protocol with the highest search time. At the end, the search time range and mean of the entire experiment.

There are also other 7-cards distributions where the genetic approach can search for protocols. Considering the constraints among a, b, c cards presented in [1], we obtain just two 7-cards scenarios where there exist protocols to search, namely: 3.3.1 and 4.2.1. Another case is 2.4.1. Then, there is no protocol where first a makes the announcement. But swapping the roles of a and b we can apply a 4.2.1 protocol.

The first experiment we did was a 30-runs from 1 to 10 different combinations of weights. We can represent that as (`SC` $30, 500, 200, wI, wS, 3, 3, 1$) where:

- SC stands for scenario configuration
- 500 is the maximun number of generations allowed to evolve
- 200 is the initial population size
- 30 means thirty runs of the algorithm
- wI is the weight for informativeness
- wS is the weight for security
- $3, 3, 1$ is the deal distribution

Notice that wI and wS will vary from 1 to 10, generating different scenarios configurations in order to search for weight with the fastest result. The first three results from a 30-run sample experiments with $wI = 4$ and $wS = 7$ are depicted in Figure 4.

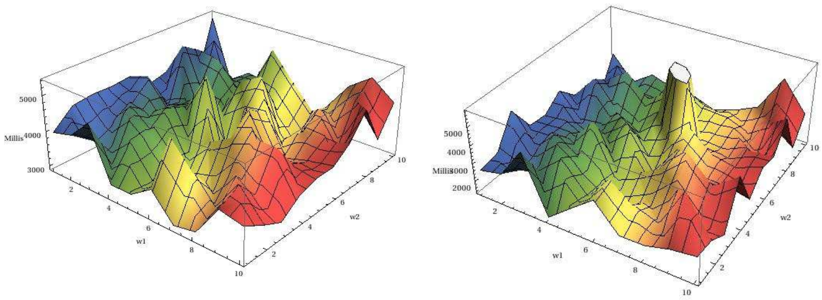
```
[[0, 2, 5], [0, 3, 4], [1, 3, 5], [1, 4, 6], [2, 3, 6]]:
58.0, 58.0, 73, 6464, 4, 0, 500, 200
[[0, 1, 6], [0, 2, 3], [1, 2, 4], [2, 5, 6], [3, 4, 5]]:
58.0, 58.0, 24, 2587, 4, 0, 500, 200
[[0, 1, 5], [0, 3, 4], [1, 2, 6], [2, 4, 5], [3, 5, 6]]:
58.0, 58.0, 48, 4478, 4, 0, 500, 200
. . .
```

Fig. 4. Partial output of SC $30, 500, 200, 4, 7, 3, 3, 1$

Each two lines represent the execution of the serach for a 3.3.1 scenario. For example, considering the first protocol result, those figures mean:

- `[[0, 2, 5], [0, 3, 4], [1, 3, 5], [1, 4, 6], [2, 3, 6]]` is the announcement sequence. So a annouces $\{025, 034, 135, 146, 236\}$.
- `58.0` means the value the fitness functions assigned to that protocol.
- `58.0` (the second occurrence) represents the maximum fitness value that can be reached for any protocol.
- `73` is the generation where this protocol was found.
- `6464` is the time of searching in milliseconds.
- `4` means the minimum number of cards b can learn using this protocol.
- `0` is the maximum number of cards c can learn.
- `500` is the maximun number of generations allowed to evolve.
- `200` is the initial population size.

We executed a 30-runs experiment varying both weights from 1 to 10 generating a total of $30 \times 10 \times 10 = 3000$ protocol results. Figure 5 depicts information about the runtime for different weights for 3.3.1: on the left, the relation between the differents weights assignments and the search mean time of those; on the right, the standard deviation extracted from the first. At first sight one can infer there is not a regular relation among those parameters. Hence we think there is no use in extracting the best weights for ignorance and informativeness from this experiment and scale them up to larger deals.



(a) weighing and mean time (b) weighing and deviation time

Fig. 5. 3.3.1 weighing and mean stats graphics

6 Conclusions and future work

We confirmed the possibiliy of modelling unconditionally secure protocols search using Genetic Algorithms. The statistical analyzer showed the non-regular relation between weighing protocol (main) requirements and the mean time of searching. Althouth the genetic engine has been used for small deals, it has now been studied in order to improve the time/memory efficiency to use it for larger deals where several announcements comprise a protocol and a huge number of operations is presumed to be executed. In fact considering the exponential nature of the search space based on the cards distribution, our actual concern is focussed on a possible future guidance of the fitness function and the other variation operators i.e: think that a 5.5.1 scenario with a binary announcement representation has a search space cardinality around 2^{462}. Using the RHC_1 we can observe the epistatic nature of the search space where a slight modification of the chromosome leads to a huge penalty on the solution fitness assignment. We project new features regarding not just the statistical analysis over the protocols found but the search for symmetry properties in protocols, features like a non-exhaustive fitness function and alternative protocol representation that can give a clue about a possible analytic solution for larger and more complex deals.

References

1. Albert, M., Aldred, R., Atkinson, M., van Ditmarsch, H., Handley, C.: Safe communication for card players by combinatorial designs for two-step protocols. Australasian Journal of Combinatorics 33, 33–46 (2005)
2. Albert, M., Cordón, A., van Ditmarsch, H., Fernández, D., Joosten, J., Soler, F.: Secure communication of local states in interpreted systems. In: Abraham, A., Corchado, J., González, S., De Paz Santana, J. (eds.) International Symposium

on Distributed Computing and Artificial Intelligence. pp. 117–124. Advances in Intelligent and Soft Computing, Vol. 91, Springer (2011)
3. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
4. van Ditmarsch, H.: The Russian cards problem. Studia Logica 75, 31–62 (2003)
5. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic, Synthese Library, vol. 337. Springer (2007)
6. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning about Knowledge. MIT Press, Cambridge MA (1995)
7. Fischer, M., Wright, R.: Bounds on secret key exchange using a random deal of cards. Journal of Cryptology 9(2), 71–99 (1996)
8. Gochet, P., Gribomont, P.: Epistemic logic. In: Gabbay, D.M., Woods, J. (eds.) Logic and the Modalities in the Twentieth Century. Handbook of the History of Logic, vol. 7, pp. 99–195. North-Holland (2006)
9. Hernández-Antón, I., Soler-Toscano, F., van Ditmarsch, H.P.: Unconditionally secure protocols with genetic algorithms. In: PAAMS (Special Sessions). pp. 121–128 (2012)
10. Kumar, A., Ghose, M.K.: Overview of information security using genetic algorithm and chaos. Information Security Journal: A Global Perspective 18(6), 306–315 (2009)
11. Menezes, A.J., Vanstone, S.A., Oorschot, P.C.V.: Handbook of Applied Cryptography. CRC Press, Inc., Boca Raton, FL, USA, 1st edn. (1996)
12. Ocenasek, P.: Evolutionary approach in the security protocols design. In: Blyth, A. (ed.) EC2ND 2005, pp. 147–156. Springer London (2006)
13. Plaza, J.: Logics of public communications. In: Emrich, M., Pfeifer, M., Hadzikadic, M., Ras, Z. (eds.) Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems: Poster Session Program. pp. 201–216. Oak Ridge National Laboratory (1989)
14. Plaza, J.: Logics of public communications. Synthese 158(2), 165–179 (Sep 2007), `http://dx.doi.org/10.1007/s11229-007-9168-7`
15. Sivanandam, S.N., Deepa, S.N.: Introduction to Genetic Algorithms. Springer Publishing Company, Incorporated, 1st edn. (2007)
16. Whitley, D.: A genetic algorithm tutorial. Statistics and Computing 4, 65–85 (1994)
17. Zarza, L., Pegueroles, J., Soriano, M.: Evaluation function for synthesizing security protocols by means of genetic algorithms. In: Second international Conference on Availability, Reliability and Securiy (ARES'07). IEEE (2007)

Markov Logic Networks for Spatial Language in Reference Resolution

Casey Kennington

Dialogue Systems Group, CITEC
Faculty of Linguistics and Literary Studies
Universität Bielefeld, Bielefeld, Germany
ckennington@cit-ec.uni-bielefeld.de
www.caseyreddkennington.com

Abstract. This paper presents an approach for automatically learning reference resolution, which involves using natural language expressions, including spatial language descriptions, to refer to an object in a context. This is useful in conversational systems that need to understand the context of an utterance, like multi-modal or embodied dialogue systems. *Markov Logic Networks* are explored as a way of jointly inferring a reference object from an utterance with some simple utterance structure, and properties from the real world context. An introduction to MLNs, with a small example, is given. Reference resolution and the role of spatial language are introduced. Different aspects of combining an utterance with properties of a context are explored. It is concluded that MLNs are promising in resolving a contextual reference object.

Keywords: reference resolution, spatial language, markov logic networks

1 Introduction

When we speak in a dialogue setting, we are often co-located (situated) in space and refer to objects in that space. Besides using salient properties to refer to an object, like its color, shape, and size, the language which is often used to refer to those objects make up *spatial language*; that is, where the object is relative to some frame of reference, or as relative to other objects in that space. Understanding spatial language has application in fields like dialogue systems, multi-modal systems, and robotics, and makes up a part of natural language understanding (NLU) in situated environments.

Work done in reference resolution has focused mainly on word-level approaches (see [16]), without specific focus on spatial language. [20] construct a visual grammar for reference resolution with some degree of success. [18] looks at incremental reference resolution, but also only looks at words. In this paper, we extend the work of reference resolution by focusing on real world properties and bridge them with utterances that use spatial language, with a small amount of linguistic structure. We created a statistical model trained on generated data and apply Markov Logic Networks (MLNs, [14]) as the machine learning technique in the experiments. It is shown that these models are significantly better than the baseline and that spatial language can be learned and applied to reference resolution using MLNs.

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see http://ceur-ws.org/, 2012, pp. 54–64.

Plan of this paper: In the following section, MLNs are defined and a simple example is given. The last part of the section will define reference resolution, with particular attention to spatial language and how it pertains to the task of this paper. In section 2, the domain, task, data, and procedure are explained. The experiment section then shows how well the system performs in reference resolution, as well as some visual representations of how well the system learned about spatial language.

1.1 Markov Logic Networks

Markov Logic Networks have recently received attention in language processing fields like coreference resolution [3], semantic role labeling [13], and web information extraction [17]. As defined by [14], a Markov Logic Network is a first-order knowledge base with a weight attached to each formula (see also Markov Random Fields [10], which make up MLNs). This knowledge base is a declaration of predicates and typed arguments, along with weighted first-order logic (FOL) formulas which define the type of relations between those predicates. This becomes a *template* to a MLN network. *Evidence* in the form of instances of the predicates can also be given either for training the network, or for direct inference. MLNs are a type of graphical model, where the nodes are not directed, an example of which can be found in Figure 2.

As an example MLN, consider the following statement: *If you write something down, you are twice as likely to remember it.* We can represent this by the predicates $Write(person)$ and $Remember(person)$. This can be formulated as: 2 $Write(x) \Rightarrow Remember(x)$. This defines a relationship between *Write* and *Remember*. The argument in both *Write* and *Remember* have the type *person* and for this example, we will introduce only one constant: *Mary*. Building all possible formulas given this information requires all combinations of positive and negative predicates in the formula, resulting in four formulas, all with weight 2, which will be referred to as *worlds*. It is then necessary to determine which of those four formulas are satisfied by the original formula of $Write(x) \Rightarrow Remember(x)$. It is satisfied when $Write$ is negative or when *Remember* is positive, or both. The only time it is not satisfied is when $Write(Mary) \Rightarrow \neg Positive(Mary)$.

After a set of worlds is created, one can perform inference by calculating a probability:

$$P(X = x) = \frac{1}{Z}\, e^{\sum_j w_j f_j(x)} \tag{1}$$

Where Z, also known as the partition function, is the normalizing constant given by:

$$Z = \sum_{x\ in\ X} e^{\sum_j w_j f_j(x)} \tag{2}$$

Where j indexes over the formulas. Here w_j is defined as the weight of the corresponding formula (in our example, the weight was 2), and f_j is a function that returns 1 if the formula is satisfiable, and 0 if it is not satisfiable. After the worlds and their satisfiabilities are identified, the next step is to determine Z. If our evidence is $Write(Mary)$, Z can be computed by finding which of the four formulas satisfy the evidence $Write(Mary)$, which results in one world that is satisfiable by the original formula, and one that is not. Given this set of worlds, the numerator of the probability is found by identifying the formulas which satisfy a query, for example, $Remember(Mary)$?, which results in only one satisfiable formula. In this case, the probability of $Remember(Mary)$ given the evidence $Write(Mary)$ is $\frac{e^2}{e^2+e^0} = 0.88$.

This simple example only considered a very small set of possible worlds. It doesn't take many more formulas, predicates, arguments, or constants to make computing MLNs intractable for large domains. In fact, inference alone is NP-Hard [15], and determining clause satisfiability is NP-Complete [21]. Most of the current research done in MLN attempts to find better ways to approach these problems of intractability, with a large degree of success. Furthermore, only inference was discussed. Inference is performed when the weights are given. There are also mechanisms for learning the weights given training data, but the details of how that is done will not be discussed here. For this paper, I use the discriminative learning approach [21] to automatically learn the formula weights. Methods for training MLNs can also be found in [5] and [12]. A book by Pedro Domingos and Daniel Lowd [6] offers a good introduction to MLN inference, weight learning, and contains several examples.

1.2 Reference Resolution

Reference resolution involves the linking of natural language expressions to contextually given entities [18]. Visual properties are often used to this end, properties such as color, size, and shape. However, we often use spatial relationships with other objects to aid in that reference. As spatial language plays a very important role in the reference resolution in this paper, spatial language will now be discussed.

Spatial Language

Learning the meaning of spatial relationships has been done in navigational direction tasks using reinforcement learning [22]. Spatial language has also been studied in psycholinguistic research, as well as practical applications like robotics (see *Spatial Language and Dialogue* [4] and *Language and Spatial Cognition* [8] by Annette Herskovit). When humans use spatial language, they use properties of the objects to which they are referring, such as color, shape, relative position to another salient object, or relative position with respect to some axis [23].

Spatial language involves the language humans use to describe space and the objects in that space. Humans require a common understanding of *absolute* and *relative* spatial descriptions in order to communicate effectively. Words such as *left*, *right*, *top*, *bottom*, and *middle* can represent absolute descriptions. Relative means that objects identified relative to another, perhaps more visually salient object. This is where prepositions are often used; *on top of*, *below*, *next to*, *to the left of*, *beside*. Even if we can gesture by pointing or looking at an object, we still usually need to be able to articulate the linguistic description of the object such that the hearer of the utterance can uniquely identify the object to which we are referring. This linguistic description becomes even more important when a human is interfaced with a computer and language is the only medium for communication from human to computer, which is the experimental setting for this paper.

It is essential to establish a frame of reference when using spatial language. It has been shown that alignment of the reference point is an ongoing process across utterances, depending on the context [24]. [11] and [22] use two main categories for the frame of reference: *egocentric* where the speaker is the frame of reference, and *allocentric* were the coordinates are not based on the speaker. In this paper, we will assume a single frame of allocentric reference. Specifically, both the hearer and the speaker use the same frame of reference.

2 Two-Dimensional Spatial Learning

In this section, the domain, task, data, and procedure of the experiment which will use MLN to automatically learn reference resolution, particularly using spatial language, are defined. By giving accuracies of successfully referred objects, we show that MLNs can be used successfully in reference resolution, as well as some figures that show the extent that spatial language was learned.

2.1 Domain, Task, Data, and Procedure

Domain

In this study, we used the *Pentomino* puzzle piece domain. A Pentomino board is visually represented in rows and columns with pieces viewable by a human. On the computer side, pieces are identified with unique arbitrary identifiers. The properties of the pieces which are visually distinguishable by a human (color, shape, row, column, relation to other pieces) are accessible also to the computer. In this way, the board and pieces effectively become a shared visual context between the human and the computer, as described in [20]. An example Pentomino board can be seen in Figure 1.

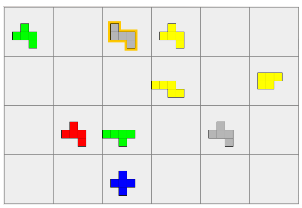


Fig. 1. Pentomino Board Example

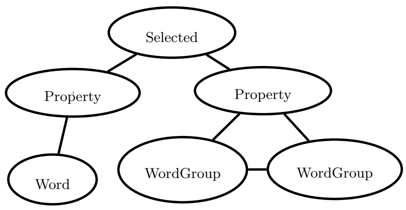


Fig. 2. MLN Relations

Perspective is not being studied in this paper, so assumptions need to be made about the frame of reference. The Pentomino board defines a fixed frame of reference which is a board on a computer screen. Also, given the nature of the utterances used in this paper (both generated from a corpus), egocentric descriptions such as *my left* were not used. The utterances used involve three components to each spatial description: *target*, a possible *reference object*, and *spatial term(s)* [2]. The goal is to see if these can be learned automatically to identify the reference object.

Task

The Pentomino board as implemented in the InPro toolkit [1, 19] was used.[1] The larger goal of NLU in Pentomino is to understand what action which is to be performed on the board (rotate, delete, move, mirror, or select a piece), which results in a change of the board's state (e.g., rotating a piece means visually making it appear with a different orientation). This experiment will focus only on identifying the piece to which the user is referring, not the action or change of board state.

Data

The training and evaluation data were automatically generated. This was a natural choice because one of the goals of this paper is to determine how well MLNs perform on language data. We can infer that if MLNs can't be made to work well on simplistic, automatically generated data with simple relational structures, then they will not work on real speech data with more complex structures. Further, the use of automatically generated data was motivated by the fact that the perspective is fixed, making the utterances of a finite type. Also, the task is reference resolution so the verb can be assumed to be a command-form *select* (as in "Select the piece ..."), thus reducing the possible syntactic structures. This puts focus on the spatial language and words that describe other properties of a piece, which can be generated to be representative of typical real-world language use. Finally, we don't generate actual utterances, but a simple semantic representation that defines relationships between words and word groups (phrases), though we will refer to them as the utterance. This kind of semantic representation can be obtained from most syntactic or semantic formalisms, though no syntactic or semantic parsing is done here.

The training data were generated by creating one thousand boards. Each board was randomly assigned 3-5 rows and 3-5 columns (number of rows and columns need not be the same), creating a small variability in the dimensions of the boards which allowed for generality. The number of pieces was also randomly assigned, between 4 and 6 pieces for training where the maximum number of pieces must be no larger than the number fields in the smallest possible board. Each piece was assigned a random color, shape, row, and column, and of course multiple pieces could not occupy the same space. After each piece was in place, one was chosen randomly to be the *selected* piece, the piece that was referred. A pseudo-utterance that described that piece was then generated in the form of a simple semantic representation. The pseudo-utterance contained words that described the shape, color, absolute, and relative descriptions. There were a total of 5 colors and 7 shapes. I used the following absolute spatial descriptions: *left*, *right*, *top*, *bottom*, *middle*, *corner*, and the following relative spatial descriptions: *above*, *higher than*, *on top of*, *next to*, *beside*, *below*, *under*, *beneath*, *on the left of*, and *on the right of*. The shape, color, and all absolute spatial descriptions were treated as a bag of words. The words in a relative spatial description were treated as a word group, which corresponds to a small amount of linguistic structure (grouped, for example, as prepositional phrases).

An example of this is given in Figure 3. Here, the selected piece is a red cross somewhere near the middle, below a gray cross which is in the middle of the top row. The generated pseudo-utterance would be something like *select the middle red cross below the gray cross in the top middle*, where *red*, *middle*, and *cross* are treated as simple words, and *below the gray cross in the top middle* is treated as a word group.

[1] `http://sourceforge.net/projects/inprotk/`

The evaluation set was similarly generated, but used 100 boards, with row and column lengths between 3 and 7, and the number of pieces was between 4 and 9. The range of rows and columns, and the range of possible number of pieces was larger than in training. For each evaluation experiment, the evaluation set was randomly generated, so each one was different.

Part of the effort in using MLNs is determining what should be specified as predicates (e.g. Color, Shape, Row, etc) and what should be learned automatically. In this paper, only one abstract predicate, *Property*, that implicitly does the typing into different features was used. The example in Figure 3 shows a single *Piece*, and multiple *Property* predicates for that piece. Where numbers would cause confusion, they can simply be annotated with a unique property type (e.g. column 1 = 1C). The properties used as represented in Figure 3 were type, color, % horizontal from center, % vertical from center, row, and column. The predicates *Word* and *WordGroup* represent the words, and group of words, respectively, as previously explained. The final argument for all the predicates is the board identifier, which separated states of the board and corresponding selected pieces and utterances from other boards. The second argument in *WordGroup*, groupID, is a unique identifier that is the same across a group of words. In Figure 3, the identifier is simply 1, but if more *WordGroups* existed, then they would be represented by a different number. A typical representation of a board would have many pieces and more word groups for the utterance.

Figure 3 also shows the actual MLN template that was used for training. Lines 1-5 are the predicate definitions and argument types and 6-7 define the relations via FOL formulas. The + before an argument in a formula tells the MLN to learn from each constant in the training data, which is necessary when dealing with natural language because it must learn how to interpret individual words as individual symbolic features. The result is that words and word groups are mapped to properties.

MLN template

1 $Piece(piece, boardID)$
2 $Property(piece, prop, boardID)$
3 $Word(word, boardID)$
4 $WordGroup(gWord, groupID, boardID)$
5 $Selected(piece, boardID)$
6 $Word(+w, b) \wedge$
$Property(p, +p1, b) \Rightarrow$
$Selected(p, b)$
7 $WordGroup(+w1, e, b) \wedge$
$WordGroup(+w2, e, b) \wedge$
$Property(p, +p1, b) \Rightarrow Selected(p, b)$

example

```
Piece(tile-7,-1)
Property(tile-7,X,-1)
Property(tile-7,red,-1)
Property(tile-7,0H,-1)
Property(tile-7,-67V,-1)
Property(tile-7,2R,-1)
Property(tile-7,1C,-1)
Word(red,-1)
Word(cross,-1)
Word(center,-1)
WordGroup(below,1,-1)
WordGroup(gray,1,-1)
WordGroup(cross,1,-1)
WordGroup(top,1,-1)
WordGroup(middle,1,-1)
```

Fig. 3. MLN template and example for *select the middle red cross below the gray cross in the top middle*

Procedure

The *Alchemy* system [5] for MLN learning and inference was used for these experiments.[2] The board and pseudo-utterances were represented as evidence to the MLN, as well as which piece was selected. Discriminative training with `Select` as the predicate to query was used. As MLN is a way of defining a relation between various predicates, those predicates need to represent meaningful information from the Pentomino board, as well as the utterance and how the utterance relates to the board, as shown in Figure 3.

To test, a board is similarly randomly generated as in training, and a piece is again randomly selected. However, the selected piece is what was being inferred about by the MLN, so it was kept hidden. The pseudo-utterance was generated in the same way as in training. Using the state of the board and the utterance, the MLN system then inferred which piece was being described by that utterance. If the piece with the highest probability matched the selected piece, it was marked as correct. If there was a tie, then the first one returned by the query was chosen. This is therefore a simple measure of accuracy. In this scenario, the majority class baseline would be 25%, where 4 is the minimum number of possible pieces during evaluation.

2.2 Experiment: Absolute and Relative Spatial Understanding

Table 1 shows various training and evaluation settings that were used. Each row represents the parts of the utterance and board properties that were represented. The **Full** column was trained on one set of boards with all possible descriptions (shape, color, relative spatial, absolute spatial, row, column) and that trained model was used to evaluate for both rows in the column. Even though, words like *row* and *column* are often used to distinguish pieces in this kind of setting, those words are left out of evaluation unless specifically stated because they are easy features to learn as they map easily to their corresponding piece properties. The **Ind** column (short for Individual) in the table shows the results when each row's setting was trained individually, and subsequently evaluated.

Description	**Full%**	**Ind%**
shape, color	70	92
absolute spatial with rows and columns	87	97
absolute spatial	65	69
relative spatial	35	59
shape, color, absolute spatial	**92**	**100**
absolute and relative spatial	34	82
all	58	88

Table 1. Various `Select` Accuracy Results; **Full** represents a single trained model with all features and evaluated only on the features of the row, **Ind** represents a different model trained and evaluated on the features of the corresponding row.

[2] `http://alchemy.cs.washington.edu/`

Though the main goal of this paper is to show that a MLN can effectively learn spatial language reference resolution, it is also useful to see how a MLN performs in different settings, as shown in Table 1. Overall, alchemy and MLN perform above baseline in all areas. Most interesting to note is the fact that when all types were used (shape, color, absolute, and relative spatial), the accuracy was only 58%. This is possibly due to the fact that it was trained on a system which included the rows and columns (information that was not used at evaluation). This is evidenced by the fact that the *Ind* column for the same row gave 88% and was trained on similar settings, only minus rows and columns, forcing the model to learn how to resolve without rows and columns as piece properties. It is further interesting to note that shape, color, and absolute spatial received high accuracies for both systems. However, a system with relative spatial knowledge will be more robust to real language input in a real-world environment, despite the somewhat lower probability.

2.3 Board Distributions

Another way of showing how well the MLN learned about spatial language is to look at a graphical representation of the probabilities of each piece in the board as a gradient, where the darker in color, the higher the probability. Some of these gradient boards (using a 5x5 board) are displayed, where all fields are taken by a piece with the same color and shape, thus nullifying them as distinguishing features. First, a look at absolute spatial language. An example of absolute *top-left*, and absolute *centre-right* are represented in Figures 4 and 5 respectively. These show that the absolute language was well learned, especially for a specific point like *top-left* in Figure 5.

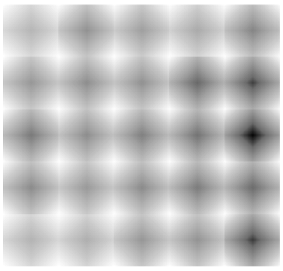


Fig. 4. Center Right

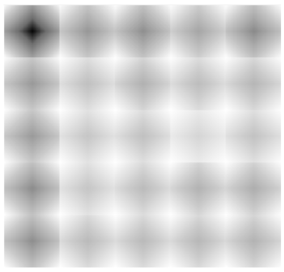


Fig. 5. Top Left

Relative spatial language is a little more difficult to visualize in a gradient, but certain relative relations can be isolated by looking at a gradient map with its corresponding board, where the board is not completely filled. The board and gradient map for the relation *above* can be found in Figures 6 and 7. For a piece to be *above* another piece, it simply needed to be in a higher row than that piece, regardless of the column. The notion of *above* is learned, but the distribution over the pieces is somewhat unexpected. Any piece that is above another piece should have some probability, with the darker gradients starting at the higher pieces, decreasing with the rows. The concept of *above* here is generally learned, though there is need for improving the model when it comes to relative spatial relationships.

After the submission of this paper, we used the principles that were learned here and applied them to real, non-generated Pentomino data collected in a Wizard-of-Oz study [7, 18] in a situated setting of natural language understanding. That work resulted in a paper [9] in which we used MLNS not only for reference resolution, but to predict a semantic frame which also included the action to take on the referred piece, and the desired state of the board after the action was complete (for example, the utterance *rotate the blue cross clockwise* would have *rotate* as the action, *the blue cross* as the referred piece, and the resulting state of the board would be that it appears 90 degrees to the right). Experiments were performed on hand-annotated speech, as well as automatically transcribed speech, evaluated incrementally (word-level increments), and on full utterances. We concluded that information from the visual context (pentomino board), the utterance (which included a syntactic context-free parser and corresponding semantic representation), and previous discourse context perform well, in terms of frame accuracy, in the Pentomino domain.

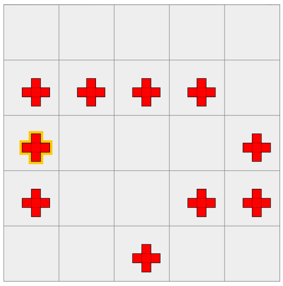


Fig. 6. above Board

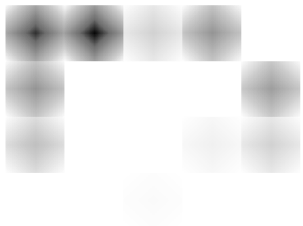


Fig. 7. above Map

3 Conclusions and Future Work

Markov Logic Networks are effective in reference resolution in a two-dimensional domain like Pentomino. They can effectively learn a mapping between an utterance and real world object properties, including utterances that contain spatial language descriptions.

Future work involves using what was learned via this task and implementing it into a larger natural language understanding framework. We will continue to use MLNs, and further incorporate other real-world information, such as eye gaze and gestural information from the human. We will extend this to domains beyond Pentomino, domains which use real world spatial representations, and apply it in interactive settings.

References

1. Baumann, T., Schlangen, D.: The InproTK 2012 Release. In: Proceedings of Human Language Technologies: The 2012 Annual Conference of the North American Chapter of the Association for Computational Linguistics on - NAACL '12 (2012)

2. Carlson, L.A., Hill, P.L.: Formulating Spatial Descriptions across Various Dialogue Contexts. In: Spatial Language and Dialogue, pp. 89–103 (2009)
3. Chen, F.: Coreference Resolution with Markov Logic. Association for the Advancement of Artificial Intelligence (2009)
4. Coventry, K.R., Tenbrink, T., Bateman, J. (eds.): Spatial language and dialogue, vol. 3. Oxford University Press (2009)
5. Domingos, P., Kok, S., Poon, H., Richardson, M.: Unifying logical and statistical AI. American Association of Artificial Intelligence (2006)
6. Domingos, P., Lowd, D.: Markov Logic An Interface Layer for Artificial Intelligence. Morgan & Claypool (2009)
7. Fernández, R., Lucht, T., Schlangen, D.: Referring under restricted interactivity conditions. In: Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue. pp. 136–139 (2007)
8. Herskovits, A.: Language and Spatial Cognition. Cambridge University Press (2009)
9. Kennington, C., Schlangen, D.: Markov Logic Networks for Situated Incremental Natural Language Understanding. In: Proceedings of SIGdial 2012. Association for Computational Linguistics, Seoul, Korea (2012)
10. Kindermann, R., Snell, J.L.: Markov random fields and their applications (1980)
11. Levinson, S.C.: Space in Language and Cognition. Explorations in Cognitive Diversity, vol. 5. Cambridge University Press (2003)
12. Lowd, D., Domingos, P.: Efficient weight learning for Markov logic networks. Knowledge Discovery in Databases: PKDD 2007 pp. 200–211 (2007)
13. Meza-Ruiz, I., Riedel, S.: Jointly identifying predicates, arguments and senses using Markov logic. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics on - NAACL '09. p. 155. No. June, Association for Computational Linguistics, Morristown, NJ, USA (2009)
14. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning 62(1-2), 107–136 (2006)
15. Roth, D.: On the hardness of approximate reasoning. Artificial Intelligence 82(1-2), 273–302 (1996)
16. Roy, D.: Grounding words in perception and action: computational insights. Trends in Cognitive Sciences 9(8), 389–396 (Aug 2005)
17. Satpal, S., Bhadra, S., Rajeev, S.S., Prithviraj, R.: Web Information Extraction Using Markov Logic Networks. Learning pp. 1406–1414 (2011)
18. Schlangen, D., Baumann, T., Atterer, M.: Incremental Reference Resolution: The Task, Metrics for Evaluation, and a Bayesian Filtering Model that is Sensitive to Disfluencies. In: Proceedings of SIGdial 2009 the 10th Annual SIGDIAL Meeting on Discourse and Dialogue. pp. 30–37. No. September (2009)
19. Schlangen, D., Skantze, G.: A general, abstract model of incremental dialogue processing. Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics on - EACL '09 (April), 710–718 (2009)
20. Siebert, A., Schlangen, D.: A Simple Method for Resolution of Definite Reference in a Shared Visual Context. In: Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue. pp. 84–87. No. June, Association for Computational Linguistics (2008)
21. Singla, P., Domingos, P.: Discriminative Training of Markov Logic Networks. Computing 20(2), 868–873 (2005)

22. Vogel, A., Jurafsky, D.: Learning to Follow Navigational Directions. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistic. pp. 806–814 (2010)
23. Vorwerg, C.: Consistency in Successive Spatial Utterances. In: Coventry, K.R., Tenbrink, T., Bateman, J.A. (eds.) Spatial Language and Dialogue, chap. 4, pp. 40–55. Oxford University Press (2009)
24. Watson, M.E., Pickering, M.J., Branigan, H.P.: Alignment of Reference Frames in Dialogue. In: Cognitive Science Society (2004)

Using Conditional Probabilities and Vowel Collocations of a Corpus of Orthographic Representations to Evaluate Nonce Words

Özkan Kılıç

Department of Cognitive Science, Informatics Institute, Middle East Technical University, Ankara, Turkey
e-mail: okilic@ii.metu.edu.tr

Abstract. Nonce words are widely used in linguistic research to evaluate areas such as the acquisition of vowel harmony and consonant voicing, naturalness judgment of loanwords, and children's acquisition of morphemes. Researchers usually create lists of nonce words intuitively by considering the phonotactic features of the target languages. In this study, a corpus of Turkish orthographic representations is used to propose a measure for the nonce word appropriateness for linearly concatenative languages. The conditional probabilities of orthographic co-occurrences and pairwise vowel collocations within the same word boundaries are used to evaluate a list of nonce words in terms of whether they would be rejected, moderately accepted or fully accepted. A group of 50 Turkish native speakers were asked to evaluate the same list of nonce words. Both the method and the participants displayed similar results.

Keywords: Nonce words, Orthographic representations, Conditional probabilities.

1 Introduction

Nonce words are frequently employed in linguistic studies to evaluate areas such as well-formedness [1], morphological productivity [2] and development [3], judgment of semantic similarity [4], and vowel harmony [5]. Nonce words are also used to understand the process of adopting loan words. The majority of loaned words undergo certain phonetic changes to more resemble the lexical entries of the language into which they will be adopted [6]. For example, *television* in Turkish becomes *televizyon /televızjon/* because */jon/* is more frequent than */ʒın/* in Turkish[1]. Similarly, *train* is adopted as *tren /tren/* because, similar to diphthongs, vowel-to-vowel co-occurrences are not usually allowed in Turkish non-compound words. This phenomenon shows that the speakers of a language are aware of the possible sound frequencies and collocations of their native languages, and they can make judgements on the naturalness of loan

[1] In the METU-Turkish Corpus, there are 181 occurrences with the segment */ʒın/* of which only 30 are at the terminating word boundaries. On the other hand, there are 5,945 occurrences with the segment */jon/* of which 3,190 are at the terminating word boundaries, excluding the word *televizyon*.

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see http://ceur-ws.org/, 2012, pp. 65–72.

words, recently invented words and nonce words by using their knowledge of the existing Turkish lexis. Thus, the acceptability of nonce words is a logical decision based on known-word statistics.

The acceptability of nonce words can be investigated by experimental investigations through phonotactic properties or factor-based analysis [7]. In the experimental investigations, it is observed that the participants accepted or rejected nonce words according to probable combinations of sounds [1, 8]. In factor-based analysis, the acceptability of nonce words is evaluated through the co-occurrences of syllables or consonant clusters locally [9] or non-locally [10–12] or through nucleus-coda combination probabilities [13].

In this study, the acceptability of nonce words was assessed using the conditional probabilities of the bigram co-occurrences of the orthographic representations locally and the pairwise collocations of the vowels within the same word boundaries. Similar methods within the context of phonotactic modeling had been used for Finnish vowel harmony [14]. Yet in this study, the local bigram phonotactic modeling was used to evaluate Turkish nonce words. Two threshold values were set for the decision to reject, moderately accept and fully accept. The threshold values were computed according to the length of each input string. For the evaluation of the conditional and collocation probabilities, the METU-Turkish Corpus containing about two million words was employed [15]. The list of nonce words was created intuitively. The same list of nonce words evaluated by the method was also given to 50 Turkish native speakers to judge the level of acceptability of each word. The 25 male and 25 female Turkish native speakers, had an average age is 31.26 ($s = 4.11$).The results from the native speakers were very similar to the results provided by the statistical method. In this paper, brief information about Turkish language and plausibility of conditional probabilities will be given then details of the method and the results will be presented.

2 Turkish Language and Conditional Probability

Turkish has 8 vowels and 21 consonants, and it is agglutinative with a considerably complex morphology [16, 17]. While communicating, the word internal structure in Turkish is required to be segmented because Turkish morphosyntax plays a central role in semantic analysis. For example, although Turkish is considered as an SOV language, the sentences are usually in a free order. Thus, the subject and object of a verb can only be determined by the morphological markers as in (1) rather than the word order.

(1) *Köpek adam-ı ısırdı.*	*Köpeğ-i adam ısırdı.*
Dog man-Acc bit	Dog-ACC man bit
The dog bit the man.	The man bit the dog.

The description of Turkish word structure depends heavily on morphophonological constraints and morphotactics. In Turkish morphotactics, the continuation of a morpheme is determined by the preceding morpheme or by the stem as in (2).

(2) *ev-de-ki*	**ev-ki-de*
house-Loc-Rel	
The one in the house	

These morphotactic constraints in Turkish are captured by statistical models based on conditional probabilities [18, 19]. In addition to morphotactics, the morphophonology of Turkish needs a brief explanation because nonce words have to mimic this morphophonology.

Vowel harmony is dominantly effective in Turkish morphophonology in order to preserve the roundedness and the frontness of vowels within the same word boundaries. While a morpheme with a vowel is concatenated to a string, its vowel is modified with respect to the roundedness and frontness properties of the most recent vowel in the string as in (3).

(3)	*ev-ler*	*oda-lar*	*bil-di*	*duy-du*
	house - Plu	room - Plu	know - Past	hear - Past
	houses	rooms	knew	heard

Another important phenomenon in Turkish morphophonology is voicing. If some of the strings terminating with the voiceless consonant, *'p, t, k, ç'*, are followed by the suffixes starting with vowels, then the consonants are voiced as *'b, d, ğ, c'* as in (4).

(4)	*sonuç*	*sonuc-um*	*kanat*	*kanad-ı*
	result	result -1S.Poss	wing	wing - Acc
		my result		he wing

Consonant assimilation is also important in Turkish morphophonology. The initial consonants of some morphemes undergo an assimilation operation if they are attached to the strings terminating in the voiceless consonants, *'p, t, k, ç, f, s, ş, h, g',* as in the surface forms of the Turkish past tense -*DI* in (5).

(5)	*at-tı*	*konuş-tu*
	throw - Past	speak - Past
	threw	spoke

The final Turkish morphophonological phenomena that need to be briefly mentioned are deletion and epenthesis occurring as in (6).

(6)	*hak*	*hakk-ım*	*isim*	*ism-im*
	right	right - 1S.Poss	name	name - 1S.Poss
		my right		my name

The Turkish morphophonological phenomena described above occur in the co-occurrences of the orthographic representations in the concatenating positions except in vowel harmony and the deletion. This results in high conditional probabilities evaluated using the frequencies of the pairs of consecutive orthographic representations. Since the vowel harmony and deletion take place after or before the concatenation positions, their pairwise collocations within the same word boundaries are also required to be utilized in the statistical model.

The transition probability between A and B is simply based on the conditional probability statistics as in (7).

(7) $P(B|A)$ = *(frequency of AB) / (frequency of A)*

Infants are reported to successfully discriminate speech segments using transitional probabilities of syllable pairs [20, 21]. Adults also make use of transitional probabilities between word classes to acquire syntactic rules [22]. Similarly, transition probabilities are dominantly used in unsupervised morphological segmentation and disambiguation [18, 19], [23–25].

Statistical approaches to linguistics support the empiricist view; and they provide an explanatory account of linguistic phenomena such as the decrease in performance errors and language variations. Considering the properties of the Turkish language, using the conditional probabilities of orthographic representations and the collocations of vowels within the same word boundaries is a plausible method to decide whether nonce words or loan words will be *rejected*, *moderately accepted* or *accepted*

3 The Method

Let s be a string such that $s = u_1 u_2 \dots u_n$, where u_i is a letter in the Turkish alphabet. The string s is unified with the empty strings σ and ε such that $s = \sigma u_1 u_2 \dots u_n \varepsilon$, where σ denotes the initial word boundary and ε denotes the terminal word boundary. The overall transition probability of the string s is evaluated from the METU-Turkish Corpus using Formula 1.

$$P_t(s) = \prod_{1}^{n+1} P(u_i|u_{i-1}) \tag{1}$$

For example, using the Formula 1, $P(a|\sigma)$ gives the probability of the strings starting with the letter a, and $P(b|a)$ estimates the probability of the substring ab in the corpus. Now let v be a subset of the string s such that $v = u_{i,1} u_{j,2} \dots u_{k,m}$ where $u_{k,m}$ is the m^{th} vowel in the k^{th} location of the string s. The overall vowel collocations of the string s are estimated from the substring of vowels v using Formula 2.

$$\begin{aligned} P_c(v) &= \prod_{2}^{m} \frac{g(v_{i-1}v_i)}{f(v_{i-1})} \qquad if\ |v| > 1 \\ P_c(v) &= \frac{f(v_i)}{CorpusSize} \qquad if\ |v| = 1 \end{aligned} \tag{2}$$

In the Formula 2, the function $f(v_i)$ gives the frequency of the words that contain the vowel v_i as a substring in the corpus. The function $g(v_{i-1}v_i)$ gives the frequency of words in which the vowels v_{i-1} and v_i are collocating not necessarily in immediately consecutive positions but within the same word boundaries. The acceptability probability of the string s is calculated by $P_a(s) = P_t(s)P_c(v)$. The acceptability decision of the string s in the method is made by using the Formula 3.

$$\begin{aligned} &Accept \quad if \quad P_a(s) \geq 10^{-(t+v)} \\ &Moderately\ accept \quad if \quad 10^{-(t+v+1)} \leq P_a(s) < 10^{-(t+v)} \\ &Reject \quad if \quad 10^{-(t+v+1)} > P_a(s) \end{aligned} \tag{3}$$

where t is the number of transitions (which is *the length of the string* + 1) and v is the number of the vowel collocations (which is *the number of the vowels* - 1) in the string. If the string s has only one vowel, then $v = 1$.

The method was applied to the list of nonce words given in the following section. The same list was also given to the 50 Turkish native speakers to evaluate the acceptability of each item. The comparison of the results from the method and the native speakers is given below.

4 Results

The nonce word *talar* is evaluated as in (8)

(8)

$$\begin{aligned} P_a(talar) &= P_t(\sigma talar\varepsilon)xP_c(aa) \\ &= P(t|\sigma)P(a|t)P(l|a)P(a|l)P(r|a)P(\varepsilon|r)xP_c(aa) \\ &= 7.66e-06xP_c(aa) = 7.66e-06 * 4.75e-01 = 3.63e-06 \end{aligned}$$

Since $P_a(talar) \geq 10^{-(6+1)}$, in which 6 conditional probability estimations and 1 vowel collocation are evaluated, the nonce word *talar* is accepted. The word list was evaluated by the 50 selected Turkish speakers. The distribution of the native speaker responses and the results of the method are given in Table 1.

For 82% of the words the Turkish native speaker's responses are in agreement with the results from the method. The method failed to simulate the responses from the participants in 18% of the results.

5 Discussions and Conclusion

The acceptability of loan words and nonce words is mainly determined by the phonological properties of the target language and the current approaches are syllable-based [7–13]. Since there are no lexical entries for nonce words, the method in this study tries to estimate the acceptability of the words using the bigram conditional probabilities and collocations of the orthographic representations within the word boundaries, which is a simplified way of inducing Turkish morphophonology.

The nonce word *ülü* was rejected by the method but accepted by the participants. A possible reason might be that the nonce word *ülü* sounds similar to an existing Turkish word *ölü* 'death'. Similarly, the responses for the nonce word *nort* were in disagreement. This nonce word has a similar pronunciation to an English word *north* and the most of the participants also knew English as a foreign language. Therefore, the participants might also make use of their foreign language knowledge to evaluate nonce words.

Although the method does not assume to utilize any property of Turkish phonology and it does not implement any phonologic filtering mechanism, it is able to mimic, in a remarkable way, a large number of the responses from the participants. Indeed, this study does not propose that acceptability is based on raw orthographical representations rather than syllables and phonemes. Instead, it underlines that simple pairwise conditional properties and vowel collocations from a corpus can give an estimation of

Table 1. The results of the method and the results of the participants (Bold text indicates a strong similarity of the results)

		Responses of the Participants		
Nonce Words	Results of the Method	Reject	Moderately Accept	Accept
öğtar	**Reject**	96%	4%	
söykıl	**Reject**	96%	4%	
talar	**Accept**			100%
telüti	**Reject**	64%	28%	8%
prelüs	**Reject**	84%	14%	2%
katutak	**ModeratelyAccept**	8%	50%	42%
par	**Accept**		14%	86%
öçgöş	**Reject**	100%		
jeklürt	**Reject**	100%		
böşems	**Reject**	88%	12%	
trüğat	**Reject**	96%	4%	
cakeyas	**Reject**	92%	8%	
çörottu	**Reject**	74%	16%	10%
döyyal	**Reject**	78%	22%	
efföl	**Reject**	92%	8%	
aznı	Reject	32%	60%	8%
fretanit	**Reject**	64%	30%	6%
erttiçe	**ModeratelyAccept**	36%	64%	
goytar	Reject	38%	52%	10%
hekkürük	Reject	41%	47%	12%
henatiya	**ModeratelyAccept**	36%	64%	
taberarul	**Reject**	84%	16%	
gövük	Reject	30%	44%	26%
sör	**ModeratelyAccept**		78%	22%
perolus	**Reject**	84%	16%	
kletird	**Reject**	98%	2%	
ojuçı	**Reject**	100%		
ürtanig	**Reject**	94%	6%	
lezğaji	**Reject**	100%		
lamafi	**ModeratelyAccept**		64%	36%
nort	Reject	38%	42%	20%
netik	**Accept**		18%	82%
meşipir	ModeratelyAccept		24%	76%
oblan	**ModeratelyAccept**		58%	42%
öftik	**Reject**	62%	34%	4%
özola	**ModeratelyAccept**	32%	60%	8%
ayora	Accept		72%	28%
sengri	**ModeratelyAccept**	32%	68%	
sakkütan	**Reject**	58%	34%	8%
şepilt	**Reject**	78%	22%	
şür	**ModeratelyAccept**		78%	22%
puhaptı	**ModeratelyAccept**	38%	44%	18%
upapık	**Reject**	54%	28%	18%
ülü	Reject	28%	52%	20%
yukta	**ModeratelyAccept**		74%	26%
zerafip	**Reject**	54%	34%	12%
upgur	**Reject**	70%	16%	14%
kujmat	**Reject**	90%	10%	
lertic	**Reject**	94%	6%	
düleri	Accept		64%	36%

the acceptability of a list of nonce words. This can be used by researchers that need an evaluation for the nonce words for their studies when no phonologically annotated corpus with syllables exists.

6 Limitations and development

The method needs to be tested with larger word lists. The method is successful because there is a close correspondence between phonotactics and orthotactics in Turkish. It requires improvements in terms of the morphophonological properties of target languages. The method uses exact orthographic representations. Thus, it requires an additional phonological similarity measure for the representations to increase the success rate.

The threshold values for the acceptability decisions depend on word lengths. They also need to be improved with respect to the target languages. The method also needs to be tested and adapted for the languages with ablaut or umlaut phenomena such as English and German, and the templatic languages such as Arabic and Hebrew.

References

1. Hammond, M. : Gradience, phonotactics, and the lexicon in English phonology. Int. J. of English Studies **4** (2004) 1–24
2. Anshen, F., Aronoff, M.: Producing morphologically complex words. Linguistics **26** (1988) 641–655
3. Dabrowska, E.: Low-level schemas or general rules? The role of diminutives in the acquisition of Polish case inflections. Language Sciences **28** (2006) 120–135
4. MacDonald, S., Ramscar, M. : Testing the distributional hypothesis: The influence of context on judgements of semantic similarity. Proc. of the 23rd Annual Conference of the Cognitive Science Society, University of Edinburgh (2001)
5. Pycha, A., Novak, P., Shosted, R., Shin, E.: Phonological rule-learning and its implications for a theory of vowel harmony. Proc. of WCCFL 22, G. Garding and M. Tsujimura (Eds.) (2003) 423–435
6. Kawahara, S: OCP is active in loanwords and nonce words: Evidence from naturalness judgment studies. Lingua (to appear)
7. Albright, A.: From clusters to words: Grammatical models of nonce word acceptability. Handout of talk presented at 82nd LSA, Chicago, January 3 (2008)
8. Shademan, S.: From clusters to words: Grammatical models of nonce word acceptability. Grammar and Analogy in Phonotactic Well-formedness Judgments. Ph. D. thesis, University of California, Los Angeles (2007)
9. Hay, J., Pierrehumbert, J., Beckman, M.: Speech perception, well-formedness and the statistics of the lexicon. In: J. Local, R. Ogden, and R. Temple (Eds.), Phonetic Interpretation: Papersbin Laboratory Phonology VI. Cambridge: Cambridge University Press (2004)
10. Frisch, S. A., Zawaydeh, B. A.: The psychological reality of OCP-Place in Arabic. Language **77** (2001) 91–106
11. Koo, H., Callahan, L.: Tier-adjacency is not a necessary condition for learning phonotactic dependencies. Language and Cognitive Processes **77** (2011) 1–8
12. Finley, S.: Testing the limits of long-distance learning: learning beyond a three-segment window. Cognitive Science **36** (2012) 740–756

13. Treiman, R., Kessler, B., Knewasser, S., Tincoff, R., and Bowman, M.: English speakers' sensitivity to phonotactic patterns. In: M. B. Broe and J. Pierrehumbert (eds.), Papers in Laboratory Phonology V: Acquisition and the Lexicon. Cambridge: Cambridge University Press (2000) 269–282
14. Goldsmith, J., Riggle, J.: Information theoretic approaches to phonological structure: the case of Finnish vowel harmony. Natural Language & Linguistic Theory (to appear)
15. Say, B., Zeyrek, D., Oflazer, K., Özge, U.: Development of a corpus and a treebank for present-day written Turkish. Proc. of the Eleventh International Conference of Turkish Linguistics (2002)
16. Göksel, A., Kerslake, C.: Turkish: A Comprehensive Grammar. Routledge: London and New York (2005)
17. Lewis, G: Turkish Grammar, Second edition. Oxford: University Press (2000)
18. Kılıç, Ö., Bozşahin, C.: Semi-supervised morpheme segmentation without morphological analysis. Pro. of the LREC 2012 Workshop on Language Resources and Technologies for Turkic Languages, İstanbul, Turkey (2012)
19. Yatbaz, M. A., Yuret, D.: Unsupervised morphological disambiguation using statistical language models. Pro. of the NIPS 2009 Workshop on Grammar Induction, Representation of Language and Language Learning, Whistler, Canada (2009)
20. Aslin, R.N., Saffran, J.R., Newport, E.L.: Computation of conditional probability statistics by human infants. Psychological Science **9** (1998) 321–324
21. Gomez, R. L.: Variability and detection of invariant structure. Psychological Science **13** (2002) 431–436
22. Kaschak, M. P., Saffran, J. R.: Idiomatic syntactic constructions and language learning. Cognitive Science **30** (2006) 43–63
23. Creutz, M., Lagus, K.: Unsupervised models for morpheme segmentation and morphology learning. ACM Tran. on Speech and Language Processing **4(1)** (2007)
24. Bernhard, D.: Unsupervised morphological segmentation based on segment predictability and word segments alignment. Proc. of 2nd Pascal Challenges Workshop (2006) 19–24
25. Demberg, V.: A language-independent unsupervised model for morphological segmentation. Ann. Meet. of Assoc. for Computational Linguistics **45(1)** (2007) 920–927

A Dynamic Probabilistic Approach to Epistemic Modality*

Manuel Križ

University of Vienna

Abstract. This paper develops a very simple probabilistic dynamic formalism for a semantics of epistemic modal expressions that is essentially expressivist in nature, with a notion of information states based on probability spaces. It differs from existing approaches to this task (in particular [22] and [12]) in certain features that make it more appropriate for modelling actual conversational behavior, especially as regards ignorance and unawareness. It will be argued that the latter is important for an adequate treatment of the alleged non-monotonicity of conditionals. Directions for further research that follow from the approach will be pointed out.

Introduction

Recent work ([19], [20], [11]) has emphasized the importance of probability in epistemic modality in natural language. At the same time, there is a tradition in dynamic semantics (e. g. [16], [6], more recently [13]) which is successful in dealing with the subjective, non-truth conditional aspects of their meaning and its non-monotonicity in conversation. In this paper, we make a first attempt at doing the obvious thing and bringing them together. We will explore a possibility that has also been hinted at in [22], and developped quite fully, though with a different focus, in [12]: we will give a semantics for epistemic modals that is both probabilistic and essentially dynamic, using sets of probability functions to model information. In section 1, we will define our simple technical apparatus. In the next section, we will explore some applications to the behavior of epistemic modals in discourse and some linguistically relevant aspects of internal states of speakers and hearers. Section 3 concerns the non-monotonic properties of conditionals, and in the final section, we point to some further limitations of our model and directions for future research that follow from our approach.

1 A Toy Model

Definition 1. *A* ***probability space over possible worlds*** *is a triple* $\langle S, \mathscr{S}, P\rangle$*, where* S *is a set of worlds,* $\mathscr{S}$ *is some Boolean algebra over* S*, and* P *is a probability function on* $\mathscr{S}$*.*

*My thanks are due to Viola Schmitt, Daniel Büring, and especially Johannes Schmitt for stimulating and encouraging conversations.

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see `http://ceur-ws.org/`, 2012, pp. 73–83.

Let our language $\mathcal{L}$ be a simple propositional language with modal operators $\Diamond$, $\Box$ and $\triangle$ (the latter for *probably*) and a conditional connective $\succ$. We call a formula *non-modal* if it contains none of these.

Definition 2. *A* ***model*** *for* $\mathcal{L}$ *is a pair* $\langle W, [\![\cdot]\!]\rangle$*, where* W *is a set of possible worlds and* $[\![\cdot]\!]$ *an interpretation function that assigns to every non-modal formula of the language a subset of* W*, subject to the usual conditions for* $\wedge$ *and* $\vee$*.*

Definition 3. *We define an* ***information state*** *(or* ***context****)* c *to be a pair* $\langle \mathscr{S}, \mathscr{P}\rangle$*, where* $\mathscr{S}$ *is a boolean algebra over* W *and* $\mathscr{P}$ *is a set of functions* P *such that* $\langle W, \mathscr{S}, P\rangle$ *is a probability space.*

A preliminary definition of update, which serves to illustrate what properties we want it to have, is given in the following:

Definition 4. *(to be revised) If* ϕ *is a non-modal formula, then the* ***update*** *of a context* c *is defined as follows:*[1]

$$\begin{aligned}
c[\phi] &:= \{P(\cdot|\phi)|P \in c \wedge P(\phi) \neq 0\}\\
c[\Diamond\phi] &:= \{P \in c|P(\phi) > 0\}\\
c[\Box\phi] &:= \{P \in c|P(\phi) = 1\}\\
c[\triangle\phi] &:= \{P \in c|P(\phi) > .5\}\\
c[\phi \succ \psi] &:= \{P \in c|\{P\}[\phi][\psi] = \{P\}[\phi]\}
\end{aligned}$$

Definition 5. *A formula* ϕ *is* ***accepted*** *in an information state* c *(*$c \vDash \phi$*) if* $c[\phi] = c \neq \varnothing$*. It is* ***consistent*** *with* c *if* $c[\phi] \neq \varnothing$ *and inconsistent if it is not consistent. It is* ***compatible*** *with* c *if there is a sequence* Δ *(which may include* ϕ*) of formulae such that successive update with the members of* Δ *yields a* c' *in which* ϕ *is accepted.* ϕ *is incompatible with* c *if it is not compatible.*

The recursive definition of negation in such a system is a tricky matter due to the qualitative difference between conditionalization and eliminative update. It is not even clear that the definition of negation really needs to be given recursively instead of as a set of rules for negative update, as in [13] (cf. also [10]). However, in our case a recursive definition *can* be given by adapting the procedure developed in [12].

We use a slightly simplified version of Schmitt's notion of a Bayesian closure of a set of probability functions. It is the set that contains all the functions in the original set, and all those that are the result of conditionalizing a function in the original set on some non-modal sentence in the language.

Definition 6. *The* ***Bayesian closure*** c^{cl} *of a context* c *is defined as*

$$\{P|\exists P' \in c\colon\ \exists\phi \in \mathcal{L}\colon\ P(\cdot) = P'(\cdot|\phi)\}.$$

Schmitt proves that the function $(\cdot)^*$ is the reverse of the Bayesian closure function if c fulfills the condition of *weak regularity*: if a function in c assigns 1 to any formula ϕ, then all functions in c do. Our contexts are, in general, not weakly regular, but as will be clear in a moment, that is not a problem for our final definition of update.

[1] We write $P(\phi)$ for $P([\![\phi]\!])$. Furthermore, since $\mathscr{S}$ is not affected by updates at this point, we will for ease of exposition pretend that a context is just a set of probability functions.

Definition 7. *The *-function returns for every set C of probability functions that is closed under conditionalization in the above sense the unique weakly regular set of probability functions that C is the Bayesian closure of.*

$$C^* = \{P \in C | \neg \exists P' \in C\colon\ \exists \phi \in \mathcal{L}\colon\ P(\cdot) = P'(\cdot|\phi)\}$$

On the level of Bayesian closures, a recursive definition for negation can be given. These are Schmitt's update rules:

Definition 8. *The update of a Bayesian closure C with a formula is defined as follows, where ϕ is a non-modal formula and ψ is an arbitrary formula.*

$$\begin{aligned}
C \uparrow \phi &:= \{P \in C | \exists P' \in C\colon\ P(\cdot) = P'(\cdot|\phi)\} \\
C \uparrow \neg\psi &:= \{P \in C | \exists y \subseteq C\colon\ P \in y \wedge y^{cl} \uparrow \phi = \varnothing\} \\
C \uparrow \Diamond\phi &:= \{P \in C | C \uparrow \neg\phi \neq C\} \\
C \uparrow \Box\phi &:= \{P \in C | C \uparrow \phi = C\} \\
C \uparrow \triangle\phi &:= \{P \in C | \forall P \in C^*\colon\ P(\phi) > .5\} \\
C \uparrow (\phi \succ \psi) &:= \{P \in c | (C \uparrow \phi) \uparrow \psi = C \uparrow \phi\}
\end{aligned}$$

But Bayesian closures are not what correctly represents our information, and furthermore, the update with modals here is only a test. What we will do is to make the update of a context distributive: we take the singleton of every function in P, form the Bayesian closure of it, apply Schmitt's rules to it, feed it to the *-function, and collect all the results.

Definition 9. *The update of a context c with an arbitrary formula ϕ (written $c[\phi]$) is given by the following:*

$$c[\phi] = \bigcup \{\mathscr{P} | \exists P \in c\colon\ \mathscr{P} = (\{P\}^{cl} \uparrow \phi)^*\}$$

The effects of this update are generally as in Definition 4. Now the negation of a formula ϕ is accepted in c iff ϕ is inconsistent with c.[2]

Norm of Assertion. A speaker S may assert ϕ iff ϕ is accepted in their information state c_S.

Norm of Contradiction. A hearer H may contradict a speaker's assertion of ϕ iff ϕ is incompatible with their information state c_H.

2 Some Applications

2.1 Question Sensitivity

These definitions allow us to capture elegantly a number of linguistic phenomena. The first (perhaps more of a cognitive phenomenon) is what is called *question sensitivity*

[2] We cannot, however, derive the fact, pointed out in [14], that the negation of a conditional is actually the negation of the consequent. Neither are the neg-raising properties of *probable* explained. This is a defect that the present approach shares with all comparable ones that I am aware of.

in [21]. There are people who don't know about the existence of a town by the name of Topeka. It seems very plausible to say that they are entirely *insensitive* to the question of whether it's raining in Topeka. The possibility of encoding this falls out directly from our use of probability spaces. Nothing forces $\mathscr{S}$ to be the whole power set of W, so there may well be propositions that the probability functions in $\mathscr{P}$ are not defined on.[3] Note that this means that sensitivity is not closed under logical consequence, although due to our use of proper probability spaces over possible worlds, we do, of course, still incur the problem of logical omniscience and, as it were, logical omnisensitivity.

2.2 Ignorance about Possibilities

Using sets of probability functions is the standard way of capturing that probability assignments are typically vague (cf. e. g. [7], among many others): someone who says that it's probably raining doesn't have to have a specific probability, say, .7, in mind.[4] But there is an interesting special case: writing $\mathscr{P}(\phi)$ for the set of values assigned to ϕ by some $P \in \mathscr{P}$, we can consider the possibility that $\mathscr{P}(\phi) = [0, 1]$. An agent with such an information state seems strange at first: she is sensitive to ϕ, and she doesn't have any evidence to exclude it. Still, she doesn't believe that ϕ is possible. But we can use this to make sense of certain acts of communication that Yalcin ([21]) subsumes under question sensitivity, but which we would like to distinguish from for reasons that will become clear later (cf. footnote 10). For instance, in well-known examples from [1], a person who is anxiously awaiting the results of John's cancer test may be disposed to say "I don't know whether John might have cancer, we're still waiting for the test results." Of course, John's having cancer is consistent and compatible with her information state, and she has considered the question, which means she is sensitive to it. In fact, the person could just as well say "Yes, John might have cancer, that's why they're running a test." But by saying she doesn't know whether John might have cancer, she portrays herself as accepting neither that he might nor that he might not have cancer; in our terms, this means that she presents herself as being in an information state where the range of probabilities assigned to John's having cancer is $[0, 1]$ (or $[0, 1)$). Similarly, a speaker who assigns probabilities $(0, 1]$ to ϕ is disposed to say that she doesn't know how probable ϕ is (at least if the interval is dense).

It could be suggested that possibility operators embedded under *know* have no semantic effect (at least at the level of at-issue content), so that an agent who says she doesn't know if something is possible is just expressing that she is ignorant as to whether it does in fact obtain. However, evidence for an analysis along the lines presented above comes from the fact that while the agent may, in a given situation, either say that something is possible or that she doesn't know if it's possible, she can't say both at the same time; she has to decide how to present herself:

---

[3] A reviewer suggested that this might just be a case of a familiarity presupposition of a proper name. But that is not an *alternative* explanation: the fact that his probability function is undefined for sentences involving the name is just an effect of the agent's being unfamiliar with it! In addition, it is a well-known fact from the literature of belief that it does often not seem appropriate to ascribe beliefs about a matter that the agent is simply not thinking of, even if she would have no trouble forming them were the proposition to come to his mind. See section 3 for some further discussion of this.

[4] Naturally, there is also higher-order vagueness, which will be ignored.

(1) ??John might have cancer, so they ran a test to rule that out. I haven't seen the results yet, so I don't know if he might have cancer.

Our analysis is similar in spirit to the treatment that the phenomenon receives in [17]. However, Willer, using a non-probabilistic system, has to introduce special-purpose machinery: for him, information states are sets of sets of worlds, i. e. sets of Stalnakerian contexts, which is not something that is readily interpretable. In contrast, in our probabilistic framework, we have the requisite technical apparatus in place because it also models the vagueness of probability. The only question that remains is what it means for an agent to assign to a proposition a vague probability represented by an interval that include both extreme (0 or 1) and intermediate values. Isn't there simply a fact of the matter about whether or not their evidence excludes the possibility? Yes, there is; and if pushed, the agent will probably agree that the proposition under discussion is a possibility after all. But by presenting herself, for the purpose of the conversation, as assigning to ϕ (e. g. that John has cancer) such a weird set of probability values, she expresses a disposition: should an issue come up the resolution of which depends on whether ϕ is possible or not (e. g. the question of whether certain precautions should be taken), the agent would not just assume the possibility, but rather try to get hold of the lab report before deciding anything. Only if she could not obtain the lab report would she resort to assuming that John's having cancer is, after all, possible, and proceed to action directly.[5] The crucial thing here is that an agent's information state only encodes her dispositions (or the dispositions she wants to portray herself as having) at a given time, and it may change not only as a result of utterances of others, but also be manipulated by processes *internal* to the agent. It is, however, very unclear to what extent such internal processes are amenable to logical modeling; they certainly aren't at *this* point.

2.3 Contradiction and the Search for Evidence

The reader may have been puzzled by the norm of contradiction given above; the usual way to state it is, of course, to say that a hearer may (or even must) contradict an assertion of ϕ if ϕ is inconsistent with their information state. There are, however, some cases where a hearer can neither accept an assertion nor contradict it. Assume that A assign's to John's having been there probabilities in the interval $[.6, .9]$.

(2) A: Smith is probably the murderer.
B: He MUST be.
A^1: Why?
A^2: ??I see.
A^3: ??No.

We can make sense of this if we assume that A is in an information state where the range of probabilities assigned to John's having been there (which we abbreviate as χ) is, say, $[.6, .8]$. If he updated with $\Box\chi$, he would end up with the absurd context, so he cannot just accept B's assertion. On the other hand, he cannot contradict it either, because his information state is, presumably, not incompatible with $\Box\chi$: from

[5] Note that in order for this to work technically, the person's probability estimate about the contents of the lab report also have to be totally vague. But that seems about right: she is not disposed to speculate about the lab report. Rather, if questions about its content come up, she would try to settle them by having a look at it.

an information state that accepts the negation of a modal formula, one can, in the usual case, reach one that accepts its unnegated version through update with certain non-modal formulae, i. e. conditionalization on the right kind of evidence. That is why, in the most typical case, the only route open to A is to *ask* for such evidence.

As predicted, unmodalized assertion behaves differently:

(3) A: John was probably there.
B: Yes, he was there.
A^1: I see.

To be sure, there are cases that can be updated with $\Box\phi$ directly, viz. if the hearer's information state contains a P such that $P(\phi) = 1$. In that case, we would expect her to just accept the assertion. Indeed, if a hearer is very weakly opinionated about ϕ, or thinks it possible that ϕ follows from what she knows, then such a reaction seems appropriate intuitively. Only when the hearer does have an opinion about the probability of ϕ, the question for evidence is necessary.

Of course, the classic kind of examples of "subjective" use of epistemic modals are covered by our theory as well, as well as the usual asymmetries.

(4) A: Your keys might be in the car.
B: No, they can't be, I still had them when we came into the house.
B': #Okay, but *I* know that they're not there.
(after [3])

Here, speaker A asserts a possibility on the basis of her own information state, and speaker B, following the norm of rejection, denies it. Her information state assigns probability 0 to the keys' being in the car, so she cannot update in any way to make A's assertion accepted in his information state, thus fulfilling the norm of rejection. What she cannot do is accept A's assertion and point out that she, on the other hand, knows better — such a reaction is just not sanctioned by the norms of conversation.

(5) a. A: John might be/is probably in the garden. B: He's not.
b. #A: John isn't in the garden. B: He might be.
etc.

Our system also includes the explanation from [6] for why the sequence in (5a) is infelicitous when uttered by a single speaker: it is not possible to be in an information state that accepts both of these statements, so the norm of assertion would be violated.[6] But with two speakers involved, there is no problem, because A can update with the information from B and end up in a state where she no longer accepts his initial assertion.

In (5b), this is not an option. Of course, (5b) is not an impossible sequence in actual discourse, but the point is that B is prompting A to *revise* her information state — by removing the information that John isn't in the garden — and not merely to update.

[6] Contra Yalcin ([22]), who considers sequences like (5a) dubious without taking into account the question of whether they are uttered by one or two speakers. That is why Yalcin is hesitant about recommending a system like the present one, where the update with non-modal formulae is non-eliminative, although his formalism could accommodate that in a way similar to our own.

2.4 A Word on Worlds

The technical setup we have seen might look very similar to that of [22]. However, we believe it is only as similar as any system that deals with the same phenomena will be. The crucial difference is that for Yalcin, information states are not sets of probability functions, but sets of pairs $\langle s, P\rangle$, where s is a set of worlds, the "live possibilities", and P is a probability function so that $P(s) = 1$. For him, the only expression that properly uses a probability function is *probably*; possibility and necessity modals merely quantify over the elements of s, and update with a non-modal formula restricts s (with concomitant conditionalization of P). By having suitable variation within a context, it is possible to reproduce the total vagueness we used to model a person who claimed not to know if something is possible. But insensitivity is now lost on us; or rather, it is restricted to *probable*. There is, however, no way that $\Diamond\phi$ or $\Box\phi$ could be undefined in Yalcin's system.

Yalcin's reason for sticking with a quantificational semantics for modals, is given in [18]: there are events that have, as a mathematical fact, probability 0 without being impossible: if we pick a number from the interval $[0, 1]$ completely at random, then the probability of that number being any particular number is 0; still, it is, by hypothesis, possible for every number to be picked. We have gone with [11] in identifying possibility with non-zero probability. Lassiter's reply to Yalcin's objection is that in natural language, there is always granularity. If we want to assess the probability of the randomly picked number being .5, they do not think of it as absolutely precise; there will always be some numbers that are indistinguishable from it. We can, as it were, be arbitrarily, but not infinitely, precise. Therefore, natural language statements are never about a continuous sample space in the intended way — only about arbitrarily precise sample spaces. We are sympathetic to this point of view; but we may note that, as a last resort, we would even be prepared to postulate infinitesimal probabilities in order to harvest the benefits of doing away with quantification over worlds. Primarily, this allows us the notion of insensitivity we have developed, which will be put to further use in the next section. A secondary reason to disfavor possible worlds is that they may eventually get in the way of avoiding logical omniscience (cf. [4], where probabilities are assigned directly to sentences).

3 Conditionals and Sensitivity

There is a well-known kind of conjunctions of conditionals which are not reversible, sometimes called *Sobel sequences*. The classic example is this:

(6) If the USA were to throw its nukes into the sea tomorrow, there would be war. Of course, if the USA and all the other superpowers were to throw their nukes into the sea tomorrow there would be peace.

(7) #If the USA and all the other superpowers were to throw their nukes into the sea tomorrow, there would be peace. Of course, if the USA were to throw its nukes into the sea tomorrow, there would be war.

Sobel sequences are generally discussed in the context of counterfactual conditionals, but they are just as possible with epistemic conditionals as with counterfactuals.[7] In

[7] The existence of indicative Sobel sequences is in fact acknowledged in [5].

a few thousand years, a historian researching the history of the twentieth century, and so oblivious of human nature that she actually considers the possibility of any country throwing its weapons into the sea, could utter:

(8) If the USA threw its nukes into the sea, there was war. Of course, if the USA and all the other superpowers threw their nukes into the sea, there was peace.

(9) #If the USA and all the other superpowers threw their nukes into the sea, there was peace. Of course, if the USA threw its nukes into the sea, there was war.

Intuitively, it is clear what happens here: at first, the speaker didn't even consider the possibility that all superpowers could throw their weapons into the sea. Then it comes to her attention and she corrects herself.

The traditional analysis (e. g. [2], [5]) is formulated in terms of domain of quantification: the domain of quantification for a conditional is not the whole set of possible worlds, but a certain subset not including any possibilities that one considers too remote or just forgets to think about, and while the domain can be widened in the course of a conversation, it cannot (at least not without significantly more effort) be narrowed, hence the irreversibility.[8]

Theories based on domain widening have a problem: they cannot distinguish between disregarding the possibility of ϕ (i. e. failure to consider it) and accepting that ϕ is impossible. Assume that in a context[9] c_1, a speaker has accepted *if* ϕ*,* ψ. At the same time, she is disposed to accept *if* ϕ *and* χ*, then* $\neg\psi$ (which is the situation in Sobel sequences), update with which would put her in the information state c_2. On the domain restriction theory, it follows that in c_1, no χ-worlds are in the relevant domain. But then in c_1, $\neg\Diamond\chi$ is accepted, and so is *if* ϕ*,* $\neg\chi$. But this is not what we want: a speaker who has failed to consider a possibility is not necessarily disposed to accept its negation.

Conversely, domain restriction theories predict that propositions deemed to be impossible can still feature as the antecedent of a subsequent conditional. But sequences like the following have a strong contradictory flavor:

(10) #The USA can't have thrown their weapons into the sea, because human nature wouldn't have allowed them to; of course, if they did, there was peace.

It suggests itself to apply the ideas in [8] to this, where pragmatic halos are assumed for domains of modal quantification. We do not doubt that the empirical connection between conditionals and imprecision is correct (cf. also [9]), but such an analysis is only as good as the pragmatic halos framework, which has trouble incorporating negation and providing an explanation for the directionality effects.

In our approach, there is a notion of insensitivity that captures the fact that disregarding a possibility is to disregard its negation and various combinations of it with

[8] Schulz's ([13]) analysis derives the non-monotonicity of counterfactual conditionals in a different way and fails to predict this dynamic, directional effect.

[9] Of course, in these theories, a context is something different than in ours.

others as well.[10] While we cannot analyze complete Sobel sequence, we believe that the manner of failure here is very interesting.

Our update rules do not change $\mathscr{S}$, which means that we cannot model *becoming* sensitive to a question in the course of a conversation. This is obviously inadequate. In fact, when a question is mentioned in a conversation that we weren't sensitive to, we adjust our information state often (though not always) silently. In our formalization, this would mean that if an agent is insensitive to ϕ and ϕ (or some modalized version of it) is mentioned, they extend $\mathscr{S}$ to the smallest superset of itself that contains $[\![\phi]\!]$ and still fulfills all the closure conditions that come with being part of a probability space. Of course, the probability functions in $\mathscr{P}$ would have to be altered so as to assign some value to ϕ. This could be done by just *extending* them (which will result in some constraints on the probability of the new ϕ), but this is quite obviously inadequate: upon becoming aware of a possibility, we sometimes *lower* the probability we assign to other possibilities. This is also what is needed in Sobel sequences. For assume that in $c, \vDash \phi \succ \psi$ is accepted and ϕ and ψ are non-modal formulae, so that $\mathscr{P}(\psi|\phi) = 1$. As long we are not allowed to alter existing probability assignments, that will not change no matter what probability we assign to the new possibility χ, since $\mathcal{P}(\phi|\phi \wedge \chi) = 1$, and so $\mathcal{P}(\psi|\phi \wedge \chi)$. But the conditional that prompted us to become sensitive to χ was $(\phi \wedge \chi) \succ \psi$, with which our context is then inconsistent.

What is missing is a representation of the background knowledge on the basis of which we alter probability assignments when we become aware of an additional possibility; presumably the kind of information that is encoded in *generic* conditionals.[11] The modeling of this is a challenging task: one may suspect with [10] that this is actually an issue of activation in the associative memory of the speaker/hearer.

4 Further Limitations and Outlook

There are other issues we cannot treat with our simplistic approach; in particular, the evidential effect of modals. While $\Box\phi$ and ϕ have different update effects and deniability conditions, they still have identical acceptance conditions, and so should be assertable by an agent in the same circumstances (and be interchangeable under attitude operators and in the consequent of conditionals). However, as noted in [15], this is not true:

(11) Context: *Rain is falling outside and the agent is looking out of the window.*
  a. It's raining outside.
  b. #It must be raining outside.

Evidential effects are equally present under attitude operators:

(12) a. Peter believes that aliens were here.

[10] This is the reason why we want to distinguish insensitivity to ϕ from $\mathscr{P}(\phi) = [0, 1]$. The first conjunct of a Sobel sequence behaves, in a way, as if the probability of the disregarded possibility were 0; this couldn't be captured if insensitivity were just total vagueness of the probability. On our conception, however, insensitivity does not survive even mention of the possibility (and neither do conditionals survive the mention of an exception!), which makes it unfit for the explanation of the DeRose kind of examples.

[11] Of course, one could just use an ordered set of contexts with successively larger event spaces $\mathscr{S}$, but that would be entirely uninteresting.

b. Peter believes that aliens must have been here.

(12b) clearly implies that Peter's belief about the presence of aliens is inferred, and that he neither actually saw, nor, indeed, hallucinated them. (12a) is neutral about either possibility. That $\Box\phi$ and ϕ are therefore not in complementary distribution either is what makes the incorporation of the evidential impact of modals very non-trivial. One can of course extend information states to include a special body of directly evidenced propositions, but the intricate interplay of implicatures and presuppositions with respect to these will then have to be explored. This we leave to future research as well.

In summary, we have demonstrated that the probabilistic dynamic approach has a number of useful features. Modeling an information state as a vague probability space over possible worlds, we have provided a conspicuous picture of the workings of modals in conversation regarding acceptance, rejection and to some extent the information conferred by them, and makes use of the formal possibility of a totally vague probability distribution to model an agent who presents herself having no commitment at all as to the probability of a proposition. Furthermore, a superior account of question sensitivity falls out directly from the formal tools that are employed.

We have reached the limits of our simple model at two points, giving rise to questions for further research. First, the evidential effect of modals is not captured, which points to the necessity of extending information states to designate some propositions of special status so that evidential presuppositions may be formulated. Second, while we have seen that our notion of insensitivity is in fact superior to that employed in standard accounts of Sobel sequences of conditionals, we have not ourselves provided an analysis of these. It turned out that in order to treat the dynamics of sensitivity that are involved, a representation of generic probabilistic knowledge is needed, and it stands to reason that this ultimately leads us deep into cognitive science.

References

1. DeRose, K.: Epistemic possibilities. The Philosophical Review 100, 581–605 (1991)
2. von Fintel, K.: Counterfactuals in a dynamic context. In: Kenstowicz, M. (ed.) Ken Hale: A Life in Language. MIT Press, Cambridge (2001)
3. von Fintel, K., Gillies, A.: 'might' made right. In: Weatherson, B., Egan, A. (eds.) Epistemic Modality. Oxford University Press (2011)
4. Gaifman, H.: Reasoning with limited resources and assigning probabilities to arithmetical statements. Synthese 140, 97–119 (2004)
5. Gillies, A.: Counterfactual scorekeeping. Linguistics and Philosophy 30, 329–360 (2007)
6. Groenendijk, J., Stokhof, M., Veltman, F.: Coreference and modality in multi-speaker discourse. In: Kamp, H., Partee, B. (eds.) Context Dependence in the Analysis of Linguistic Meaning. I. M. S., Stuttgart (1997)
7. Joyce, J.M.: How probabilities reflect evidence. Philosophical Perspectives 19, 153–178 (2005)
8. Klecha, P.: Shifting modal domains: An imprecision-based account. Paper presented at LSA 2012 (2012)
9. Križ, M., Schmitt, V.: Semantic slack in plural predication. Talk at the Semantics Workshop, Vienna (May 2012)
10. van Lambalgen, M., Stenning, K.: Human Reasoning and Cognitive Science. MIT Press (2008)

11. Lassiter, D.: Measurement and Modality: The Scalar Basis of Modal Semantics. Ph.D. thesis, NYU (2011)
12. Schmitt, J.: Iffy Confidence. Ph.D. thesis, USC (2012)
13. Schulz, K.: Minimal Models in Semantics and Pragmatics: Free Choice, Exhaustivity, and Conditionals. Ph.D. thesis, University of Amsterdam (2007)
14. Stalnaker, R.: A defense of conditional excluded middle. In: Stalnaker, R., Harper, W., Pearce, G. (eds.) Ifs: Conditionals, Belief, Decision, Chance and Time, pp. 87–104. Reidel, Dordrecht (1981)
15. Swanson, E.: How not to theorize about the language of subjective uncertainty. In: Weatherson, B., Egan, A. (eds.) Epistemic Modality. Oxford University Press (2011)
16. Veltman, F.: Defaults in update semantics. Journal of Philosophical Logic 25(3), 221–261 (1996)
17. Willer, M.: Dynamics of epistemic modality. Philosophical Review (forthcoming)
18. Yalcin, S.: Epistemic modals. Mind 116, 983–937 (2007)
19. Yalcin, S.: Modality and Enquiry. Ph.D. thesis, MIT (2008)
20. Yalcin, S.: Probability operators. Philosophy Compass 5(11), 916–37 (2010)
21. Yalcin, S.: Nonfactualism about epistemic modality. In: Weatherson, B., Egan, A. (eds.) Epistemic Modals. Oxford University Press (2011)
22. Yalcin, S.: Context probabilism. Proceedings of the 18th Amsterdam Colloquium (2012)

Computing with Numbers, Cognition and the Epistemic Value of Iconicity

José Gustavo Morales

National University of Córdoba & National Research Council (CONICET)

Abstract. *In my paper I will rely on research by Grosholz (2007) considering, in particular, her thesis of the irreducibility of iconic representation in mathematics. Against this background, my aim is to discuss the epistemic value of iconicity in the case of different representations of numbers in elementary arithmetic. In order to make my point, I will bring in a case study selected from Leibniz's work with notations and the lessons Leibniz draws in the context of his number-theoretic considerations.*

Keywords: Iconic Representation, Notations, Visual Reasoning, Numbers, Epistemic Value

1 Introduction

According to the standard view in twentieth century philosophy of logic and mathematics, reasoning in the formal sciences is best characterized as purely syntactic so that 1) the body of mathematical knowledge can be seen as built up systematically and 2) a purely syntactic presentation guarantees formal rigor and transparency by making *explicit* all relevant epistemic contents in proving mathematical results. Full-explicitness goes hand in hand with the elimination of any reference to the context of work.[1] From such assumptions follows that figures and, more generally, iconic ingredients are to be eliminated from a fully systematic presentation.

This view of the role of representation in the formal sciences has been called into question by a number of recent investigations. According to a critical line of research developed by Grosholz (2007), the research mathematician engages in formal reasoning that is broadly conceived as sets of problem-solving activities which make use of an irreducible variety of modes of representation or working tools such as notations, tables, figures, etc. but also discursive reasoning in natural language. Such modes of representation depend on the specific context of work as well as acquired cognitive abilities of the agent which include background knowledge that remains largely *implicit*. As the critics point out "syntactic reconstructions" of formal reasoning may be complemented by formal semantics but the price to pay is the elimination of discursive language and forms of know-how relevant to the successful use of notations and other working tools, as well as rich dialogical aspects of mathematical practice leading to innovation. In particular, the syntactic view requires that figures be eliminated in favor of a one dimensional formal reconstruction which is purely symbolic.

[1]For a discussion of the standard view see [6], [5], and [4].

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see http://ceur-ws.org/, 2012, pp. 84–92.

2 Aim of My Paper

In my paper I will rely on research by Grosholz (2007) considering, in particular, her thesis of the irreducibility of iconic representation in mathematics. Against this background, my aim is to discuss the epistemic value of iconicity in the case of different representations of numbers in elementary arithmetic. In order to make my point, I will bring in a case study selected from Leibniz's work with notations and the lessons Leibniz draws in the context of his number-theoretic considerations. The reason for my selection of this particular case study is twofold. Firstly, throughout his work as a mathematician, Leibniz emphasizes the importance of visual reasoning while relying on a variety of tools which display rich iconic aspects in the implementation of problem-solving activities. Secondly, Leibniz discusses the peculiar iconicity of representations of numbers; in particular, he illuminates the issue of the epistemic value of different numerical systems by discussing the cognitive benefits of the binary system vis-à-vis the system of Arabic numerals. For Leibniz some notations are more fruitful than others, moreover, simplicity and economy is amongst the epistemic virtues of notational systems. In the case under consideration – my case study – Leibniz argues for the view that the iconic aspects present in binary notation reveal structural relations of natural numbers that are concealed in other numerical modes of representation such as the system of Arabic numerals.

3 The Idea of Iconic Representation

Let me start by focusing on the idea of iconic representation. Representations may be iconic, symbolic and indexical depending upon their role in reasoning with signs in specific contexts of work.[2] According to the received view representations are iconic when they resemble the things they represent. In the case of arithmetic this characterization appears as doubtful because of its appeal to a vague idea of similarity which would seem untenable when representations of numbers are involved. But Grosholz argues that in mathematics iconicity is often an irreducible ingredient, as she writes,

> In many cases, the iconic representation is indispensable. This is often, though not always, because shape is irreducible; in many important cases, the canonical representation of a mathematical entity is or involves a basic geometrical figure. At the same time, representations that are 'faithful to' the things they represent may often be quite symbolic, and the likenesses they manifest may not be inherently visual or spatial, though the representations are, and articulate likeness by visual or spatial means [6, p. 262].

In order to determine whether a representation is iconic or symbolic, the discursive context of research needs to be taken into account in each particular case, in other words, iconicity cannot simply be read off the representation in isolation of the context of use. We find here a more subtle understanding of "iconicity" than the traditional view. Let me focus on the idea that representations "articulate likeness by visual or spatial means" in the case of arithmetic. Grosholz suggests that even highly abstract symbolic reasoning goes hand in hand with certain forms of visualizations. To many this sounds polemical at best. Granted to the critic that "shape is irreducible" in

[2]This tripartite distinction goes back to Charles Peirce's theory of signs. For a discussion of the distinction, see [6, p. 25].

geometry as it is the case with geometrical figures. But what is the role of iconicity in the representation of numbers, and more generally, what is involved in visualizing in arithmetic?

Giardino (2010) offers a useful characterization of the cognitive activity of "visualizing" in the formal sciences. In visualizing, she explains, we are decoding articulated information which is embedded in a representation, such articulation is a specific kind of spatial organization that lends unicity to a representation turning it intelligible. In other words, spatial organization is not just a matter of physical display on the surface (paper or table) but "intelligible spatiality" which may require substantial background knowledge:

> (...) to give a visualization is to give a contribution to the organization of the available information (...) in visualizing, we are referring also to background knowledge with the aim of getting to a global and synoptic representation of the problem [3, p. 37].

According to this perspective, the ability to read off what is referred to in a representation depends on some background knowledge and expertise of the reader. Such cognitive act is successful only if the user is able to decode the encrypted information of a representation while establishing a meaningful relationship between the representation and the relevant background knowledge which often remains implicit. The starting poing of this process is brought about by representations that are iconic in a rudimentary way, namely, they have spatial isolation and organize information by spatial and visual means; and they are indivisible things. Borrowing Goodman's terms we may say that representations have 'graphic suggestiveness'.

4 The Role of Iconicity in the Representation of Numbers

Against this background, I will bring in my case study in order to consider the role that iconicity plays in the representation of natural numbers both by Arabic numerals $(0, 1, \ldots, 9)$ and by binary numerals $(0, 1)$. In a number of fragments, Leibniz discusses both notational systems. He compares them with regard to usefulness for computation and heuristic value leading to discovery of novelties. I begin by asking about the interest in choosing a particular representation of numbers in the context of arithmetical problem-solving activities. Once more, I will rely here on the Leibnizian view as discussed by Grosholz. According to this view the use of different modes of representation in the formal sciences allows us to explore "productive and explanatory conditions of intelligibility for the things we are thinking about" [7, p. 333].[3] The objects of study of mathematics are abstract ("intelligible"), but they are not transparent but problematic, and they are inexhaustible requiring mathematical analysis which will clarify and further develop conceptual structures aided by the appropriate working tools.[4] In the case of number-theoretic research different modes of representation may reveal different aspects of things leading to discovery of new properties and the design of more elaborated tools.

[3]The roots of this perpective are, as Grosholz recognizes, in Leinbiz's theory of expression developed around 1676-1684.

[4]See [6, p. 47 and p. 130].

5 Representation of Numbers

From this perspective, let us now consider the case of the representation of natural numbers. A natural number is, Grosholz writes, "either the unit or a multiplicity of units in one number" [6, p. 262]. Following this idea a very iconic representation of six could look like this:

$$//////$$

On the one hand, this "picture" contains – or shows – the multiplicity of units contained in the number six. On the other hand, the unity of the number six is expressed by some strategy of differentiation from the rest of objects surrounding it on the page (or surface). In this particular case, the six strokes are spatially organized to achieve this aim. But this works only with small numbers and if we want to represent, for instance, the number twenty-four, the iconicity of the strokes collapses partly because the reader cannot easily visualize the number that was meant to be thereby represented. In opposition to this rudimentary representation of numbers by means of strokes, take the Arabic numeral "6" which does not reveal the multiplicity of units contained in the natural number six but exhibits instead the unity of the number itself. Arabic numerals exhibit each number as a unity and just for this reason they are iconic too. If what is at stake are big numbers, representation by strokes becomes useless and the weaker iconicity of Arabic numerals appears as more productive in problem solving activities such as basic computing with numbers. In other words, iconicity is a matter of degree depending on each context of work as well as background knowledge relevant to the problem-solving activity. When it comes to arithmetical operations the "graphic suggestiveness" of strokes may not be the last word while the "maneuverability" of Arabic numerals seems more decisive as it allows us to operate with precision and easiness. In Arabic notation each individual mark is "dense" in the sense that in each character there is a lot of information codified in highly compressed way.

6 Leibniz and His Preference for the Binary System

In this section, I will look at Leibniz's discussion of the binary notation broadly exposed in "Explication de l'arithmetique binaire" (1703). I consider this case study of great interest because it brings to light specific aspects which are central to the topic of my paper, in particular, the epistemic value of iconicity in the representation of number systems in specific problem-solving contexts of work. Leibniz emphasizes what he sees as the cognitive virtues of his favorite notational system in arithmetic, the binary notation. This notation displays some properties of numbers by means of only two characters, namely, 0 and 1 and the following four rules: $1+1=10, 1+0=1$ (addition) and $1 \cdot 1 = 1, 1 \cdot 0 = 0$ (multiplication). In binary notation when we reach two, we must start again; thus, in this notation two is expressed by 10, four by 100, eight by 1000 and so on.

Time and again Leibniz points to the values of this binary notation, economy and simplicity of the system. All of arithmetic can be expressed by only two characters and a few rules for the manipulation of them. Having presented the law for the construction of the system, Leibniz explains the benefits of it comparing it with the more familiar decimal system:

> Mais au lieu de la progression de dix en dix, j'ai employé depuis plusieurs années la progression la plus simple de toutes, qui va de deux en deux; ayant trouvé qu'elle sert à la perfection de la science des Nombres [10, Vol. VII, p. 223].

For Leibniz, the economy and simplicity of his binary system seems to run contrary to the decimal system of Arabic numeration. Simplicity and easiness go hand in hand, as in every operation with binary notation the elements of the system are made fully explicit, while in Arabic notation we must always appeal to memory:

> Et toutes ces opérations sont si aisées (...) [o]n n'a point besoin non plus de rien apprendre par coeur ici, comme il faut faire dans le calcul ordinaire, ou il faut scavoir, par exemple, que 6 et 7 pris ensemble font 13; et que 5 multiplié par 3 donne 15 (...) Mais ici tout cela se trouve et se prouve de source... [10, Vol. VII, p. 225].

Consider the case of three multiplied by three. In order to solve this case by means of the decimal system, Leibniz argues, we need to appeal to memory; we must recall the multiplication table for 3 which gives us the correct outcome, and the same goes for any numeral of the Arabic system from 0 to 9. In contrast, the same operation made within the binary system always makes explicit all applications of the rules for any operation we perform. In this case we do not need to rely upon memory but only on the combination of characters fully deployed on the page. Thus, in decimal we know by memory that $3 \cdot 3 = 9$ while in binary notation we "see" it (Fig. 1), where "11" expresses the natural number three and "1001" stands for the number nine.

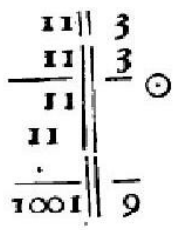


Fig. 1. Leibniz, *Mathematische Schriften*, VII, ed. Gerhardt, p. 225.

Moreover, Leibniz insists, the binary system reveals structural relations among characters, and in the same text "Explication de l'arithmetique binaire", Leibniz goes on to note that in virtue of the economy and simplicity of the binary system we are able to easily *visualize* structural relations between numbers thus uncovering novelties. The example he presents here to illustrate his point is that of geometric progression of ratio two:

> On voit ici d'un coup d'oeil la raison d'une *propriété célébre de la progression Géométrique double* en Nombres entiers... [10, Vol. VII, p. 224, italics included in quoted edition].

Let us take the following geometric progression "deux-en-deux" of natural numbers $7, 14, 28$. Next, we express those numbers as sums of powers of two. Accordingly, within the decimal system of Arabic numerals we then have the following progressions:

a)
$$4 + 2 + 1 = 7$$

b)
$$8 + 4 + 2 = 14$$

c)
$$16 + 8 + 4 = 28$$

Finally, we proceed to decompose a), b) and c) into powers of two:

a')
$$2^0 + 2^1 + 2^2 = 2^0 \cdot 7$$

b')
$$2^1 + 2^2 + 2^3 = 2^1 \cdot 7$$

c')
$$2^2 + 2^3 + 2^4 = 2^2 \cdot 7$$

In the first case, the three lines do not provide any information about the pattern that lead to a), b), c). Instead, those lines require familiarity with all elements of the system as well as familiarity with the operation in question (addition).

Similarly in the second case, the three lines do not provide any information about the pattern underlying the progression. In each of the three lines, the right side of a'), b'), c') does not indicate the outcome. We must find out how to express the outcome as power of two. In all three cases we find that the outcome cannot be expressed as power of two and that it is therefore necessary to introduce a new element: the factor 7. Of course, we must know the multiplication table for seven as well.

Let us now go back to the binary notation and consider how the spatial distribution of a), b), c) is expressed by binaries (see Fig. 2).

In opposition to the decimal system of Arabic numerals, within the binary system it is unnecessary to analyze the case in two separate steps. This is so because the characters and the order they exhibit on the page make visible the pattern underlying the progression. We only need to know the rules for addition and the system characters ("0" and "1").

Finally, Leibniz points to another feature of binaries in relation with the construction of the system. It is the simplicity and economy of the binary that according to him brings forth a remarkable periodicity and order. In making this point the author again emphasizes visual aspects and spatial configuration of characters:

> (...) les nombres étant réduits aux plus simples principes, comme 0 & 1, il paroît partout un ordre merveilleux. Par exemple, dans la *Table* même *des Nombres*, on voit en chaque colonne régner des périodes qui recommencent toujours [10, Vol. VII, p. 226, italics included in quoted edition].

100	1000	10000
10	100	1000
1	10	100
——	——	——
111	1110	11100

Fig. 2. Geometric progression of ratio two as expressed by binaries.

Leibniz groups together numbers that fall under $2^1, 2^2, 2^3$, etc. I include below a segment of a larger table used by Leibniz to show three groups of numbers (surrounded by vertical and horizontals lines), namely, 0, 1; 2, 3 and 4 – 7. Each group is a cycle which is iterated in the next cycle, and so on ad infinitum as we can easily see.[5]

7 Conclusion

For Leibniz mathematical research starts with the search for suitable signs ("characters") and the design of good notations (or "characteristics") by means of which structural relations of intelligible objects of study could be explored; a good "characteristic" should allow us to uncover different aspects of things by means of a sort of reasoning with signs. When Leibniz remarks this in a brief text of 1683-1684 his example of a "more perfect characteristic" is the binary notation vis-à-vis the decimal system.[6] In order to understand Leibniz preference for the binary system, we recall here, it is useful to focus on the importance of visual reasoning in problem-solving contexts of work. According to Leibniz, problem-solving activities and the discovery of new properties is the goal of mathematical analysis in the case of number theory, one of the areas of research he was most interested in pursuing. Such goal strongly motivates the design of notational systems with a view to obtain fruitful results. As already pointed out, for Leibniz the binary system is characterized by its simplicity and economy so that in each operation every element ("0", "1") is displayed for the eye to see without any need to rely on memory as in the case of operating with Arabic numerals. Leibniz observed that in certain contexts of work the spatial distribution of such binary elements reveals patterns which are relevant to the resolution of the prob-

[5] [10, Vol. VII, p. 224].

[6] Exempli gratia perfectior est característica numerorum bimalis quam decimales vel alia quaecunque, quia in bimali – ex characteribus – omnia demonstrari possunt quae de numeris asseruntur, in decimali vero non item. [2, p. 284].

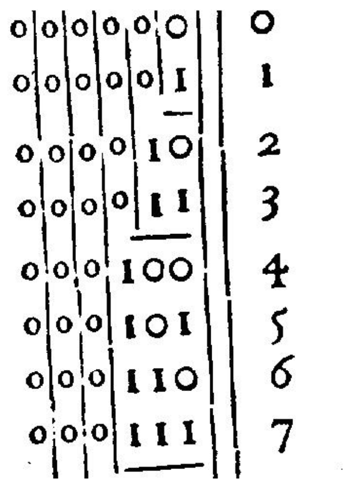


Fig. 3. Leibniz, *Mathematische Schriften*, VII, ed. Gerhardt, p. 225.

lem under consideration, or to the discovery of new properties that would otherwise remain hidden. Such is the case of the geometrical progression "deux-en-deux", which as we have just noted, can be easily seen only when expressed by means of the binary system. No doubt that for practical considerations of everyday life the decimal system of Arabic numerals may be easier to calculate with, nonetheless Leibniz was fascinated by the binary as facilitating algorithmic structures which like the calculus engaged the issue of infinite series.

To conclude, one of the things we can learn from my case study is that in computing with numbers – a way of "reasoning with signs" – we always require systems of signs or characters but some of them are more fruitful than others, some are easier to calculate with but beyond the specific epistemic virtues they may have, all of them include important iconic features that are most relevant to cognition. This conclusion, in particular, calls into question the old idea that working with algorithmic structures – computing with numbers – is a purely mechanical affaire which excludes iconicity.

Acknowledgments.

I should like to thank Norma Goethe for extended comments and suggestions. Many thanks go also to two anonymous referees for helpful observations.

References

1. Bender, A.: Fingers as a Tool for Counting – Naturally Fixed or Culturally Flexible? *Front. Psychology* 2:256. doi: 10.3389/fpsyg.2011.00256 (2011)
2. Couturat, L.: *Leibniz, Opuscules et Fragments Inédits.* Paris, Alcan (1903). Reprinted by Olms, Hildesheim (1961)
3. Giardino, V.: Intuition and Visualization in Mathematical Problem Solving. Published online: 9 February 2010 – Springer Science+Business Media B. V. Topoi, 29:29–39 (2010)

4. Goethe, N. B., Friend, M.: Confronting Ideals of Proof with the Ways of Proving of the Research Mathematician. Studia Logica 96 (2), Dordrecht: Springer, p. 273–288 (2010)
5. Goethe, N. B.: Revisiting the Question about Proof: Philosophical Theory, History, and Mathematical Practice. Manuscrito, Vol. 31 (1), CLE, Campinas S.P. p. 361–386 (2008)
6. Grosholz, E.: *Representation and productive ambiguity in mathematics and the sciences.* Oxford New York (2007)
7. Grosholz, E.: Reference and Analysis: The Representation of Time in Galileo, Newton, and Leibniz. *Arthur O. Lovejoy Lecture* , Journal of the History of Ideas, Volume 72, Number 3 (July 2011) pp. 333–350 (2010)
8. Ippoliti, E.: *Inferenze Ampliative. Visualizzazione, Analogia e Rappresentazioni Multiple.* Morrisville, North Carolina, USA (2008)
9. Lakoff, G., Núñez.: *Where Mathematics Comes From. How the Embodied Mind Brings Mathematics into Being.* Basic Books, The Perseus Books Group (2000)
10. Leibniz GW.: E.: *Mathematische Schriften.* Vol. I – VII (1849–1863). Reprinted by Olms, Hildesheim (1962)

The Force of Innovation
Emergence and Extinction of Messages in Signaling Games

Roland Mühlenbernd[1], Jonas David Nick[1], and Christian Adam[1]

University of Tübingen

Abstract. Lewis [L1] invented *signaling games* to show that meaning convention can arise simply from regularities in communicative behavior. The precondition for the emergence of such conventions are so-called *perfect signaling systems*. In a series of articles the emergence of such signaling systems was addressed by combining signaling games with learning dynamics; and not uncommonly researchers examined the circumstances aggravating the emergence of perfect signaling. It could be shown that especially by increasing the number of states, messages and actions for a signaling game perfect signaling becomes more and more improbable. This paper contributes to the question how the capability of innovation through emergence of new messages and extinction of unused messages would change these outcomes. Our results show that innovation in fact supports the emergence of perfect signaling.

1 Introduction

With the objective to explore the evolution of semantic meaning, signaling games recently became a leading model for this purpose. In line with this trend researchers used simulations to explore agents' behavior in repeated signaling games. Within this field of study two different lines of research are apparent: i) the simulation of a repeated 2-player signaling game combined with agent-based learning dynamics, in the majority of cases with the dynamics *reinforcement learning* (e.g. [B1], [BZ1], [S1]) and ii) evolutionary models by simulating population behavior, wherein signaling games are usually combined with the population-based *replicator dynamics* (e.g. [HH1], [HSRZ1]). To fill the gap between both accounts, recent work deals with applying repeated signaling games combined with agent-based dynamics on social network structures or at least multi-agents accounts (e.g. [Z1], [W1], [M1], [MF1]). With this paper we want to make a contribution to this line of research.

Barrett ([B1]) could show that i) for the simplest variant of a signaling game, called *Lewis game*, combined with a basic version of the learning dynamic *reinforcement learning* in a 2-player repeated game conventions of meaningful language use emerge in any case, but ii) by extending the domains[1] of the signaling game those conventions become more and more improbable. Furthermore the number of possible perfect signaling systems increases dramatically. This let surmise the motive that up to now researchers applied only the simple variant Lewis game on population and keep the hands off domain-extended signaling games. Because if even two players fail to learn

[1] With domains we refer to the sets of states, messages and action, which will be introduced in the following section.

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see http://ceur-ws.org/, 2012, pp. 93–??.

perfect signaling from time to time, multiple players will not only have this problem, but also be confronted with an environment evolving to Babylon, where a great many of different signaling systems may evolve.

With this article we will show that by extending the learning dynamics to allow for innovation we can observe i) an improvement of the probability that perfect signaling emerges for domain-extended signaling games and ii) a restriction of the number of evolving perfect signaling systems in a population, even if the number of possible systems is huge. This article is divided in the following way: in Section 2 we'll introduce some basic notions of repeated signaling games, reinforcement learning dynamics and multi-agent accounts; in Section 3 we'll take a closer look at the variant of reinforcement dynamics we used - a derivative of Bush-Mosteller reinforcement; Section 4 is about how implementing innovation of new and extinction of unused messages significantly improves our results; we'll finish with some implications of our approach in Section 5.

2 Signaling Games and Learning

A signaling game $SG = \langle \{S, R\}, T, M, A, Pr, U \rangle$ is a game played between a sender S and a receiver R. Initially, nature selects a state $t \in T$ with prior probability $\Pr(t) \in \Delta(T)^2$, which the sender observes, but the receiver doesn't. S then selects a message $m \in M$, and R responds with a choice of action $a \in A$. For each round of play, players receive utilities depending on the actual state t and the response action a. We will here be concerned with a variant of this game, where the number of states is on par with the number of actions ($|T| = |A|$). For each state $t \in T$ there is exactly one action $a \in A$ that leads to successful communication. This is expressed by the utility function $U(t_i, a_j) = 1$ if $i = j$ and 0 otherwise. This utility function expresses the particular nature of a signaling game, namely that because successful communication doesn't depend on the used message, there is no predefined meaning of messages. A signaling game with n states and n messages is called an $n \times n$-game, whereby n is called the *domain* of the game.

2.1 Strategies and Signaling Systems

Although messages are initially meaningless in this game, meaningfulness arises from regularities in behavior. Behavior is defined in terms of strategies. A *behavioral sender strategy* is a function $\sigma : T \to \Delta(M)$, and a *behavioral receiver strategy* is a function $\rho : M \to \Delta(A)$. A behavioral strategy can be interpreted as a single agent's probabilistic choice or as a population average. For a 2×2-game, also called Lewis game, exactly two isomorphic strategy profiles constitute a perfect signaling system. In these, strategies are pure (i.e. action choices have probabilities 1 or 0) and messages associate states and actions uniquely, as depicted in Figure 1.

It is easy to show that for an $n \times n$-game the number of perfect signaling systems is $n!$. This means that while for a Lewis game we get the 2 signaling systems as mentioned above, for a 3×3-game we get 6, for a 4×4-game 24, and for a 8×8-game more than 40,000 perfect signaling systems. Moreover for $n \times n$-games with $n > 2$ there is a possibility of partial *pooling equilibria*, which transmit information in a fraction of all possible cases.

[2] $\Delta(X) : X \to \mathbb{R}$ denotes a probability distribution over random variable X.

$$L_1: \begin{tikzcd} t_1 \arrow[r] & m_1 \arrow[r] & a_1 \\ t_2 \arrow[r] & m_2 \arrow[r] & a_2 \end{tikzcd} \qquad L_2: \begin{tikzcd} t_1 \arrow[dr] & m_1 \arrow[dr] & a_1 \\ t_2 \arrow[ur] & m_2 \arrow[ur] & a_2 \end{tikzcd}$$

Fig. 1. Two perfect signaling systems of a 2×2-game, consisting of a pure sender and receiver strategy.

2.2 Models of Reinforcement Learning

The simplest model of reinforcement learning is *Roth-Erev reinforcement* (see [RE1]) and can be captured by a simple model based on urns, known as *Pólya urns*, which works in the following way: an urn contains balls of different types, each type corresponding to an action choice. Now drawing a ball means to perform the appropriate action. An action choice can be successful or unsuccessful and in the former case, the number of balls of the appropriate act will be increased by one, such that the probability for this action choice is increased for subsequent draws. All in all this model ensures that the probability of making a particular decision depends on the number of balls in the urn and therefore on the success of past action choices. This leads to the effect that the more successful an action choice is, the more probable it becomes to be elected in following draws.

But Roth-Erev reinforcement has the property that after a while the learning effect[3] slows down: while the number of additional balls for a successful action is a static number α, in the general case $\alpha = 1$, as mentioned above, the overall number of balls in the urn is increasing over time. E.g. if the number of ball in the urn at time τ is n, the number at a later time $\tau + \epsilon$ must be $m \geq n$. Thus the learning effect is changing from $^{\alpha}/_{n}$ to $^{\alpha}/_{m}$ and therefore can only decrease over time.

Bush-Mosteller reinforcement (see [BM1]) is similar to Roth-Erev reinforcement, but without slowing the learning effect down. After a reinforcement the overall number of balls in an urn is adjusted to a fixed value c, while preserving the ratio of the different balls. Thus the number of balls in the urn at time τ is c and the number at a later time $\tau + \epsilon$ is c and consequently the learning effect stays stable over time at $^{\alpha}/_{c}$.

A further modification is the adaption of *negative reinforcement*: while in the standard account unsuccessful actions have no effect on the urn value, with negative reinforcement unsuccessful communication is punished by decreasing the number of balls leading to an unsuccessful action.

By combining Bush-Mosteller reinforcement with negative reinforcement, the resulting learning dynamic follows the concept of *lateral inhibition*. In particular, a successful action will not only increase its probability, but also decrease the probability of competing actions. In our account lateral inhibition applies to negative reinforcement as well: for an unsuccessful action the number of the appropriate balls will be decreased, while the number of each other type of ball will be increased.

[3] The learning effect is the ratio of additional balls for a successful action choice to the overall number of balls.

2.3 Applying Reinforcement Learning on Repeated Signaling Games

To apply reinforcement learning to signaling games, sender and receiver both have urns for different states and messages and make their decision by drawing a ball from the appropriate urn. We assume the states are equally distributed. The sender has an urn $\mho_t$ for each state $t \in T$, which contains balls for different messages $m \in M$. The number of balls of type m in urn $\mho_t$ designated with $m(\mho_t)$, the overall number of balls in urn $\mho_t$ with $|\mho_t|$. If the sender is faced with a state t she draws a ball from urn $\mho_t$ and sends message m, if the ball is of type m. Accordingly the receiver has an urn $\mho_m$ for each message $m \in M$, which contains balls for different actions $a \in A$, whereby the number of balls of type a in urn $\mho_m$ designated with $a(\mho_m)$, the overall number of balls in urn $\mho_m$ with $|\mho_m|$. For a receiver message m the receiver draws a ball from urn $\mho_m$ and plays the action a, if the ball is of type a. Thus the sender's behavioral strategy σ and receiver's behavioral strategy ρ can be defined in the following way:

$$\sigma(m|t) = \frac{m(\mho_t)}{|\mho_t|} \quad (1) \qquad \rho(a|m) = \frac{a(\mho_m)}{|\mho_m|} \quad (2)$$

The learning dynamics are realized by changing the urn content dependent on the *communicative success*. For a Roth-Erev reinforcement account with a positive update value $\alpha \in \mathbb{N} > 0$ and a lateral inhibition value $\gamma \in \mathbb{N} \geq 0$ the following update process is executed after each round of play: if communication via t, m and a is successful, the number of balls in the sender's urn $\mho_t$ is increased by α balls of type m and reduced by γ balls of type $m' \neq m$. Similarly, the number of balls in the receiver's urn $\mho_m$ is increased by α balls of type a and reduced by γ balls of type $a' \neq a$.

Furthermore for an account with negative reinforcement urn contents also change in the case of unsuccessful communication for the negative update value $\beta \in \mathbb{N} \geq 0$ in the following way: if communication via t, m and a is unsuccessful, the number of balls in the sender's urn $\mho_t$ is decreased by β balls of type m and and increased by γ balls of type $m' \neq m$; the number of balls in the receiver's urn $\mho_m$ is decreased by β balls of type a and increased by γ balls of type $a' \neq a$. The lateral inhibition value γ ensures that the probability of an action can get zero and it speeds up the learning process.

We extended the Bush-Mosteller reinforcement for applying it to games with more than two messages. The content of the appropriate sender and receiver urns will be adjusted to a predefined value in the following way: for the given value c of fixed urn content it is assumed that before a round of play the urn content of all sender and receiver urns $|\mho| = c$. After a round of play it may be the case that the urn content $|\mho| = d \neq c$. Now the number n_i of each type of ball i is multiplied by c/d.[4] For two messages the Bush-Mosteller is equivalent to our extension by setting the learning parameter of the original model to $\phi = \frac{c \cdot \alpha}{c + \alpha}$.

2.4 Multi-Agent Accounts

It is interesting not only to examine the classical 2-players sender-receiver game, but the behavior of agents in a society (e.g. [Z1], [W1], [M1], [MF1]): more than 2 agents interact with each other and switch between sender and receiver role. In this way an agent can learn a sender and a receiver strategy as well. Now if such a combination forms a signaling system, it is called a *signaling language* and the corresponding agent

[4] In this account urn contents and numbers of balls are real numbers

Game	RFR
2×2	0%
3×3	9.6%
4×4	21.9%
8×8	59.4%

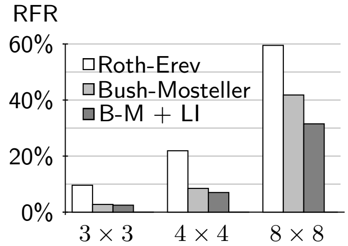


Fig. 2. *Left:* Barrett's results for different $n \times n$ games. *Right:* Comparison of different learning dynamics: Barrett's results of Roth-Erev reinforcement, results for Bush-Mosteller reinforcement without and with lateral inhibition.

is called a *learner*. Thus the number of different possible signaling languages is defined by the number of possible signaling systems and therefore for a $n \times n$-game there are $n!$ different languages an agent can learn. Furthermore if an agent's combination of sender and receiver strategy forms a pooling system, it is called a *pooling language*. After all it is easy to show that the number of possible pooling languages outvalues the number of possible signaling languages for any kind of $n \times n$-game.

3 Simulating Bush-Mosteller

Barrett (see [B1]) simulated repeated signaling games with Roth-Erev reinforcement in the classical sender-receiver variant and computed the *run failure rate* (RFR). The RFR is the proportion of runs not ending with communication via a perfect signaling system. Barrett started 10^5 runs for $n \times n$-games with $n \in \{2, 3, 4, 8\}$. His results show that 100% (RFR = 0) of 2×2-games were successful. But for $n \times n$-games with $n > 2$, the RFR increases rapidly(Figure 2, left).

To compare different dynamics, we started two lines of simulation runs for Bush-Mosteller reinforcement in the sender-receiver variant with urn content parameter $c = 20$ and reinforcement value $\alpha = 1$. For the second line we additionally used lateral inhibition with value $\gamma = {}^1\!/_{|T|}$. We tested the same games like Barrett and correspondingly 10^5 runs per game. In comparison with Barrett's findings our simulation outcomes i) resulted also in a RFR of 0 for the 2×2-game, but ii) revealed an improvement with Bush-Mosteller reinforcement for the other games, especially in combination with lateral inhibition (see Figure 2, right). Nevertheless, the RFR is never 0 for $n \times n$-games with $n > 2$ and gets worse for increasing n-values, independent of the dynamics.

To analyze the behavior of agents in a multi-agent account, we started with the smallest group of agents in our simulations: three agents arranged in a complete network. In contrast to our first simulations all agents communicate as sender and as receiver as well and can learn not only a perfect signaling system, but a signaling language. Furthermore it was not only to examine if the agents have learned a language, but how many agents learned one. With this account we started between 500 and 1000 simulation runs with Bush-Mosteller reinforcement ($\alpha = 1$, $c = 20$) for $n \times n$-games with $n = 2 \ldots 8$. Each simulation run stopped, when each agent in the network has learned a signaling language or a pooling language. We measured the percentage of simulation runs ending with no, one, two or three agents, which have learned a signaling language.

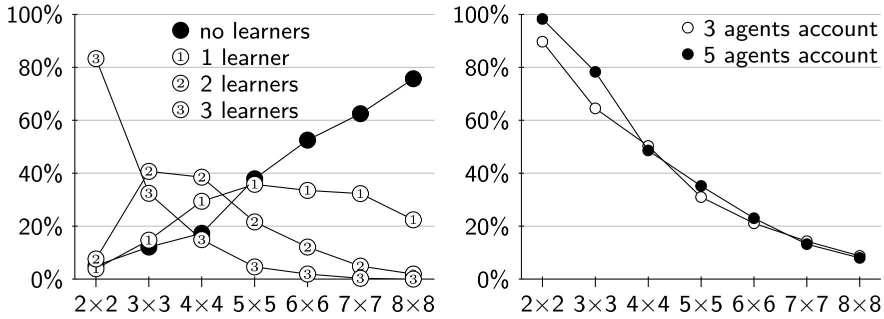


Fig. 3. Left: Percentage of simulation runs ending with a specific number of learners os signaling languages in a network with three agents for different $n \times n$-games with $n = 1 \ldots 8$. Right: Average percentage of agents learning a signaling language over all runs for different $n \times n$-games with $n = 1 \ldots 8$. Comparison of the results of a complete network of 3 agents (white circles) and 5 agents (black circles).

We got the following results: for a 2×2-game, all three agents have learned the same signaling language in more than 80% of all simulation runs. But for a 3×3-game in less than a third of all runs agents have learned a signaling language; in more than 40% of all runs two agents have learned a signaling language and the third one a pooling language. And it gets even worse for higher $n \times n$-games. E.g. for an 8×8-game in almost 80% of all runs no agents have learned a signaling language and never have all agents learned a signaling language. Figure 3 (left) depicts the distribution of how many agents have learned a signaling language (no learner, only one learner, two learners or all three agents are learners of a signaling language) for $n \times n$-games for $n = 2 \ldots 8$.[5]

In addition we were interested in whether and how the results would change by extending the number of agents. Thus in another line of experiments we tested the behavior of a complete network of 5 agents for comparison with the results of the 3 agents account. Figure 3 (right) shows the average number of agents who learned a signaling language per run for different $n \times n$-games. As you can see for 2×2-games and 3×3-games the enhancement of population size leads to a higher average percentage of agents learning a signaling language. But for games with larger domains the results are by and large the same.

The results for the classical sender-receiver game reveal that by extending learning accounts the probability of the emergence of perfect signaling systems can be improved but nevertheless is never one for an $n \times n$-game, if n is large enough. Furthermore the results for the multi-agent account with only three agents show that even for a 2×2-game not in any case all agents learn a language. And for games with larger domains, results get worse. Furthermore results don't get better or worse by changing the number of agents, as shown in a multi-agent account with 5 agents. But how could natural languages arise by assuming them having emerged from $n \times n$-games with a

[5] Note: further tests with Bush-Mosteller reinforcement in combination with negative reinforcement and/or lateral inhibition revealed that in the same cases the results could be improved for 2×2-games, but were in any case worse for all other games with larger domains.

huge n-value and in a society of much more interlocutors? We'll show that by allowing for the extinction of unused messages and the emergence of new messages, perfect signaling systems emerge for huge n-values and multiple agents in any case. In other words, we'll show that stabilization needs innovation.

4 Innovation

The idea of innovation in our account is that messages can become extinct and new messages can emerge, thus the number of messages during a repeated play can vary, whereas the number of states is fixed. The idea of innovation and extinction for reinforcement learning applied on signaling games stems from Skyrms (2010), whereby to our knowledge it is completely new i) to combine it with Bush-Mosteller reinforcement plus negative reinforcement and ii) to use it for multi-agent accounts.

The process of the emergence of new messages works like this: additionally to the balls for each message type each sender urn has an amount of *innovative balls* (according to Skyrms we call them *black balls*). If drawing a black ball the sender sends a completely new message, not ever used by any agent of the population. Because the receiver has no receiver urn of the new message, he chooses a random action. If action and state matches, the new message is adopted in the *set of known messages* of both interlocutors in the following way: i) both agents get a receiver urn for the new message, wherein the balls for all actions are equiprobable distributed, ii) both agents' sender urns are filled with a predefined amount of balls of the new message and iii) the sender and receiver urn involved in this round are updated according to the learning dynamic. If the newly invented message doesn't lead to successful communication, the message will be discarded and there will be no change in the agents strategies.

As mentioned before, messages can become extinct, and that happens in the following way: because of lateral inhibition, infrequently used or unused messages' value of balls in the sender urns will get lower and lower. At a point when the number of balls of a message is 0 for all sender urns, the message isn't existent in the active use of the agent (i.o.w. she cannot send the message anymore), and will also be removed from the agent's passive use by deleting the appropriate receiver urn. At this point the message isn't in this agent's set of known messages. Besides, there is no other interference between sender and receiver urn of one agent.

Some further notes:

- it is possible that an agent can receive a message that is not in her set of known messages. In this case she adopts the new message like described for the case of innovation. Note that in a multi-agent setup this allows for a spread of new messages
- the black balls are also affected by lateral inhibition. That means that the number of black balls can decrease and increase during runtime; it can especially be zero
- a game with innovation has a dynamic number of messages starting with 0 messages, but ends with $|M| = |T|$. Thus we call an innovation game with n states and n ultimate messages an $n \times n^*$-game

4.1 The Force of Innovation

The total number of black balls of an agent's sender urns describes his personal *force of innovation.* Note that black balls can only increase by lateral inhibition in the case

Game	$2 \times 2^*$	$3 \times 3^*$	$4 \times 4^*$	$5 \times 5^*$	$6 \times 6^*$	$7 \times 7^*$	$8 \times 8^*$
3 agents	1,052	2,120	4,064	9,640	21,712	136,110	> 500,000
5 agents	2,093	5,080	18,053	192,840	> 500,000	> 500,000	> 500,000

Table 1. Runtime Table for $n \times n^*$-games with $n = 2 \ldots 8$; for a complete network of 3 agents and 5 agents.

of unsuccessful communication and decrease by lateral inhibition in the case of successful communication. This interrelationship leads to the following dynamics: successful communication lowers the personal force of innovation, whereas unsuccessful communication raises the personal force of innovation. If we define the global *force of innovation* for a group of connected agents X as the average personal force of innovation over all $x \in X$, then the following holds: the better the communication between agents in a group X, the lower the global force of innovation of this group and vice versa. In other words, this account realizes a plausible social dynamics: if communication works, then there is no need to change and therefore a low (or zero) value of the force of innovation, whereby if communication doesn't work, the force of innovation rises.

4.2 Learning Languages by Innovation: A Question of Time

We could show in section 3 that the percentage of agents learning a signaling language in a multi-agent context is being decreased by increasing the domain size of the game. To find out whether innovation can improve these results we started simulation runs with the following settings:

- network types: complete network with 3 agents and with 5 agents
- learning dynamics: Bush-Mosteller reinforcement with negative reinforcement and lateral inhibition value ($\alpha = 1$, $\beta = 1$, $\gamma = {}^1/_{|T|}$) and innovation
- initial state: every urn of the sender is filled with black balls and the receiver does not have any a priori urn.
- experiments: 100 simulation runs per $n \times n^*$-game with $n = 2 \ldots 8$
- break condition: simulation stops if the communicative success of every agents exceeds 99% or the runtime passes the runtime limit of 500,000 communication steps (= runtime)

These simulation runs gave the following results: i) for the 3-agents account in combination with $n \times n^*$-games for $n = 2 \ldots 7$ and the 5-agents in combination with $n \times n^*$-games for $n = 2 \ldots 5$ all agents have learned a signaling language in each simulation run and ii) for the remaining account-game combinations all simulation runs exceeded the runtime limit (see Table 1). We expect that for the remaining combination all agents will learn a signaling language as well, but it takes extremely long.

All in all we could show that the integration of innovation and extinction of messages leads to a final situation where all agents have learned the same signaling language, if the runtime doesn't exceed the limit. Nevertheless we expect the same result for account-game combinations where simulations steps of these runs exceeded our limit for a manageable runtime.

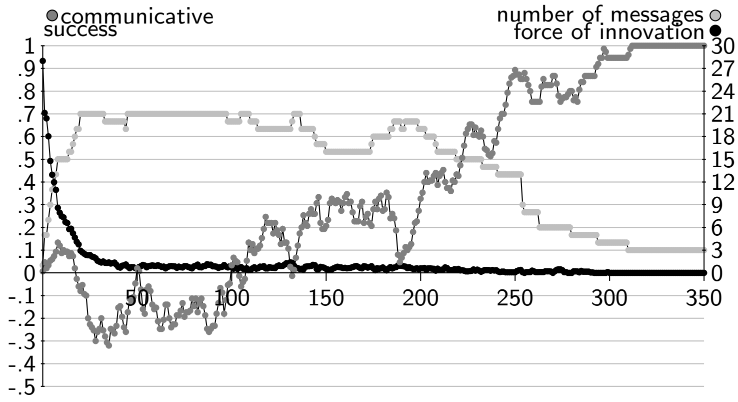


Fig. 4. Simulation run of a $3 \times 3^*$-game with innovation in a 3-agents population. Communicative success, number of used messages and force of innovation of all agents in the population; number of simulation steps at x-axis.

4.3 The Development of Signaling Languages by Innovation

As our experiments in the last section showed, by applying Bush-Mosteller reinforcement learning with innovation all agents learn the same signaling language for a small group of agents and any $n \times n^*$-game with $n = 2 \ldots 7$. Let's take a closer look at how a $3 \times 3^*$-game develops during a simulation run by analyzing i) one randomly chosen agent's parameters and ii) parameters of the whole population. Three parameters are of interest to us:

– *communication success:* utility value averaged over the last 20 communication steps averaged over all agents in the population
– *number of messages in use:* number of actually used messages in the whole population
– *force of innovation:* absolute number of black balls averaged over all agents

Figure 4 shows the resulting values for the whole population: in the beginning all the agents try out a lot of messages, which reduces the number of black balls in the urns because balls for the new messages are added and then the urn content is normalized. Note that for the first communication steps the force of innovation drops rapidly, while the number of messages rises until it reaches 21 messages here. As you can see in the course of the success-graph, the work is not done here. Once they have more or less agreed on which messages might be useful, the agents are trying them out and it is only when finally a subset of those messages is probabilistically favored that the success is increasing, while the number of known messages decreases, until the success finally reaches a perfect 1 on average, while the number of messages equals that of the states (3) and the force of innovation is zero.

What you can see in the figures as well is that even though there is no one-to-one correspondence between the number of messages and the average success, their graphs do show some sort of mirroring on the micro level. The interrelationship of innovation force and average success is not well visible in Figure 4, because of the

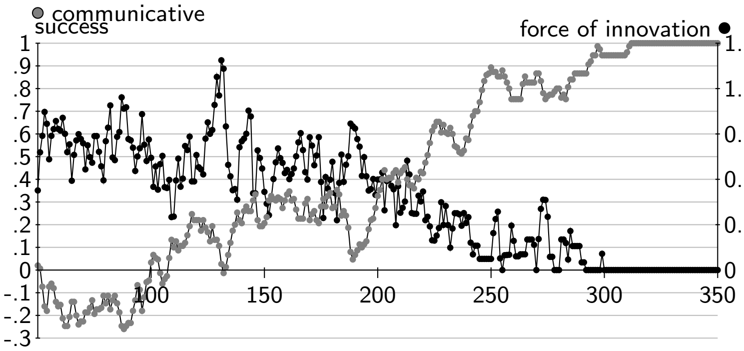


Fig. 5. Simulation run of a $3 \times 3^*$-game with innovation in a 3-agents population. Comparison of communicative success and force of innovation; number of simulation steps at x-axis.

coarse scaling of the force of innovation value. Figure 5 shows the force of innovation and the communication success between step 50 and 350 of the simulation run, already depicted in Figure 4, whereas the force of innovation value is 20 times more fine-grained. Here the interrelationship between both values is clearly recognizable, one measure's peak is simultaneously the other measure's valley. Admittedly the mirroring is not perfect, but it improves by increasing the number of agents.

5 Conclusion and Outlook

Let's recap: We started out with comparing Roth-Erev and Bush-Mosteller reinforcement, finding that Bush-Mosteller yields better results for repeated signaling games. Extending Bush-Mosteller with lateral inhibition lead to even better results, but far from perfect. And results were even worse for multi-agent account with 3 or 5 agents: with increasing n less agents develop a signaling language in the first place, especially pooling strategies turned out to be a common outcome. In a next step we extended the classical Bush-Mosteller reinforcement by adding negative reinforcement and therefore achieving lateral inhibition, innovation and extinction. We found that these tweaks result in perfect communication between 3 agents in $n \times n^*$-games for $n < 8$ and between 5 agents for $n < 6$, since higher values for n or the number of agents require much higher runtime that exceed our limit. Especially the force of innovation seems to be responsible for this achievement, since it makes sure that new messages are introduced when communication is not successful, while the combination of negative reinforcement and lateral inhibition takes care of all unused or useless messages to become extinct. Consequently, the result is an agreement on one single perfect signaling language with no other messages that might interfere.

The purpose of this direction of research is mostly about finding reasonable extensions of simple learning algorithms that lead to more explanatory results, assuming that more sophisticated learning dynamics might be more adequate to eventually describe human language behavior. We think the extensions we introduced in this article are of that kind, especially negative reinforcement, since we're rather certain that failure has a learning-effect, and innovation and extinction, because it seems unreasonable to assume that all messages are available right from the start and that everything is kept in the lexicon, even if it has only once been successfully used. Further research in this direction should clarify how memory restrictions could be modeled and how sender and receiver roles of one agent should influence each other. What remains to be shown is that our results in fact hold for higher numbers of agents and states. It would further be interesting to see what influence different, again more realistic network-types (say small-world or scale-free networks) have on the results and what happens if two or more languages interact.

References

CE1. Clarke, F., Ekeland, I.: Nonlinear oscillations and boundary-value problems for Hamiltonian systems. Arch. Rat. Mech. Anal. **78** (1982) 315–333

L1. Lewis, David: Convention. Cambridge: Harvard University Press (1969)

B1. Barret, Jeffrey A.: The Evolution of Coding in Signaling Games. Theory and Decision 67 (2009), pp. 223–237

BZ1. Barret, Jeffrey A., Zollman, Kevin J. S.: The Role of Forgetting in the Evolution and Learning of Language. Journal of Experimental and Theoretical Artificial Intelligence 21.4 (2009), pp. 293–309

BM1. Bush, Robert, Mosteller, Frederick: Stochastic Models of Learning. New York: John Wiley & Sons (1955)

HH1. Hofbauer, Josef, Huttegger, Simon M.: Feasibility of communication in binary signaling games. Journal of Theoretical Biology 254.4 (2008), pp. 843–849

HSRZ1. Huttegger, Simon M., Skyrms, Brian, Rory, Smead, Zollman, Kevin J.: Evolutionary dynamics of Lewis signaling games: signaling systems vs. partial pooling. Synthese 172.1 (2010), pp. 177–191

HZ1. Huttegger, Simon M., Zollman, Kevin J.: Signaling Games: Dynamics of Evolution and Learning. Language, Games, and Evolution. Ed. by Anton Benz et al. LNAI 6207. Springer (2011), pp. 160–176

M1. Mühlenbernd, Roland: Learning with Neighbours. Synthese 183.S1 (2011), pp. 87–109

MF1. Mühlenbernd, Roland, Franke, Michael: Signaling Conventions: Who Learns What Where and When in a Social Network. Proceedings of EvoLang IX (2011)

RE1. Roth, Alvin, Erev, Ido: Learning in extensive-form games: experimental data and simple dynamic models in the intermediate term. Games and Economic Behaviour 8 (1995), pp. 164–212

S1. Skyrms, Brian: Signals: Evolution, Learning & Information. Oxford: Oxford University Press (2010)

W1. Wagner, Elliott: Communication and Structured Correlation. Erkenntnis 71.3 (2009), pp. 377–393

Z1. Zollman, Kevin J. S.: Talking to neighbors: The evolution of regional meaning. Philosophy of Science 72.1 (2005), pp. 69–85.

Existential Definability of Modal Frame Classes

Tin Perkov

Polytechnic of Zagreb, Croatia
tin.perkov@tvz.hr

Abstract. A class of Kripke frames is called modally definable if there is a set of modal formulas such that the class consists exactly of frames on which every formula from that set is valid, i. e. globally true under any valuation. Here, existential definability of Kripke frame classes is defined analogously, by demanding that each formula from a defining set is satisfiable under any valuation. This is equivalent to the definability by the existential fragment of modal language enriched with the universal modality. A model theoretic characterization of this type of definability is given.

Keywords: modal logic, model theory, modal definability

1 Introduction

Some questions about the power of modal logic to express properties of relational structures are addressed in this paper. For the sake of notational simplicity, only the *basic propositional modal language* (BML) is considered in this paper. Let Φ be a set of *propositional variables*. The syntax of BML is given by

$$\varphi ::= p \mid \bot \mid \varphi_1 \vee \varphi_2 \mid \neg\varphi \mid \Diamond\varphi,$$

where $p \in \Phi$. We define other connectives and $\Box$ as usual. Namely, $\Box\varphi := \neg\Diamond\neg\varphi$.

Only the Kripke semantics is considered in this paper. The basic notions and results are only briefly recalled here (see [1] for details if needed). A *Kripke frame* for the basic modal language is a relational structure $\mathfrak{F} = (W, R)$, where $W \neq \emptyset$ and $R \subseteq W \times W$. A *Kripke model* based on a frame $\mathfrak{F}$ is $\mathfrak{M} = (W, R, V)$, where $V : \Phi \to 2^W$ is a mapping called *valuation*. For $w \in W$, we call $(\mathfrak{M}, w)$ a *pointed model*.

The *truth* of a formula is defined locally and inductively as usual, and denoted $\mathfrak{M}, w \Vdash \varphi$. Namely, a formula of a form $\Diamond\varphi$ is *true* at $w \in W$ if $\mathfrak{M}, u \Vdash \varphi$ for some u such that Rwu. A valuation is naturally extended to all modal formulas by putting $V(\varphi) = \{w \in W : \mathfrak{M}, w \Vdash \varphi\}$.

We say that a formula is *globally true* on $\mathfrak{M}$ if it is true at every $w \in W$, and we denote this by $\mathfrak{M} \Vdash \varphi$. On the other hand, a formula is called *satisfiable* in $\mathfrak{M}$ if it is true at some $w \in W$.

If a formula φ is true at w under any valuation on a frame $\mathfrak{F}$, we write $\mathfrak{F}, w \Vdash \varphi$. We say that a formula is *valid* on a frame $\mathfrak{F}$ if we have $\mathfrak{M} \Vdash \varphi$ for any model $\mathfrak{M}$ based on $\mathfrak{F}$. This is denoted $\mathfrak{F} \Vdash \varphi$. A class $\mathcal{K}$ of Kripke frames is *modally definable* if there is a set Σ of formulas such that $\mathcal{K}$ consists exactly of frames on which every formula from Σ is valid, i. e. $\mathcal{K} = \{\mathfrak{F} : \mathfrak{F} \Vdash \Sigma\}$. If this is the case, we say that $\mathcal{K}$ is *defined* by Σ and denote $\mathcal{K} = \mathrm{Fr}(\Sigma)$.

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see http://ceur-ws.org/, 2012, pp. 104–111.

Model theoretic closure conditions that are necessary and sufficient for an elementary class of frames (i. e. first-order definable property of relational structures) to be modally definable are given by the famous Goldblatt-Thomason Theorem.

Theorem (Goldblatt-Thomason [3]). *An elementary class $\mathcal{K}$ of frames is definable by a set of modal formulas if and only if $\mathcal{K}$ is closed under surjective bounded morphisms, disjoint unions and generated subframes, and reflects ultrafilter extensions.*

All of the frame constructions used in the theorem – bounded morphisms, disjoint unions, generated subframes and ultrafilter extensions – are presented in detail in [1] (the same notation is used in this paper). Just to be clear, we say that a class $\mathcal{K}$ *reflects* a construction if its complement $\mathcal{K}^c$, that is the class of all Kripke frames not in $\mathcal{K}$, is closed under that construction.

Now, an alternative notion of definability is proposed here as follows.

Definition. A class $\mathcal{K}$ of Kripke frames is called *modally* $\exists$*-definable* if there is a set Σ of modal formulas such that for any Kripke frame $\mathfrak{F}$ we have: $\mathfrak{F} \in \mathcal{K}$ if and only if each $\varphi \in \Sigma$ is satisfiable in $\mathfrak{M}$, for any model $\mathfrak{M}$ based on $\mathfrak{F}$. If this is the case, we denote $\mathcal{K} = \mathrm{Fr}_\exists(\Sigma)$.

The definition does not require that all formulas of Σ are satisfied at the same point – it suffices that each formula of Σ is satisfied at some point.

In the sequel, a notation $\mathrm{Mod}(F)$ is used for a class of structures defined by a first-order formula F. Similarly, if $\Sigma = \{\varphi\}$ is a singleton set of modal formulas, we write $\mathrm{Fr}_\exists(\varphi)$ instead of $\mathrm{Fr}_\exists(\{\varphi\})$.

Example 1. It is well-known that the formula $p \to \Diamond p$ defines reflexivity, i. e. $\mathrm{Fr}(p \to \Diamond p) = \mathrm{Mod}(\forall x Rxx)$. Now, it is easy to see that $\mathrm{Fr}_\exists(p \to \Diamond p)$ is the class of all frames such that $R \neq \emptyset$, that is $\mathrm{Fr}_\exists(p \to \Diamond p) = \mathrm{Mod}(\exists x \exists y Rxy)$. This class is not modally definable in the usual sense, since it is clearly not closed under generated subframes. Note that the condition $R \neq \emptyset$ is $\exists$-definable also by a simpler formula $\Diamond\top$.

The main result of this paper is the following characterization.

Theorem 1. *Let $\mathcal{K}$ be an elementary class of Kripke frames. Then $\mathcal{K}$ is modally $\exists$-definable if and only if it is closed under surjective bounded morphisms and reflects generated subframes and ultrafilter extensions.*

This is an analogue of a similar characterization of existentially definable Kripke model classes, given in [6].

2 First and second-order standard translations

The starting point of correspondence between first-order and modal logic is the *standard translation*, a mapping that translates each modal formula φ to the first-order formula $ST_x(\varphi)$, as follows:

$ST_x(p) = Px$, for each $p \in \Phi$,
$ST_x(\bot) = \bot$,
$ST_x(\neg\varphi) = \neg ST_x(\varphi)$,
$ST_x(\varphi \vee \psi) = ST_x(\varphi) \vee ST_x(\psi)$,
$ST_x(\Diamond\varphi) = \exists y(Rxy \wedge ST_y(\varphi))$.

Clearly, we have $\mathfrak{M}, w \Vdash \varphi$ if and only if $\mathfrak{M} \models ST_x(\varphi)[w]$, and $\mathfrak{M} \Vdash \varphi$ if and only if $\mathfrak{M} \models \forall x ST_x(\varphi)$. But, validity of a formula on a frame generally is not first-order expressible, since we need to quantify over valuations. We have a second-order standard translation, that is, $\mathfrak{F} \Vdash \varphi$ if and only if $\mathfrak{F} \models \forall P_1 \ldots \forall P_n \forall x ST_x(\varphi)$, where $P_1, \ldots, P_n$ are monadic second-order variables, one for each propositional variable occurring in φ. So, the notion of modal definability is equivalent to the definability by a set of second-order formulas of the form $\forall P_1 \ldots \forall P_n \forall x ST_x(\varphi)$. However, in many cases a formula of this type is equivalent to a first-order formula. Namely, this holds for any *Sahlqvist formula* (the definition is omitted here – see [7] or [1]), for which a first-order frame correspondent is effectively computable. On the other hand, the Goldblatt-Thomason Theorem characterizes those first-order properties that are modally definable.

Now, $\exists$-definability is clearly also equivalent to the definability by a type of second-order formulas – those of the form $\forall P_1 \ldots \forall P_n \exists x ST_x(\varphi)$. Consider another example of a modally $\exists$-definable class.

Example 2. The condition $F = \exists x \forall y (Rxy \to \exists z Ryz)$ is not modally definable, since it is not closed under generated subframes, but it is modally $\exists$-definable by the formula $\varphi = p \to \Box\Diamond p$.

To prove this, we need to show $\mathrm{Fr}_\exists(\varphi) = \mathrm{Mod}(F)$. But $\mathfrak{F} = (W, R) \in \mathrm{Fr}_\exists(\varphi)$ if and only if $\mathfrak{F} \models \forall P \exists x (Px \to \forall y (Rxy \to \exists z (Ryz \wedge Pz)))$. So in particular, under the assignment which assigns the entire W to the second-order variable P, we get $\mathfrak{F} \models \exists x \forall y (Rxy \to \exists z Ryz)$, thus $\mathfrak{F} \in \mathrm{Mod}(F)$. The reverse inclusion is proved similarly.

Other changes of quantifiers or the order of first and second-order quantifiers would result in other types of definability, perhaps also worthy of exploring. In fact, this has already been done by Venema [9] and Hollenberg [5], who consider *negative definability*, which corresponds to second-order formulas of the form $\forall x \exists P_1 \ldots \exists P_n ST_x(\neg\varphi)$. A class of frames negatively defined by Σ is denoted $\mathrm{Fr}^-(\Sigma)$. A general characterization of negative definability has not been obtained, and neither has a characterization of elementary classes which are negatively definable – it even remains unknown if all negatively definable classes are in fact elementary. But, to digress a little from the main point of this paper, we easily get the following fairly broad result.

Proposition 1. *Let φ be a modal formula which has a first-order local correspondent, i. e. there is a first-order formula $F(x)$ such that for any frame $\mathfrak{F} = (W, R)$ and any $w \in W$ we have $\mathfrak{F}, w \Vdash \varphi$ if and only if $\mathfrak{F} \models F(x)[w]$. (In particular, this holds for any Sahlqvist formula.)*

Then we have $\mathrm{Fr}^-(\varphi) = \mathrm{Mod}(\forall x \neg F(x))$.

Proof. We have $\mathfrak{F} \in \mathrm{Fr}^-(\varphi)$ if and only if $\mathfrak{F} \models \forall x \exists P_1 \ldots \exists P_n ST_x(\neg\varphi)$ if and only if $\mathfrak{F} \not\models \exists x \forall P_1 \ldots \forall P_n ST_x(\varphi)$. But this means that there is no $w \in W$ such that $\mathfrak{F} \models \forall P_1 \ldots \forall P_n ST_x(\varphi)[w]$. The latter holds if and only if $\mathfrak{F}, w \Vdash \varphi$, which is by assumption equivalent to $\mathfrak{F} \models F(x)[w]$. The fact that such w does not exist, is equivalent to $\mathfrak{F} \in \mathrm{Mod}(\forall x \neg F(x))$. $\Box$

So for example, since $p \to \Diamond p$ locally corresponds to Rxx, we have that $p \to \Diamond p$ negatively defines irreflexivity, which is not modally definable property, since it is not preserved under surjective bounded morphisms.

3 Model-theoretic constructions

This section can be used, if needed, for a quick reference of the basic facts about constructions used in the main theorem. Otherwise it can be omitted.

A *bisimulation* between Kripke models $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}' = (W', R', V')$ is a relation $Z \subseteq W \times W'$ such that:

(at) if wZw' then we have: $w \in V(p)$ if and only if $w' \in V'(p)$, for all $p \in \Phi$,

(forth) if wZw' and Rwv, then there is v' such that vZv' and $R'w'v'$,

(back) if wZw' and $R'w'v'$, then there is v such that vZv' and Rwv.

The basic property of bisimulations is that (at) extends to all formulas: if wZw' then $\mathfrak{M}, w \Vdash \varphi$ if and only if $\mathfrak{M}', w' \Vdash \varphi$, i. e. $(\mathfrak{M}, w)$ and $(\mathfrak{M}', w')$ are modally equivalent. We get the definition of bisimulation between frames by omitting the condition (at).

A *bounded morphism* from a frame $\mathfrak{F} = (W, R)$ to $\mathfrak{F}' = (W', R')$ is a function $f : W \to W'$ such that:

(forth) Rwv implies $R'f(w)f(v)$,

(back) if $R'f(w)v'$, then there is v such that $v' = f(v)$ and Rwv.

Clearly, a bounded morphism is a bisimulation that is also a function.

A *generated subframe* of $\mathfrak{F} = (W, R)$ is a frame $\mathfrak{F}' = (W', R')$ where $W' \subseteq W$ such that $w \in W'$ and Rwv implies $v \in W'$, and $R' = R \cap (W' \times W')$. A *generated submodel* of $\mathfrak{M} = (W, R, V)$ is a model based on a generated subframe, with the valuation $V'(p) = V(p) \cap W'$, for all $p \in \Phi$. It is easy to see that the global truth of a modal formula is preserved on a generated submodel.

To define the ultraproducts and ultrafilter extensions, we need the notion of ultrafilters. An *ultrafilter* over a set $I \neq \emptyset$ is a family $U \subseteq \mathcal{P}(I)$ such that:

(1) $I \in U$,

(2) if $A, B \in U$, then $A \cap B \in U$,

(3) if $A \in U$ and $A \subseteq B \subseteq I$, then $B \in U$,

(4) for all $A \subseteq I$ we have: $A \in U$ if and only if $I \setminus A \notin U$.

The existence of ultrafilters is provided by a fact that any family of subsets which has the finite intersection property (that is, each finite intersection is non-empty) can be extended to an ultrafilter (see e. g. [2]).

Let $\{\mathfrak{M}_i = (W_i, R_i, V_i) : i \in I\}$ be a family of Kripke models and let U be an ultrafilter over I. The *ultraproduct* of this family over U is the model $\prod_U \mathfrak{M}_i = (W, R, V)$ such that:

(1) W is the set of equivalence classes f^U of the following relation defined on the Cartesian product of the family: $f \sim g$ if and only if $\{i \in I : f(i) = g(i)\} \in U$,

(2) $f^U R g^U$ if and only if $\{i \in I : f(i) R_i g(i)\} \in U$,

(3) $f^U \in V(p)$ if and only if $\{i \in I : f(i) \in V_i(p)\} \in U$, for all p.

The basic property of ultraproducts is that (3) extends to all formulas.

Proposition 2. *Let $\{\mathfrak{M}_i : i \in I\}$ be a family of Kripke models and let U be an ultrafilter over I.*

Then we have $\prod_U \mathfrak{M}_i, f^U \Vdash \varphi$ if and only if $\{i \in I : \mathfrak{M}_i, f(i) \Vdash \varphi\} \in U$, for any f^U. Furthermore, we have $\prod_U \mathfrak{M}_i \Vdash \varphi$ if and only if $\{i \in I : \mathfrak{M}_i \Vdash \varphi\} \in U$.

This is an analogue of the Łoś Fundamental Theorem on ultraproducts from the first-order model theory (see [2] for this, and [1] for the proof of the modal analogue). The Łoś Theorem also implies that an elementary class of models is closed under ultraproducts.

An ultraproduct such that $\mathfrak{M}_i = \mathfrak{M}$ for all $i \in I$ is called an *ultrapower* of $\mathfrak{M}$ and denoted $\prod_U \mathfrak{M}$. From the Łoś Theorem it follows that any ultrapower of a model is elementarily equivalent to the model, that is, the same first-order sentences are true on $\mathfrak{M}$ and $\prod_U \mathfrak{M}$. Definition of an ultraproduct of a family of frames is obtained by omitting the clause regarding valuation.

Another notion needed in the proof of the main theorem is *modal saturation*, the modal analogue of *ω-saturation* from the classical model theory. The definition of saturation is omitted here, since we only need some facts which it implies. Most importantly, saturation implies a converse of the basic property of bisimulations, which generally does not hold. In fact, modal equivalence between points of modally saturated models is a bisimulation. Also, we use the fact that any ω-saturated Kripke model is also modally saturated (see [1] for proofs of these facts).

Finally, the *ultrafilter extension* of a model $\mathfrak{M} = (W, R, V)$ is the model $\mathfrak{ue}\mathfrak{M} = (\mathrm{Uf}(W), R^{\mathfrak{ue}}, V^{\mathfrak{ue}})$, where $\mathrm{Uf}(W)$ is the set of all ultrafilters over W, $R^{\mathfrak{ue}}uv$ holds if and only if $A \in v$ implies $m_\diamond(A) \in u$, where $m_\diamond(A)$ denotes the set of all $w \in W$ such that Rwa for some $a \in A$, and $u \in V^{\mathfrak{ue}}(p)$ if and only if $V(p) \in u$. The basic property is that this extends to any modal formula, i. e. we have $u \in V^{\mathfrak{ue}}(\varphi)$ if and only if $V(\varphi) \in u$ (see [1]). From this it easily follows that the global truth of a modal formula is preserved on the ultrafilter extension. Another important fact is that the ultrafilter extension of a model is modally saturated (see [1]).

The ultrafilter extension of a frame $\mathfrak{F} = (W, R)$ is $\mathfrak{ue}\mathfrak{F} = (\mathrm{Uf}(W), R^{\mathfrak{ue}})$.

4 Proof of the main theorem

In this section Theorem 1 is proved in detail. Arguments and techniques used in the proof are similar to the ones used in the proof of Goldblatt-Thomason theorem as presented in [1], so the reader might find it interesting to compare these proofs to note analogies and differences.

Proof (of Theorem 1). Let $\mathcal{K} = \mathrm{Fr}_\exists(\Sigma)$. Let $\mathfrak{F} = (W, R) \in \mathcal{K}$ and let f be a surjective bounded morphism from $\mathfrak{F}$ to some $\mathfrak{F}' = (W', R')$. Take any $\varphi \in \Sigma$ and any model $\mathfrak{M}' = (W', R', V')$ based on $\mathfrak{F}'$. Put $V(p) = \{w \in W : f(w) \in V'(p)\}$. Then V is a well defined valuation on $\mathfrak{F}$. Put $\mathfrak{M} = (W, R, V)$. Since $\mathfrak{F} \in \mathcal{K}$, there exists $w \in W$ such that $\mathfrak{M}, w \Vdash \varphi$. But then $\mathfrak{M}', f(w) \Vdash \varphi$. This proves that $\mathcal{K}$ is closed under surjective bounded morphisms.

To prove that $\mathcal{K}$ reflects generated subframes and ultrafilter extensions, let $\mathfrak{F} = (W, R) \notin \mathcal{K}$. This means that there is $\varphi \in \Sigma$ and a model $\mathfrak{M} = (W, R, V)$ based on $\mathfrak{F}$ such that $\mathfrak{M} \Vdash \neg\varphi$. Let $\mathfrak{F}' = (W', R')$ be a generated subframe of $\mathfrak{F}$. Define $V'(p) = V(p) \cap W'$, for all p. Then we have $\mathfrak{M}' \Vdash \neg\varphi$, which proves $\mathfrak{F}' \notin \mathcal{K}$, as desired. Also, $\mathfrak{ue}\mathfrak{M}$ is a model based on the ultrafilter extension $\mathfrak{ue}\mathfrak{F}$ and we have $\mathfrak{ue}\mathfrak{M} \Vdash \neg\varphi$, which proves $\mathfrak{ue}\mathfrak{F} \notin \mathcal{K}$.

For the converse, let $\mathcal{K}$ be an elementary class of frames that is closed under surjective bounded morphisms and reflects generated subframes and ultrafilter extensions. Denote by Σ the set of all formulas that are satisfiable in all models based on all frames in $\mathcal{K}$. Then $\mathcal{K} \subseteq \mathrm{Fr}_\exists(\Sigma)$ and it remains to prove the reverse inclusion.

Let $\mathfrak{F} = (W, R) \in \mathrm{Fr}_\exists(\Sigma)$. Let Φ be a set of propositional variables that contains a propositional variable p_A for each $A \subseteq W$. Let $\mathfrak{M} = (W, R, V)$, where $V(p_A) = A$ for all $A \subseteq W$. Denote by Δ the set of all modal formulas over Φ which are globally true on $\mathfrak{M}$. Now, for any finite $\delta \subseteq \Delta$ there is $\mathfrak{F}_\delta \in \mathcal{K}$ and a model $\mathfrak{M}_\delta$ based on $\mathfrak{F}_\delta$ such that $\mathfrak{M}_\delta \Vdash \delta$. Otherwise, since Δ is closed under conjunctions, there is $\varphi \in \Delta$ such that $\neg\varphi \in \Sigma$, thus $\neg\varphi$ is satisfiable in $\mathfrak{M}$, which contradicts $\mathfrak{M} \Vdash \Delta$.

Now, let I be the family of all finite subsets of Δ. For each $\varphi \in \Delta$, put $\hat{\varphi} = \{\delta \in I : \varphi \in \delta\}$. The family $\{\hat{\varphi} : \varphi \in \Delta\}$ clearly has the finite intersection property, so it can be extended to an ultrafilter U over I. But for all $\varphi \in \Delta$ we have $\{\delta \in I : \mathfrak{M}_\delta \Vdash \varphi\} \supseteq \hat{\varphi}$ and $\hat{\varphi} \in U$, thus $\{\delta \in I : \mathfrak{M}_\delta \Vdash \varphi\} \in U$, so the Proposition 2 implies $\prod_U \mathfrak{M}_\delta \Vdash \varphi$. The model $\prod_U \mathfrak{M}_\delta$ is based on the frame $\prod_U \mathfrak{F}_\delta$. Since $\mathcal{K}$ is elementary, it is also closed under ultraproducts, so $\prod_U \mathfrak{F}_\delta \in \mathcal{K}$. It remains to prove that there is a surjective bounded morphism from some ultrapower of $\prod_U \mathfrak{F}_\delta$ to a generated subframe of $\mathfrak{ue}\mathfrak{F}$. Then the assumed properties of $\mathcal{K}$ imply that $\mathfrak{F} \in \mathcal{K}$, as desired.

The classical model theory provides that there is an ω-saturated ultrapower of $\prod_U \mathfrak{M}_\delta$ (cf. [2]). Let $\mathfrak{M}_\Delta$ be such an ultrapower. We have that $\mathfrak{M}_\Delta$ is modally saturated. Also, it is elementarily equivalent to $\prod_U \mathfrak{M}_\delta$, so using standard translation we obtain $\mathfrak{M}_\Delta \Vdash \Delta$. The model $\mathfrak{M}_\Delta$ is based on a frame $\mathfrak{F}_\Delta$, which is an ultrapower of $\prod_U \mathfrak{F}_\delta$. Now define a mapping from $\mathfrak{F}_\Delta$ to $\mathfrak{ue}\mathfrak{F}$ by putting $f(w) = \{A \subseteq W : \mathfrak{M}_\Delta, w \Vdash p_A\}$.

First we need to prove that f is well-defined, i. e. that $f(w)$ is indeed an ultrafilter over W.

(1) We easily obtain $W \in f(w)$, since $p_W \in \Delta$ by the definition of V.

(2) If $A, B \in f(w)$, then $\mathfrak{M}_\Delta, w \Vdash p_A \wedge p_B$. Clearly, $\mathfrak{M} \Vdash p_A \wedge p_B \leftrightarrow p_{A\cap B}$. Thus $\mathfrak{M}_\Delta \Vdash p_A \wedge p_B \leftrightarrow p_{A\cap B}$, so $\mathfrak{M}_\Delta, w \Vdash p_{A\cap B}$, i. e. $A \cap B \in f(w)$.

(3) If $A \in f(w)$ and $A \subseteq B \subseteq W$, then from the definition of V it follows $\mathfrak{M} \Vdash p_A \to p_B$. But then also $\mathfrak{M}_\Delta \Vdash p_A \to p_B$, hence $\mathfrak{M}_\Delta, w \Vdash p_B$, so $B \in f(w)$.

(4) For all $A \subseteq W$ we have $\mathfrak{M} \Vdash p_A \leftrightarrow \neg p_{W\setminus A}$, which similarly as in the previous points implies $A \in f(w)$ if and only if $W \setminus A \notin f(w)$, as desired.

Assume for the moment that we have: $u = f(w)$ if and only if $(\mathfrak{ue}\mathfrak{M}, u)$ and $(\mathfrak{M}_\Delta, w)$ are modally equivalent. Since $\mathfrak{ue}\mathfrak{M}$ and $\mathfrak{M}_\Delta$ are modally saturated, the modal equivalence between their points is a bisimulation. So f is a bisimulation, but it is also a function, which means that it is a bounded morphism from $\mathfrak{F}_\Delta$ to $\mathfrak{ue}\mathfrak{F}$. But then the corestriction of f to its image is a surjective bounded morphism from an ultrapower of $\prod_U \mathfrak{F}_\delta$ to a generated subframe of $\mathfrak{ue}\mathfrak{F}$, which we needed.

So to conclude the proof, it remains to show that $u = f(w)$ holds if and only if $(\mathfrak{ue}\mathfrak{M}, u)$ and $(\mathfrak{M}_\Delta, w)$ are modally equivalent. Let $u = f(w)$. Then we have $\mathfrak{ue}\mathfrak{M}, u \Vdash \varphi$ if and only if $V(\varphi) \in u$, which is by the definition of f equivalent to $\mathfrak{M}_\Delta, w \Vdash p_{V(\varphi)}$. But the definition of V clearly implies $\mathfrak{M} \Vdash \varphi \leftrightarrow p_{V(\varphi)}$, so also $\mathfrak{M}_\Delta \Vdash \varphi \leftrightarrow p_{V(\varphi)}$, which provides the needed modal equivalence.

For the converse, the assumption implies that we have $\mathfrak{ue}\mathfrak{M}, u \Vdash p_A$ if and only if $\mathfrak{M}_\Delta, w \Vdash p_A$, for all $A \subseteq W$. This means that $V(p_A) = A \in u$ if and only if $A \in f(w)$, i. e. $u = f(w)$.

□

5 Conclusion: link to the universal modality

Although the approach of this paper is to define ∃-definability as a metalingual notion, it should be noted that it can be included in the language itself. That is, the satisfiability of a modal formula under any valuation on a frame can be expressed by a formula of the modal language enriched with the universal modality (BMLU). The syntax is an extension of the basic modal language by new modal operator $A\varphi$, and we can also define its dual $E\varphi := \neg A \neg \varphi$. We call A the *universal modality*, and E the *existential modality.* The semantics of the new operators is standard modal semantics, with respect to the universal binary relation $W \times W$ on a frame $\mathfrak{F} = (W, R)$. This means that the standard translation of universal and existential operators is as follows (cf. [4] and [8]):

$ST_x(E\varphi) = \exists y ST_y(\varphi)$,
$ST_x(A\varphi) = \forall y ST_y(\varphi)$.

Now, let $\mathcal{K}$ be a class of Kripke frames. Clearly, $\mathcal{K}$ is modally $\exists$-definable if and only if it is definable by a set of formulas of the existential fragment of BMLU, i. e. by a set of formulas of the form $E\varphi$, where φ is a formula of BML. This immediately follows from the clear fact that for any frame $\mathfrak{F}$ and any φ we have $\mathfrak{F} \Vdash E\varphi$ if and only if $\mathfrak{F} \models \forall P_1 \dots \forall P_n \exists y ST_y(\varphi)$, where $P_1, \dots, P_n$ correspond to propositional variables that occur in φ, and the letter holds if and only if φ is satisfiable under any valuation on $\mathfrak{F}$.

Goranko and Passy [4] gave a characterization that an elementary class is modally definable in BMLU if and only if it is closed under surjective bounded morphisms and reflects ultrafilter extension. So, from the main theorem of this paper we conclude that reflecting generated subframes, not surprisingly, is what distinguishes existential fragment within this language, at least with respect to elementary classes. Also, the usual notion of modal definability clearly coincides with the universal fragment of BMLU, hence the Goldblatt-Thomason Theorem tells us that closure under generated subframes and disjoint unions is essential for this fragment.

As for some further questions that might be worth exploring, for example, similarly to the notion of $\pm$-definability from [5], we can say that a class of frames is *modally* $\forall\exists$-*definable* if there is a pair (Σ_1, Σ_2) of sets of formulas such that a class consists exactly of frames on which every formula from Σ_1 is valid and every formula from Σ_2 is satisfiable under any valuation, and try to obtain a characterization theorem. This also coincides with a fragment of BMLU, and generalizes both usual modal definability and $\exists$-definability. Furthermore, we may be able to obtain general characterization theorems for these fragments, without the assumption of the first-order definability.

On the other hand, a question to be addressed is which modally $\exists$-definable classes are elementary, and is there an effective procedure analogous to the one for Sahlqvist formulas, to obtain a first-order formula equivalent to a second-order translation $\forall P_1 \dots \forall P_n \exists x ST_x(\varphi)$ for some sufficiently large and interesting class of modal formulas.

References

1. J. Blackburn, M. de Rijke, Y. Venema: Modal Logic. Cambridge University Press (2001)
2. C. C. Chang, H. J. Keisler: Model Theory. Elsevier (1990)
3. R. I. Goldblatt, S. K. Thomason: Axiomatic classes in propositional modal logic. In: J. Crossley (ed.) Algebra and logic, pp. 163–173. Springer (1974)
4. V. Goranko, S. Passy: Using the Universal Modality: Gains and Questions. Journal of Logic and Computation, 2, 5-30 (1992)
5. M. Hollenberg: Characterizations of Negative Definability in Modal Logic. Studia Logica, 60, 357–386 (1998)
6. T. Perkov: Towards a generalization of modal definability. In: D. Lassiter, M. Slavkovik (eds.) New Directions in Logic, Language, and Computation, pp 130–139. Springer (2012)
7. H. Sahlqvist: Completeness and correspondence in the first and second order semantics for modal logic. In: S. Kanger (ed.) Proceedings of the Third Scandinavian Logic Symposium, Uppsala 1973, pp. 110–143. Amsterdam: North-Holland (1975)
8. J. van Benthem: The range of modal logic. Journal of Applied Non-Classical Logics, 9, 407–442 (1999)

9. Y. Venema: Derivation rules as anti-axioms in modal logic. Journal of Symbolic Logic, 58, 1003–1034 (1993)

Spotting and Improving Modularity in Large Scale Grammar Development

Simon Petitjean

LIFO, University of Orleans

Abstract. XMG (eXtensible MetaGrammar) is a metagrammar compiler which has already been used for the design of large scale Tree Adjoining Grammars and Interaction Grammars. Due to the heterogeneity in this field (different grammar formalisms, different languages, etc), a particularly interesting aspect to explore is modularity. In this paper, we discuss the different spots where this modularity can be considered in a grammar development, and its integration to XMG.

1 Introduction

Nowadays, a lot a applications have to deal with languages and consequently need to manipulate their descriptions. Linguists are also interested in this kind of resources, for study or comparison. For these purposes, formal grammars production has became a necessity. Our work focuses on large scale grammars, that is to say grammars which represent a significant part of the language.

The main issue with these resources is their size (thousands of structures), which causes their production and maintenance to be really complex and time consuming tasks. Moreover, these resources have some specificities (language, grammatical framework) that make each one unique.

Since a handwriting of thousands of structures represents a huge amount of work, part of the process has to be automatized. A totally automatic solution could consist in a acquisition from treebanks, which is a widely used technique. Semi automatic approaches are alternatives that give an important role to the linguist: they consist in building automatically the whole grammar from information on its structure. The approach we chose is based on a description language, called metagrammar [1]. The idea behind metagrammars is to capture linguistic generalization, and to use abstractions to describe the grammar.

The context that initially inspired metagrammars was the one of Tree Adjoining Grammars (TAG) [8]. This formalism consists in tree rewriting, with two specific rewriting operations: adjunction and substitution. An adjunction is the replacement of an internal node by a an auxiliary tree (one of its leaf nodes is labelled with $\star$ and called foot node) with root and foot node having the same syntactic category as the internal node. A substitution is the replacement of a leaf node (marked with $\downarrow$) by a tree with a root having the same syntactic category as this leaf node. The principle is to apply these operations to a set of elementary trees to match the sentence we want to parse. TAG is said to have a extended domain of locality, because those operations (especially adjunction) and the depth of the trees allow to represent long distance relations between nodes: two nodes of the same elementary tree can after derivation

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see `http://ceur-ws.org/`, 2012, pp. 112–120.

end up at an arbitrary distance from each other. Here, we will only manipulate LTAG (lexicalized-TAG), which means each elementary tree is associated with at least one lexical element.

What can we do to lower the amount of work implied by the conception of the grammar ? Let us take a look at some rules:

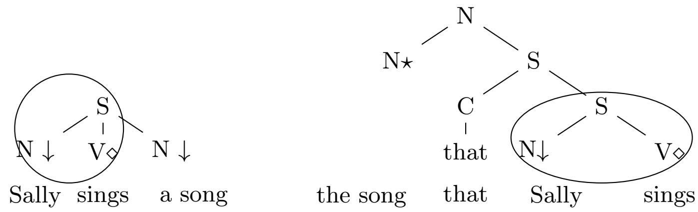


Fig. 1. Verb with canonical subject and canonical or extracted object

Those two trees share some common points: part of the structure is the same (the subject is placed before the verb in both circled parts), and the agreement constraints, given in feature structures associated to nodes (not represented here), are similar. This kind of redundancy is one of the key motivations for the use of abstractions. These abstractions are descriptions of the redundant fragments we can use everywhere they are needed.

Metagrammars are based on the manipulation of those linguistic generalizations. They consist in generating the whole grammar from an abstract description, permitting to reason about language at an abstract level. The metagrammatical language we will deal with here is XMG (eXtensible MetaGrammar) [1], introduced in [4]. A new project, XMG-2 [2], started in 2010 to achieve the initial goal of the compiler, extensibility, which has not been realized yet: XMG-1 only supports tree based grammars (two formalisms, Tree Adjoining Grammars and Interaction Grammars), and includes two levels of description, the syntactic one and the semantic one. Our goal is to go towards two levels of modularity: we want it to be possible to assemble a grammar in a modular way, thanks to a metagrammar assembled in a modular way.

We will begin pointing out the modularity on the grammar side in section 2. In section 3, we will focus on a new level of modularity, a metagrammatical one. In section 4, we will give an overview of what has been done, and what remains to be done. Finally, we will conclude and give some perspectives.

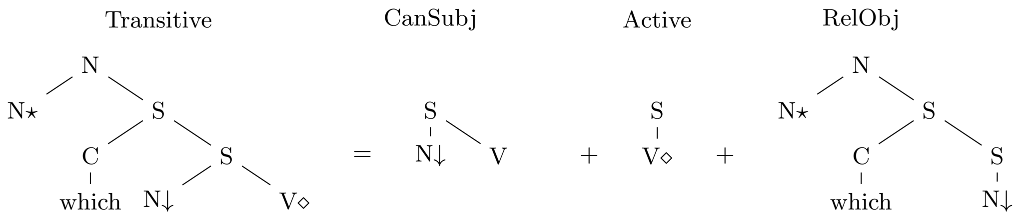
2 Assembling grammars in a modular way

XMG consists in defining fragments of the grammar, and controlling how these fragments can combine to produce the whole grammar. The following figure shows the intuition of the combination of fragments to produce a tree for transitive verbs. It is done by combining three tree fragments, one for the subject (in its canonical form, that

[1] https://sourcesup.cru.fr/xmg/

[2] https://launchpad.net/xmg

we noticed redundant previously), one for the object (relative) and one for the active form.



To build a lexicon, the metagrammar is first executed in an indeterministic way to produce descriptions. Then these descriptions are solved to produce the models which will be added to the lexicon.

2.1 The control language and the dimension system

The main particularity of XMG is that it allows to see the metagrammar as a logical program, using logical operators.

The abstractions (possibly with parameters) we manipulate are called classes. They contain conjunctions and disjunctions of descriptions (tree fragments descriptions for TAG), or calls to other classes. This is formalized by the following control language:

$$\begin{array}{rcl} \mathit{Class} & := & \mathit{Name}[p_1, \ldots, p_n] \rightarrow \mathit{Content} \\ \mathit{Content} & := & \langle \mathit{Dim} \rangle \{ \mathit{Desc} \} \ \mid \ \mathit{Name}[\ldots] \ \mid \ \mathit{Content} \vee \mathit{Content} \\ & & \mid \ \mathit{Content} \wedge \mathit{Content} \end{array}$$

For example, we can produce the two trees of the figure 1 by defining the tree fragments for canonical subject, verbal morphology, canonical object and relativized object, and these combinations:

$$\begin{array}{rl} \mathit{Object} & \rightarrow \mathit{CanObj} \ \vee \ \mathit{RelObj} \\ \mathit{Transitive} & \rightarrow \mathit{CanSubj} \ \wedge \ \mathit{Active} \ \wedge \ \mathit{Object} \end{array}$$

This part of metagrammar says that an object can either be a canonical object or a relative object, and that the transitive mode is created by getting together a canonical subject, an active form and one of the two object realizations.

Notice that descriptions are accumulated within dimensions, which allow to separate types of data. Sharing is still possible between dimensions, by means of another dimension we call interface. In XMG's TAG compiler for example, the *syn* dimension accumulates tree descriptions while the *sem* dimension accumulates predicates representing the semantics. Each dimension comes with a description language, adapted to the type of data it will contain. For each type of description we need to accumulate, we have to use a different description languages. The first version of XMG provides a tree description langague (for TAG or Interaction Grammars) associated with the *syn* dimension and a language for semantics associated with the *sem* dimension.

A tree description language

For trees in TAG, we use the following tree description language:

$$
\begin{aligned}
Desc \ := \ & x \rightarrow y \ \mid \ x \rightarrow^{+} y \ \mid \ x \rightarrow^{*} y \ \mid \ x \prec y \ \mid \ x \prec^{+} y \ \mid \ x \prec^{*} y \ \mid \ x[f{:}E] \\
& \mid \ x(p{:}E) \mid \ Desc \ \wedge \ Desc
\end{aligned}
$$

where x and y are node variables, $\rightarrow$ and $\prec$ dominance and precedence between nodes ($^{+}$ and * respectively standing for transitive and reflexive transitive closures). ':' is the association between a property p or a feature f and an expression E. Properties are constraints specific to the formalism (the fact that a node is a substitution node for example), while features contain linguistic information, such as syntactic categories, number or gender.

When accumulated, the tree description in the syntactic dimension is still partial. The TAG elementary trees that compose the grammar are the models for this partial description. They are built by a tree description solver, based on constraints to ensure the well-formedness of the solutions. XMG computes minimal models, that is to say models where only the nodes of the description exist (no additional node is created). Here is a toy metagrammar, composed of three description classes (representing canonical subject, relative object, active form) and one combination class (transitive mode):

$$
\begin{aligned}
CanSubj \rightarrow & \langle syn \rangle \{ (s_1[cat : S] \rightarrow v_1[cat : V]) \wedge (s_1 \rightarrow n_1(mark : subst)[cat : N]) \\
& \wedge (n_1 \prec v_1) \} \\
RelObj \rightarrow & \langle syn \rangle \{ (n_2[cat = N] \rightarrow n_3(mark = adj)[cat = N]) \wedge (n_2 \rightarrow s_2[cat = S]) \\
& \wedge (n_3 \prec s_2) \wedge (s_2 \rightarrow c) \wedge (s_2 \rightarrow s_1[cat = S]) \wedge (c \prec s_1) \\
& \wedge (c \rightarrow wh[cat = wh]) \wedge (s_1 \rightarrow n_1[cat = n]) \} \\
Active \rightarrow & \langle syn \rangle \{ (s_1 \rightarrow v_2[cat : V]) \} \\
Transitive \rightarrow & CanSubj \wedge RelObj \wedge Active
\end{aligned}
$$

The minimal models for the classes named CanSubj, Active and Object are the trees with matching names on the previous figure. The tree Transitive is a minimal model for the description accumulated in class Transitive.

A language for semantics

To describe semantics, we use another description language, which is:

$$
SemDesc \ := \ \ell : p(E_1, ..., E_n) \mid \neg \ell : p(E_1, ..., E_n) \mid E_i << E_j \mid E
$$

where ℓ is a label for predicate p (of arity n) and $<<$ is a scope-over relation for dealing with quantifiers. To add binary relations to the semantic dimension, we can use a class of this type:

$$
BinaryRel[Pred, X, Y] \rightarrow \langle sem \rangle \{ Pred(X, Y) \}
$$

When instantiated with $Pred{=}love$, $X{=}John$, $Y{=}Mary$, calling the class $BynaryRel$ accumulates the predicate $love(John, Mary)$.

2.2 Principles

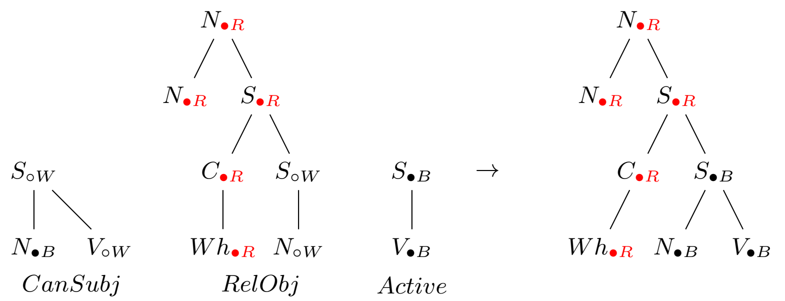
Some additional sets of constraints we call principles are available. Their goal is to check some properties in the resulting models of the compilation, they are consequently dependent from the target formalism. For example, in TAG, the color principle is a way to forbid some fragments combination, by associating colors to each node.

When unifying nodes, their colors are merged: a red node must not unify, a white node has to unify with a black node, creating a black node, and a black node can only unify with white nodes. The only valid models are the ones in which every node is colored either in red or black. The following table shows the results of colors unifications.

	$\bullet_B$	$\bullet_R$	$\circ_W$	$\perp$
$\bullet_B$	$\perp$	$\perp$	$\bullet_B$	$\perp$
$\bullet_R$	$\perp$	$\perp$	$\perp$	$\perp$
$\circ_W$	$\bullet_B$	$\perp$	$\circ_W$	$\perp$
$\perp$	$\perp$	$\perp$	$\perp$	$\perp$

Fig. 2. Unification rules for colors.

For example, if we consider our previous example, the colored trees of the metagrammar are the following:



The tree description solver (ignoring the colors) will produce models where the nodes labelled S of CanSubj and Active unify with any of the two nodes labelled S in RelObj, where the nodes labelled V do not unify, etc. But when filtering with the colors principle, the only remaining model is the one of the right, which is linguistically valid, contrary to the others.

We can also cite the rank principle: we use it to add constraints on the ordering of nodes in the models of the description. In French for example, clitics are necessarily ordered, so we associate a rank property to some nodes, with values that will force the right order.

3 Assembling metagrammars in a modular way

The main aim of the XMG-2 project is to make it possible for the linguist to design new metagrammatical scopes, that can accomodate any linguistic theory. A simple way to realize this ambition is to provide a set of bricks the user can pick to build the compiler he needs. Those bricks could be used to design new dimensions, with new description languages or new principles.

3.1 A modular architecture

XMG compiler comes with a modular processing chain. Most of this chain is a standard compiling chain, including a tokenizer for the metagrammar, a parser, an unfolder, etc.

The particularity of XMG is to make it possible to chose the modules that suits the best his metagrammar. By this mean, descriptions accumulated in different dimensions can be handled differently. For example, the end of the processing chain for TAG is a tree description solver, that builds the grammar's elementary trees from the descriptions accumulated in the syntactic dimension. The user can chose the kind of output the compiler will produce: he can interactively observe the grammar he produced, or produce an XML description of the grammar. This description can be used by a parser (for example TuLiPA [9] [3] for TAG, or LeoPar [4] for IG).

3.2 Representation modules

As we wish to build a tool which is as universal as possible, being independent from the formalism is a priority. To achieve this goal, we need to be able to describe any type of structure into XMG. We saw the dimension system was useful to separate syntax from semantics. It could also be used to separate tree descriptions from constraints based descriptions, as long as we have a dedicated dimension, with a dedicated description language.

In [6], description languages for two formalisms, namely Lexical Functional Grammars (LFG) and Property Grammars (PG), are proposed. Here, we will focus on Property Grammars, because they differ from TAG in many aspects. PG are not based on tree rewriting but on a local constraints system: the properties. A property concerns a node and applies constraints over its children nodes. One of the interesting aspects of PG is the ability to analyse non grammatical utterances. When parsing a utterance, its grammaticality score is lowered at every violated property. Here, we will consider these six properties:

Obligation	$A : \triangle B$	at least one B child
Uniqueness	$A : B!$	at most one B child
Linearity	$A : B \prec C$	B child precedes C child
Requirement	$A : B \Rightarrow C$	if a B child, then also a C child
Exclusion	$A : B \not\Leftrightarrow C$	B and C children are mutually exclusive
Constituency	$A : S$	children must have categories in S

A real size PG consists in a inheritance hierarchy of linguistic constructions. These constructions are composed of feature structures and a set of properties. Variables are manipulated on both sides, and can be used to share data between them. Figure 3 represents a part of the hierarchy built in [7] for French. The V-n construction of the figure says that in verbs with negation in French, negation implies the presence of an adverb *ne* labelled with category $Adv - ng$ (*ne*) and/or an adverb labelled with category $Adv - np$ (like *pas*). We also have a uniqueness obligation over these adverbs,

[3] `https://sourcesup.cru.fr/tulipa/`

[4] `http://wikilligramme.loria.fr/doku.php?id=leopar:leopar`

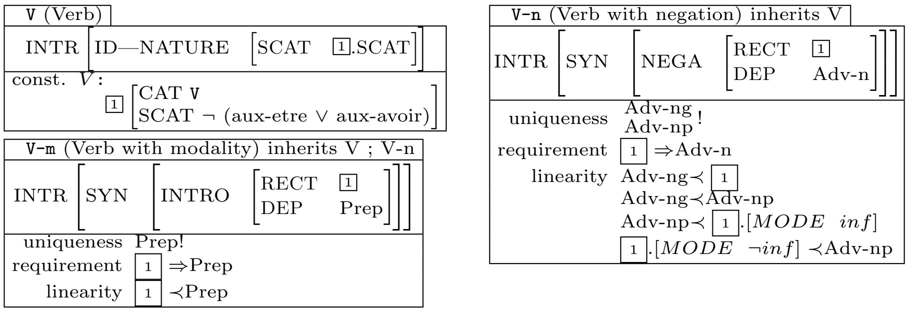


Fig. 3. Fragment of a PG for French (basic verbal constructions)

and an linear order must be respected (*ne* must come before *pas*). When the mode of the verb is infinitive, the verb must be placed after the adverbs.

To describe a PG, we need to be able to represent encapsulations, variables, feature structures, and properties. We can notice that XMG classes can be seen as encapsulations, and that variables and features structures were already used for TAG descriptions. Considering that, the XMG description language for PG can be formalized this way:

$$\begin{array}{rcl} Desc_{PG} & := & x = y \;\mid\; x \neq y \mid\; [f{:}E] \;\mid\; \{P\} \mid\; Desc_{PG} \;\wedge\; Desc_{PG} \\ P & := & A : \triangle B \;\mid\; A : B! \;\mid\; A : B \prec C \;\mid\; A : B \Rightarrow C \;\mid\; A : B \not\Leftrightarrow C \;\mid\; A : B \end{array}$$

where x, y correspond to unification variables, $=$ to unification, $\neq$ to unification failure, $:$ to association between the feature f and some (possibly complex) expression E, and $\{P\}$ to a set of properties. Note that E and P may share unification variables.
The translation of the linguistic construction for V-m in XMG would be:

$$\begin{array}{rl} V{-}m \;\rightarrow & (Vclass \vee V{-}n) \wedge \langle PG \rangle \{[\mathsf{INTR}{:}[\mathsf{SYN}{:}[\mathsf{INTRO}{:}[\mathsf{RECT}{:}X, \mathsf{DEP}{:}\mathtt{Prep}]]]] \\ & \wedge\, (V : \mathtt{Prep}!) \wedge (V : X \Rightarrow \mathtt{Prep}) \wedge (V : X \prec \mathtt{Prep})\} \end{array}$$

Here, inheritance is made possible by calls of classes. The control language even allows to do disjunctive inheritance, like it happens in class V-m. The end of the compilation process for PG will differ from TAG's one. We don't need any solver for descriptions, the accumulation into PG dimension is the grammar. To get the properties solved for a given sentence, the solution is to use a parser as a post processor for the compiler.

Nevertheless, including a specific representation module to the compiler can be seen as an ad-hock solution. That is why allowing the linguist to build his own description language (for example, choosing to use feature structures, dominance relations between nodes, open unification, etc), would be an essential feature.

3.3 Principle bricks

The notion of principles defined in XMG was too restrictive for our aims. Their specificity for the target formalism, for example, is incompatible with the multi-formalism ambition. An interesting way to handle principles is the one of [3], both allowing the linguist to create his own principles or to use a subset of the ones already defined. An example is the tree principle, which states that the solution models must be trees. What we aim to provide is a meta-principles library: generic and parametrizable principles the user can pick and configure. For example, the color principle provided for TAG could be an implementation of a generic polarity principle, parametrized with the table of figure 2. Another example of meta-principle is called unicity and was already implemented in XMG-1. It is used to check the uniqueness of a specific attribute-value pair in each solution, and thus is not specific to any linguistic theory.

3.4 Dynamic definition of a metagrammar

To build his own metagrammatical scope, one should only have to select the dimensions he needs and the properties he wants to check on them. Building a dimension would consist in picking bricks out from a library to create a new description language. With this feature, a user could redefine the property grammars description we proposed earlier. The advantage here is that the specific part of the compiler is written automatically, and new features could be added just for experiments. Defining the principles would just consist in taking meta-principles out from the library and instantiate them.

Building a metagrammar compiler in this way allows to deal with a large range of linguistic theories, or even to quickly experiment while creating a new grammar formalism.

4 Current state of the work

XMG project started in 2003 with a first tool, that has been used to produce large TAG grammars for French [2], German [10] and English, and a large Interaction Grammar for French [11]. The compiler was written is Oz/Mozart, a language which is not maintained any more and not compatible with today's architectures (64 bits). It was also important to restart from scratch, in order to build a compiler more in adequation with its ambitions : modularity and extensibility.

Consequently, a new implementation started in 2010, in YAP (Yet Another Prolog) with bindings with Gecode for constraints solving. XMG-2 is currently the tool used for modeling the syntax and morphology of Ikota, a bantu language [5], and is getting close to total compatibility with the previous large metagrammars. It also includes a dimension for basic property grammar description. The work focuses now on a parser generator which, from a description of a description language, produces the parser rules for this language. The first application could be the dynamic generation of a language dedicated to morphologic descriptions. We also wish to implement quickly some generic principles, beginning with the tree principle.

5 Conclusion

In this paper, we showed how modularity, together with a metagrammatical approach, eases the development of a large scale grammar. This modularity is essential for reaching the main goal of XMG, that is to say extensibility. Getting to that means taking a big step towards multi-formalism and multi-language grammar development, and then offers new possibilities for sharing data between different types of grammar, or even for comparing them.

Now, what we would like to create is a way to express the definition of dimensions and meta-principles. This could begin by formalizing a description language for description languages. We also aim to provide more checking tools to the user, beginning with the type checking of the properties and the feature structures we manipulate in a lot of grammar formalisms.

References

1. Candito, M.: A Principle-Based Hierarchical Representation of LTAGs. In: Proceedings of COLING 96. Copenhagen, Denmark (1996)
2. Crabbé, B.: Représentation informatique de grammaires fortement lexicalisées : Application à la grammaire d'arbres adjoints. Ph.D. thesis, Université Nancy 2 (2005)
3. Debusmann, R.: Extensible Dependency Grammar: A Modular Grammar Formalism Based On Multigraph Description. Ph.D. thesis, Saarland University (4 2006)
4. Duchier, D., Le Roux, J., Parmentier, Y.: The Metagrammar Compiler: An NLP Application with a Multi-paradigm Architecture. In: Proceedings of the 2nd Oz-Mozart Conference, MOZ 2004. Charleroi, Belgium (2004)
5. Duchier, D., Magnana Ekoukou, B., Parmentier, Y., Petitjean, S., Schang, E.: Describing Morphologically-rich Languages using Metagrammars: a Look at Verbs in Ikota. In: Workshop on "Language technology for normalisation of less-resourced languages", 8th SALTMIL Workshop on Minority Languages and the 4th workshop on African Language Technology. Istanbul, Turkey (2012), http://hal.archives-ouvertes.fr/hal-00688643/en/
6. Duchier, D., Parmentier, Y., Petitjean, S.: Cross-framework grammar engineering using constraint-driven metagrammars. In: CSLP'11. Karlsruhe, Allemagne (2011), http://hal.archives-ouvertes.fr/hal-00614661/en/
7. Guénot, M.L.: Éléments de grammaire du français pour une théorie descriptive et formelle de la langue. Ph.D. thesis, Université de Provence (2006)
8. Joshi, A.K., Schabes, Y.: Tree adjoining grammars. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages. Springer Verlag, Berlin (1997)
9. Kallmeyer, L., Maier, W., Parmentier, Y., Dellert, J.: Tulipa - parsing extensions of tag with range concatenation grammars (June 2009), first Polish-German Workshop on Research Cooperation in Computer Science
10. Kallmeyer, L., Lichte, T., Maier, W., Parmentier, Y., Dellert, J.: Developing a tt-mctag for german with an rcg-based parser. In: LREC. ELRA (2008)
11. Perrier, G.: A French Interaction Grammar. In: RANLP. Borovets, Bulgaria (2007), http://hal.inria.fr/inria-00184108/en/

Questions of Trust

Jason Quinley[1] and Christopher Ahern[2]

[1] University of Tübingen
jason.quinley@student.uni-tuebingen.de
[2] University of Pennsylvania
cahern@ling.upenn.edu

Abstract. We consider the application of Game Theory in the modeling of different strategies of politeness. In particular, we examine how differences in the linguistic form of requests and proposals map onto the structure of the game being played by interlocutors. We show how considerations of social wants [1, 2] and coordination and cooperation motivate these differences. First, we adapt the notion of other-regarding preferences [3, 4] to show how linguistic systems with the ability to encode politeness strategies allow for requests and cooperation between a wider range of individuals. Finally, we connect the distinction between requests and proposals to the notion of a self-enforcing equilibrium [5].

1 Introduction

Questions in their many forms are central to social interaction. Asking someone for a dollar, or if they would like to see a movie, are commonplace yet revelatory. The indicate how we use language not only to convey information, but also to negotiate relationships. A clear case of these distinctions can be had in the use of the modals *will* and *would* in the following requests:

(1) Will/Would you lend me a dollar?
(2) Will/Would you open the door?
(3) Will/Would you turn that music down?
(4) Will/Would you marry me?

Consider asking these questions of a stranger. It would be impolite to omit the modal. Moreover, between the two modals we also sense a difference in effect. In the first two cases *would* is the more polite form of request. In the third either is acceptable, modulo the degree to which the music affects the speaker. In the last, *will* seems the more appropriate form. Moreover, *would* allows for comedic response: I would if you were rich/handsome/x!

Why are such questions necessary? One reason is that scarcity and ambiguity drive interaction. We have neither unlimited resources nor unlimited information with which to achieve our ends. This leads to the need for cooperation, and with it, strategies to address its fragility. As humans have access to language, they are availed of multiple avenues of cooperation. Studying these allows for a fruitful combination of theories of language and rational interaction.

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see http://ceur-ws.org/, 2012, pp. 121–132.

In what follows, we examine the use of modals in requests and proposals. We show how modals, and other politeness strategies, when thought of in terms of other-regarding preferences, allow for expanded interaction between individuals. We also show how and why the use of the modal *will* in requests is binding, whereas *would* is not necessarily so. We begin by presenting the relevant notions from politeness- and game theory, then turn to our analysis of requests in these terms and suggest future directions.

2 Politeness Theory and Speech Acts

Beginning from Goffman's [1] notion of *face*, Brown and Levinson [2] articulated an ur-theory of politeness, which has prompted much subsequent theoretical and empirical work. *Face* is the term given to an individual's basic needs, characterized broadly as the need for autonomy (negative face) and acceptance (positive face). Broadly, positive face can be thought of as the wants of the individual, including the desire that those wants be desirable to or approved of by others. Negative face includes both the freedom of action and the freedom from imposition.

Preferences of one agent may conflict with those of others, incentivizing them to make requests, issue threats, or offer proposals. In cases where a request must be made, speakers must commit a *face-threatening act* (*FTA*). In order to mitigate the weight of a FTA, speakers may use several strategies, as laid out in Figure 1.

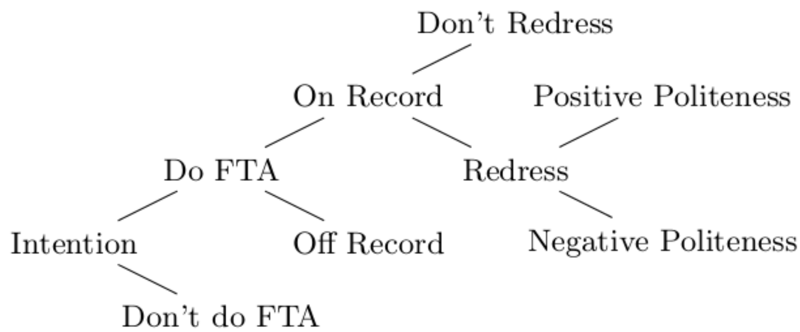


Fig. 1. Brown and Levinson's Politeness Strategies: As we move upwards on the graph, the potential for a *face-threatening act* (FTA) increases.

At the two extremes, a speaker might avoid making the FTA altogether, or state it in a direct manner. In between there are various degrees of deference to the hearer's face wants: indirect speech that is "off the record" and addressing the hearer's positive or negative face. As a concrete example, consider the situation of having left one's wallet at the office while going out to lunch with a group. Here the relevant FTA might be taken as requesting some money from a friend. The various strategies of doing so could be implemented as:

(1) **Don't do FTA:** Don't ask for money.
(2) **Off Record:**[3] "Oh no! I forgot my wallet in my office!"
(3) **Negative Politeness:** "You don't have to, but would you mind lending me a bit of money?"
(4) **Positive Politeness:** "Congratulations on the raise! Want to lend me some money."

The goal of the speaker is to craft the appropriate message to convey the intent and the weight of the FTA. The greater an imposition a FTA carries, the more care needs to be taken. However, too much politeness is inappropriate given certain FTAs. It would seem odd to be asked, rather circuitously, "Excuse me Sir/Ma'am, but I was hoping that it might be possible if it's not too much trouble that you would be able to tell me the time." Similarly, when expediency is called for, "Please, if you could, move out of the way of that speeding car," would be inappropriate. Thus we might think of different forms of politeness as strategic responses to situations where face may be threatened. Again, the goal of the speaker is to select the appropriate form for the FTA in question; neither too much nor too little deference can be paid. With this notion of strategy in mind we turn to the game-theoretic framework that will figure in our analysis.

3 Basic Game Theory

Game theory gives a mathematical model of strategic interaction between agents. We begin by presenting canonical examples from the field. Crucially, we focus on the difference between cooperation and coordination in sequential play. Sequential play allows for an optimal outcome under rational behavior in cases of coordination, but not for cooperation.

Formally, a sequential game is a tuple $\langle N, O, A_j, U_i \rangle$. N is the set of players in the game. O is a sequence over N that determines the order of play; for $j \in O$, A_j is the set of actions available to the jth player in the order of play. Finally, U_i is a preference for player i over the set of possible paths of play. The payoffs are represented as numeric values, where higher values are taken to be more preferred outcomes. The Prisoner's Dilemma (PD) offers the canonical example of choosing between cooperation and defection. The game reflects a scenario wherein two prisoners must choose between cooperatively staying silent (C), telling the police nothing about their crime, or defecting on each other (D) and confessing the details to the police. Jointly, both prisoners do better if they remain silent, but individually, they do better by ratting out their accomplice. We represent the structure of the Prisoner's Dilemma in *extensive form* in Figure 2. Each node in the game is labeled with the letter of the player whose turn it is to take an action, $O = \langle X, Y \rangle$. The payoffs, as determined by the utility functions, are listed as (U_X, U_Y) at the bottom of the tree.

We use the notion of a *rollback equilibrium* to examine expected behavior in the game. The reasoning proceeds as follows. We begin by considering the lowest nodes in the game and putting in bold the best action available. For any node where Y can make a choice, she should choose D as it is always the better option. Knowing that

[3] See Pinker et al. [6] as well as Mialon [7] for game-theoretic treatments of indirect speech.

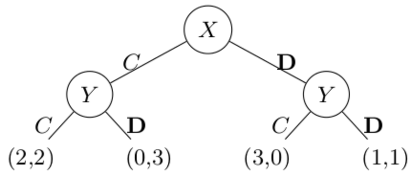


Fig. 2. A Sequential PD

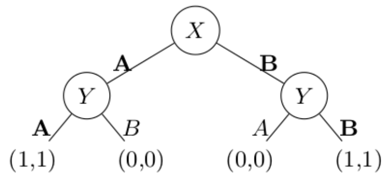


Fig. 3. A Sequential PCG

this is the case, X should always choose D, as it always the best option based on what Y will do. That is, cooperation will only ever be met with defection, so it should never be explored. We will refer to those instances where players have diverging interests but come together to yield the optimal outcome for all as instances of cooperation. In instances of cooperation, as in the PD, reaching the best outcome for the players as a group requires some sacrifice in terms of individual payoff. That is, each player must forgo the temptation payoff of defecting in order to maintain cooperation.

In contrast, in cases of coordination, players' incentives do not conflict. Consider the case of a Pure Coordination Game (PCG) in Figure 3. Here players are ambivalent between the actions they take, they only prefer to take the same action. An example might be a scenario where two friends want to meet up for lunch at noon. If one player suggests a restaurant, then the other should indeed go to that restaurant. If X plays A (B), then Y should play A (B). Sequential games allow for the optimal outcome for both players in pure coordination games.

These game structures allow us to distinguish between the notions of cooperation and coordination. We might think of the first as exemplified in the Prisoner's Dilemma, and the latter in the case of Pure Coordination Games. With this background in place, we now turn to the analysis of modals in requests and examine the different rationales for polite behavior.

4 Trust and Modals

In this section we show how face-addressing forms allow for requests between a wider range of individuals. This serves as a broad motivation for using such forms with strangers. We then turn to a distinction between requests in general and marriage proposals in particular, where we argue for a distinction between the different sorts of speech acts involved as they relate to the notion of self-enforcing equilibria.

4.1 Requests as Extended Trust Games

Quinley [8] adapts Trust Games [9] as a model of requests. We borrow techniques and insights from this approach and introduce Extended Trust Games to capture the sequential dynamics of requests. We note the effects of repetition, reputation, and observation on polite forms in requests, but suggest that they are not sufficient to fully explain the use of politeness strategies. Instead, we propose other-regarding preferences as a means to explain the use of modals and other forms of linguistic politeness in a variety of situations.

Trust games are an appropriate model for requests due to several factors. First, individuals are rarely if ever entirely self-sufficient. Moreover, agents possess different aptitudes and abilities, and this asymmetry prompts requests. Requests entail a loss of face on the part of the requester; so to speak, the requester makes a face "payment" to the requestee. Finally, the requestee is not obligated to grant the request, presenting the agent in need with the risk of both a loss of face and having their request denied.

Trust games depict a scenario where Player X has an initial option to defer to Player Y for a potentially larger payoff for both. We extend this notion further, incorporating a third step in the order of play. Here the play of the game is shown in extensive form in Figure 4 and consists of the first player asking or not asking for some favor, the second player granting or not granting the request, and the requester thanking or not thanking the requestee. We considered a more detailed motivation of the utility structure below.

If X does not ask ($\neg A$), then the status quo remains and X is left to her own devices. Let c_x be the cost to X to achieve the desired outcome. Let c_y be the cost to Y to achieve the same outcome. As noted before, assume an asymmetry in ability or disposition such that $c_y < c_x$; Y is in a better position than X to bring about X's desired state of affairs. If X does ask (A) for help, using a polite request, Y should experience some boost in self-esteem based on the attention received. That is, by acting in accordance with Y's face wants, X increases Y's face. Let the amount of face paid by X to Y in the request be f_r. Let m_r be a multiplicative factor that acts upon f_r to determine the payoff to Y. If talk is cheap, then flattery is certainly sweet; a little bit of face goes a long way, so we assume that $m_r > 1$. Even if Y chooses not to grant the request, Y still comes away with some benefit based on the face paid by X, $m_r f_r$. If Y denies the request ($\neg G$), X has incurred the face cost of asking without receiving any benefits, and must also bear the cost of performing the action, c_x. If Y chooses to grant the request (G), then Y incurs some cost of the action, but still receives the benefit of face from X. Let the benefit to X of Y granting the request be b_x. In general, we assume that $b_x < c_x$. If the request is granted, then X has an opportunity to express to Y some sort of thanks (T) or not ($\neg T$). This expression of thanks, again, comes at some cost f_t, and, again, carries with it some face benefit to Y as determined by a factor $m_t > 1$.

We are faced with the same problem as the prisoner's dilemma; requests are FTAs that require cooperation. X prefers $\neg T$ to T, Y prefers $\neg G$ to $\neg T$, and X prefers $\neg A$ to $\neg G$. Thus, if we are only maximizing individual utility, it never makes sense in a one-shot scenario to ask, grant, or thank, even though both players might prefer the interaction under certain assumptions. We thus consider the effect of repetition, reputation, and observation on the outcome.

4.2 Repetition and Reputation

Under various conditions, repetition engenders cooperation [10]. More specifically, with a given probability of another round of play, group welfare becomes individual welfare; i.e. a PD becomes a Stag Hunt [11]. In a Stag Hunt, players' interest are highly aligned, and the only pitfall is the possibility of mis-coordination. Importantly, in a Stag Hunt players wish to coordinate, but may not necessarily know how when playing simultaneously. In Figure 5, the Stag Hunt structure assumes that players have aligned preferences, and shared preferences over outcomes.

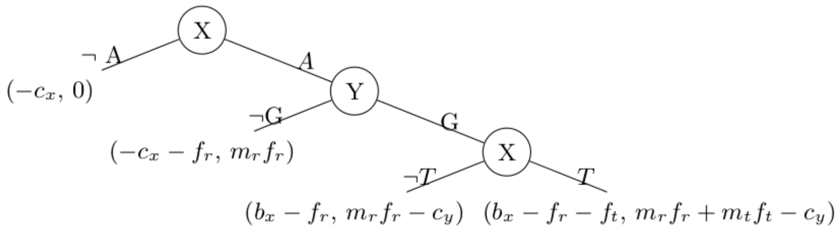


Fig. 4. Request Trust Game: Player X can choose to Ask (A) something from Player Y, who can then choose to Grant (G) the favor. Player X can choose to Thank (T) or not Thank ($\neg T$) player Y.

	Stag	*Rabbit*
Stag	4,4	0,1
Rabbit	1,0	1,1

Fig. 5. Stag Hunt (SH): in strategic form

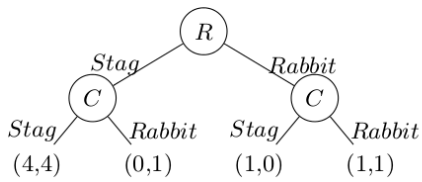


Fig. 6. Stag Hunt (SH): in extensive form

Here, as in the case of coordination games, sequential play allows the players to achieve the optimal outcome. That is, if the first player plays *Stag*, then the second player should as well. Repetition transforms a Prisoner's Dilemma to a Stag Hunt. However, repetition cannot be all there is to the outcome of the interactions we consider here. People are polite to strangers they will never see again.

The effects of reputation and observation on different strategies in trust games are explored in Quinley [8]. Namely, asking requests of other agents is rational when there is a sufficient likelihood that the request will be granted based on the requestee's reputation. Or, in the case here, granting requests is rational when there is sufficient likelihood that X will play T. If Y has sufficient experience or knowledge about the behavior of X with regards to $Pr(T)$, then this suffices to render granting the request rational strategy or not. The novel contribution of Quinley is the inclusion of face effects due to third-party observation. In line with experimental results [12], such observation, framed as a loss (gain) in face for Y when denying (granting) a request, is shown to ensure that requests are asked and granted by and large. This can be extended similarly to X's actions when choosing to thank Y or not.

4.3 Other-Regarding: Reciprocity Without Repetition

While reputation and observation offer rationales for asking and granting requests, they are unlikely to explain all of the behavior we observe. Modals are used to make requests in one-shot interactions where nothing is known about the other individual and there are no third-party observers. In fact, the polite use of modals is even more expected in these sorts of situations. This suggests that reputation and observation are not alone in explaining the behavior observed in requests A rationale for politeness strategies in such situations can be found when we consider other-regarding preferences.

There exists a wealth of theoretical work on [13–15], and behavioral [16, 17] and neurobiological evidence [18] of other-regarding preferences. Here we adapt the notion of sympathy as advanced by Sally [3, 4] to explain the observed behavior. The central notion is that of a sympathy distribution over the payoffs of all the agents involved in the game. For each agent, there is a distribution, $\delta_i \in \Delta(U)$, such that $\sum_j \delta_i(U_j) = 1$, which determines how much that agent cares about her own payoffs and those of others. For example, the perfectly self-interested agent of classical Economic theory is such that $\delta_i(U_j) = 0$ for all $j \neq i$. A selfless agent would be such that $\delta_i(U_i) = 0$.

Here we consider the limiting case of a single interlocutor. Based on the sympathy distribution and the utility function U of the original game, we define a new utility function V.

$$V_i = \delta_i(U_i) \cdot U_i + (1 - \delta_i(U_i)) \cdot U_j \tag{1}$$

The impact of other-regarding preferences can be seen in the following. Consider what values would suffice to make thanking rational for X in the Extended Trust Game of Figure 4. Namely, we wish to determine the condition under which the sympathy distribution of X renders thanking (T) preferable to not thanking ($\neg T$). This holds just when:

$$\begin{aligned} V_x(\neg T) &< V_x(T) \\ \frac{1}{1+m_t} &< \delta_x(U_y) \end{aligned} \tag{2}$$

Given that $m_t > 1$, the highest threshold will be bounded from above by $\frac{1}{2}$. As m_t increases, the threshold approaches 0. The greater the benefit to Y for thanking, the less X has to care about Y's payoff to do so. This undoes some of the unraveling effect of divergent preferences. We move on to determine the conditions on Y's preferences that suffice to allow for cooperation. That is, we wish to determine when Y prefers T to $\neg G$.

$$\begin{aligned} V_y(\neg G) &< V_y(T) \\ \frac{(c_y - m_t f_t)}{(c_y - m_t f_t) + b_x + c_x - f_t} &< \delta_y(U_x) \end{aligned} \tag{3}$$

There several important points to consider. First, the thresholds we have outlined here are the conditions under which the underlying game of cooperation is transformed into one of coordination. That is, if these thresholds are surpassed, then the game is one of coordination rather than cooperation, and we should expect requests to be made, granted, and thanks expressed. If X's condition on thanking is not met, then the request should be granted if Y prefers $\neg T$ to $\neg G$, which is true just when:

$$\begin{aligned} V_y(\neg G) &< V_y(\neg T) \\ \frac{c_y}{c_y + b_x + c_x} &< \delta_y(U_x) \end{aligned} \tag{4}$$

Otherwise, we should expect requests not to be made.

Second, we find that if $m_t f_t > c_y$, then the request should be granted for anyone, regardless of the sympathy distribution. We might be tempted to think of T in terms of expressing a future commitment to cooperation. While T has this flavor, it does not have this force; thanks, like talk, are cheap. For the expression of thanks to outweigh the cost, c_y, would require either something particularly important to Y, or some strong guarantee on the part of X. Again, future guarantees are not available in the case of single interactions with strangers.

Third, note that $c_x - f_t > c_y - m_t f_t$ given that $c_x > c_y$ and $m > 1$. Moreover, note that $b_x > c_x$, and thus $b_x > c_y - m_t f_t$. As such, as we collapse the non-fixed values towards zero, we see that $\frac{1}{3}$ serves as an upper bound on the threshold. The use of a face addressing form, such as the polite use of modals, allows for a lower threshold of other-regarding preferences for requestees compared to requesters. This makes intuitive sense as requesters are more inherently self-interested.

Finally, the use of politeness strategies that address face allow for a lower threshold than a system without such forms. Consider a faceless Trust Game, where $f_t = f_r = 0$ for the payoffs in Figure 4. We can think of this as a system where no transfers of face are possible. The structure of the game reduces to a choice on the part of Y between granting or not granting the request. The corresponding threshold of other-regarding preference can be given as follows:

$$\begin{aligned} V_{y'}(\neg G) &< V_{y'}(G) \\ \frac{c_y}{c_y + b_x + c_x} &< \delta_{y'}(U_{x'}) \end{aligned} \tag{5}$$

From Eq. (3) and (5) we know that a system with face requires a lower sympathy threshold than one without face just when:

$$\begin{aligned} \delta_y(U_x) &< \delta_{y'}(U_{x'}) \\ \frac{c_y}{b_x + c_x} &< m_t \end{aligned} \tag{6}$$

Given $c_y < c_x$, we know that $\frac{c_y}{b_x + c_x} < 1$. Since, $m_t > 1$, it is always the case that a system with face requires a lower threshold than a system without, thus allowing for requests between a wider range of individuals. In this sense, when considered in the context of other-regarding preferences, face allows for cooperation by smoothing out the payoffs of the interlocutors. By "investing" in each other's face, we can guarantee cooperation more easily, even with people we do not know.

4.4 Proposals and Credible Signaling

In contrast with requests, proposals encode an interaction potentially to the benefit of both participants. Returning to marriage, we noted the use of modals in certain contexts differs. For purposes of both humor and invoking the undercurrent of common knowledge, we observed that *would* allows for a certain amount of disavowal whereas *will* does not. For example, the following dialogues can be completed for comedic effect:

Xavier: Would you marry me?
Yvonne: *I would...if you were rich.*
Xavier: **Sigh**
(or)
Yvonne: *Yes!!!*
Xavier: *Woah, I was just asking hypothetically!*

Xavier: Would you like to see a movie?
Yvonne: *Yeah, there are a few I'd like to see.*
Xavier: *Great! When can I pick you up?*
Yvonne: *Oh! I didn't realize you meant with you.*
(or)
Yvonne: *Yeah! When do you want to go?*
Xavier: *Oh! I didn't mean with me, just in general.*

We argue that *will* and *would*, for the most part, have the same illocutionary force. However, they differ in that *would* allows for disavowal. To tease out how they do differ, we consider the notion of self-enforcing equilibria.

Aumann considers the game in Figure 7 with pre-play communication. The game has two *Nash Equilibria*: combinations of actions from which neither player can profitably deviate from unilaterally. The equilibria are (C, C) and (D, D). It would seem that both players should settle on playing C, since it is the *payoff dominant* equilibrium. However, this outcome is not guaranteed, even with communication. Suppose both players agree to play C. Suppose X pauses to think about Y. If Y does not trust him, then Y will play D despite the agreement to play C. Y would still want X to play C regardless of what Y does. So, just because both players have agreed to play C, it does not mean that they will; the agreement and the associated equilibrium are not self-enforcing.

	C	D
C	3,3	0,2
D	2,0	1,1

Fig. 7. Aumann's Game

	M_r	M_i
A_r	$v - f_n,\ v + f_n$	$-f_n - f_p, f_n$
A_i	$0, -f_p$	0,0

Fig. 8. Adjusted Aumann's Game

In light of the dialogues above, we might think of the strategies available to X(avier)as either asking for information (A_i) or asking as a request (A_r). Similarly, think of the strategies available to Y(vonne) as interpreting the question as asking for information (M_i) or as a request (M_r). We motivate the utility structure as follows. Suppose that (A_i, M_i) results in some baseline payoff where both players receive 0. Now, suppose that X intends the question as a request, A_r, but Y takes it as a request for information, M_i. X has made some effort to address Y's negative face, and thus is out some effort, f_n, which is transferred to Y. Moreover, X is embarrassed by the miscommunication and loses some amount of positive face because Y does not have the same wants as him. Similarly, if Y assumes a request, but X does not, then Y loses some amount of positive face. Finally, when X intends a request and Y interprets it as such, then both achieve some payoff, v, modulo a transfer of negative face. These payoffs are given in Figure 8.

On the reasonable assumption that $v > f_n$, there are two pure Nash equilibria: (A_i, M_i), (A_r, M_r). The payoff dominant equilibrium, (A_r, M_r), is not self-enforcing. X prefers for Y to play M_r regardless of what X intends to do; Y prefers for X to play A_r regardless of what Y intends to do. Thus, we can predict the disavowals that occur. However, by and large, we do commit ourselves to making requests, A_r, with *would* and this is because other-regarding preferences transform the payoff structure. The use of *would* is self-enforcing just in case $0 < \delta_x(U_y)$ and $\frac{f_n}{2f_n+f_p} < \delta_y(U_x)$. There are two things to note. First, the comedy of the dialogues above stems from the mismatch between a generally expected amount of sympathy and that displayed. Second, the disavowal on the part of the speaker seems far crueler than what could be an honest mistake on the part of the hearer, as predicted by the fact that $\delta_x(U_y) < \delta_y(U_x)$.

The crucial distinction between *would* and *will*, and why *will* is the appropriate choice for a marriage proposal is evidenced by the effect of not paying negative face to the hearer, as in Figure 10. That is, in a marriage proposal, (A_r, M_r) is a self-enforcing equilibrium much like the classical Stag Hunt, where both players benefit by coordinating on the payoff-dominant choice.

	Stag	*Rabbit*
Stag	4,4	0,1
Rabbit	1,0	1,1

Fig. 9. Stag Hunt (SH): in strategic form

	M_r	M_i
A_r	v, v	$-f_p$,0
A_i	0,$-f_p$	0,0

Fig. 10. Marriage Game

Thus, using the modal *will* ignores the listener's negative face, but renders the request self-enforcing. This aligns perfectly with our intuition that one cannot back out after asking "Will you marry me?". Moreover, this reasoning about face and other-regarding preferences provides a rationale for why commissive speech acts are possible, and the form they take.

4.5 Summary

We have shown that the transfer of face via politeness strategies with other-regarding preferences allows requests and trust between a wider range of individuals. Specifically, we have shown the necessary amount of sympathy between two individuals that suffices to transform a game of cooperation into one of coordination, and that face lowers this threshold. In addition, we have shown that *would* and *will* differ fundamentally in terms of illocutionary force, and the underlying structure of the interaction. *would* allows for disavowal and is not necessarily self-enforcing, whereas *will* as a commissive speech act commits the speaker to a course of action. In parallel to results from dynamic epistemic logic [19], saying *will* creates common knowledge between the participants of the hearer's commitment to future action, and thus it is only rational in the case that both participants have a benefit towards taking that action and that the action cannot be repeated.

5 Conclusion

This work follows in the vein of approaches to pragmatics and politeness from a strategic viewpoint. It defines the conditions under which politeness strategies are rational in those situations where repetition, reputation, and observation do not hold. A central result is that a system with face allows for a greater level of trust between agents with other-regarding preferences. Also, it outlines how the modals *will* and *would* map onto fundamentally different game structures and predicts both the humorous possibilities of denial and the real power of socially-binding statements. Future directions include extending the current analysis to threats such as *Will you cut that music out!* and requests for information *Will you be here later?*, and providing a broader theoretical framework for the description of speech acts. The results presented here demonstrate the growing ability of game-theoretic methods to model pragmatic phenomena, including politeness. Moreover, though reciprocity and coordination existed outside of and prior to language, language nonetheless serves as an efficient tool for managing them in relationships.

References

1. Goffman, E.: Interaction Ritual: Essays on Face-to-Face Behavior. 1st pantheon books edn. Pantheon, New York (1982)
2. Brown, P., Levinson, S.C.: Politeness: Some universals in language use. Cambridge University Press, Cambridge (1978)
3. Sally, D.: A general theory of sympathy, mind-reading, and social inter- action, with an application to the prisoners' dilemma. Social Science Information **39**(4) (2000) 567–634
4. Sally, D.: On sympathy and games. Journal of Economic Behavior and Organization **44**(1) (2001) 1–30
5. Aumann, R.J.: 34. In: Nash Equilibria Are Not Self-Enforcing. Elsevier (1990) 667–677
6. the National Academy of Sciences of the United States of America: The logic of indirect speech, the National Academy of Sciences of the United States of America (2007)
7. Mialon, H., Mialon, S.: Go figure: The strategy of nonliteral speech. (2012)
8. Quinley, J.: Trust games as a model for requests. ESSLLI 2010 and ESSLLI 2011 Student Sessions. Selected Papers (2012) 221–233
9. Berg, J., Dickhaut, J., McCabe, K.: Trust, reciprocity, and social history. Games and Economic Behavior **10**(1) (July 1995) 122–142
10. Mailath, G.J., Samuelson, L.: Repeated games and reputations: long-run relationships. Oxford University Press, Oxford (2006)
11. Skyrms, B.: The Stag Hunt and the Evolution of Social Structure. Cambridge University Press, Cambridge (2004)
12. Fehr, E., Fischbacher, U.: Third-party punishment and social norms. Experimental 0409002, EconWPA (September 2004)
13. Rabin, M.: Incorporating fairness into game theory and economics. American Economic Review **83**(5) (December 1993) 1281–1302
14. Fehr, E., Schmidt, K.M.: A theory of fairness, competition and cooperation. CEPR Discussion Papers 1812, C.E.P.R. Discussion Papers (March 1998)

15. Levine, D.K.: Modeling altruism and spitefulness in experiment. Review of Economic Dynamics **1**(3) (July 1998) 593–622
16. Fehr, E., Schmidt, K.M.: Theories of fairness and reciprocity - evidence and economic applications. CEPR Discussion Papers 2703, C.E.P.R. Discussion Papers (February 2001)
17. Camerer, C.: Behavioral Game Theory: Experiments in Strategic Interaction. Princeton University Press, Princeton (2003)
18. Fehr, E.: 15. In: Social Preferences and the Brain. Academic Press (2008)
19. Baltag, A., Moss, L.S., Solecki, S.: The logic of public announcements, common knowledge and private suspicions. Technical Report TR534, Indiana University, Bloomington (November 1999)

Checking Admissibility in Finite Algebras

Christoph Röthlisberger*

Mathematics Institute, University of Bern
Sidlerstrasse 5, 3012 Bern, Switzerland
christoph.roethlisberger@math.unibe.ch

Abstract. Checking if a quasiequation is admissible in a finite algebra is a decidable problem, but the naive approach, i.e., checking validity in the corresponding free algebra, is computationally unfeasible. We give an algorithm for obtaining smaller algebras to check admissibility and a range of examples to demonstrate the advantages of this approach.

1 Introduction

Rules and axioms are the building blocks of a logic. Axioms are the assumptions of the logic, whereas rules are used to derive new facts from previously derived facts. Rules are usually formulated as IF-THEN statements, e.g. "IF x is an integer and x is positive THEN $x+1$ is a natural number". More generally, a rule is a set of premises followed by a conclusion. In logic, the premises and the conclusion are formulas. In algebra they are usually equations, as in the cancellation rule "IF $x+y=x+z$ THEN $y=z$". Axioms are rules without a premise and can be read as, e.g., "x + y = y + x always holds". In algebra one often uses Σ to denote a finite set of equations and calles the rule "IF Σ THEN $\varphi \approx \psi$", written $\Sigma \Rightarrow \varphi \approx \psi$, a quasiequation. A quasiequation $\Sigma \Rightarrow \varphi \approx \psi$ is called valid in the finite algebra $\mathbf{A}$ if whenever every equation in Σ is true in $\mathbf{A}$ for a specific choice of elements of $\mathbf{A}$ for the variables occuring in $\Sigma \cup \{\varphi \approx \psi\}$, then also $\varphi \approx \psi$ is true in $\mathbf{A}$ for this choice.

Checking validity in finite algebras (similarly, derivability in finite-valued logics) has been studied extensively in the literature, and may be considered a "solved problem" in the sense that there exist both general methods for obtaining proof systems for checking validity (tableaux, resolution, multisequents, etc.) and standard optimization techniques for such systems (lemma generation, indexing, etc.) (see, e.g., [1, 12, 24]). A rule which can be added to a given system without producing new valid equations is called *admissible*. This notion was introduced by Lorenzen in 1955 [18], but the property of being admissible was already used by Gentzen twenty years earlier [7]. Admissibility has been studied intensively in the context of intermediate and transitive modal logics and their algebras [6, 8, 9, 13, 15, 21], leading also to proof systems for checking admissibility [2, 10, 14], and certain many-valued logics and their algebras [5, 16, 17, 19, 21], but a general theory for this latter case has so far been lacking.

Showing the admissibility of rules can play an important role in establishing completeness results. That means for example, that one proves the admissibility of the cut-rule "IF x = y and y = z THEN x = z" to show that the system can derive the same equations without the cut-rule. Moreover, in some cases adding admissible rules to a system can simplify or speed up reasoning in this system.

*Supported by Swiss National Science Foundation grant 20002_129507.

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see http://ceur-ws.org/, 2012, pp. 133–141.

Often it is possible to transform logical settings to algebraic settings and vice versa (see, e.g., [3]). In this sense rules and logics correspond to quasiequations and classes of algebras satisfying the same quasiequations, respectively. In this work we concentrate on the question, whether a given quasiequation is admissible in a finite algebra. This corresponds to the question, whether the quasiequation holds in a corresponding free algebra on countably infinitely many generators. Although it is well known that admissibility is decidable in finite algebras, the naive approach is computationally unfeasible. We give an algorithm to answer this question in a more efficient way.

This paper (based on joint research with my supervisor [20]) focuses on procedural aspects of the given problem and its solution. Necessary algebraic definitions are provided, so that also readers without experience in universal algebra are able to understand the text.

2 Validity and Admissibility

Let us first recall some basics from universal algebra. A *language* is a set of operation symbols $\mathcal{L}$ such that to each operation symbol $f \in \mathcal{L}$ a nonnegative integer $\mathrm{ar}(f)$ is assigned called the *arity of* f. An $\mathcal{L}$*-algebra* $\mathbf{A}$ is an ordered pair $\mathbf{A} = \langle \mathrm{A}, \{f_1^{\mathbf{A}}, \dots, f_k^{\mathbf{A}}\}\rangle$ such that A is a set, called the *universe of* $\mathbf{A}$, and each $f_i^{\mathbf{A}}$ is an operation on $\mathbf{A}$, corresponding to an operation symbol $f_i \in \mathcal{L}$. We often omit superscripts when describing the operations of an algebra. Let $\mathbf{A}$ and $\mathbf{B}$ be two algebras of the same language. Then $\mathbf{B}$ is a *subalgebra* of $\mathbf{A}$, written $\mathbf{B} \leq \mathbf{A}$, if $\mathrm{B} \subseteq \mathrm{A}$ and every operation of $\mathbf{B}$ is the restriction of the corresponding operation of $\mathbf{A}$. For $\{a_1, \dots, a_k\} \subseteq \mathrm{A}$ the smallest subalgebra of $\mathbf{A}$ containing $\{a_1, \dots, a_k\}$ is denoted by $\langle a_1, \dots, a_k\rangle$. We use the letters x, y, z, possibly indexed, to denote variables.

Example 1. Let $\mathcal{L} = \{\to, \mathrm{e}\}$ be a language with $\mathrm{ar}(\to) = 2$ and $\mathrm{ar}(\mathrm{e}) = 0$. Define the algebra $\mathbf{S}_{\mathbf{4}}^{\to\mathrm{e}} = \langle\{-2, -1, 1, 2\}, \to, \mathrm{e}\rangle$ with the operations

$$x \to y = \begin{cases} \max\{-x, y\} & x \leq y \\ \min\{-x, y\} & \text{otherwise} \end{cases} \qquad \text{and} \qquad \mathrm{e} = 1.$$

The algebra $\mathbf{S}_{\mathbf{2}}^{\to\mathrm{e}} = \langle\{-1, 1\}, \to, \mathrm{e}\rangle$ is a subalgebra of $\mathbf{S}_{\mathbf{4}}^{\to\mathrm{e}}$, i.e., $\mathbf{S}_{\mathbf{2}}^{\to\mathrm{e}} \leq \mathbf{S}_{\mathbf{4}}^{\to\mathrm{e}}$.

Example 2. Let $\mathcal{L}$ consist of one operation symbol $\star$ with arity 1. Then consider the algebra $\mathbf{P} = \langle\{a, b, c, d\}, \star\rangle$ where the unary operation $\star$ is described by the diagram below. The algebra $\langle\{a, b, d\}, \star\rangle$ is then clearly a subalgebra of $\mathbf{P}$.

$$\begin{tikzcd}
\mathbf{P} & a \arrow[d, shift right] & \\
& b \arrow[u, shift right] & \\
c \arrow[ur] & & d \arrow[ul]
\end{tikzcd}$$

The set $\mathrm{Tm}_{\mathcal{L}}$ of $\mathcal{L}$-terms is inductively defined: every variable is an $\mathcal{L}$-term and if $\varphi_1, \dots, \varphi_n$ are $\mathcal{L}$-terms and the operation symbol $f \in \mathcal{L}$ has arity n, then also $f(\varphi_1, \dots, \varphi_n)$ is an $\mathcal{L}$-term. We denote the term algebra over countably infinitely many variables by $\mathbf{Tm}_{\mathcal{L}}$ (i.e., for each $f \in \mathcal{L}$ with $\mathrm{ar}(f) = n$, $\varphi_1, \dots, \varphi_n \in \mathrm{Tm}_{\mathcal{L}}$, $f^{\mathbf{Tm}_{\mathcal{L}}}(\varphi_1, \dots, \varphi_n)$ is just the $\mathcal{L}$-term $f(\varphi_1, \dots, \varphi_n)$) and let φ, ψ stand for arbitrary members of the universe $\mathrm{Tm}_{\mathcal{L}}$. An $\mathcal{L}$-*equation* is a pair of $\mathcal{L}$-terms, written $\varphi \approx \psi$. If Σ is a finite set of $\mathcal{L}$-equations, we call $\Sigma \Rightarrow \varphi \approx \psi$ an $\mathcal{L}$-*quasiequation.* As usual, if the language is clear from the context we may omit the prefix $\mathcal{L}$.

Example 3. Terms in the language of Example 1 are, e.g., x, $x \to x$ or $(x \to \mathrm{e}) \to y$ whereas terms corresponding to Example 2 have the form $\star(x)$ or $\star(\star(\star(y)))$. The following is a quasiequation in the language of Example 1

$$\{x \approx y \to x,\ \ x \to \mathrm{e} \approx y\} \quad \Rightarrow \quad x \approx \mathrm{e}.$$

A *homomorphism* h between two algebras $\mathbf{A}$ and $\mathbf{B}$ of the same language $\mathcal{L}$ is a map $h\colon \mathrm{A} \to \mathrm{B}$ between their universes that preserves all the operations, i.e., for all $a_1, \dots, a_n \in \mathrm{A}$ and every operation $f \in \mathcal{L}$ with $\mathrm{ar}(f) = n$, $h(f^{\mathbf{A}}(a_1, \dots, a_n)) = f^{\mathbf{B}}(h(a_1), \dots, h(a_n))$. The homomorphism $h\colon \mathbf{A} \to \mathbf{B}$ is called *surjective* if for all $b \in \mathrm{B}$, there exists an $a \in \mathrm{A}$ such that $h(a) = b$. Two algebras $\mathbf{A}$ and $\mathbf{B}$ are said to be *isomorphic*, if there exists a surjective homomorphism $h\colon \mathbf{A} \to \mathbf{B}$ with $h(a) \neq h(b)$ for all $a \neq b$. The algebra $\mathbf{C}$ with the universe $\mathrm{C} = \{h(a) : a \in \mathrm{A}\} \subseteq \mathrm{B}$ and the restrictions of the operations of $\mathbf{B}$ to C as operations is called a *homomorphic image of* $\mathbf{A}$, written $\mathbf{C} \in \mathbb{H}(\mathbf{A})$.

We say that the quasiequation $\Sigma \Rightarrow \varphi \approx \psi$ is *valid in* $\mathbf{A}$ or "holds in $\mathbf{A}$", written $\Sigma \models_{\mathbf{A}} \varphi \approx \psi$, if for every homomorphism $h\colon \mathbf{Tm}_{\mathcal{L}} \to \mathbf{A}$, $h(\varphi') = h(\psi')$ for all $\varphi' \approx \psi' \in \Sigma$ implies $h(\varphi) = h(\psi)$.

Example 4. The quasiequation of Example 3 is not valid in $\mathbf{S_4^{\to e}}$ since the homomorphism $h\colon \mathbf{Tm}_{\mathcal{L}} \to \mathbf{S_4^{\to e}}$ with $h(x) = -1$ and $h(y) = 1$ satisfies $h(x) = h(y \to x)$ and $h(x \to \mathrm{e}) = h(y)$, but not $h(x) = h(\mathrm{e})$.

We also need the well-known fact (see, e.g., [4]) that taking homomorphic images and subalgebras preserves equations and quasiequations, respectively.

Lemma 1. *Let* $\mathbf{A}$ *be an algebra and* $\Sigma \cup \{\varphi \approx \psi\}$ *a finite set of equations. Then*

(a) $\models_{\mathbf{A}} \varphi \approx \psi$ *implies* $\models_{\mathbf{B}} \varphi \approx \psi$ *for all* $\mathbf{B} \in \mathbb{H}(\mathbf{A})$.
(b) $\Sigma \models_{\mathbf{A}} \varphi \approx \psi$ *implies* $\Sigma \models_{\mathbf{B}} \varphi \approx \psi$ *for all* $\mathbf{B} \leq \mathbf{A}$.

For a nonnegative integer m, let $\mathbf{F_A}(m)$ denote the *free algebra with* m *generators of the* $\mathcal{L}$-*algebra* $\mathbf{A}$, i.e., the algebra of equivalence classes $[\varphi]$ of $\mathcal{L}$-terms φ containing at most m variables $x_1, \dots, x_m$ such that two terms φ and ψ belong to the same class if and only if $\models_{\mathbf{A}} \varphi \approx \psi$. The free algebra of the algebra $\mathbf{A}$ has the same language as $\mathbf{A}$ and for $\mathcal{L}$-terms $\varphi_1, \dots, \varphi_n$ and the operation f with $\mathrm{ar}(f) = n$ we have $f^{\mathbf{F_A}(m)}([\varphi_1], \dots, [\varphi_n]) = [f^{\mathbf{A}}(\varphi_1, \dots, \varphi_n)]$.

Lemma 2 ([21], [4]). *Let* $\mathbf{A}$ *be a finite* $\mathcal{L}$-*algebra and* $\Sigma \cup \{\varphi \approx \psi\}$ *a finite set of* $\mathcal{L}$-*equations. Then*

(a) $\mathbf{F_A}(m)$ *is finite for all* $m \in \mathbb{N}$.
(b) $\models_{\mathbf{F_A}(|\mathrm{A}|)} \varphi \approx \psi$ *if and only if* $\models_{\mathbf{A}} \varphi \approx \psi$.
(c) $\Sigma \models_{\mathbf{F_A}(|\mathrm{A}|)} \varphi \approx \psi$ *if and only if* $\Sigma \models_{\mathbf{F_A}(\mathrm{k})} \varphi \approx \psi$, $|\mathrm{A}| \leq k \in \mathbb{N}$.

Intuitively an $\mathcal{L}$-quasiequation $\Sigma \Rightarrow \varphi \approx \psi$ is admissible in an $\mathcal{L}$-algebra $\mathbf{A}$, if every substitution (i.e., every homomorphism from the term algebra to the term algebra), that makes every equation of Σ hold in $\mathbf{A}$, also makes $\varphi \approx \psi$ hold in $\mathbf{A}$. More formally, an $\mathcal{L}$-quasiequation $\Sigma \Rightarrow \varphi \approx \psi$ is called *admissible in* $\mathbf{A}$, if for every homomorphism $\sigma\colon \mathbf{Tm}_{\mathcal{L}} \to \mathbf{Tm}_{\mathcal{L}}$:

$$\models_{\mathbf{A}} \sigma(\varphi') \approx \sigma(\psi') \text{ for all } \varphi' \approx \psi' \in \Sigma \quad \text{implies} \quad \models_{\mathbf{A}} \sigma(\varphi) \approx \sigma(\psi).$$

Quasiequations admissible in the n-element algebra $\mathbf{A}$ are, equivalently, quasiequations valid in $\mathbf{F_A}(n)$.

Lemma 3 ([19]). *$\Sigma \Rightarrow \varphi \approx \psi$ is admissible in $\mathbf{A}$ iff $\Sigma \models_{\mathbf{F_A}(|\mathrm{A}|)} \varphi \approx \psi$.*

If a quasiequation $\Sigma \Rightarrow \varphi \approx \psi$ is valid in an algebra $\mathbf{A}$, then it is also admissible in $\mathbf{A}$. However, the other direction is not true in general. We say that $\mathbf{A}$ is *structurally complete*, if admissibility and validity coincide for $\mathbf{A}$, i.e., $\Sigma \Rightarrow \varphi \approx \psi$ is admissible in $\mathbf{A}$ if and only if $\Sigma \models_{\mathbf{A}} \varphi \approx \psi$.

Example 5. Consider the two-valued Boolean algebra $\mathbf{2} = \langle \{0,1\}, \wedge, \vee, \neg, 1, 0\rangle$. Suppose that a $\{\wedge, \vee, \neg, 1, 0\}$-quasiequation $\Sigma \Rightarrow \varphi \approx \psi$ is not valid in $\mathbf{2}$, i.e., there exists a homomorphism $h\colon \mathbf{Tm}_{\mathcal{L}} \to \mathbf{2}$ such that $h(\varphi') = h(\psi')$ for all $\varphi' \approx \psi' \in \Sigma$ and $h(\varphi) \neq h(\psi)$. Define the homomorphism $\sigma\colon \mathbf{Tm}_{\mathcal{L}} \to \mathbf{Tm}_{\mathcal{L}}$ by sending each variable x to 1, if $h(x) = 1$ and to 0, if $h(x) = 0$. It follows immediately that $\models_{\mathbf{2}} \sigma(\varphi') \approx \sigma(\psi')$ for all $\varphi' \approx \psi' \in \Sigma$, but $\not\models_{\mathbf{2}} \sigma(\varphi) \approx \sigma(\psi)$. So $\Sigma \Rightarrow \varphi \approx \psi$ is not admissible in $\mathbf{2}$, hence $\mathbf{2}$ is structurally complete.

3 A Procedure to Check Admissibility

To check if a given quasiequation $\Sigma \Rightarrow \varphi \approx \psi$ is admissible in a finite algebra $\mathbf{A}$, it suffices by Lemma 3 to check whether the quasiequation is valid in the free algebra $\mathbf{F_A}(|\mathrm{A}|)$, which we know is always finite. The validity of quasiequations in finite algebras is well studied and decidable (see, e.g., [1, 12, 24]). However, even free algebras on a small number of generators can be very large. E.g., the free algebra $\mathbf{F}_{\mathbf{S_4^{\to e}}}(2)$ has 453 elements, where $\mathbf{S_4^{\to e}}$ is the algebra of Example 1. We therefore seek smaller algebras $\mathbf{B}$ such that, as for $\mathbf{F_A}(|\mathrm{A}|)$:

$$\Sigma \Rightarrow \varphi \approx \psi \text{ is admissible in } \mathbf{A} \quad \iff \quad \Sigma \models_{\mathbf{B}} \varphi \approx \psi.$$

Proposition 1. *Let $\mathbf{A}, \mathbf{B}$ be $\mathcal{L}$-algebras such that $\mathbf{B}$ is a subalgebra of $\mathbf{F_A}(|\mathrm{A}|)$ and $\mathbf{A}$ is a homomorphic image of $\mathbf{B}$. Then $\Sigma \Rightarrow \varphi \approx \psi$ is admissible in $\mathbf{A}$ iff $\Sigma \models_{\mathbf{B}} \varphi \approx \psi$.*

Proof. Let $\Sigma \Rightarrow \varphi \approx \psi$ be admissible in $\mathbf{A}$. So $\Sigma \models_{\mathbf{F_A}(|\mathrm{A}|)} \varphi \approx \psi$ by Lemma 3 and then $\Sigma \models_{\mathbf{B}} \varphi \approx \psi$ by Lemma 1. For the other direction suppose that $\Sigma \models_{\mathbf{B}} \varphi \approx \psi$ and $\models_{\mathbf{A}} \sigma(\varphi') \approx \sigma(\psi')$ for all $\varphi' \approx \psi' \in \Sigma$. Then $\models_{\mathbf{F_A}(\mathrm{n})} \sigma(\varphi') \approx \sigma(\psi')$ for all $\varphi' \approx \psi' \in \Sigma$ by Lemma 2 and therefore $\models_{\mathbf{B}} \sigma(\varphi') \approx \sigma(\psi')$ for all $\varphi' \approx \psi' \in \Sigma$ by Lemma 1. But then $\models_{\mathbf{B}} \sigma(\varphi) \approx \sigma(\psi)$ and since $\mathbf{A}$ is a homomorphic image of $\mathbf{B}$, $\models_{\mathbf{A}} \sigma(\varphi) \approx \sigma(\psi)$ by Lemma 1. ◻

Note that every subalgebra $\mathbf{B}$ of a subalgebra $\mathbf{C}$ of the free algebra $\mathbf{F_A}(|\mathrm{A}|)$, i.e., $\mathbf{B} \leq \mathbf{C} \leq \mathbf{F_A}(|\mathrm{A}|)$, is a subalgebra of $\mathbf{F_A}(|\mathrm{A}|)$. So since $\mathbf{F_A}(m_1) \leq \mathbf{F_A}(m_2)$ for all $m_1 \leq m_2$ (see, e.g., [4]), we possibly do not need $|\mathrm{A}|$ generators. This suggests the following procedure when $\mathbf{A}$ is finite:

(i) Find the smallest free algebra $\mathbf{F_A}(m)$ such that $\mathbf{A} \in \mathbb{H}(\mathbf{F_A}(m))$.

(ii) Compute subalgebras $\mathbf{B}$ of $\mathbf{F_A}(m)$, increasing in their size, and check for each whether $\mathbf{A} \in \mathbb{H}(\mathbf{B})$.

(iii) Derive a proof system for a smallest $\mathbf{B}$ with the properties of (ii).

Steps (i) and (ii) of the procedure have been implemented using macros implemented for the Algebra Workbench [23]. Step (iii) can be implemented directly making use of a system such as MUltlog/MUltseq [11, 22].

We now give some explanation how to implement the first two steps of the procedure. For a given $\mathbf{A}$, we want to find the smallest free algebra $\mathbf{F_A}(m)$ ($m \le |\mathrm{A}|$) such that $\mathbf{A}$ is a homomorphic image of $\mathbf{F_A}(m)$. The idea is to calculate first $\mathbf{F_A}(0)$ and to check whether $\mathbf{A} \in \mathbb{H}(\mathbf{F_A}(0))$. Stop if this is the case, otherwise calculate $\mathbf{F_A}(1)$ and check whether $\mathbf{A} \in \mathbb{H}(\mathbf{F_A}(1))$ and so on.

Suppose that, given a finite algebra $\mathbf{A} = \langle \{a_1, \dots, a_n\}, f_1, \dots, f_k \rangle$, we want to calculate the elements of $\mathbf{F_A}(m)$. Recall that the elements of the free algebra can be seen as equivalence classes of terms. Therefore we need to know all the terms that are definable using the given generators. To decide whether two terms φ and ψ are the same, i.e., $\models_{\mathbf{A}} \varphi \approx \psi$, we would have to check all the possible homomorphisms $h\colon \mathbf{Tm}_{\mathcal{L}} \to \mathbf{A}$. So we simulate the truth table checking by storing the elements by sequences of elements of $\mathbf{A}$.

We represent the m generators of $\mathbf{F_A}(m)$ by sequences $\langle \pi_i(\bar{a}_1), \dots, \pi_i(\bar{a}_{n^m}) \rangle$ where $\bar{a}_1, \dots, \bar{a}_{n^m}$ are the elements of $\mathbf{A}^m$ and π_i, $i = 1, \dots, m$ the i-th projection-map from $\mathbf{A}^m$ to $\mathbf{A}$ and collect them in a set G. Then we run the function DefinableTerms (see Fig. 1) to get the elements of $\mathbf{F_A}(m)$, stored as sequences of length n^m in the set F.

```
function DefinableTerms(G, {f_1, ..., f_k})
    F ← G
    repeat
        F_0 ← F
        for all f ∈ {f_1, ..., f_k} do
            F ← F ∪ { ⟨f(g_1), ..., f(g_ar(f))⟩ : g_1, ..., g_ar(f) ∈ F }
        end for
    until F_0 == F
    return F
end function
```

Fig. 1. Algorithm to generate all the definable terms, given a set of generators G

Example 6. Suppose that we want to calculate the elements of the free algebra for the algebra $\mathbf{P} = \langle \{a, b, c, d\}, \star \rangle$ defined in Example 2. We certainly need generators for the free algebra since $\mathbf{P}$ has no constants, i.e., no nullary operations. So the first step will be to calculate $\mathbf{F_P}(1)$: our generator is the sequence (a, b, c, d). Running the function DefinableTerms$((a, b, c, d), \{\star\})$ gives us $F = \{(a, b, c, d), (b, a, b, b), (a, b, a, a)\}$. It is easy to see that there cannot be a surjective homomorphism from $\mathbf{F_P}(1)$ to $\mathbf{A}$ since $\mathbf{A}$ has four elements. So we have to calculate the algebra $\mathbf{F_P}(2)$ with generators $(a, a, a, a, b, b, b, b, c, c, c, c, d, d, d, d)$ and $(a, b, c, d, a, b, c, d, a, b, c, d, a, b, c, d)$, which will give us a six element algebra that fulfills the requirement of the homomorphism.

The second step of the procedure requires us to calculate subalgebras of the free algebra, increasing in their size. We could, at least theoretically, check $\mathbf{A} \in \mathbb{H}(\mathbf{B})$ for all $\mathbf{B} \leq \mathbf{F_A}(m)$. But since we are interested in the smallest algebras with this property, we generate the subalgebras by increasing their size and always testing whether they satisfy the property. The principles for the calculation of subalgebras are those of DefinableTerms defined in Fig. 1. Using the generating elements as arguments for the operations, we increase the set of "reached" elements as long as we get new elements.

We first calculate all the one-generated subalgebras of the free algebra $\mathbf{F_A}(m)$, i.e., $\langle\varphi\rangle$ for $\varphi \in \mathrm{F}_{\mathbf{A}}(m)$ and store their sizes $|\langle\varphi\rangle|$. Now we know that the size of the two-generated subalgebra $\langle\varphi_1, \varphi_2\rangle$ of $\mathbf{F_A}(m)$ is at least $\max\{|\langle\varphi_1\rangle|, |\langle\varphi_2\rangle|\}$. Suppose that $k = \min\{|\langle\varphi\rangle| \colon \varphi \in \mathrm{F}_{\mathbf{A}}(m)\}$. If there is more than one $\varphi \in \mathrm{F}_{\mathbf{A}}(m)$ with $|\langle\varphi\rangle| = k$, then we generate all the algebras (increasing the number of generators) $\langle\varphi_1, \ldots, \varphi_r\rangle$ with $\max\{|t_1|, \ldots, |t_r|\} \leq k$, again testing if there exists a surjective homomorphism to $\mathbf{A}$ and storing their sizes. We then proceed similarly for $k' = \min\{|\langle\varphi\rangle| \colon \varphi \in \mathrm{F}_{\mathbf{A}}(m), |\langle\varphi\rangle| > k\}$. As soon as we find an algebra $\mathbf{B}$ with $\mathbf{A} \in \mathbb{H}(\mathbf{B})$ we have an upper-bound for the size of the algebras to test (note that this upper-bound always exists since it cannot exceed the size of the free algebra $\mathbf{F_A}(m)$). However, we then have to continue until we know that every combination of generators will lead to a subalgebra $\mathbf{B}'$ with $\mathrm{B} \leq \mathrm{B}'$.

It is not hard to see that step (i) is sound and terminating since we use the operations of the generating algebra $\mathbf{A}$ to calculating new, finite sequences of elements of $\mathbf{A}$. So there are at most $|\mathrm{A}|^{|\mathrm{A}|^m}$ sequences. The soundness of step (ii) is, similar to the previous step, given by the construction of the subalgebras and the used bounds of the cardinalities of the algebras. The algorithm terminates since there are only finitely many subalgebras of the free algebra.

Example 7. Consider the algebra $\mathbf{S_3^{\to\neg}} = \langle\{-1, 0, 1\}, \to, \neg\rangle$, where $\to$ is defined as in Example 1 and $\neg x = -x$. Note that an equation of the form $\varphi \approx \varphi \to \varphi$ holds in $\mathbf{S_3^{\to\neg}}$ iff φ is a theorem of the $\{\to, \neg\}$-fragment of the logic RM. Now, following our procedure, we obtain:

(i) $\mathbf{S_3^{\to\neg}} \notin \mathbb{H}(\mathbf{F}_{\mathbf{S_3^{\to\neg}}}(1))$, but $\mathbf{S_3^{\to\neg}} \in \mathbb{H}(\mathbf{F}_{\mathbf{S_3^{\to\neg}}}(2))$.
(ii) $\mathbf{F}_{\mathbf{S_3^{\to\neg}}}(2)$ has 264 elements and the smallest subalgebras $\mathbf{B} \leq \mathbf{F}_{\mathbf{S_3^{\to\neg}}}(2)$ with $\mathbf{S_3^{\to\neg}} \in \mathbb{H}(\mathbf{B})$ have 6 elements.

The fact that we first check the smaller subalgebras is useful here: We only had to check the 264 one-generated algebras, 15 two-generated and 3 three-generated algebras rather than all the 5134 possible subalgebras of $\mathbf{F}_{\mathbf{S_3^{\to\neg}}}(2)$.

Example 8. Small changes in the universe or language of an algebra can dramatically change the size and structure of its free algebra. Consider the algebra $\mathbf{S_4^{\to\neg e}} = \langle\{-2, -1, 1, 2\}, \to, \neg, \mathrm{e}\rangle$ where $\to$ and $\neg$ are defined as in Example 7 and $\mathrm{e}=1$. Although this algebra is only slightly different to $\mathbf{S_4^{\to e}}$ and $\mathbf{S_3^{\to\neg}}$ of the Examples 1 and 7, the appropriate free algebra is much smaller and only needs one generator:

(i) $\mathbf{S_4^{\to\neg e}} \notin \mathbb{H}(\mathbf{F}_{\mathbf{S_4^{\to\neg e}}}(0))$, but $\mathbf{S_4^{\to\neg e}} \in \mathbb{H}(\mathbf{F}_{\mathbf{S_4^{\to\neg e}}}(1))$.
(ii) $\mathbf{F}_{\mathbf{S_4^{\to\neg e}}}(1)$ has 18 elements and the smallest subalgebras $\mathbf{B} \leq \mathbf{F}_{\mathbf{S_4^{\to\neg e}}}(1)$ with $\mathbf{S_4^{\to\neg e}} \in \mathbb{H}(\mathbf{B})$ have 6 elements.

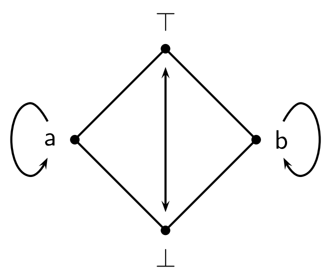
Example 9. In some cases, it is possible to establish structural completeness results for the algebra $\mathbf{A}$ using the described procedure. The smallest $\mathbf{B} \leq \mathbf{F_A}(m)$ with $\mathbf{A} \in \mathbb{H}(\mathbf{B})$ may be an isomorphic copy of $\mathbf{A}$ itself. In particular, known structural completeness results have been confirmed for

$\mathbf{Ł_3^{\rightarrow}} = \langle \{0, \frac{1}{2}, 1\}, \rightarrow_{\mathrm{L}} \rangle$	the 3-element Komori C-algebra
$\mathbf{B_1} = \langle \{0, \frac{1}{2}, 1\}, \min, \max, \neg_{\mathrm{G}} \rangle$	the 3-element Stone algebra
$\mathbf{G_3} = \langle \{0, \frac{1}{2}, 1\}, \min, \max, \rightarrow_{\mathrm{G}} \rangle$	the 3-element positive Gödel algebra
$\mathbf{S_3^{\rightarrow}} = \langle \{-1, 0, 1\}, \rightarrow_{\mathrm{S}} \rangle$	the 3-element implicational Sugihara monoid

where $x \rightarrow_{\mathrm{L}} y = \min(1, 1 - x + y)$, $x \rightarrow_{\mathrm{G}} y$ is y if $x > y$, otherwise 1, $\neg_{\mathrm{G}} x = x \rightarrow_{\mathrm{G}} 0$, and $\rightarrow_{\mathrm{S}}$ is the operation $\rightarrow$ of $\mathbf{S_3^{\rightarrow\neg}}$ from Example 7. A new structural completeness result has also been established for the psuedocomplemented distributive lattice $\mathbf{B_2}$ obtained by adding a top element to the 4-element Boolean algebra.

Example 10. There are even cases where we do not need any generators for the free algebra since the constants alone suffice. Consider the 5-valued Post algebra $\mathbf{P_5} = \langle \{0, 1, 2, 3, 4\}, \wedge, \vee, ', 0, 1 \rangle$ where $\langle \{0, 1, 2, 3, 4\}, \wedge, \vee \rangle$ builds a 5-chain with $0 < 4 < 3 < 2 < 1$ and $0' = 1, 1' = 2, 2' = 3, 3' = 4, 4' = 0$. This algebra builds the algebraic counterpart to the Post logic P_5. Running our procedure we recognize that $\mathbf{P_5}$ is isomorphic to $\mathbf{F_{P_5}}(0)$, i.e., $\mathbf{P_5}$ is also structurally complete.

Example 11. Consider the algebra $\mathbf{D_4} = \langle \{\bot, a, b, \top\}, \wedge, \vee, \neg, \bot, \top \rangle$, called the 4-element De Morgan algebra, consisting of a distributive bounded lattice with an involutive negation defined as shown below and also its constant-free case called the 4-element De Morgan lattice $\mathbf{D_4^L} = \langle \{\bot, a, b, \top\}, \wedge, \vee, \neg \rangle$.



Following our procedure, we obtain:

(i) $\mathbf{D_4} \notin \mathbb{H}(\mathbf{F_{D_4}}(1))$ and $\mathbf{D_4^L} \notin \mathbb{H}(\mathbf{F_{D_4^L}}(1))$, but $\mathbf{D_4} \in \mathbb{H}(\mathbf{F_{D_4}}(2))$ and $\mathbf{D_4^L} \in \mathbb{H}(\mathbf{F_{D_4^L}}(2))$.

(ii) $\mathbf{F_{D_4}}(2)$ has 168 elements, $\mathbf{F_{D_4^L}}(2)$ has 166 elements and the smallest subalgebras of the free algebras, for which $\mathbf{D_4}$ and $\mathbf{D_4^L}$ are homomorphic images, have 10 and 8 elements, respectively.

Example 12. Similar results were also obtained in [19] for Kleene algebras and lattices generated by the 3-element chains $\mathbf{C_3} = \langle \{\top, a, \bot\}, \wedge, \vee, \neg, \bot, \top \rangle$ and $\mathbf{C_3^L} = \langle \{\top, a, \bot\}, \wedge, \vee, \neg \rangle$ where $\neg$ swaps $\bot$ and $\top$ and leaves a fixed. In both cases the smallest subalgebra of the free algebra, for which $\mathbf{C_3}$ and $\mathbf{C_3^L}$ are homomorphic images, is a 4-element chain.

Example 13. Consider the 3-valued Łukasiewicz algebra $\mathbf{Ł_3} = \langle \{0, \frac{1}{2}, 1\}, \rightarrow, \neg \rangle$ with $x \rightarrow y = \min(1, 1 - x + y)$ and $\neg x = 1 - x$. Following our procedure:

(i) $\mathbf{Ł_3} \notin \mathbb{H}(\mathbf{F_{Ł_3}}(0))$, but $\mathbf{Ł_3} \in \mathbb{H}(\mathbf{F_{Ł_3}}(1))$.

$\mathbf{A}$	$\lvert\mathbf{A}\rvert$	Quasivariety $\mathbb{Q}(\mathbf{A})$	$\lvert$Free algebra$\rvert$	$\lvert$Output algebra$\rvert$
$\mathbf{Ł_3}$	3	algebras for $Ł_3$ (Ex. 13)	$\lvert\mathbf{F_A}(1)\rvert = 12$	6
$\mathbf{B_1}$	3	Stone algebras (Ex. 9)	$\lvert\mathbf{F_A}(1)\rvert = 6$	3
$\mathbf{C_3}$	3	Kleene algebras (Ex. 12)	$\lvert\mathbf{F_A}(1)\rvert = 6$	4
$\mathbf{Ł_3^{\to}}$	3	algebras for $Ł_3^{\to}$ (Ex. 9)	$\lvert\mathbf{F_A}(2)\rvert = 40$	3
$\mathbf{C_3^L}$	3	Kleene lattices (Ex. 12)	$\lvert\mathbf{F_A}(2)\rvert = 82$	4
$\mathbf{S_3^{\to\neg}}$	3	algebras for $\mathrm{RM}^{\to\neg}$ (Ex. 7)	$\lvert\mathbf{F_A}(2)\rvert = 264$	6
$\mathbf{S_3^{\to}}$	3	algebras for $\mathrm{RM}^{\to}$ (Ex. 9)	$\lvert\mathbf{F_A}(2)\rvert = 60$	3
$\mathbf{G_3}$	3	algebras for G_3 (Ex. 9)	$\lvert\mathbf{F_A}(2)\rvert = 18$	3
$\mathbf{D_4^L}$	4	De Morgan lattices (Ex. 11)	$\lvert\mathbf{F_A}(2)\rvert = 166$	8
$\mathbf{D_4}$	4	De Morgan algebras (Ex. 11)	$\lvert\mathbf{F_A}(2)\rvert = 168$	10
$\mathbf{P}$	4	$\mathbb{Q}(\mathbf{P})$ (Ex. 2)	$\lvert\mathbf{F_A}(2)\rvert = 6$	6
$\mathbf{S_4^{\to\neg e}}$	4	$\mathbb{Q}(\mathbf{S_4^{\to\neg e}})$ (Ex. 8)	$\lvert\mathbf{F_A}(1)\rvert = 18$	6
$\mathbf{B_2}$	5	$\mathbb{Q}(\mathbf{B_2})$ (Ex. 9)	$\lvert\mathbf{F_A}(1)\rvert = 7$	5
$\mathbf{P_5}$	5	algebras for P_5 (Ex. 10)	$\lvert\mathbf{F_A}(0)\rvert = 5$	5

Table 1. Algebras for checking admissibility

(ii) $\mathbf{F_{Ł_3}}(1)$ has 12 elements and the smallest subalgebras $\mathbf{B} \leq \mathbf{F_{Ł_3}}(1)$ with $\mathbf{Ł_3} \in \mathbb{H}(\mathbf{B})$ have 6 elements.

Note that our procedure does not necessarily find the smallest algebra $\mathbf{B}$ for checking admissibility in $\mathbf{A}$. I.e., there may be an algebra $\mathbf{C}$ with $\mathrm{C} \leq \mathrm{B}$ and

$$\Sigma \Rightarrow \varphi \approx \psi \text{ is admissible in } \mathbf{A} \quad \Longleftrightarrow \quad \Sigma \models_{\mathbf{C}} \varphi \approx \psi.$$

Example 14. Following our procedure for the algebra $\mathbf{P}$ defined in Example 2:

(i) $\mathbf{P} \notin \mathbb{H}(\mathbf{F_P}(1))$, but $\mathbf{P} \in \mathbb{H}(\mathbf{F_P}(2))$.
(ii) $\mathbf{F_P}(2)$ has 6 elements and the smallest subalgebra $\mathbf{B} \leq \mathbf{F_P}(2)$ with $\mathbf{P} \in \mathbb{H}(\mathbf{B})$ is $\mathbf{F_P}(2)$ itself.

However, $\mathbf{P}$ can be embedded into $\mathbf{F_P}(1) \times \mathbf{F_P}(1)$; that is, $\mathbf{P}$ is structurally complete (see, e.g., [5]). But this means that a quasiequation $\Sigma \Rightarrow \varphi \approx \psi$ is admissible in $\mathbf{P}$ iff $\Sigma \Rightarrow \varphi \approx \psi$ is valid in $\mathbf{P}$.

This last issue, but also possibilities of improving the given procedure for checking admissibility (e.g., ruling out symmetric cases of generators when calculating a free algebra) will be the subject of future work. The given examples for our procedure are summarized in Table 1.

References

1. M. Baaz, C. G. Fermüller, and G. Salzer. Automated Deduction for Many-Valued Logics. In *Handbook of Automated Reasoning*, volume II, chapter 20, pages 1355–1402. Elsevier Science B.V., 2001.
2. S. Babenyshev, V. Rybakov, R. A. Schmidt, and D. Tishkovsky. A Tableau Method for Checking Rule Admissibility in S4. In *Proceedings of UNIF 2009*, volume 262 of *ENTCS*, pages 17–32, 2010.

3. W. J. Blok and D. Pigozzi. *Algebraizable Logics*. Number 396 in Memoirs of the American Mathematical Society volume 77. American Mathematical Society, 1989.
4. S. Burris and H. P. Sankappanavar. *A Course in Universal Algebra*, volume 78 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1981.
5. P. Cintula and G. Metcalfe. Structural Completeness in Fuzzy Logics. *Notre Dame Journal of Formal Logic*, 50(2):153–183, 2009.
6. P. Cintula and G. Metcalfe. Admissible Rules in the implication-negation fragment of intuitionistic logic. *Annals of Pure and Applied Logic*, 162(10):162–171, 2010.
7. G. Gentzen. Untersuchungen über das Logische Schliessen. *Math. Zeitschrift*, 39:176–210,405–431, 1935.
8. S. Ghilardi. Unification in Intuitionistic Logic. *Journal of Symbolic Logic*, 64(2):859–880, 1999.
9. S. Ghilardi. Best Solving Modal Equations. *Annals of Pure and Applied Logic*, 102(3):184–198, 2000.
10. S. Ghilardi. A resolution/tableaux algorithm for projective approximations in IPC. *Logic Journal of the IGPL*, 10(3):227–241, 2002.
11. A. J. Gil and G. Salzer. Homepage of MUltseq. `http://www.logic.at/multseq`.
12. R. Hähnle. *Automated Deduction in Multiple-Valued Logics*. OUP, 1993.
13. R. Iemhoff. On the Admissible Rules of Intuitionistic Propositional Logic. *Journal of Symbolic Logic*, 66(1):281–294, 2001.
14. R. Iemhoff and G. Metcalfe. Proof Theory for Admissible Rules. *Annals of Pure and Applied Logic*, 159(1–2):171–186, 2009.
15. E. Jeřábek. Admissible Rules of Modal Logics. *Journal of Logic and Computation*, 15:411–431, 2005.
16. E. Jeřábek. Admissible rules of Łukasiewicz logic. *Journal of Logic and Computation*, 20(2):425–447, 2010.
17. E. Jeřábek. Bases of admissible rules of Łukasiewicz logic. *Journal of Logic and Computation*, 20(6):1149–1163, 2010.
18. P. Lorenzen. *Einführung in die operative Logik und Mathematik*, volume 78 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1955.
19. G. Metcalfe and C. Röthlisberger. Admissibility in De Morgan algebras. *Soft Computing*, to appear.
20. G. Metcalfe and C. Röthlisberger. Unifiability and Admissibility in Finite Algebras. In *Proceedings of CiE 2012*, volume 7318 of *LNCS*, pages 485–495. Springer, 2012.
21. V.V. Rybakov. *Admissibility of Logical Inference Rules*, volume 136 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, Amsterdam, 1997.
22. G. Salzer. Homepage of MUltlog. `http://www.logic.at/multlog`.
23. M. Sprenger. Algebra Workbench. `http://www.algebraworkbench.net`.
24. R. Zach. Proof theory of finite-valued logics. Master's thesis, Technische Universität Wien, 1993.

Scalar Properties of Degree Modification in Karitiana: Evidence for Indeterminate Scales

Luciana Sanchez Mendes

Universidade de São Paulo (Fapesp/Capes/CNPq) and Université Paris 8
sanchez.mendes@gmail.com

Abstract. The aim of this article is to give evidence for the existence of a new parameter for the typology of degree predicates: the indeterminacy of scales. The arguments are based in the analysis of the semantic behavior of the degree modifier *pitat* in Karitiana - a native brazilian language.

1 Introduction

This paper focusses on the scalar properties of a specific verbal construction in Karitiana: degree modification with the adverb *pitat* 'a lot' in sentences like (1)[1].

(1) Taso ∅-na-pytim'adn-∅ pitat.
man 3-DECL-work-NFUT a.lot
'The man worked a lot'

Pitat is a degree modifier that has some particular characteristics. It only combines with atelic verbal predicates, and the modified sentences are adequate in a range of situations related to a high degree in many dimensions, such as duration in time, number of occurrences, intensity, speed, and distance.
The main claims of this paper are (i) *pitat* in Karitiana does not behave like other degree modifiers such as *beaucoup* in French or *a lot* in English; (ii) the data from Karitiana support a degree-scale semantics for degree modiification, and (iii) the traditional typology of gradable predicates based on the closure of the scales and their relation with a standard of comparison can be improved by introducing a further dimension: the distinction between determinate and indeterminate scales.

2 Degree Modification in Karitiana

The aim of this Sect. is to present the distribution of *pitat* 'a lot' in Karitiana. Only atelic verbal predicates can be modified by *pitat*. Sentences (2) and (3) with activity and stative predicates are grammatical. Accomplishment and achievement predicates as in sentences (4) and (5), on the other hand, cannot be modified by *pitat*.

[1] Karitiana is a native language of the Arikén family, Tupi stock, spoken by about 320 people on a demarcated area in the northwest of Brazil (Storto and Vander Velden [2005]).

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see http://ceur-ws.org/, 2012, pp. 142–149.

(2) Milena ∅-na-aka-t i-tarak-t pitat.
Milena 3-DECL-COP-NFUT PART-walk-ABS. a.lot
'Milena walked a lot'

(3) Inacio ∅-na-aka-t i-osedn-∅ pitat.
Inacio 3-DECL-COP-NFUT PART-be.happy-ABS. a.lot
'Inacio was happy a lot'

(4) *Inacio ∅-na-aka-t i-tat-∅ pitat Porto Velho pip.
Inacio 3-DECL-COP-NFUT PART-go-ABS. a.lot Porto Velho to
'Inacio went a lot to Porto Velho city'

(5) *Inacio ∅-na-aka-t i-horop-∅ pitat ep opy ty.
Inacio 3-DECL-COP-NFUT PART-reach-ABS. a.lot tree top OBL
'Inacio reached a lot the top of the tree'

The distribution above is similar to what we find with *a lot* in English (as described in Caudal and Nicolas [2005]).

(6) Yanning walked a lot.

(7) *Yanning ate his pancake a lot.

(Caudal and Nicolas [2005], p.5)

Nevertheless *pitat* has an unexpected characteristic: it can be easily used in a much wider range of interpretations. Thus, sentence (2) can be used to describe the following situations: (i) Milena walked for a long time; (ii) Milena walked a lot of times; (iii) Milena walked in high speed; (iv) Milena walked for a long distance; (v) Milena walked with pleasure. And sentence (3) felicitously describes situations in which: (i) Inacio was happy for a long time; (ii) Inacio was happy a lot of times; (iii) Inacio was very happy. So, an appropriate analysis for *pitat* has to account for these two characteristics: (i) the distribution with only atelic predicates; and (ii) the diversity of possible interpretations. The analysis proposed in the next Sects. attemps to capture these two features.

3 Degree Modification in Karitiana Cannot be Explained by the Mass/Count Distinction in the Verbal Domain

Degree modifiers conveying high degree are usually sensitive to the mass/count distinction in the domain they modify. *Much* and *many* are the classical example (Chierchia [1998]). One could be tempted to suggest that the distribution of *pitat* can be explained by this regularity. The degree adverb *beaucoup* in French, for example, was investigated by Doetjes [2007] in these terms. The author follows Bach [1986] in arguing that the same mass/count distinction we find in the nominal domain can be found in the different types of verbal predicates. Roughly the proposal is that telic verbal phrases (accomplishments and achievements) can be considered countable predicates and atelic (activities and states) can be taken as massive.
Beaucoup has a different behavior depending on the kind of predicate that it modifies. When it is used with a telic predicate, as in example (8), the sentence has an iterative

interpretation, by which many events occurred. In sentences with atelic predicates, as in (9), the sentence can have an iterative interpretation or a degree interpretation (Doetjes [2007]).

(8) Pierre va beaucoup au Louvre.
Pierre goes a.lot to.the Louvre
'Pierre goes to the Louvre a lot (many times)'

(9) Il a plu beaucoup.
It has rained a.lot
'It has rained a lot (many times or intensively)'

According to Doetjes [2007], the iterative interpretation of sentences with *beaucoup* has its origin in the count feature of the predicate. So in sentences like (8) the fact that only iterative interpretation is available is explained by the count nature of the predicate *aller au Louvre* 'to go to the Louvre'. Regarding the sentences with atelic predicates, like *pleuvoir* 'to rain' in (9), when they have a degree interpretation, it is on account of the massive nature of the verbal predicate, but when they have an iterative interpretation, it is because the massive predicate shifted from mass to count. This type–adjustment is the price to pay for the assumption that the iterative interpretation in sentences with degree adverbs has its origin in the count nature of the predicate.

This analysis is not adequate to explain the Karitiana data. Firstly, there are more than only two readings associated with degree modification in Karitiana. All the interpretations available for sentences with *pitat* are equally important and none of them should be explained by an exception rule, like 'shift the predicate' from mass to count. It is not the case that one of the readings is available by one rule and the others by another one. *Beaucoup* has a binary behavior (concerning its interpretations), so it makes sense to capture it by a binary rule (mass/count distinction). *Pitat* has not a binary behavior with atelic verbs, then it should not be explained by a property like the mass/count distinction.

As the next Sect. will show, the iterative interpretation can be considered as being part of the degree modification. I will argue that iterativity in sentences with *pitat* is built on one of the possible scales associated with degree modification, and it is not an operation on the verbal domain that competes with the degree one.

4 Degree Modification in Karitiana and Degree-Scale Structures

The aim of this Sect. is to argue that scalar structures are the proper way to deal with degree modification in atelic constructions in Karitiana. It will be shown that a degree-scale semantics can account for both the distribution and meaning of *pitat*.

Degree modification can be understood as an operation on gradable predicates. Following Kennedy [1999], I will consider gradable predicates as predicates that have a degree argument and a scalar structure[2]. Kennedy and McNally [2005] argue that there are two parameters that are crucial to the typology of degree predicates: the closure of

[2]A scale is a set of degrees ordered along a dimension. It can be understood metaphorically as a ruler formed by degrees ordered in a certain dimension (that can be, for instance, weight, temperature, length).

their scales and their relation to a standard of comparison.
The first parameter divides the scales into open and closed ones. Open scales do not have a minimum or a maximum degree lexically determined. The adjective *high*, for example, has an open scale since it has no lexically defined minimum or maximum degree. On the other hand, closed scales have a well determined minimum and maximum degree. For example, *full* and *empty* are closed scales adjectives. The scales related with these adjectives have a minimum degree, associated to *empty* and a maximum one associated to *full*.
The second important parameter of the typology of degree predicates is described by their relation to the context. Relative degree predicates are dependent on a contextual standard of comparison in order to be interpreted; absolute ones, by contrast, do not have a context dependent standard of comparison. For instance, the adjective *high* is a relative predicate because its denotation in a sentence varies according to the context. On the other hand, an adjective such as *closed* does not have the standard of comparison defined by the context. Kennedy and McNally [2005] claim that there is an relation between the parameters - gradable adjectives that have totally open scales have relative standards, whereas gradable adjectives that have totally or partially closed scales have absolute standards.
Since I am dealing with degree modification of verb phrases, this typology, which is widely used in the studies of gradable adjectives, must be apllied to the verbal domain. Caudal and Nicolas [2005] assume that there is a relation between event structure and scale closure. They apply the distinction open/closed to the verbal domain and claim that telic verbal predicates, since they have a final point given by the *telos*, can be considered as having closed scales. On the other hand, atelic predicates are open scale predicates since they do not have a lexically defined end point (telos).
As it was shown in Sect. 2, *pitat* can be used only with atelic verbal predicates. We may rephrase the restriction by saying that it applies only to open scale predicates. This is precisely the same restriction of the degree modifier *very* in English as described in Kennedy and McNally [2005][3].

(10) Kim was very worried by the diagnosis.

(11) ??Beck is very acquainted with the facts of the case.

So far this is the distribution of *pitat* according to a theory that assumes scalar structures. In what follows, I will develop a proposal to account for how the multiple readings associated to the sentences with pitat are built.
Intuitively, there is a difference between degree modification of the adjectival and in the verbal domain. When one says 'Mary is very beautiful', it is clear that the dimension of the scale involved in the interpretation of the sentence is easily made available by the adjective: the sentence is evaluated relative to a scale of beauty. But when one say in Karitiana '*Taso napytim'adn pitat*' ('The man worked a lot'), the proper scale for the evaluation of the truth conditions of the sentence is not obvious, but must be filled by context.
This intuition can be formally captured by the theory of degree and scalar structures. The scales formed in constructions with adjectives are scales available in the lexicon (cf. Kennedy [1999]). This explains the similarity between the adjectives and the scales related to them. *High* is related to the scale of height, *happy* is associated to the hap-

[3] See Kennedy and McNally [2005] for details.

piness scale, and so on. Activity verbs, in turn, like the verb *to work* do not lexically encode a scale. There is no "workness" scale lexically associated to the verbal predicate. However, this does not mean that verbal constructions of this type cannot be associated to a scale. The proposal submit is that the scales in these cases are contextually constructed rather than given by the lexicon.

I assume, following Dowty [1979], that activities are dynamic predicates involving complex changes, that is a combination of changes in several possible dimensions[4].The variety of the dimensions associated to complex change predicates is responsible for the variable range of scales related to these predicates and therefore for the multiple readings of the sentences in which they appear.

The variety of dimensions can be formally captured by the tools provided in the works on degree-scales by what Kennedy and McNally [2005] called *indeterminacy.* Indeterminacy is the capacity of a predicate to be compatible with scales of various dimensions. The different measurable dimensions of an event denoted by an atelic verbal predicate – as duration in time, number of occurrences, number of participants, intensity, etc. – can be used to fill in the dimensions of the scales.

4.1 Formalization Proposal for Degree Modification with *Pitat*

This Sect. intends to present my proposal of formalization for the degree modifier *pitat.* As stated before the idea I adopt is that scales associated to atelic predicates (activity and states) are not given in the lexicon, but are provided by context. Since *pitat* only modifies atelic predicates, it only operates on contextual scales. Since scales are sets of degrees, this suggests that the degree that this adverb selects is not present in the lexical representation of the predicates it modifies. See below the traditional lexical entry for *to walk* in (12) and compare to *beautiful* in (13).

(12) $[\![\textit{walk}]\!] = \lambda \text{e. walk(e)}$[5]

(13) $[\![\textit{beautiful}]\!] = \lambda \text{d. } \lambda \text{x. beauty(x)} = \text{d}$

However, in order to be modified by *pitat*, the degree argument must be present somehow in verbs like *tarak* 'to walk'. I assume following Caudal and Nicolas [2005] that atelic predicates may have a degree argument, although it is not present in the lexicon. Piñón [2000] claims that there is possible to give a degree argument to verbal predicates in the course of the semantic composition by a degree function. Following his idea I postulate a function **DegP** – in (14) – that takes a simple predicate of events and returns a relation between degrees and events[6]. To capture the indeterminacy of the scale, a measure function μ is used as a variable for dimensions (cf. Krifka [1998], Thomas [2009]).

[4]In fact, I extend the idea of complex change predicates to all activity and stative verbs in Karitiana since they have the same behavior in *pitat*'s constructions.

[5]I assume a neo-davidsonian semantic of events that consider verbs as predicates of events (cf. Parsons [1990]). Furthermore, I follow Kratzer [1996] in the assumption that the external argument is inserted in the syntax.

[6]The crucial difference between the degree function I postulate and the one seggested in Piñón [2000] is that his formula maps events, objects and degrees of accomplishment predicates. The degree function I suggested in (10) can be applied only to atelic verbal predicates and it says nothing to the relation between the objects and

(14) $[\![\text{DegP}]\!] = \lambda P_{\langle s,t\rangle}.\ \lambda d.\ \lambda e.\ P(e)\ \&\ \mu\ (e) = d$

In (10) μ can be replaced by temporal duration, event cardinality, speed, distance or intensity[7]. The DegP function has a double role. Besides adding a degree argument to the predicate, it also functions as a restriction in the domain of *pitat*. Since *pitat* can occur only in sentences with predicates of open scales, I assume that DegP is a function that can be applied only to open scale predicates. I propose the following lexical entry for *pitat*:

(15) $[\![\textit{pitat}]\!] = \lambda G_{\langle d,\langle s,t\rangle\rangle}.\ \lambda e.\ \exists d.\ [\ d \geq N\ \&\ G\ (d)\ (e)\ ]$
where: N = normal degree of the scale

So the formalization proposed here exploits the idea in Caudal and Nicolas [2005] that the degree modifiers can restructure or introduce scales during the derivation combining lexical, syntatic and semantic information.

4.2 Some Consequences of the Proposal

In this Sect., some consequences of the proposal presented above are discussed. The first one relies on the iterative versus degree interpretations (as discussed in Sect. 3). Returning to Karitiana data, with the assumption of indeterminacy of scales it is possible to explain the iterative reading as part of the degree modification, without postulating a type-shifting rule on the predicate as suggested by Doetjes [2007]. Thus unlike Doetjes [2007] iterativity is considered a subtype of degree modification, and not an operation that competes with it. In fact, in the proposal assumed in that there is no degree interpretation. There is a degree modification and there are iterative and intensity interpretations that are generated by the degree operation.

The second consequence of the idea that constructions involving gradable adjectives have scales whose dimension is lexically specified; while constructions involving verbs have contextual ones is that degree modification in the former constructions is less complex than in the latter. The application of a degree adverb like *very* to an adjective like *beautiful* is just an operation of modification on the lexicalized scale of beauty. The degree modification of a verb, on the other hand, involves a few more steps. For something like *to run a lot* be interpreted, it is necessary that the appropriate scale of the verb is formed and only then the construction can be evaluated. In this very distinction lies the difference between English *very* and *a lot*. Insofar as *very* is a modifier that can be applied only to lexical scales, *a lot* is an adverb that forces the predicate to have a degree argument in order to be used. The assumption that degree modification in the adjectival domain is less complex than modification in the verbal domain is supported by other works in the degree-scale literature. Bochnak [2010] analyzes the degree modifier *half* and posits some semantic functions that are necessary to the modification of verbs and are not used to modify adjectives[8].

The last consequence that will be discussed is the importance of indeterminacy to the

---

the events or degrees. Its purpose is to give a degree argument and a variable of scale to the predicate.

[7] In the way the formula is given it is over-generating because all verbs can be associated to any dimension. The next step of the investigation will be to find lexical constraints to fix this, since, of course, not all verbs have distance and high speed readings, for instance.

[8] See Bochnak [2010] for details.

typology of degree predicates. Starting from the idea that constructions involving gradable adjectives such as *beautiful* have scales available in the lexicon, while constructions involving verbs like the ones I am dealing with have scales built in the context, the first issue that arises is a problem in the description of the distribution of *pitat*. Previously I claimed that *pitat* selects open scale predicates. But, in fact, the predicates that are modified by *pitat* do not have a scalar structure before the modification. So it is necessary to slightly reformulate that claim. In fact, *pitat* selects predicates that can have an indeterminate scale and this is the relevant property for both its distribution and meaning. The indeterminacy of scales is as important to the scales' typology as their closure and their relation to the standard of comparison. This is the main theoretical contribution of this work: besides the well-know parameters of open/closed and relative/absolute predicates, the indeterminate/determinate property is also important.
In fact, since there is a one-to-one correspondence between the two parameters described in Kennedy and Mcnally [2005], there is no reason to consider them as two different parameters. They can be treated as a single parameter described in two different ways. The relative vs. absolute distinction is a characteristic of degree predicates whereas the open vs. closed distinction is a characteristic of their scalar structure. On the ther hand, indeterminacy is an independent parameter since it allows for crossed combinations with the other(s).
The table resumes the proposal with examples (adjectives and verbs)[9]:

	(In)determinacy	**Adjctive**	**Verb**
Open Scale	Determinate	high	to melt (atelic reading)
Relative Predicate	Indeterminate	big	to run
Closed Scale Absolute Predicate	Determinate	full	to melt (telic reading)

The proposal is that if a gradable predicate is absolute and it has a closed scale, it necessarily has a determinate scalar strucutre, there is, its scale is given by lexicon. If a predicate is relative, it can have a determinate or an indeterminate scale. The adjective *high* is the classical example os an open scale adjective with determinate scale (scale of height). The verbs trated in this paper as *to run*, on the other hand, are open scale predicates with indeterminate scales (scales of speed, distance, iterativity, intensity, etc.).

5 Conclusions

This paper argues that the number of parameters that classify degree predicates must be enlarged in order to include also the determinate/indeterminate distinction. Thus the degree modification with *pitat* in Karitiana can be properly analized in a degree-scale semantics. The analysis proposed assumes that atelic predicates are verbs of complex changes that are composed by changes in several possible dimensions (speed, distance, iterativity, intensity) whichh are made available by the scales' inderteminacy. The proposal has some consequences. First, iterative interpretations in sentences with *pitat* are properly described by a degree modification operation, without the postulation of a type-shifting rule on the predicates. Secondly, the predictions that degree modification in the verbal domain is in a certain way more complex than in the adjectival domain is in accordance with other works in the same field. And finally, the

[9]For degree achievements ambiguity see Hay et al. [1999]

investigation of degree modification with *pitat* in Karitiana helped to reach a new theoretical claim: the determinacy of scales is as important to their typology as their closure and their dependence from the contextual standart.

References

1986. Bach, E.: The Algebra of Events. Linguistics and Philosophy **9** (1986) 5–16
2010. Bochnak, M. R.: Two sources of scalarity within the verb phrase. Paper presented at Workshop on Subatomic Semantics of Event Predicates. (2010)
2005. Caudal, P., Nicolas, D.: "Types of degrees and types of event structures". In C. Maienborn and A. Wöllstein, eds., Event arguments: foundations and arguments. Max Niemeyer Verlag, Tübingen. (2005) 277–299
1998. Chierchia, G.: Plurality of mass nouns and the notion of semantic parameter. In S. Rothstein, ed., Events and Grammar. Kluwer, Dordrecht. (1998) 53–103
2007. Doetjes, J.: Adverbs and quantification: degree versus frequency. Lingua **117** (2007) 685–720
1979. Dowty, D.R.: Word Meaning and Montague Grammar. Reidel, Dordrecht. (1979)
1999. Hay, J., Kennedy, C., Levin, B.: Scalar Structure Underlies Telicity in Degree Achievements, SALT **9** (1999) 127–144.
1999. Kennedy, C.: Projecting the adjective: The syntax and semantics of gradability and comparison. NewYork: Garland. [Santa Cruz: University of California, Santa Cruz dissertation, 1997.] (1999)
2005. Kennedy, C., McNally, L.: Scale Structure, Degree Modification, and the Semantic Typology of Gradable Predicates. Language **81** (2005) 345–381
1996. Kratzer, A.: "Severing the External Argument from its Verb". In Rooryck, J. and L. Zaring. (eds.) Phrase Structure and the Lexicon. Dordrecht, Kluwer. (1996) 109–137
1998. Krifka, M.: "The Origins of telicity". In Susan Rothstein (ed.) Events and Grammar. Dordrecht: Kluwer. (1998) 197–236
1990. Parsons, T.: Events in the Semantics of English: A Sttudy in Subatomic Semantics. MIT Press: Cambridge, MA. (1990)
2000. Piñón, C.: Happening gradually. In Conathan, L. J.; Good, J.; Kavitskaya, D. Wulf, A. B. and Yu, A. C. L. (eds.) Proceedings of the Twenty-Sixth Annual Meeting of the Berkeley Linguistics Society. (2000) 445-456
2005. Storto, L., Vander Velden, F.F.: Karitiana. Povos Indígenas do Brasil. http://www.socioambiental.org/pib/epi/karitiana/karitiana.shtm (2005)
2009. Thomas, G.: Comparison across domains in Mbya. In Proceedings of WSCLA 14 Vancouver: Department of Linguistics, University of British Columbia. (2009)

PCFG Extraction and Pre-typed Sentence Analysis

Noémie-Fleur Sandillon-Rezer
nfsr@labri.fr

LaBRI, CNRS
351 cours de la Libération, 33405 Talence

Abstract. We explain how we extracted a PCFG (probabilistic context-free grammar) from the Paris VII treebank. First we transform the syntactic trees of the corpus in derivation trees. The transformation is done with a generalized tree transducer, a variation from the usual top-down tree transducers, and gives as result some derivation trees for an AB grammar, which is a subset of a Lambek grammar, containing only the left and right elimination rules. We then have to extract a PCFG from the derivation tree. For this, we assume that the derivation trees are representative of the grammar. The extracted grammar is used, through a slightly modified CYK algorithm that takes in account the probabilities, for sentences analysis. It enables us to know if a sentence is include in the language described by the grammar.

1 Introduction

This article describes a method to extract a PCFG from the Paris VII treebank [1]. The first step is the transformation of the syntactic trees of the treebank into derivation trees representative of an AB grammar [13], which corresponds to the elimination rules of Lambek Calculus, as shown in Fig. 1. We chose an AB grammar because we want our approach to be potentially compatible with some usual learning algorithms, like the one of Buszkowski and Penn [4] or Kanazawa [12]. Once we have the derivation trees, we extracted the PCFG from them, and use it for sentence analysis. This analysis helps us to know how we can improve our grammar and all the processing line used to get it, by analyzing why some correct sentences cannot be parsed or why some incorrect ones are still parsed. In a more long-viewed aim, parsing french sentences can be use for grammatical checking, and with semantic information over a lexicon, the grammar could be used for generate coherent sentences.

$$\frac{A/B \quad B}{A}\ [/E] \quad \frac{B \quad B\backslash A}{A}\ [\backslash E]$$

Fig. 1. Elimination rules of Lambek Calculus. An AB grammar is composed of instantiations of these two rules, where A and B are Lambek types.

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see `http://ceur-ws.org/`, 2012, pp. 150–159.

The Paris VII treebank [1] contains sentences from the newspaper *Le Monde*, analyzed and annotated by the *Laboratoire de linguistique de Paris VII*. The flat shape of trees does not allow the direct application of a usual learning algorithm, so we decided to use a generalized tree transducer. For our work, we use a subpart of the treebank, on a parenthesized form, composed by $12,351$ sentences. Even if the whole treebank was in an **XML** form, the parenthesized form is easier to treat with the transducer. The 504 sentences left aside will be an evaluation treebank that we use as a control group.

Another new treebank, Sequoia [5], which is composed by $3,200$ sentences coming from different horizons, will also be used for experimentation. It is annotated using the same convention as the French Treebank.

This article will firstly overfly the transducer we use to transform syntactic trees into derivation trees, then we will focus on PCFG extraction. In a third part, we will detail the experimental results, obtained by using our PCFG to find the best analysis for a sentence, via the CYK algorithm [19].

2 Generalized Tree Transducer

It exists many way to make a syntactic analysis of a treebank, as we can see with the work of Hockenmaier [8], or Klein and Manning [10], but they were not applicable over the French Treebank or they did not gave simple AB grammar.

The transducer we created is the central point of the grammar extraction process. Indeed, the binarization of syntactic trees parametrize the extracted grammar. We based our works on usual derivation rules of an AB grammar used in computational linguistic and the annotations of the treebank itself [2]. The annotations give two types of information about the trees :

POS-tag: the **P**art-**O**f-**S**peech tags, booked for the pre-terminal nodes, indicates the POS-tag of the daughter node. For example *NC* will be used for Common Noun, *DET* for a determinant, etc.

Phrasal types: the nodes which are not a terminal or a pre-terminal node are annotated with their syntactic categories and sometime the role of the node. A *NP-SUJ* node will correspond to a Noun Phrase used as a subject for the sentence.

For the usual derivation rules, they are instantiation of elimination rules of Lambek calculus (see Fig. 1). We based ourselves on other annotation methods : a *NP* node will have the type np, a sentence type will be s, a preposition phrase taken as an argument will have the type pp, and so on.

A transducer, like defined by TATA [7], is an automaton which read an input and write something on output. It can be applied to trees and transform the shape of them. The transducers have some feature especially important for our work. Indeed, they are non erasing (it ensures us that we do not lost informations during the transduction), linear (the transduction will not change the order of the words in the sentence) and ϵ-free (gives more control over the transduction). The G transducer, developed by Sandillon-Rezer in [17], has additional features that feet better with the global shape of the treebank:

Recursivity: Our transducer can apply a rule to a node, looking only on its label but with an arbitrary arity. It generalize the usual definition of transduction rule, by matching node with an arbitrary number of daughters. The study of each specific case of the rule can transform the recursive rule into a set of ordinary rules.

Parametrization: We allow the rules to have some generic nodes which replace a node from a finite set of nodes. This quantification is equivalent to write each instantiation of the generic node.

Priority system: As our transducer needs to be deterministic, we decided to apply the rules always in a given order. It ensures us to have only one output tree.

The transduction rules have been written from a systematic analysis of the different shapes we could find in the treebank. As an example, a syntactic tree from the treebank and its transduction is shown Fig. 2.

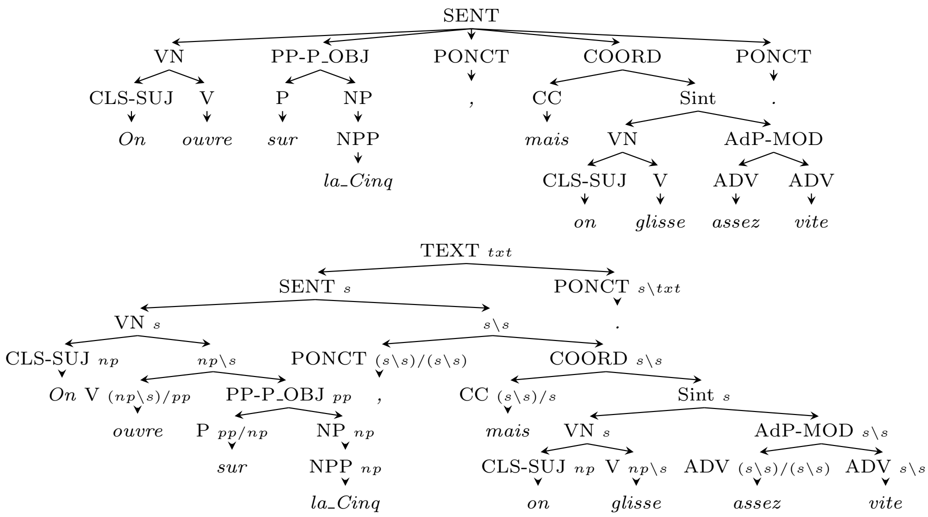


Fig. 2. Syntactic tree and derivation tree for the sentence: *"On ouvre sur la Cinq, mais on glisse assez vite."* (We open on *la Cinq*, but we slip fast enough.).

3 Grammar extraction

Even if the lexicon, extracted from the derivation trees, is representative enough of an AB grammar, and gives a probabilistic distribution of different types for words, it limits the sentence analysis to the sole lexicon of the treebank. This is the reason we decided to extract a PCFG from the derivation trees.

The output trees give both syntactic (we keep the initial labels) and structural information over sentences. We decided to proceed to a preprocessing step, in order for the user to control the extracted information. The only part of information we always keep is the type of nodes. The extracted grammar is a PCFG, the probabilities are computed based on their root node. We remind that our grammar is defined by a tuple

$\{N, T, S, R\}$, where N is the set of non-terminal symbols (internal nodes of trees), T the set of terminal symbols (typed leaves), S the initial symbol[1] and R the set of rules.

The extraction algorithm parses the trees and stores the derivation rules it sees. A rule is composed by a root and one or two daughters, then this is the usual case of a right or left elimination rule ($a \rightarrow a/b \quad b$ or $a \rightarrow b \quad a\backslash b$). Otherwise, it is only a type transmission which appears when a noun phrase is composed by a proper name only; or at the pre-terminal node level when the POS-tag node transmits type to the leaf. The probabilities are computed on a root related group.

The table 1 summarizes the grammars potentially generated. Each one presents useful information: the first one, from the derivation trees without preprocessing step, keeps the syntactic informations given by the treebank. The others are more useful for the application of a sentence analysis algorithm, like CYK (see section 4), on non-typed sentences. The table 2 shows some extracted rules.

Table 1. Extracted grammar. $n_i \in N$ and $t_i \in T$

Shape of trees	**Extracted rules**	**Specification**	**Number of rules**
Raw derivation trees	$n_1 \rightarrow n_2 \quad n_3$ $n_1 \rightarrow n_2$ $n_1 \rightarrow t_1$	Easy normalization in CNF: just need to remove some unary rules.	$63,368$
Removal of unary chains and labels exept POS-tags and words	$n_1 \rightarrow n_2 \quad n_3$ $n_1 \rightarrow t_1$	The grammar is in CNF.	$59,505$
Removal of unary chains and labels. No difference between N and T.	$n_1 \rightarrow n_2 \quad n_3$	The words do not even appear, we only have the skeleton of trees.	$3,494$

4 Sentence analysis

The analysis process can be subdivided in two parts. On the one hand, we have to type the words, while staying as close as possible to standard Lambek derivations. On the other hand, the needed rules must belong to the input grammar.

4.1 Word typing

By gathering the leaves of the derivation trees, we have a lexicon, as we can see Fig. 3. However, a typing system based only on this lexicon reduces the possibility of parsing to the sentences composed by words from the French Treebank. We decided to type words with the Supertagger (see Moot [15, 16]), which enabled us to validate the Supertagger results and to analyze sentences which did not occur in the Paris VII treebank.

The Supertagger is trained with the lexicon extracted from the transduced derivation trees.

[1] $S =$ *TXT:txt* or *txt*, depending of the preprocessing step.

Table 2. Example of rules extracted from various input trees.

Raw derivation trees		
Rule example	*NP:np* $\rightarrow$ *NPP:np*	$1.01e^{-1}$
	NP:np $\rightarrow$ *DET:np/n NC:n*	$2.02e^{-1}$
Removal of unary chains and labels but POS-tags and words		
Rule example	$:(np\backslash s_i)/(np\backslash s_p) \rightarrow VINF:(np\backslash s_i)/(np\backslash s_p)$	$9.53e^{-1}$
	$:(np\backslash s)/(np\backslash s_p) \rightarrow CLR:cl_r \;\; :cl_r\backslash((np\backslash s)/(np\backslash s_p))$	$2.88e^{-2}$
Removal of unary chains and labels.		
Rule example	$np \rightarrow np/n \; n$	$8.02e^{-1}$
	$s \rightarrow np \; np\backslash s$	$3.81e^{-1}$
	$s \rightarrow s \; s\backslash s$	$2.65e^{-1}$
	the sentence has a "sentence modifier" at its end.	
	$s \rightarrow np\backslash s_p \; (np\backslash s_p)\backslash s$	$1.13e^{-3}$
	the type $np\backslash s_p$ *corresponds to a past participle, used as an argument of the whole sentence.*	

```
5968:le:det: - 5437:np/n, 140:(n\n)/n, 79:(s/s)/n, 68:(s\s)/n
```

Fig. 3. Extract of the lexicon. It gives the occurrence in the derivation trees, the POS-tag of the words and the associated types. Here, the determiner "le" occurs 5968 times in the derivation trees, and the most probable type is np/n, the usual one for a determiner at the beginning of a noun phrase. The three other types correspond to noun phrases used as modifiers.

4.2 Typed sentence analysis

We decided to use the CYK [19] algorithm, already tested and considered as a reference, with a probabilistic version for the parsing of sentences. We removed the typing step done initially by CYK with the rules $n_1 \rightarrow t_1$ 1, replaced by the Supertagger work. We use the simplest grammar (of $3,494$ rules) for the analysis. The first test, to assure the correct running of the program, was to re-generate the trees from the transduced sentences with the grammar extracted from the derivation trees. Then, we tested the analysis with sentences typed by the Supertagger, using the grammar extracted from the main treebank (see results section 5.

The derivation trees corresponding to the sentence *"Celui-ci a importé à tout va pour les besoins de la réunification."* ("This one imported without restraint for the need of reunification need.") are shown in Fig. 4. We took the two most probable trees, typed by the Supertagger and analyzed with the grammar extracted from the main treebank. Two types of information are relevant to choose the best trees: we took into account the probability and the complexity of types. However, it is known that the comparison between two trees that do not have the same shape or leaves is complex. The main difference between the two trees is the prepositional phrase attachment; the most probable tree is more representative of the original treebank.

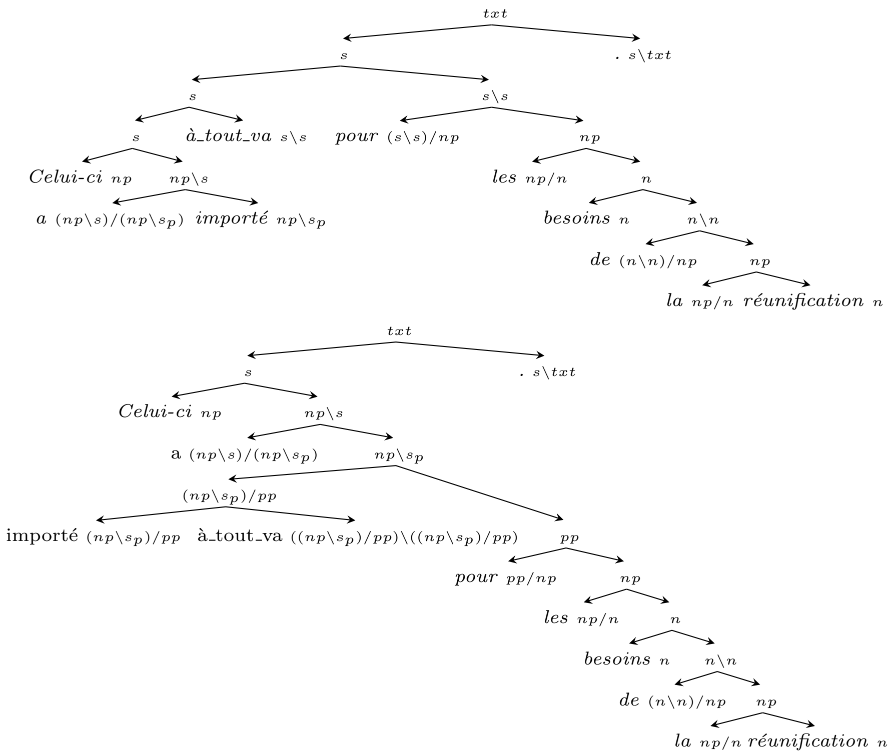


Fig. 4. Probability of the first tree: $5.12e^{-05}$; probability of the second one: $2.29e^{-08}$.

5 Results and evaluation

5.1 *G*-Transducer

From now, the transducer treats at least 88% of the corpora (see Table 3 for details). The lower percentage on the evaluation treebank can be explained by the study of the remaining sentences: they are colloquial and complex. On Sequoia, the better results can be explain by the greater simplicity of sentences.

Table 3. Coverage of the *G*-Transducer

Treebank	Number of sentences	Transduced sentences	Percentage
Main treebank	12,348	11,456	92.6%
Evaluation treebank	504	446	88.5%
Sequoia	3200	3014	94.2%

The use of rules, for the main treebank, is summarized in Table 4. We note that even if many rules are used infrequently, they do not have a real weight in the global use of the rules. Some of the important rules are shown in the Table 5 in the Tregex [14] parenthesized way. We were surprised too by the few occurrences of the rule that treats the determiner at the beginning of a noun phrase, despite the amount of *NP* in the treebank, but there is a rule which treats only the case of a noun phrase composed by a determinant and a noun, called $8,892$ times.

The derivation trees of the three corpora allow us to extract three different grammars, and in addition, lexicon were created, containing words and their formulas. The lexicon covers 96.6% of the words of the Paris VII treebank, i.e. $26,765$ words on $27,589$, and for Sequoia, it covers 95.1%, i.e. $18,350$ word on $19,284$.

Table 4. Summary of the rule usage.

Number of rules	Minimal and maximal occurrence.	Number of applications
1,148	between 1 and 20	$5,818$
473	between 21 and $1,000$	$68,440$
41	between $1,001$ and $10,000$	**125,405**
4	greater than $10,000$	$60,779$

Table 5. Some important rules of the transducer, including the four most important.

input pattern	output pattern	applications
(*NP:* NC PP*)	(*NP:* NC:n PP:n**)	$17,767$
(*NP:* DET tree*)	(*NP:* DET:np/n NP:n*)	$16,232$
(*PP:* P NP*)	(*PP:* P:*/np NP:np*)	$16,037$
(*SENT tree PONCT*)	(*TEXT:txt SENT:s PONCT:s\\ txt*)	$10,819$
(*NP:* tree* (*COORD CC NP*))	(*NP:** (*:* NP:** (*COORD:** CC:(**)/np NP:np*)))	$2,511$
(*SENT:* NP-SUJ VN NP-OBJ*)	(*SENT:* NP-SUJ:np* (*:np* VN:(np*)/np NP-OBJ:np*))	$1,820$

$$\dfrac{NP:* \quad \dfrac{CC:(*\backslash *)/np \quad NP:np}{COORD:*\backslash *}[/E]}{NP:*}[\backslash E]$$

Fig. 5. Rule for the coordination between two *NP*, as shown in Table 5.

5.2 Sentence parsing

The parsing of typed sentences is done with the grammar extracted from the main treebank, which is the most covering one. Each treebank has been divided in two part, the transduced one and the non-transduced one. For the Supertagger, we used a β equal to 0.01: even if the supertagging and the parsing steps are slower, the results are much better than a β of 0.05. The results are gathered in Table 6. We note that non-transduced sentences are nevertheless analyzed, even if the results are less accurate than for the other sentences.

We also tested the precision of the Supertagger (for the whole tests, see [18]): the Supertagger can adjust the number of types given to a word, with the β parameter. It enables to select formula which the confidence of the Supertagger is at least β times the confidence of the first supertag. We summarize the time spent to analyze the fragment of 440 sentences of the evaluation corpus, the effectiveness of the algorithm by modifying β in Table 7. The high number of types is due to the limitations of AB-grammar. When the Supertagger is used for multimodal categorial grammars, the average number of formulas is around 2.4 with a β equal to 0.01 and 4.5 with $\beta = 0.001$; the correctness is better too, with respectively a rate of 98.2% and 98.8%.

Table 6. Results.

Origin of sentences	Number of sentences	**Success Rate**
Main treebank	11,456	90.1%
Evaluation treebank	446	83.6%
Sequoia treebank	3,014	95.5%
Non transduced main treebank	892	62.5%
Non transduced evaluation treebank	98	52.0%
Non transduced Sequoia treebank	198	94.4%

Table 7. Summary of time spent, given the average number of types for a word. The correctness of types is evaluated by the Supertagger.

Average number of types		Correctness	Execution time	Analyzed sentences
1	($\beta = 1$)	76.9%	0.31 sec	30.2%
1.8	($\beta = 0.1$)	87.0%	0.78 sec	65.7%
2.4	($\beta = 0.05$)	88.9%	1.22 sec	72.7%
4.5	($\beta = 0.01$)	91.7%	4.27 sec	83.6%
10.6	($\beta = 0.001$)	93.4%	20.18 sec	96.8%
19.2	($\beta = 0.0001$)	93.8%	55.32 sec	98.0%

6 Conclusion and prospects

In this article, we have briefly introduced the G-transducer principle, that we used to transform syntactic trees into derivation trees of an AB grammar. Then we explained how we extracted a PCFG and used it in the sentence analysis. The experimental results of the CYK algorithm used with typed sentences enabled us to compare the annotation from the transducer and the Supertagger.

However, the work is still ongoing, and opens many horizons. Of course, we want to extend the coverage of the transducer to exceed 95% and simplify the types of words. The main problem is that only complex cases remain, but we should be able to find derivation trees, given that we can analyze a part of them with the CYK algorithm. In order to improve parsing precision, we intend to integrate modern techniques such as those of [3, 20] into our parser. Using the Charniak method [6], we would like to transform our grammar into a highly lexicalized grammar.

Given that AB grammars may seem limiting in the case of a complex language, we wish to transform our transducer into a tree to graph transducer. This way, we would be able to use the whole Lambek calculus.

The *XML* version of the Treebank gives more informations on the words, like the tense of verbs. A major evolution would be to reflect this information into our transducer, even if it implies many transformation for it.

Our work and programs are available on [21], under *GNU General Public Licence.*

References

1. Abeillé, A., Clément, L., Toussenel, F.: Building a treebank for French. Treebanks, Kluwer, Dordrecht (2003).
2. Abeillé, A., Clément, L., Toussenel, F.: Annotation Morpho-syntaxique. http://llf.linguist.jussieu.fr (2003).
3. Auli M. and Lopez A.: Efficient CCG Parsing: A* versus Adaptive Supertagging. In Proceedings of the Association for Computational Linguistics (2011).
4. Buszkowski, W., Penn, G.: Categorial grammars determined from linguistic data by unification. Studia Logica (1990).
5. Candito M. and Seddah D.: Le corpusSequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical TALN'2012 proceedings, Grenoble, France (2012).
6. Charniak E. A maximum-entropy-inspired parser. In Proceedings of the 1st Annual Meeting of the North American Chapter of the ACL (NAACL), Seattle (2000).
7. Comon, H., Dauchet, M., Jacquemard, F., Lugiez, D., Tison, S., Tommasi, M.: Tree automata techniques and applications (1997).
8. Hockenmaier J.: Data and Models for Statistical Parsing with Combinatory Categorial Grammar. PhD thesis, 2003.
9. Hopcroft J.E. and Ullman J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley Publishing Company (1979).
10. Klein D. and Manning C.: Accurate Unlexicalized Parsing. In Proceedings of the Association for Computational Linguistics (2003).
11. Knuth D.E.: The Art of Computer Programming Volume 2: Seminumerical Algorithms (3rd ed.) Addison-Wesley Professional (1997).
12. Kanazawa, M.: Learnable Classes of Categorial Grammars. Center for the Study of Language and Information (1998).
13. Lambek, J.: The Mathematics of Sentence Structure. (1958).
14. Levy R., Andrew G.: Tregex and Tsurgeon: tools for querying and manipulating tree data structures. http://nlp.stanford.edu/software/tregex.shtml (2006).

15. Moot, R.: Automated extraction of type-logical supertags from the Spoken Dutch Corpus. Complexity of Lexical Descriptions and its Relevance to Natural Language Processing: A Supertagging Approach (2010).
16. Moot, R.: Semi-automated Extraction of a Wide-Coverage Type-Logical Grammar For French. Proceedings TALN 2010, Montreal (2010).
17. Sandillon-Rezer, N.F.: Learning categorial grammar with tree transducers. ESSLLI Student Session Proceedings (2011).
18. Sandillon-Rezer, N-F.: *Extraction de PCFG et analyse de phrases pré-typées* Recital 2012, Grenoble (2012).
19. Younger D.H.: Context Free Grammar processing in n^3. (1968).
20. Zang Y. and Clarck S.: Shift-Reduce CCG Parsing. In Proceedings of the Association for Computational Linguistics (2011).
21. Sandillon-Rezer, N.F.: http://www.labri.fr/perso/nfsr/ (2012).

An Interaction Grammar for English Verbs

Shohreh Tabatabayi Seifi

INRIA / Université de Lorraine, BP 239
54506 Vandoeuvre-les-Nancy cedex

Abstract. This paper accounts for the construction of a grammar for English verbs using Interaction Grammars. Interaction Grammar is a grammatical formalism based on the two key notions: polarities and constraint system. A polarity expresses an available resource or a lack of resource and is used to discriminate between saturated and unsaturated syntactic structures. A grammar is viewed as a system of constraints of different kinds: structural, feature and polarity constraints, which should be satisfied by the parse trees of sentences. We have developed a grammar for English verbs in affirmative clauses and finally we evaluated our grammar on the portion of a test suite of sentences, the English TSNLP, with LEOPAR parser which is a parser devoted to Interaction Grammars.

Keywords: Grammatical formalism, interaction grammar, tree description, polarity, unification grammar

1 Introduction

The goal of this work is to construct an Interaction Grammar for English verbs. Interaction Grammar(IG) [1] is a very powerful formalism which has the advantages of both Lexical Functional Grammar (LFG) [5] and Categorial Grammar (CG) [3] and [4]. From LFG, it takes the flexibility with the mechanism of unification to perform syntactic composition. From CG, it takes the resource sensitivity to control syntactic composition with polarities. It is inspired by chemical reactions. In this formalism each lexicalized elementary tree acts as a potentially active element which can participate in a reaction with another tree. Two nodes of different trees can merge if they are able to neutralise each other's active polarities and make stable combination. Whenever all the polarities are neutralised correctly to build a unique tree, the obtained tree would be the parse tree of the sentence.

Guillaume and Perrier [9] investigated the principle underlying IG formalism from more theoretical point of view. In general we can see that in IG polarities are attached to the features where as in CG they are attached to the constituents themselves. The other more essential difference lies in the frameworks of these two formalism: CG are usually formalized in generative deductive framework while IG is formalized in a model–theoretic framework. Pullum and Scholz highlighted the advantages of changing the framework [8].

Currently a very detailed French Grammar with quite high coverage has been developed in IG formalism [9]. This work is an attempt to construct a portion of English grammar using the same formalism.

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see `http://ceur-ws.org/`, 2012, pp. 160–159.

2 Interaction Grammars

In order to get a view of how Interaction Grammars work, the following notions are required.

2.1 Tree Description

A tree description is a tree like structure which uses the notion of underspecification relations to describe a family of trees instead of only one tree [7]. This allows to make an unlimited number of trees from a unique underspecified tree representation. A tree which does not have any underspecified relation is a model. For instance, one tree description with an underspecified dominance link, illustrated in the left side of Fig. 1, can produce three (and more) different models.

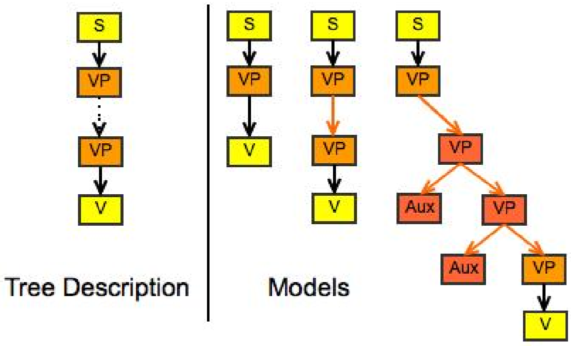


Fig. 1. A description tree with three of its models.

2.2 Feature Structure

IG associates features with the nodes of tree descriptions to put constraints and prevent building ungrammatical parse trees. Feature structure models the internal structure of the language where grammatical properties are represented. Not only do feature structures prevent building ungrammatical sentences, but also they carry valuable linguistics information. Using feature structure to construct a grammar is a salient characteristics of a formalism like LFG but unlike LFG there is no recursive feature in IG and feature structures have only one level.

2.3 Polarities and Saturated Models

Polarities are one of the basic notions in Interaction Grammars. A polarized feature is used to control the process of merging nodes of trees to be combined. Different kinds of polarities are used in IG. There are features with positive or negative polarities meaning an available resource to be consumed or an expected resource to be provided respectively. A neutral feature indicates a feature which is not to be consumed but just to participate in a simple unification process while combining with other nodes. A saturated feature is the result of combination of a positive and a negative feature and can be unified with no more positive or negative feature.

Finally there are virtual features which should be merged with real features during the parsing process including positive, negative, neutral and saturated features. Virtual polarities are used to express required contexts. Positive, negative and virtual features constitute the active polarities because they need to combine with other polarities.
In Fig. 2 features with positive (–>), negative (<–) and virtual (∼) polarities can be seen in the tree descriptions associated with the words ***been*** and ***arranged***. Saturated features indicated with symbol <==> and neutral features indicated with symbol = or == can be seen in the non-completed parse tree in the left side of Fig. 2. The horizontal arrows between nodes indicate linear precedence constraints and comes in two forms: large precedence (- - >)which means there can be several nodes between these two node and immediate precedence (–>) which obstacles locating any other node between these two nodes.

Tree descriptions along with polarized feature structure are called polarized tree descriptions or PTDs. The composition process is defined as a sequence of PTD superpositions controlled by polarities. When two nodes merge, their features will be unified according to the standard process of unification with the extra limitations coming from the rules of combination for polarities. Saturated trees are those completely specified trees which have no active polarities anymore. The goal of parsing is to generate all saturated trees from the set of input PTDs which come from a particular IG grammar.

2.4 Grammar

An IG grammar is defined by a finite set of PTDs which are called the elementary PTDs or EPTDs of the grammar. Each EPTD has one (or more) leaf which is linked with a word of a lexicon and is called an anchored node. This means that the grammar is strictly lexicalized and there is no EPTD without any anchor. In practice there is more than one EPTD for each lexical entry in the grammar with respect to its different syntactic properties.
The parser of an IG grammar has two main roles. First to select the EPTDs of the words in the input sentence from the set of EPTDs inside the grammar then to build all valid minimal models with the use of these EPTDs.

Fig. 2 and Fig. 3 show an example of building a model for the sentence *The interview* ***has been arranged***. The first figure is a snapshot of the intermediate results during parsing process; a non-completed parse tree with unsaturated nodes along with the EPTDs of the rest of the words of the sentence. The second figure shows the result yielded by combining the EPTDs of the remaining words with the non-completed tree: it is one valid model, the parse tree of the input sentence.

3 Building grammars with wide coverage

Tackling with real language problems needs a large scale lexicalized grammar which is hard to build and without using automatic tools it is an overwhelming task. The main reason for that is the huge degree of grammatical and lexical redundancy. For instance in IG, several PTDs may share same subtrees which is due to the grammatical redundancy of syntactic structures. Moreover different lexical entries share the same elementary PTDs owing to the fact that they are in the same syntactic category. These

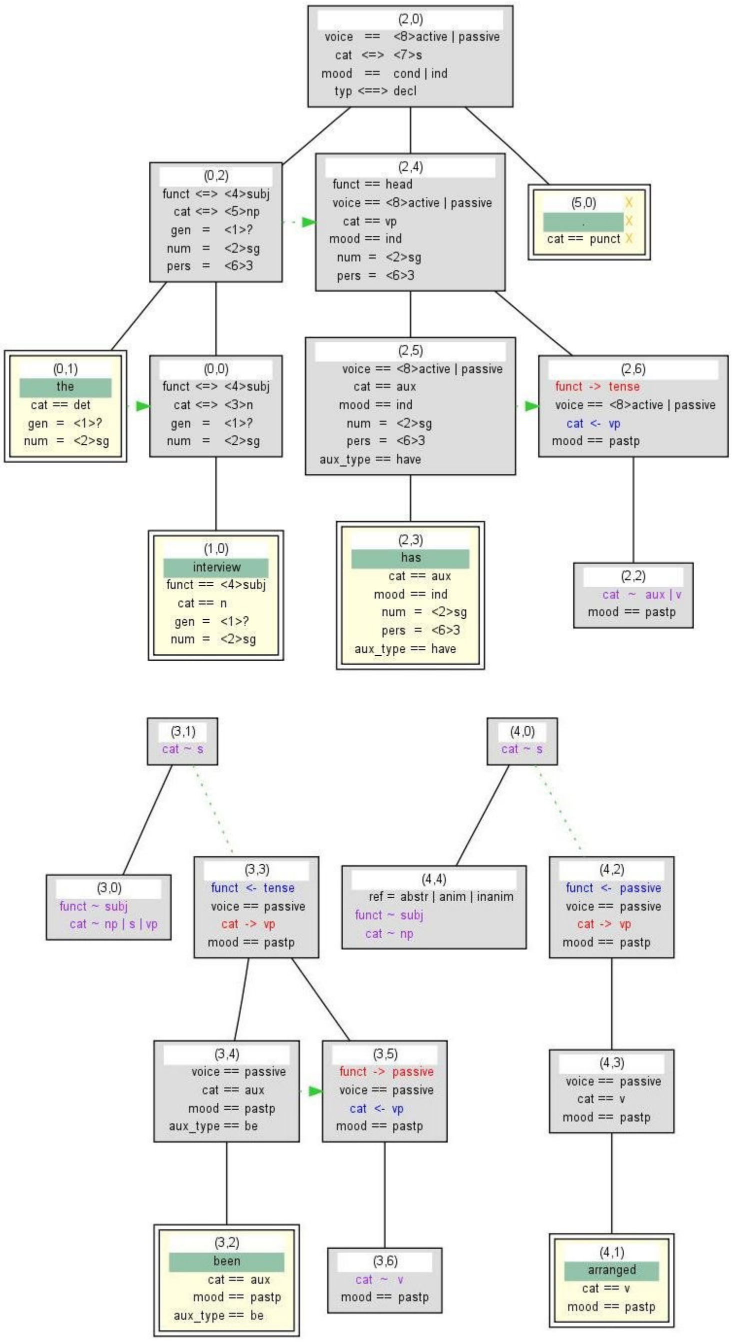


Fig. 2. A non-complete parse tree on the top with two of its EPTDs on the bottom waiting to be used to make a saturated model for the sentence *The interview* ***has been arranged***.

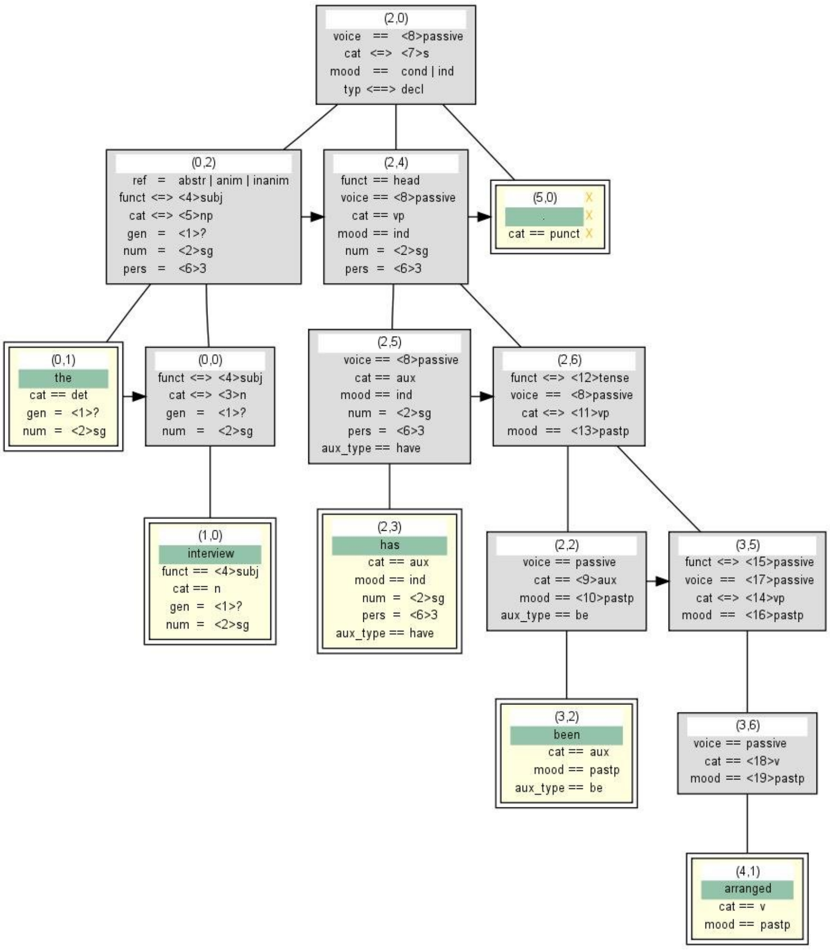


Fig. 3. A parse tree (model) for the sentence *The interview* ***has been arranged***.

redundancies turn even a small modification of the grammar into a big change in large amount of grammar trees. In order to conquer these obstacles eXtensible MetaGrammar (XMG) [6] is used as a facilitating tool to write the IG grammar. XMG is a tool for constructing large scale grammars. The main feature of XMG is to distinguish between source grammar and object grammar. A source grammar is a collection of statements written in a human readable language which produces object grammars which are usable by NLP systems.

3.1 Source grammar and object grammar

The terms source grammar and object grammar here are analogous with the source and object codes in programming languages. In the current task first we wrote the source grammar and then we compiled it with XMG into the object grammar which is encoded in XML. XMG is widely used in construction of grammars in two formalisms: IG and TAG [2].
In source grammar, each small fragment of tree structure is written in an individual class.

By use of class disjunction, conjunction and inheritance, more complex classes can be built. The compilation of terminal classes of the source grammar produces the EPTDs of the object grammar. Each EPTD does not contain any lexical entry yet, but has a detailed description of the word which is going to be nested in its anchor node.

3.2 Linking the grammar with the lexicon

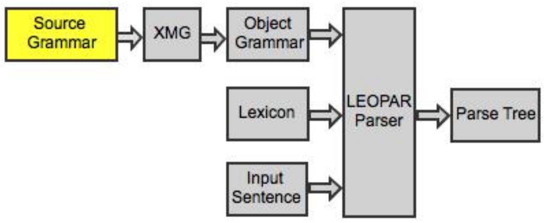


Fig. 4. The process tool chain

The characteristics of the appropriate word to be settled in an anchor node of an EPTD is described in the EPTD interface. An interface is a feature structure describing the words which are able to anchor the EPTD. Every entry in the lexicon is a pair of a word and a feature structure hence a word can anchor an EPTD if its feature structure is unifiable with the EPTD's interface in the exact same means of unification we have in node merging.

The whole process chain is illustrated in Fig. 4 and the highlighted box is the part that has been developed in our work.

4 The Grammar of English Verbs

To build a complete Interaction Grammar for a specific language which can be able to parse any possible sentence, we need a set of EPTDs for wide range of words from different categories like verb, auxiliary, noun, adjective, adverb, prepositions, determiner, pronoun etc. Our main effort was on the construction of a grammar for verbs in affirmative clauses (e.g. different tenses, moods and voices). We have used a small grammar for noun phrase, pronouns, prepositions and adjectives to provide appropriate EPTDs to parse a sentence [11].
The central focus in writing a grammar with IG formalism is to find a way to write classes in a manner that with the use of heredity, conjunction and disjunction all different grammar trees needed in a complete grammar can be obtainable.

4.1 An Sketch of The Major Modules

Five major modules contribute building the EPTDs for verbs. Module **VerbalKernel** is the kernel of all lexical and auxiliary verbs. It implements different possible voices and moods of the verbs and also it provides syntactic functions of a verb phrase (e.g. head, subject, modifier and etc). A subclass in a module is an intermediate tree description that we use along with operators such as heredity, conjunction and disjunction to build the final tree descriptions.

There are seventeen subclasses in module **VerbalKernel**. For instance we model the long distance dependency between a verb and its subject with the use of underspecification relation. All tree descriptions of the verbs which are coming out of this module have a subject node which is an empty node for imperative verbs and a non empty node for all other verbs.

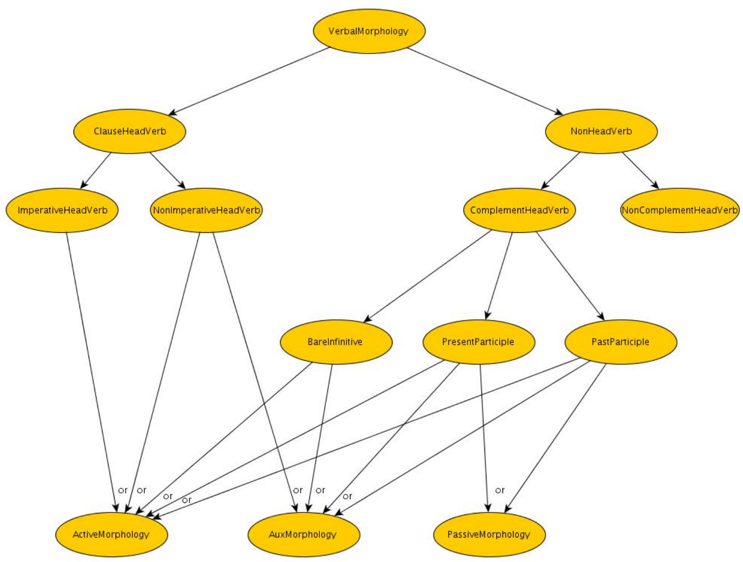


Fig. 5. A schema of relations between some of classes in the module **VerbalKernel**.

Fig. 5 aims to illustrate different relations between some of classes in the module **VerbalKernel**. Some classes are inherited from another one and some are built out of the disjunction of two or more other classes.

Whenever there is no complement nodes in the verb tree, auxiliaries are following the same pattern as non auxiliary verbs therefore class **Auxiliary** is inherited from **VerbalKernel**. The EPTD associated with the auxiliary ***been*** in Fig. 2 is an example of an auxiliary EPTD. Eight subclasses are in class auxiliary implementing all different kinds of auxiliaries. We treated **to** ,the infinite marker, like auxiliaries in the current grammar. The main reason for this approach is the complementary distribution between ***to*** and modal auxiliaries e.g. they never appear together but they appear in the exact same position in the sentence and both force the verb after them to be in the bare infinitive form.

Module **Complements** builds all different kinds of verb complements. It contains 20 different classes and all number, category and function of verb complements are defined in this module.

VerbalKernel and **Complements** will merge together in order to make a new module **Diathesis** with 20 classes which describes different diathesis along with different verb complements e.g. 13 classes for active verbs and 7 classes for passive verbs. For instance to build an EPTD for an active verb with a direct object a tree describing a predicate node with one complement of type noun phrase will merge to a tree associated with active verb and give back the appropriate PTD with active verbs with direct nominal object. Finally module **Verb** with 15 classes will implement trees for different verb families according to their subcategorization frame. For example the family of transitive verbs can anchor the tree associated with active verbs with direct object and the tree associated with passive verbs. So the class for transitive verbs will be the disjunction of two classes of module **Diathesis**. The PTDs which are coming out of module **Verb** and module **Auxiliary** are elementary PTDs and used to attach to the lexicon.

5 Evaluation Results

The English TSNLP (Test Suite for Natural Language Processing) [12] project has aimed to compose a tool for Natural Language Evaluation purposes. It is a database of phrases which carry different linguistic phenomena to be used as a reference for evaluation tasks. These phrases are divided into several categories according to different linguistic phenomena. Each category has both grammatical and ungrammatical examples to provide an infrastructure not only for determining the true positives (success in parsing the grammatical sentences), but also for counting the true negative (failure to parse ungrammatical sentences).

The English TSNLP has 15 different categories and 3 of those contain phenomena exclusively related to verbs which was the subject of this research. However those three categories also contain some other structure which was not included in the current grammar and sentences of those form were put out as well including phrasal verbs, sentences having *there* as their subject and sentences containing relative clauses. (e.g. *That she leave is requested by him.*) We have used LEOPAR parser [10] to construct the models for the input sentences and the result of evaluation of the current grammar can bee seen in table 1.

	Grammatical	Ungrammatical
Total Number	148	832
Parsing Success	115	50
Parsing Failure	33	782
Precision	86.7%	94%

Table 1. Evaluation results of the proposed IG grammar on portion of English TSNLP

5.1 Result Analysis

The major reason of failure in parsing grammatical sentences is that the construction of verbs up to now requires that every verb tree has a subject node which should be filled with a real subject while parsing the sentence. However there are situations in which a verb acts like a noun or an adjective and there is no such a subject node in its grammar tree. (e.g. *He finds the office* ***closed***.) Owing to the fact that we were focusing on construction of the verbs of the main VP of the sentence, other grammatical phenomena which are related to verbs were not properly treated and this failure should not be regarded as a weakness of the framework. In appropriate time all this structures can be constructed in the grammar which is one of the goals in the future works.

The other failure is to mistakenly parse some ungrammatical sentences. One of the main reasons, among other defeats, is that there are some sentences that are not correct because of semantic issues which is not recognizable by our grammar.

6 Conclusions and Future Works

Interaction grammars is a powerful formalism that has advantages of both unification grammars and categorial grammars at the same time. Writing a wide coverage grammar for the English language needs a huge effort and this work can be regarded as the starting point of such a project. Using tools like XMG to accomplish this goal is quite helpful and makes the writing and then the tuning of the grammar a lot more easier than before. Moreover, a high degree of factorization is possible when we separate source grammar and object grammar which leads to more efficiency.

The potential future aims of this project are first to continue to construct a complete grammar for English incorporating all different phenomena in order to parse any grammatical English sentence and second try to cope with the still open problems like coordination in the sentences within the same framework.

On the other hand English TSNLP is a relatively simple set of sentences and is not quite similar to real corpora. Therefore some steps further would be to enrich the grammar some how we would be able to cope with real corpora like newspaper or spoken language corpora where the structure of sentences are more complicated and our grammar should be able to handle several non quite grammatical sentences too .

References

1. Perrier, G.: Interaction grammars. In 18th International Conference on Computational Linguistics, CoLing 2000, Sarrebrucken, pages 600–606. (2000)
2. Crabbé, B.: Grammatical development with XMG. LACL 05. (2005)

3. Retoré, C.: The Logic of Categorial Grammars. ESSLLI 2000, Birmingham.(2000)
4. Steedman, M.: Categorial Grammars. A short encyclopedia entry for MIT Encyclopedia of Cognitive Science. (1999)
5. Bresnan, J.: Lexical-Functional Syntax. ESSLLI 2000, Birmingham.(2000)
6. Duchier, D., Le Roux, J., Parmentier,Y.: The Metagrammar Compiler: an NLP Application with a Multi-paradigm Architecture. In Second International Mozart/Oz Conference, MOZ 2004, Charleroi, Belgium, pages 175–187 (2004)
7. Marcus, M.C., Hindle, D., Fleck. M.M. : D-Theory: Talking about Talking about Trees. In 21st Annual Meeting of the Association for Computational Linguistics, pages 129–136.(1983)
8. Pullum,G.K., Scholz, B.C. : On the Distinction between Model–Theoretic and Generative–Enumerative Syntactic Frameworks. In Logical Aspects of Computational Linguistics, LACL 2001, Le Croisic, France, volume 2099 of Lecture Notes in Computer Science, pages 17–43. Springer Verlag. (2001)
9. Guillaume,B., Perrier, G.: Interaction Grammars. INRIA Research Report 6621: `http://hal.inria.fr/inria-00288376/` (2008)
10. Guillaume,B., Le Roux,J., Marchand, J., Perrier,G., Fort,K., Planul,J.:A Tool–chain for Grammarians. CoLING 08, Manchester. (2008)
11. Planul, J.: Construction d'une Grammaire d'Interaction pour l'anglais. Mémoire, présenté et soutenu publiquement le 27 juin. (2008)
12. Lehmann,S., Oepen, S., Regnier–Prost,S., Netter,K., Lux,V., Klein,J., Falkedal,K., Fouvry,F., Estival, D., Dauphin, E., Compagnion, H., Baur, J., Balkan, L., Arnold,D.: TSNLP– Test Suite for Natural language Processing. In Proceedings of COLING 1996, Kopenhagen. (1996)

Toward a Discourse Structure Account of Speech and Attitude Reports

Antoine Venant[1,2]

[1] Laboratoire Parole et Langage
[2] Ecole Normale Superieure de Cachan

Abstract. This paper addresses the question of propositional attitude reports within Segmented Discourse Representation Theory (SDRT). In line with most SDRT discussions on attitudes reports, we argue that the reported attitude should be segmented in the same way as the rest of the discourse is. We identify several issues that are raised by the segmentation of attitude reports. First, the nature of some relations crossing the boundaries between main and embedded speech remains unclear. Moreover, such constructions are introducing a conflict between SDRT's Right Frontier Constraint (RFC) and well established facts about accessibility from factual to modal contexts. We propose two solutions for adapting discourse structure to overcome these conflicts. The first one introduces a new ingredient in the theory while the second one is more conservative and relies on continuation-style semantics for SDRT.

1 Introduction

From a semantic perspective, attitudes reports require to solve several notorious puzzles. Among these, are a lot of problems triggered by definites: Substitution of directly co-referential expressions is generally not allowed under the scope of an attitude verb and neither does existential generalization(see the shortest spy problem raised by [8]). Closely related to those are effects of attitudes verbs on discourse referents availability. For instance factive epistemic verbs like *'to know'* allow referents introduced under their scope to be then referred from outside their scope, while non factive like *'to believe'* do not. These two issues are related to context which has naturally led to several accounts involving dynamic semantics such as [1, 9].

From the modeling text coherence perspective, we need to understand how reporting someone's propositional attitude interacts with the overall discourse structure. The dynamic framework of Segmented Discourse Representation Theory (SDRT) [3] allows to address both perspectives simultaneously by looking at the interaction between discourse structure and anaphoric phenomena. However there is in SDRT no semantic contribution for attitudes report that is as precise as the ones cited above and formulated within Discourse Representation Theory (DRT) [7]. Since SDRT builds over a lower-level formalism (*D*RT), and enriches it by adding rhetorical relations, one may wonder whether DRT-style accounts could be straightforwardly embedded in SDRT. One condition for this is that SDRT keeps the benefits of the work done in the chosen low-level logic, and uses its ability to handle discourse relation to model a more

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see `http://ceur-ws.org/`, 2012, pp. 170–179.

accurate interface between semantics and pragmatics.[3] We want to address then the question of how does SDRT's treatment of embedded speech acts keeps up with such a consideration.

We attach a particular attention to examples in the spirit of 1 for they involve irruption of the factive context into the modal context at the discourse level. We think that such anaphoric links are not fully modelled by the current analyses of attitude reports in SDRT. Distinguishing between Intentional/Evidential uses of reportative verbs still do not allow them in some intensional cases while DRT based approaches would very likely allow event correference from an embedded DRS to the main DRS.

Example 1. The criminal parked his car somewhere near the airport. So detectives think that afterwards he tried to get into a plane.

After briefly introducing SDRT in section 2, we argue in section 3 for segmentation of reported constructions. Section 4 deals with relations that links a reported speech act to a factual one. It shows that the discursive structure of intensional reports is closed to incoming relations, but still can bear anaphoric links to the context. On this basis it exhibits a family of relations for which RFC makes bad predictions. Section 5 presents two ways of restoring the right accessibility conditions while still benefiting from SDRT specificities.

2 Segmented Discourse Representation Structures

SDRT assumes that to analyze discourse one has to segment into meaningful units that shall be linked to each other by means of discourse relations. Each segment is called an *elementary discourse unit* (*edu*). The level of segmentation is merely the clause level (where a clause can be understood as something containing an event or a state)[4].

Each discourse unit is assigned a label (π_i,...π_n) in the language of SDRSs and a corresponding formula in a given language for representation of atomic clauses (K_1,...,K_n)[5]. These labels will serve as arguments of rhetorical relations, like $narration(\pi_i, \pi_j)$ or $explanation(\pi_i, \pi_j)$. Additional labels are associated with complex structured content made of rhetorical relations and other subordinated labels. Such labels with complex content will be called *complex discourse units* and recursively used as argument of other relations. A SDRS is a triple $\langle A, \mathcal{F}, Last\rangle$ where A is a set of labels (A), $\mathcal{F}$ a function mapping labels to contents (either lower-level language such as DRS or discourse relations in case of complex constituents) and $Last$ the information of the last segment introduced. (See [3]:p.138 for the precise definition).

SDRT makes a structural distinction between coordinating and subordinating relations. The former, like *narration*, confer an equal status to their two arguments. The latter introduce a hierarchy between the related constituents. Such a distinction allows to define the so-called Right Frontier constraint. The Right Frontier is the set of labels $RF = \{\pi \mid \pi \prec^* Last\}$ where $\prec^*$ is the transitive closure of the dominance relation $\prec$ defined by $\pi \prec \alpha$ iff α is a complex consitutent which immediately outscopes π or

[3]This is indeed what the theory aims at doing while extending DRT's definition of accessibility.

[4]How fine-grained segmentation should is still under discussions. The present work is also a contribution at this level since we argue for segmenting attitudes.

[5]This is the *lower-level* language and associated representations.

there is subordinating edge $R(\alpha, \pi)$ in some constituent γ. The Right Frontier Constraint stipulates that labels accessible for discourse continuation are those of the Right Frontier, while the ones accessible for correference have to be DRS-accessible on the right frontier.

For instance, the structure of [*John visited his friend*]$_a$. [*Then he went to the cinema*]$_b$. [*He watched* Pirates of the Caribbean]$_c$ is $narration(\pi_a, elaboration(\pi_b, \pi_c))$ elaboration is a subordinating and narration a coordinating relation, therefore the right frontier is $\{\pi_b, \pi_c\}$ and the discourse could not be felicitously continued by *They talked for a long time* which intends to attach to a.

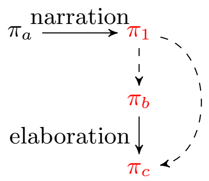


Fig. 1. SDRS example

Figure 1 gives a graphical representation of this example, with the convention that coordinating relation are drawn horrizontally, subordinating one vertically, complex constituents are linked with dashed edges to their subconstituents, and nodes of the right frontier are red.

The truth conditional content of an SDRS, as in DRT, is expressed in terms of context-change potential (*i.e* relation between world-assignment pairs), and is recursively computed using the semantic constraints associated with each rhetorical relation, finally relying on the lower-level logical forms.

In this framework, we now move to the discourse structure of attitude and speech reports.

3 Segmentation and treatment of the matrix clause

There are at least two reasons for capturing the interaction between attitudes or speech reports and discourse structure. First, we need to account for discourse phenomena both inside the reports and across their boundaries. Then the treatment of intentional and evidential uses of attitude reports in the way of [6] also require segmentation.

About the first point, example 2 is not felicitous, because the pronoun *'it'* cannot easily refer to the salmon in the given context. Such a behaviour is predicted by RFC. Therefore, even if the semantics of attitudes generally involves quantification over intensions or contents, and thus erases to some extent the structure of the logical form of the original speech act, the discourse structure of the report is needed anyway to build the logical form of the speech report.

Example 2. #John told me that Marry had a wonderful evening last night. He said [she ate salmon]$_a$ [and then won a dancing competition$_b$] [and that it was beautiful pink.]$_b$

On the other hand, in example 3 the reported speech introduces a *narration* between two events while the non-reported discourse asserts a causal relation (*result*) between the two same events. The contrast introduced by *but* is however coherent, partially because it is supported by the isomorphic structures of the reported speech and the non-reported one. SDRT treatment of contrast as a scalar relation, following [2, 3] provides such an analysis, assuming that the structure of the embedded speech is accessible.

Example 3. John says that he left after Mary did but he left because she did.

About the segmentation of the matrix, we may consider the matrix clause as nothing more than a kind of logical operator[6]. However, that would be inaccurate since the matrix clause can be fairly sophisticated. It generally includes a communication event or a mental state that can be modified by adverbs or prepositional phrases and therefore would be difficult to model as simple logical operator. Since removing the matrix-clause from the discourse representation is not an option neither segmenting attitude reports forces us to deal with this matrix-clause segment.

[6] addresses several issues raised by such a treatment of reported speech. The approach consists in segmenting apart matrix clause and reported speech and in identifying the relation between these elements themselves but also their relations with the surrounding context. It distinguishes between two uses of reportative verbs, namely *evidential* where the embedded content is asserted by the main speaker and *intensional* where the content of the report is not asserted by the main speaker. In evidential uses, the matrix clause is subordinated to the embedded content by a veridical *evidence* relation.[7] In intensional uses, the embedded content is subordinated to the matrix via a relation of attribution which is non-veridical. Such a distinction makes very profitable the separation of the matrix clause and the reported speech, accounting for cases like 4.

Example 4. (1) [The neighbours are gone.]$_a$ [John told me that]$_b$ [they went on vacation in an expensive hotel.]$_c$ [I called it this morning.]$_d$

(2) [The neighbours are gone.]$_a$ [John told me that]$_b$ [they went on vacation in an expensive hotel.]$_c$ [But he lied]$_e$.

As [6] argues, we can see in the first example above that c is asserted by the speaker since d is carrying an anaphoric link to *the hotel* eventhough it has first been introduced under the scope of the attitude[8]. On the contrary, in 4.2, the author disagree with what is reported, and the existence of *the hotel* is not ensured anywhere outside the scope of the attitude. Therefore *The hotel* shall not be referred to later in the discourse. [6] also argues that the compositional semantics of both the reported speech and the matrix clause do not change from an intensional to an evidential report. And the matrix clause can neither be deleted without loss of compositional content in the one nor the other case. But the way the two parts of speech are related can change. Fu rthermore, since

[6] This would still requires to modify the SDRT framework since all logical operators are delegated either to the lower-level logical forms or to the semantic effects of discourse relations.

[7] To be satisfied, veridical relations require their arguments to be true in the model. Non veridical relations do not have this requirement. [3]

[8] At least if we assume that d is not part of what John said here, but in that case that it would be a very odd reading.

the two first sentences are the same in both examples, the decision of choosing the one or the other might only be a matter of context, as such it is essentially information packaging, and in SDRT, this level is kept aside from the logic of information content.

Following this analysis, in the first example, d will be related to c by a veridical relation of *narration*, forcing the evidential reading. So c will be related to b with the veridical $Evidence(c, b)$ and to a with a veridical relation of *explanation*. In the second example however, continuation e is attached to the whole report with a contrast and yields an intensional reading (attaching e to the embedded clause only would entails that John said something incoherent, which is less likely the intended meaning) and b is related to c using the non-veridical $attribution(b, c)$. The two different type of structures are sketched below. (Left column is evidential, right one is intensional. We also give some of the semantics conditions associated with the two relations involved).

$$\mathcal{F}(\pi_b) = \begin{array}{|c|}\hline \phi \\ \hline A(x,\phi) \\ \hline\end{array} \qquad \mathcal{F}'(\pi_b) = \begin{array}{|c|}\hline \phi \\ \hline A(x,\phi) \\ \hline\end{array}$$

$$\mathcal{F}(\pi_{top}) = R_e(\pi_a, \pi_c) \wedge evidence(\pi_c, \pi_b) \quad \mathcal{F}'(\pi'_{top}) = R_i(\pi_a, \pi_b) \wedge attribution(\pi_b, \pi_c)$$

$$\Phi_{evidence(\pi_c,\pi_b)} \Rightarrow K_{\pi_c} \wedge K_{\pi_b} \wedge \phi \sim\hat{}\pi_c \quad \Phi_{attribution(\pi_b,\pi_c)} \Rightarrow K_{\pi_b} \wedge \phi \sim\hat{}\pi_c$$

Where $\sim$ may be understood as an equivalence relation between SDRS contents. How this content and $\sim$ are defined actually remains an open question. Basically *content* could be understood as the context change potential. However, blocking substitution of logically equivalent expressions under the scope of an attitude verb may require some amount of structure being kept in the notion of *content*([1]).

4 Relations across boundaries

As [5] remarks, the picture becomes more complicated when relations comes to cross the boundaries of an embedded speech act such as in 5.

Example 5. [Fred will go to Dax for Christmas]$_a$. [Jane claims that]$_b$ [Afterwards, he will go to Pau]$_c$.

Afterwards introduces a veridical relation of narration. If we invoke the evidential/intensional distinction and assume an evidential reading, this example does not pose any problem since the discourse producer (DP) is thought to assert the content π_c and thus can use a veridical relation for relating it to the context. However, with an intensional reading the speaker does not claim $narration(\pi_a, \pi_c)$ since he does not assert the content of π_c. But he still can commit to Jane committing to such a relation. To solve this problem, [5] set up a new paradigm for discourse analysis that examines reported relations against several sources. For instance, 5 will be analyzed as follows:

The discourse producer is certain of the main eventuallity e_a in a but he does not know anything about the one in c. Jane is attributed to be certain about the main eventuality in c, and, after the source of the narration is identified to being Jane, the picture is completed with the statements of Jane being certain of e_a too, as well as e_a and e_c being in a temporal sequence. Semantically speaking, such examples require some further discussion. First, we cannot always identify a source for a relation. Consider a two level deep embedding as in 6. Asserting $narration_{\text{Fred's wife}}(\pi_a, \pi_d)$ in this case would make us unable to distinguish between 6 and the same without b. With 6 the writer does not commit to Fred's wife committing that he will go to Pau.

Example 6. [Fred will go to Dax for Christmas]$_a$. [Jane told me that]$_b$ [according to his wife,]$_c$ [afterwards, he will go to Pau]$_d$.

Besides, interpreting $narration_J(\pi_a, \pi_c) \wedge attribution(\pi_b, \pi_c)$ requires some precisions that the framework does not provide. To this end, we may switch to a dialogical framework in which each individual would receive its own SDRS. However, reducing reported speech and other voicing effects to dialogue is not what we want to do (especially if we want to be able to account for 6). Another way to provide an interpretation would be to use [10] semantics, $R(a,b) : [\![C(Speaker(b), K_a) \wedge C(Speaker(b), K_b) \wedge C(Speaker(b), \phi_{R(a,b)})]\!]$ where C is a commitment relation and R a veridical relation. But once again the structure is misleading, and with this account, the *narration* producer must be understood as being *Jane*, which is strange since even when being reported, its producer remains the main producer. And we would end up with a fomulla entailing $C(Jane, \phi_{narration(\pi_a,\pi_b)})$ instead of something like $C(W, C(Jane, \phi_{narration(\pi_a,\pi_b)}))$ that would be needed to account for example 6 with this semantics. Our conclusion thus is that the problem originates from the structure which does not model the right scope of *attribution* which should includes the narration relation. This can be done in SDRT by introducing a complex segment for representing the embedded content[9]:

$$\begin{gathered} A = \{\pi_{top}, \pi_a, \pi_b, \gamma, \pi_c\} \\ \mathcal{F}(\pi_{top}) = attribution(\pi_b, \gamma)\ \mathcal{F}(\gamma) = narration(\pi_a, \pi_c) \end{gathered} \tag{1}$$

Equation 1 is actually missing a non-embedded left-veridical coherence relation between π_a and another segment. As it stands our structure semantics does not imply that the main discourse producer claims the content of a. However, more generally there must be some relation (R) introduced by the main producer and that links π_a with the speech act of reporting Jane's claim (at least with an intensional reading). Attribution being subordinating in the intensional case, R cannot be coordinating without the RFC being violated in 5. So it seems that R should be a subordinating like *background*. However cases like 7 are source of problems.

Example 7. (1) [The train arrived 3 hours late.] [then the company announced that] [in consequence, the passengers would be refunded]. [But as a matter of fact, they never were.]
(2) [John had a deadline at midnight yesterday.] [So we all though that afterwards he would go to bed.] [But he did not.]
(3) [Yesterday, John fell three times in a row.] [Mary then told him that] [it was probably because he drank to much.] [He did not believe her.]

All these examples involve an intensional attitude report and in all of them, lexical markers mark either a *narration* or a *result* between the first segment and the matrix clause of the report. Finally, they also all seem to support anaphoric links between the reported content and the first segment. Both *result* and *narration* are thought to be coordinating relations. So even if we use the subordinating *background* between π_a and π_b in 5, we cannot account for these links without violating RFC.

Examples in 7 thus allow us to see that the Intensional/Evidential treatment comes with the side effect of sometimes preventing from linking to the previous discourse. But

[9] Representing SDRS as directed acyclic graphs as it is often done is very confusing in this case, because a graph based representation does not distinguish which complex segment actually hosts such a cross-relation. The graphs for our structure and the problematic one are the same.

they are very specific in the sense that they enforce an explicit rhetorical link from the previous discourse on the two level (the main discourse, and the embedded one) at the same time. The problem may however be more general if these links may as well be implicit.

Example 8. (1) [The factory blew up.]$_a$ [John told me]$_b$ [there were a lot of dangerous chemicals in there.]$_c$
(2) [The factory blew up.]$_a$ [John thinks]$_b$ [there were a lot of dangerous chemicals in there.]$_c$
(3) [The factory blew up.]$_a$ [John thinks]$_b$ [there were a lot of dangerous chemicals in there.]$_c$ [But sam thinks]$_d$ [someone lighted a fire.]$_e$

Examples in 8 intend to illustrate this. The first one does not seem to require an implicit relation between a and c. The possible explanation of the explosion by the presence of chemicals is not a mandatory part of what John said. Actually John might have said that to the writer even before the explosion happened, and the writer is making the link himself from what John previously said. The two other examples on the other hands may carry such implicit links between a and the reported content b: There is at least one plausible reading for the second example involving a coordinating relation between a and b which fits very well an implicit explanation between a and c. The explosion actually made John think of a plausible explanation, which is that they are dangerous chemicals in the usine, and that these chemicals may have cause the explosion. Finally, the last example requires implicit explanation relations to make a better sense of the con trast relation that links b and c. The beliefs of John and Sam are fully compatible, unless what John and Sam respectively said is $explanation(a, c)$ and $explanation(a, e)$, in which case they are not.

All together, this threatens to make SDRT better understanding of anaphoric links in attitude reports only come at the price of some wrong predictions in some intensional cases.

5 Restoring accessibility

We have shown that SDRT damages more standard but essentially correct accounts of anaphoric links going between modal and factual contexts. An account of attitude reports in DRT for instance, would not have this behaviour. Examples like 7 would introduce reference to events in the main DRS from the modal context, which is permitted. We would like such a behaviour, but with SDRT treatment of accessibility still applying inside the reported speech. To this end, we could drop the attribution relation, falling back to a DRT like treatment. The structure of one of our problematic report in SDRT would thus be sketched by $R_{coord}(\pi_a, \pi_{att})$ with $\mathcal{F}(\pi_{att}) = K_{\pi_b} \wedge A(x, \phi) \wedge \phi \sim\hat{}\pi_c$. This structure allows referents in π_c to attach or refer to elements in π_a.[10] This builds on intensional report being " closed" discursive structures. We showed in section 4 that a relation cannot really penetrate the report from the factual context without (a "copy" of) its left argument and itself being embedded under the attitude. Moreover, attachment to the matrix clause and attachment to a complex segment made of both the matrix clause and the report are semantically and dynamically equivalent. This allows us to abstract the complete speech act of reporting under a complex segment

[10] Such an approach actually needs to slightly modify the syntax of the SDRS language

π_{att}. This approach however requires to adapt the language for inferring the relations because the intensional and evidential cases are now asymmetric. One has to state that the content of a segment π_{mat} in the evidential case is equal to a part of the content of the abstracted complex segment π_{att} in the intensional case.

That is why we propose below a more conservative approach that makes use of continuation-syle semantics [4]. Continuation style semantics represents a discourse as a $\lambda-$abstraction of type $[\![\Gamma]\!] = \gamma \to ((\gamma \to l \to t) \to l \to t)$ where γ is the type of input contexts. A discourse thus asks for (i) an input context i of type γ containing the effects of processing the previous discourse; (ii) a continuation o of type $\gamma \to l \to t$ representing the discourse to come and; (iii) a label π, the label of the SDRS representing the whole discourse.

To represent chunks of an SDRS, a language is used where every $n-$ary becomes an $n+1$-ary predicate, the extra argument stands for the label that hosts the predicate: A label π with $\mathcal{F}(\pi) = R(\pi_1, \pi_2)$ will be represented as $\exists \pi_1 \exists \pi_2 \exists \pi R(\pi_1, \pi_2, \pi)$.

We will assume that the context contains a structural representation of the SDRS for the previous discourse such that the following functions may be defined:

(1) $sel_l : \gamma \to l$ that selects a label for attachment.
(2) $\nu : \gamma \to l \to \gamma$ that performs the SDRT update operation on the context [3], defined in terms of SDRT's language for inferring relations. Given a label π, it basically picks up a relation and two other labels π_1, π_2 in the context and add the relation $R(\pi_1, \pi, \pi_2)$ to the context.

Finally, we will use the following version of the binder rule to join a discourse and a sentence:

$$[\![D.S]\!] = \lambda io\pi\, \exists\, \pi_D [\![D]\!] i(\lambda i'\, \exists \pi_S\, [\![S]\!]\, i'\, o\, \pi_S))$$

The main idea, is to refine [6] proposal of a lexical entry for attitude reports using continuation-style semantics to overcome the right-frontier problems. Since evidential and intensional readings only differ by the way the matrix clause and the embedded content are related, one simple solution is to postpone attachment of the matrix clause until the embedded content has been dealt with and all attachment to previous context have been done. But it must be performed before the following discourse is processed in order to still benefit from the intensional/evidential distinction. This might be done by modifying the continuation of the report in such a way that it proceeds to the attachment of the matrix clause before applying the real continuation.

Let us assume an attitude α in a discourse "x α that ϕ" and that syntax delivers us a parse leading to $\alpha(x, \phi)$. We add the lexical entry given in 2 for an attitude verbe α, with A a modal operator corresponding to attitude α.

$$[\![\alpha]\!] = \lambda x \lambda s \lambda io\pi_{matt} \exists\, \phi A(x, \phi, \pi_{matt}) \wedge \exists \pi_s\, \phi \sim \pi_s \wedge s\; i\; [\lambda i' o(\nu(\nu(i', \pi_{matt}), \pi_s))]\; \pi_s \tag{2}$$

Let us now have a look back to [*The train arrived late*]$_a$. [*Then the company anouced that*]$_b$ [*the passengers should thus be refunded*]$_c$.

We assume for a a lexical entry like:

$$\lambda io\pi\, \exists x\, train(x, \pi) \wedge Late(x, \pi) \wedge o\, \nu(i, \pi)$$

.

In this entry the update operation $\nu(i)$ will deliver a context i' containing the structure $\pi_a \mid F(\pi_a) = [x \mid train(x) \wedge late(x)]$, and maybe a relation linking π_a to the previous context. Assuming the lexical entry for *Thus* is

$$[\![thus]\!] = \lambda io\pi s\, s\, i\, (\lambda i' Result(sel_L(i'), \pi, sel_L(i')) \wedge o\, i')$$

We end up with the following entry for the embeded content c:

$$\lambda io\pi \exists y, z \wedge The\_Passengers(y, \pi) \wedge Be\_Refunded(y, \pi) \\ \wedge Result(sel_L(i), \pi, sel_L(i)) \wedge o\, i$$

The lexical entry for *to announce* (our α here) will be given `the_company` as its first argument and the interpretation of c as its second. Which should yield after beta reduction:

$$\lambda io\pi_{matt} \exists\, \phi A(\texttt{The\_company}, \phi, \pi_{matt}) \wedge \exists \pi_s\, \phi \sim \pi_s \\ \wedge \exists y, z \wedge The\_Passengers(y, \pi_s) \wedge Be\_Refunded(y, \pi_s) \\ \wedge Result(sel_L(i), \pi_s, sel_L(i)) \wedge o(\nu(\nu(i, \pi_{matt}), \pi_s))$$

When composing with $[\![a]\!]$, this entry will receive the context i' containing the structure $\pi_a \mid F(\pi_a) = [x \mid train(x) \wedge late(x)]$, unmodified, as input context and thus be able to select π_a as first argument for the result relation without RFC violation. Importantly, successive call to the ν function will perform the intensional/evidential choice and choose a relation to link the report to the preceding discourse before processing the continuation.

6 Conclusion

Segmenting discourse structure cannot be avoided, but as we have shown, the discourse structure of segmented reports is not straightforward. We have thus given a more precise picture of what it should be and why. It remains to give a precise semantics to those reports, and especially to decide what is the content $\hat{}\pi$ of a SDRS and what amount of structure it carries.

References

1. Asher, N.: A typology for attitude verbs and their anaphoric properties. Linguistics and Philosophy 10(2), 125–197 (1987)
2. Asher, N., Hardt, D., Busquets, J., Sabatier, I.P.: Discourse parallelism, ellipsis, and ambiguity. Journal of Semantics 18, 200–1 (2001)
3. Asher, N., Lascarides, A.: Logics of Conversation (Studies in Natural Language Processing). Cambridge University Press (Jun 2005), `http://www.worldcat.org/isbn/0521659515`
4. Asher, N., Pogodalla, S.: SDRT and Continuation Semantics. In: Onada, T., Bekki, D., McCready, E. (eds.) New Frontiers in Artificial Intelligence JSAI-isAI 2010 Workshops, LENLS, JURISIN, AMBN, ISS, Tokyo, Japan, November 18-19, 2010, Revised Selected Papers, LNCS, vol. 6797, pp. 3–15. Springer (2011), `http://hal.inria.fr/inria-00565744`
5. Danlos, L., Rambow, O.: Discourse relations and propositional attitudes. Constraint in Discourse (CID) (2011)

6. Hunter, J., Asher, N., Reese, B., Denis, P.: Evidentiality and intensionality: Two uses of reportative constructions in discourse. In: Proceedings of the Workshop on Constraints in Discourse. pp. 99–106. Maynooth: National University of Ireland (2006)
7. Kamp, H., Reyle, U.: From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory, Studies in Linguistics and Philosophy, vol. 42. Kluwer, Dordrecht (1993)
8. Kaplan, D.: Quantifying in. Synthese 19(1-2), 178–214 (1968)
9. Maier, E.: Presupposing acquaintance: a unified semantics for de dicto, de re and de se belief reports. Linguistics and Philosophy 32(5) (2010), `http://www.springerlink.com/index/10.1007/s10988-010-9065-2`
10. Vieu, L.: On the semantics of discourse relations. Constraint in Discourse (CID) (2011)

Some, Speaker Knowledge, and Subkinds*

Andrew Weir

University of Massachusetts Amherst

Abstract. I provide an analysis of sentences with *some* combining with subkind-denoting NPs, such as *Some plant is growing through the wall of my room.* In such sentences, there is epistemic uncertainty concerning the subkind, but not concerning the actual witness of the claim. I make use of the semantics proposed by [AOMB10b] for the Spanish epistemic indefinite *algún*, combined with the polysemy of common nouns between being individual-denoting and subkind-denoting [Kri95, Kra08], to provide an analysis of such sentences.

Keywords: some, epistemic indefinites, kinds, subkinds

1 Introduction

The English determiner *some*, when combined with a singular noun phrase, seems to carry a meaning of speaker ignorance as to the witness of the existential claim being made [Bec99, Far02, AOMB03].

(1) a. Some ball bearing in this pile is actually made of a different material from all the others. But they all look identical, so I can't tell which one./#Namely, that one there that's a different color from the others.
b. The hackers implanted a virus into some file on this computer. But there's no telling which file./#It's the file I'm pointing out to you right now.

In these cases, we can see that the speaker of the sentence cannot know who or what the referent of the *some NP* phrase is. Following such a phrase with an identifying statement (such as a 'namely, . . . ' statement) is infelicitous.

However, there are some cases of the use of *some*, exemplified in (2), where the meaning appears to be subtly different. The intuition about these cases is that the speaker's lack of knowledge in saying *some P* is not really connected to knowledge of which *P* is being referred to. Rather, the speaker does not know what *kind* of *P* the referent is.

(2) a. I saw some contraption in the copy room this morning.
b. I came home to find some plant growing through a hole in my wall.
c. Doctor, some growth appeared on my arm. Should I be worried?

These cases, on the reading of interest here, cannot be paraphrased as meaning (for example) 'I saw a contraption in the copy room and I don't know which contraption it

*I would like to thank Angelika Kratzer and two anonymous ESSLLI reviewers for extremely useful comments on this material. All errors are of course mine.

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see http://ceur-ws.org/, 2012, pp. 180–190.

was'. Rather, the meaning seems to be something more like 'I saw a contraption in the copy room and I don't know what kind of contraption it was'; and similarly for the other examples in (2). This is not just a matter of knowing what the name of the contraption is; I will argue that *some* can be sensitive to a lack of knowledge of a name in the case when it is in construction with a human-denoting NP, but not when it combines with a thing- or subkind-denoting NP. I will postulate a denotation for *some* based on the semantics proposed by [AOMB10b] for Spanish *algún*. The two readings, the 'unknown entity' and the 'unknown kind' reading, come about via polysemy of the common noun. Following [Kri95] and [Kra08], I will propose that a noun like *contraption* can represent either a property of individual contraptions or a property of subkinds of contraption. I will then argue that *some* can quantify over either of these; it also has built into it a partitive semantics, accounting for the fact that *some contraption is in the office* will have the truth conditions that some part of a subkind of contraption is in the office (and not a subkind itself). Firstly, I discuss the type of epistemic uncertainty that holds when *some* is used, before turning to how [AOMB10b]'s semantics for *algún* can be used to model *some*.

2 Uncertainty about things and people

We know from examples like the following (from [AOMB03]) that *some* in English is not generally incompatible with ostension. That is, a sentence like (3) is acceptable even if you can 'point to' the professor in question.

(3) Look! Some professor is dancing lambada on his table! [AOMB03, (9)]

Some other epistemic uncertainty is targeted by *some* in (3). [AOMB03] suggest the professor's name, which is possible but not the only possibility, as (4) shows:

(4) Look! Some professor wearing a name badge saying 'John Smith' is dancing the lambada![1]

[AP10] discuss cross-linguistic variation in epistemic determiners of the *some* kind (for example, Spanish *algún*, German *irgendein*, Italian *un qualche*, Romanian *vreun*). There is cross-linguistic variation in whether examples like (3) are licensed; Spanish *algún*, for example, is not licit in examples like (3). However, in cases like (3) in English, which are licit, there appear to be a variety of identification modes which *some* could indicate uncertainty about. Possibly *some* is sensitive to all the ways that there are of 'knowing who' [BL86] or 'knowing what', argued to be contextually determined by [Alo01].

If this were the end of the story, the examples of the form *I saw some contraption in the copy room*, which are the ones which motivate this paper, would not be interesting. *Some* in such examples might just denote lack of knowledge of the name of the subkind of contraption. However, I will argue that when *some* combines with NPs which denote things, rather than people, the subtleties discussed above disappear. With things, 'differentiation' – being able to distinguish the witness of the claim from other things in the extension of the NP – is the only identification mode which is relevant for *some*. Having established diagnostics to show this, I will go on to show that the 'unknown subkind' reading patterns with 'unknown things' rather than 'unknown people'.

[1] We assume that name badges are perfectly reliable. *Some* is still licensed even in that situation.

Prima facie, 'picking out' seems to be the identification mode for thing-denoting NPs as opposed to human-denoting NPs, as the below examples show.[2]

(5) a. Some professor is dancing the lambada!
  b. I saw some guy hanging about outside.

(6) a. ??Some statue is in the middle of the square. [looking at it]
  b. ??There's some letter in my mailbox. [looking at it]

Being able to 'pick out' the referent appears to delicense *some* with thing-denoting NPs in (6). One could argue that this simply represents there being many more means of 'knowing' applicable to humans than to things; so for example humans have names while things generally don't. However, even when things have names, *some* does not seem to be able to target uncertainty about the name, as the below examples show.

(7) Two diplomats from Peru are delegates to a conference you are at. One is a man and one a woman. You see them both several times, and know that they're both from Peru, but never catch their names.
  a. At dinner, I was sat across from a/some delegate from Peru.

(8) You are lost. You know that the city you're in has only two squares. You keep coming across both squares. You can tell them apart because one has a fountain and the other doesn't, but you can't see any street signs. You end up in the fountainless square in the city. Your friend phones you:
  a. A: Where are you?
  B: I'm in a/?#some square in the city.

Some appears able to signify lack of knowledge of the name in (7), but not easily in (8), even though city squares do usually have names. Given these data, I propose that the epistemic condition which *some* is sensitive to, when in construction with a thing-denoting NP, is uniformly that in (9). This will delicense *some* in situations like (8), where the speaker *can* differentiate the city squares, even though he does not know their names.[3]

(9) *Differentiation condition on 'some NP_{thing}'*
A speaker uses 'some NP_{thing}' to signal that she could not, if presented with the extension of NP, 'pick out' the witness of the existential claim.

[2]The examples in (6) are marginal rather than fully unacceptable, for reasons I will discuss in section 4.

[3]This 'differentiation' condition is subtly different from ostension, being able to 'point at' the referent. Ostension can't be exactly what is at play, because of the felicity of examples like:

(i) [A has been drugged and kidnapped; he wakes up, looks around, and exclaims:]
I'm trapped in some subway station!

Here, the poor kidnappee can point at the subway station perfectly well; what he is not in a position to do is distinguish the one he's in from other things in the extension of 'subway station'.

I do not want to speculate here about precisely how the contrast between human-denoting and thing-denoting NPs is to be modeled. An obvious way would be to posit homophony (or polysemy) between two *some*s, one which selects a [+human] argument and one which does not. That is an unattractively unparsimonious solution and hopefully further work can shed light on whether and how the two *some*s can be unified. That the distinction should be grammatically encoded, however (rather than following from some general constraint on epistemic relations towards people versus towards things), seems to be supported by contrasts like that between (7) and (8). For present purposes, it suffices to note that the contrast does exist.

3 Modeling the epistemic condition

How can we model in the semantics the 'differentiation' condition given in (9)? One way is that developed by [AOMB10b] for the Spanish indefinite determiner *algún*. *Algún* carries with it an implicature of speaker lack of knowledge, seemingly similar to the 'unknown entity' readings of English *some* discussed above. For example, asking a speaker to identify the referent of *algún NP* is an infelicitous conversational contribution, as shown in (10) ([AOMB10b]'s (8)).

(10) a. Juan tiene que estar en alguna habitación de la casa.
Juan has to be in ALGUNA room of the house
'Juan must be in a room of the house.'
b. #¿En cuál?
in which
('In which one?')

These contrast with examples where the determiner *un* is used. In such cases, there is no epistemic uncertainty, as shown in (11) ([AOMB10b]'s (10)).

(11) a. Juan tiene que estar en una habitación de la casa.
Juan has to be in UNA room of the house
'Juan must be in a room of the house.'
b. ¿En cuál?
in which
'In which one?'

[AOMB10b] propose that *algún* is a standard existential quantifier, taking a restrictor and a scope; as is *un*. However, both of these determiners also combine with a subset selection function f. The purpose of these subset selection functions is to restrict the domain of the quantifier to some subset, following proposals by [Sch02]. [AOMB10b] propose that *algún* places a presuppositional restriction on the function in question: it is an antisingleton function. *Un* does not do this. Below is a definition of an antisingleton subset selector function, and [AOMB10b]'s definitions of *un* and *algún* ([AOMB10b]'s (50, 54)).[4]

(12) Let f be a function which takes a set X and returns a subset of X.
f is an antisingleton function iff, for all X in the domain of f, $|f(X)| > 1$.

[4] Following [AOMB10b], and [HK98], I place presuppositional restrictions on the well-formedness of an expression between the colon and the period.

(13) a. $[\![\text{un}]\!] = \lambda f_{\langle \text{et,et} \rangle} \lambda P_{\langle \text{e,t} \rangle} \lambda Q_{\langle \text{e,t} \rangle} . \exists x[f(P)(x)\ \&\ Q(x)]$
b. $[\![\text{algún}]\!] = \lambda f \lambda P \lambda Q : \text{antisingleton}(f). \exists x[f(P)(x)\ \&\ Q(x)]$

[AOMB10b] argue that *algún* is in pragmatic competition with *un*. *Un* can in principle allow for an exhaustivity inference to be drawn; a speaker hearing (11) could, potentially, believe that the subset of rooms being quantified over is a singleton set. *Algún*, with its extra presupposition, precludes this possibility; on hearing a sentence like (10), the listener deduces that the speaker avoided using *un* specifically to avoid the possibility that the listener could draw the inference that the speaker was restricting the domain of rooms to a singleton set. Given this reasoning, it follows that *algún* is used to indicate that the speaker is actively unable to restrict the domain of rooms to a singleton set; that is, the speaker does not know which room Juan is in.

[AOMB10b, 16f.] argue that the 'antisingleton' presupposition is justified as *algún* cannot combine with NPs which must denote singleton sets, as in 'Juan bought *un/#algún* book that was the most expensive in the bookstore.' This seems also to be true of English examples such as *Mary bought a/?#some ring that was the most expensive in the jeweler's.* This 'antisingleton' presupposition can also, I argue, model the 'differentiation' constraint on English *some*, at least when *some* pairs with thing-denoting NPs. So, as a preliminary move, we can take over [AOMB10b]'s definition of *algún* to *some*:

(14) $[\![\text{some}]\!] = \lambda f_{\langle \text{et,et} \rangle} \lambda P_{\langle \text{e,t} \rangle} \lambda Q_{\langle \text{e,t} \rangle} : \text{antisingleton}(f). \exists x[f(P)(x)\ \&\ Q(x)]$

4 Kinds and subkinds

Having proposed this definition for *some*, I turn to the 'unknown subkind' reading of *some NP*. As noted above, the initial contrast in e.g. (2b), repeated as (15) below, would not be surprising if subkinds patterned with 'people' with respect to their combination with *some*, rather than 'things'.

(15) There's some plant growing through the wall of my room.

In this section, I will argue that subkind-denoting NPs do, however, generally pattern with 'things' in combination with *some*. The argument consists in showing that being able to distinguish subkinds from one another is sufficient to de-license *some*, even if other epistemic uncertainties (e.g. the name) remain. To show this, we turn to the following examples.

(16) *Able to distinguish subkinds*
Katniss, having grown up on her wits, is intimately familiar with all the plants in her district, and how they can be used for medicinal purposes. She's never had any formal schooling or parental teaching of herbal lore, though, so she doesn't know any of their names. She applies one to heal Gale's burns.
Gale: What's that?
Katniss: A/?#some plant that's good at soothing burns.

(17) *Unable to distinguish subkinds*
Katniss is in the Hunger Games Arena, far from home, where there are new types of plant that she's never seen before. She discovers through experimentation that one type is good for healing burns. She applies it to heal Rue's burns.
Rue: What's that?
Katniss: A/some plant that's good at soothing burns.

Here, the crucial point is that, in (16), Katniss is able to differentiate subkinds from each other, and categorize the entities (the actual plants) into each subkind, as appropriate. Even if she doesn't know the subkinds' *names*, *some* is still not licensed. However, if Katniss cannot differentiate the subkinds with certainty, as in (17), *some* on the subkind reading is again licensed.

I now turn to an analysis of common nouns which will allow us to derive the 'unknown subkind' reading. Nouns like *plant* appear to be polysemous between denoting individual plants and subkinds of plants, as shown in (18):

(18) That plant {was watered yesterday/is widespread}.

I will summarize a means of achieving this polysemy compositionally, proposed by [Kri95] and [Kra08]. We start from the assumption that a kind, for example the kind PLANT, is no more than the mereological sum of all plants in the world. See e.g. [Chi98, 349]: 'It seems natural to identify a kind in any given world (or situation) with the totality of its instances.'[5] [Kra08] argues that the noun root $\sqrt{\text{plant}}$ denotes this sum of all plants, which I will notate with ΣPlant. This root is not pronounced alone, however. The word *plant* which we actually pronounce includes a classifier, which in English is silent. There are two classifiers in English: one combines with a kind and returns a property of individuals which are parts of the kind; the other combines with a kind and returns a property of subkinds of the kind. (19), (20) show how this works. 'Π' is the part relation, as in [Lin83].

(19) (Kratzer's (2), adapted)

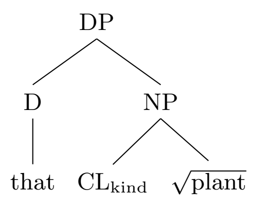
a. $[\![\sqrt{\text{plant}}]\!] = \Sigma\text{Plant}$

b. $[\![\text{CL}_{\text{ind}}]\!] = \lambda x \lambda y.\text{kind}(x)\ \&\ \text{individual}(y)\ \&\ y \Pi x$
(takes a kind and returns the property of being an individal member of that kind)

c. $[\![\text{CL}_{\text{kind}}]\!] = \lambda x \lambda y.\text{kind}(x)\ \&\ \text{kind}(y)\ \&\ y \Pi x$
(takes a kind and returns the property of being a subkind of that kind)

In (20), we see how this system can provide a subkind reading for a DP like *that plant*. The NP formed by the combination of the root $\sqrt{\text{plant}}$ and the classifier CL_{kind} is pronounced as *plant*.

(20) a.



[5] In fact, Chierchia argues that kinds are not entity-type, type e, but individual concept type, type $\langle s, e\rangle$ – a function from situations to entities. Here, I work with a fully extensional semantics, and so treat kinds simply as the entities which would, in a fuller treatment, be the result of applying a Chierchia-type kind to the world (or situation) of evaluation w_0.

b. Function Application on $[\![\mathrm{CL_{kind}}]\!]$ and $[\![\sqrt{\mathrm{plant}}]\!]$:
$\lambda y.\mathrm{kind}(\Sigma\mathrm{Plant})\ \&\ \mathrm{kind}(y)\ \&\ y\Pi(\Sigma\mathrm{Plant})$
$= \lambda y.\mathrm{kind}(y)\ \&\ y\Pi(\Sigma\mathrm{Plant})$[6]

c. $[\![\mathrm{that}]\!] = \lambda P_{\langle e,t\rangle}.\iota x[P(x)]$[7]

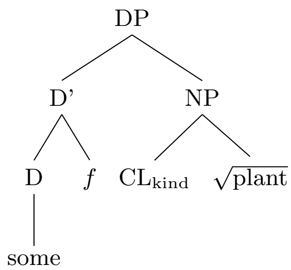
d. Function Application on (b) and (c): $\iota x[\mathrm{kind}(x)\ \&\ x\Pi(\Sigma\mathrm{Plant})]$
'The contextually salient subkind of plant'

The word *plant* can either denote the property of being an individual plant, or the property of being a subkind of plant. Given this, it is not surprising that *some* might combine with *plant* and yield an 'unknown kind' reading, rather than lack of certainty as to the witness of the existential claim. In a sentence like *Some plant is growing through my wall*, we want *some* to make an existential claim about a plant, the witness to which claim the speaker might well be able to distinguish from other members of $[\![\mathrm{plant}]\!]$; and at the same time express speaker ignorance about the *kind* of plant. Below is a revised proposal for the denotation of *some*, based on [AOMB10b]'s proposal for *algún*, but with a crucial underlined addition.

(21) $[\![\mathrm{some}]\!] = \lambda f_{\langle et,et\rangle}\lambda P_{\langle e,t\rangle}\lambda Q_{\langle e,t\rangle} : \mathrm{antisingleton}(f).\exists x[(f(P))(x)$
$\underline{\&\ \exists y[y\Pi x\ \&\ Q(y)]]}$

In this denotation, the quantifier's scope, Q, is not applied to the x that the restrictor P applies to. Rather, there is a subpart of x which Q applies to. So *some P is a Q* means that there is a part of a P that is a Q. The work done by the underlined addition is to allow *some* to felicitously combine with kind-type arguments, but to end up quantifying, not over subkinds themselves, but over instantiations of subkinds (here equated with parts of those subkinds). If we apply this *some* to a noun root combined with $\mathrm{CL_{kind}}$, we get the reading that is at issue here, as shown below. For perspicuity, I have indicated some places where I replace lambda-abstraction notation for properties (understood extensionally as sets of entities) with set notation, as the lambda notation for the set of subkinds in (23a) is not very transparent when embedded in a larger expression such as (23b).

(22)



(23) a. $[\![\mathrm{NP}]\!] = \lambda y.\mathrm{kind}(y)\ \&\ y\Pi(\Sigma\mathrm{Plant})$
$= \{\Sigma\mathrm{Ivy}, \Sigma\mathrm{Creeper}, \Sigma\mathrm{Rhododendron}, \ldots\}$ (using set notation)

[6]I abbreviate by removing from the truth conditions the restriction on ΣPlant that it be a kind. I simply assume from now on that $\mathrm{CL_{kind}}$ combines only with kinds, without writing it explicitly into its denotation.

[7]I am ignoring the deictic contribution of *that* here and simply identifying it with *the*.

b. $[\![\text{DP}]\!] = \lambda Q.\exists x[(f(\lambda y.\text{kind}(y)\ \&\ y\Pi(\Sigma\text{Plant}))(x))\ \&\ \exists z[z\Pi x\ \&\ Q(z)]]$
$= \lambda Q.\exists x[x \in f(\{\Sigma\text{Ivy}, \Sigma\text{Creeper}, \Sigma\text{Rhododendron}, \ldots\})\ \&\ \exists z[z\Pi x\ \&\ z \in Q]]$ (using set notation)
Presupposition: antisingleton(f)

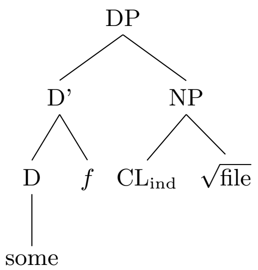
(24) $[\![\text{some plant is growing through my wall}]\!] =$
$\exists x[(f(\lambda y.\text{kind}(y)\ \&\ y\Pi(\Sigma\text{Plant}))(x))\ \&\ \exists z[z\Pi x\ \&\ \text{growsInWall}(z)]]$
'There is some subkind of plants x, and there is something z that is a part of that subkind of plants, and z is growing through my wall, and the speaker wants to signal that the set of subkinds of plants to which x belongs is not a singleton.'

The pragmatic effect of (24) is such that there is no signal that the speaker does not know which plant is at issue. It is not the case that the speaker is signaling that she cannot narrow the set of plants down to a singleton set. Rather, she is signaling that she is not whittling the set of subkinds of plant ({ΣIvy, ΣCreeper, ΣRhododendron, . . . }) down to a singleton. We use the same reasoning as used by [AOMB10b] to analyze *algún*: on hearing *some plant*, the listener deduces that the speaker chose *some* (rather than *a*, which has no anti-singleton presupposition) in order to signal that she actively could not restrict the set of possible subkinds that the plant could fall into to a singleton set.

This analysis also predicts the marginal (not fully ungrammatical) status of sentences like ??*There's some statue in the town square* (while looking at the statue). Such sentences are good exactly to the extent that we can imagine uncertainty about the subkind involved; that is, they are good to the extent that *statue* can mean *kind of statue* (see [Car77] for discussion of which nouns can easily receive a kind reading).

5 *Some* with individuals

In cases like *some file is infected*, there does seem to be an epistemic effect concerning individuals, very much parallel to Spanish *algún*. Do we need yet more *some*s in English; one to combine with properties of kinds, and one combining with properties of individuals? I will argue that this is not the case. There is only one *some* (at least when it combines with thing-denoting NPs), with the semantics given above. Below, I show the result of combining this *some* with a property of individuals, another possible meaning for a common noun.

(25) a.


b. $[\![\text{NP}]\!] = \lambda y.\text{individual}(y)\ \&\ y\Pi(\Sigma\text{File})$[8]
$= \{\text{file}_1, \text{file}_2, \text{file}_3, \ldots\}$ (in set notation)

[8] Again, I abbreviate by omitting the statement that ΣFile is a kind.

c. $[\![\text{DP}]\!] = \lambda Q \exists x[(f(\lambda y.\text{individual}(y)\ \&\ y\Pi(\Sigma\text{File})))(x)\ \&\ \exists z[z\Pi x\ \&\ Q(z)]$
$= \lambda Q \exists x[x \in f(\{\text{file}_1, \text{file}_2, \text{file}_3, \ldots\})\ \&\ \exists z[z\Pi x\ \&\ z \in Q]]$
(in set notation)
Presupposition: antisingleton(f)

(26) $[\![\text{some file is infected}]\!] = \exists x[(f(\lambda y.\text{individual}(y)\ \&\ y\Pi(\Sigma\text{File})))(x)$
$\&\ \exists z[z\Pi x\ \&\ \text{infected}(z)]]$
'There is some individual x which is a mereological part of the file-kind (i.e. x is a file), and there is a z which is a mereological part of x, and z is infected, and the speaker wants to signal that the set of individual files to which x belongs is not a singleton.'

If we say there is an individual x which is a file, and a z such that $z\Pi x$, then – because x is an individual and so has no proper mereological parts – z must be an improper mereological part of x, that is, $z = x$. The semantics then is precisely equivalent to the semantics proposed by [AOMB10b] for *algún*. A sentence like *some file is infected* means that there is an x which is a member of a subset of individual files, and x is infected, and the speaker wishes to signal that the subset of individual files is not a singleton set; that is, the speaker cannot specify which files is the witness to the existential claim. We therefore derive both the 'unknown individual' and 'unknown kind' readings with the same denotation for *some*.

6 Notes on plurality

Consider the semantics for a sentence like *some contraption is in the office* (on the 'unknown kind' reading).

(27) $[\![\text{some contraption is in the office}]\!] =$
$\exists x[(f(\lambda y.\text{kind}(y)\ \&\ y\Pi(\Sigma\text{Contraption}))(x))\ \&\ \exists z[z\Pi x\ \&\ \text{inTheOffice}(z)]]$

Let us say that one choice of contraption is 'hole punch'. Then we could have, for example, the situation where $\exists z[z\Pi(\Sigma\text{HolePunch})\ \&\ \text{inTheOffice}(z)]$ is true as one verifying instance of (27). We do not actually have any requirement that z in (27) be atomic, despite the singular morphology on the noun *contraption*. On the basis of the following examples, I suggest that in fact the 'unknown-kind' reading of *some* is indeed number-neutral.

(28) A: What's this warehouse for?
B: There's some contraption in there. There are shelves upon shelves of the things, all the same. I don't know what they are, though.

(29) [I take a delivery of 100 plants, but they are not the type I ordered; they are all the same type of plant, but I don't recognize what type.]
A: Did you get the plants you ordered?
B: They did deliver some plant. I have 100 of the things clogging up the office. But I've no idea what they are, they're not what I ordered.

Some can range over pluralities; the restriction on cases like (28), is that all the things in the plurality belong to the same subkind. If this is not the case, then the examples above become sharply bad.

(30) A: What's this warehouse for?
B: #There's some contraption in there. Three shelves for three different things, but I don't know what any of them are.

(31) [I take a delivery of 100 plants, not the type I ordered, and not all the same type of plant; I don't recognize any of the types of plants.]
A: Did you get the plants you ordered?
B: #They did deliver some plant.

These examples suggest that the semantics for *some* proposed here is on the right track; on the 'unknown kind' reading, *some NP* is number-neutral with respect to the *entities* quantified over, but makes reference to one specific subkind of the denotation of the NP. Note that *some file is infected*, the individual reading, is not number-neutral with respect to entities in the same way:

(32) Some file is infected. I don't know which one/#which ones.

The (unknown) referent of *some file* cannot be plural. However, this is accounted for by the semantics of the individual classifier $\mathrm{CL}_{\mathrm{ind}}$, which, when combined with a root like $\sqrt{\text{file}}$, returns a set of *individual* files. *Some file is infected* asserts that there is (some part of) some member of that set which is infected. We thereby achieve the result that *some file is infected* only makes reference to individuals, in contrast with the 'unknown kind' reading of a phrase like *some plant*.

7 Conclusion

I have argued that the 'unknown individual' meaning of *some file is infected* and the 'unknown subkind' reading of *some contraption is in the office* can be accounted for by a unitary analysis of *some*, whose denotation includes a partitive semantics allowing it to combine with properties of kinds. The ambiguity is not a property of *some* itself, but rather due to the polysemy of noun phrases like *plant, contraption* between subkind-type readings and individual-type readings, following [Kri95] and [Kra08].

Various questions remain. For example, does *some* with plural NPs (as in *some files are infected, namely these ones*), which has no epistemic effect, admit of the same analysis as *some* with singular NPs? [AOMB10a] propose an analysis of Spanish *algunos* (the plural form of *algún*), which also does not have an epistemic effect, where the antisingleton constraint is retained in the denotation but the epistemic effect is not present.[9] Whether the account can be transplanted to English is a question I leave for future work.

Furthermore, the question raised in section 2 concerning the source of the difference between *some* when paired with human-denoting NPs and thing-denoting NPs remains open. This 'split' does not seem to be cross-linguistic – for example, *algún* is

[9] [AOMB10a] assume a number-neutral semantics for plural, in which case the set {John, Mary, John+Mary} can be the subset of 'students' picked out by '*algunos* students'. Crucially, while this set contains only one plural individual (John+Mary), the set is not a singleton, and so is licit as the domain of *algunos*. It is possible, then, for the speaker to not narrow down the set of students to a singleton set and yet have only one witness (here, John and Mary) in mind. *Algunos* is then predicted not to have an epistemic effect. See [AOMB10a] for the full details.

not felicitous in [AOMB03]'s 'lambada professor' example. That these splits are not reproducible across languages seems to indicate that the nature of the constraint is grammatical, rather than a 'deeper' (mental/cognitive) constraint on epistemic relations. Cross-linguistic work will be crucial here (see [AP10] for an overview). I leave these as open questions, however.

References

Alo01. Maria Aloni. *Quantification under conceptual covers.* PhD thesis, University of Amsterdam, 2001.

AOMB03. Luis Alonso-Ovalle and Paula Menéndez-Benito. Some epistemic indefinites. In Makoto Kadowaki and Shigeto Kawahara, editors, *Proceedings of NELS 34*, pages 1–12. Amherst, MA: GLSA, 2003.

AOMB10a. Luis Alonso-Ovalle and Paula Menéndez-Benito. Domain restriction, modal implicatures and plurality: Spanish *algunos*. *Journal of Semantics*, 28(2):211–40, 2010.

AOMB10b. Luis Alonso-Ovalle and Paula Menéndez-Benito. Modal indefinites. *Natural Language Semantics*, 18:1–31, 2010.

AP10. Maria Aloni and Angelika Port. Epistemic indefinites cross-linguistically. Presented at NELS 41. `http://staff.science.uva.nl/~maloni/NELS2010-handout.pdf`, 2010.

Bec99. Misha Becker. The *some* indefinites. In Gianluca Storto, editor, *Syntax at Sunset 2*, number 3 in UCLA Working Papers in Linguistics, pages 1–13. 1999.

BL86. Steven E. Boër and William G. Lycan. *Knowing who.* Cambridge, MA: MIT Press, 1986.

Car77. Gregory N. Carlson. *Reference to kinds in English.* PhD thesis, University of Massachusetts Amherst, 1977.

Chi98. Gennaro Chierchia. Reference to kinds across languages. *Natural Language Semantics*, 6:339–405, 1998.

Far02. Donka Farkas. Varieties of indefinites. In Brendan Jackson, editor, *Proceedings of SALT 12*, pages 59–83. Ithaca, NY: CLC Publications, Cornell University, 2002.

HK98. Irene Heim and Angelika Kratzer. *Semantics in generative grammar.* Malden: Blackwell, 1998.

Kra08. Angelika Kratzer. On the plurality of verbs. In Johannes Dölling, Tatjana Heyde-Zybatow, and Martin Schäfer, editors, *Event structures in linguistic form and interpretation*, pages 269–300. Berlin: Walter de Gruyter, 2008.

Kri95. Manfred Krifka. Common nouns: a contrastive analysis of Chinese and English. In Gregory N. Carlson and Francis J. Pelletier, editors, *The generic book*, pages 398–411. Chicago: University of Chicago Press, 1995.

Lin83. Godehard Link. The logical analysis of plurals and mass terms: A lattice theoretic approach. In Rainer Bäuerle, Christoph Schwarze, and Arnim von Stechow, editors, *Meaning, use and the interpretation of language.* Berlin: de Gruyter, 1983.

Sch02. Roger Schwarzschild. Singleton indefinites. *Journal of Semantics*, 19(3):289–314, 2002.

Tableau-based Decision Procedure for Hybrid Logic with Satisfaction Operators, Universal Modality and Difference Modality

Michał Zawidzki[1 2]

[1] Department of Logic, University of Lodz
16/18 Kopcinskiego St., 90-232 Lodz, Poland
[2] School of Computer Science, The University of Manchester
Oxford Road, Manchester M13 9PL, United Kingdom
michal.zawidzki@gmail.com

Abstract. Hybrid logics are extensions of standard modal logics, which significantly increase the expressive power of the latter. Since most of hybrid logics are known to be decidable, decision procedures for them is a widely investigated field of research. So far, several tableau calculi for hybrid logics have been presented in the literature. In this paper we introduce a sound, complete and terminating tableau calculus $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$ for hybrid logics with satisfaction operators, universal modality and difference modality. $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$ not only uniformly covers relatively wide range of various hybrid logics but it is also conceptually simple and enables effective search for a minimal model for a satisfiable formula. $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$ exploits the *unrestricted blocking* mechanism introduced as an explicit, sound tableau rule.

Keywords: hybrid logics, modal logics, tableau algorithms, decision procedures, automated reasoning

1 Introduction

Hybrid logics are powerful extensions of modal logics which allow referring to particular states of a model without using meta-language. In order to achieve it, the language of standard modal logics is enriched with the countably infinite set of propositional expressions called *nominals* (we fix the notation NOM $= \{i, j, k, \dots\}$ to stand for the set of nominals), disjoint from the set of propositional variables PROP. Each nominal is true at exactly one world and therefore can serve both as a label and as a formula. Supplying a language with nominals significantly strengthens its expressive power. In the presented paper we also consider further modifications of hybrid logic obtained by adding the so-called satisfaction operators, the universal modality and the difference modality. The satisfaction operators of the form $@_i$ allow stating that a particular formula holds at a world labelled by i. The universal modality E expresses the fact that there exists a world in a domain, at which a particular formula holds. The difference modality D stands for the fact that a particular formula holds at a world different from the current one.

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see http://ceur-ws.org/, 2012, pp. 191–201.

Some hybrid logics additionally contain a different sort of expressions, the *state variables*, which allow quantifying over worlds, and additional operators like the *down-arrow operator* or the *state quantifiers*. However, these logics are proven to be undecidable (cf. [1]) so, in principle, they cannot be subjected to a terminating tableau-based decision procedure. We therefore confine ourselves only to the forgoing decidable hybrid logic.

In the present paper we introduce a sound, complete and terminating tableau calculus $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$ for hybrid logics with @, E and D operators. Our approach, unlike that in [7] and [3], is focused on the uniform treatment of all aforementioned logics, conceptual simplicity and minimality of models generated by $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$. Basing on [10], we introduce the *unrestricted blocking* mechanism that satisfies these conditions.

In Section 2 a characterisation of the logic $\mathcal{H}(@,\mathsf{E},\mathsf{D})$ is provided. In Section 3 we introduce the tableau calculus $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$ and we describe the decision procedure for $\mathcal{H}(@,\mathsf{E},\mathsf{D})$. In Section 4 we prove soundness and completeness of $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$ and Section 5 provides a closer look at the termination problem for $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$. We conclude the paper in Section 6.

2 Hybrid logic

Syntax

Let $\mathcal{O} \in \{@,\mathsf{E},\mathsf{D}\}$. By $\mathcal{H}(\mathcal{O})$ we will denote the hybrid logic with operator(s) $\mathcal{O}$.

We recursively define the set FORM of well-formed formulas of the logic $\mathcal{H}(@,\mathsf{E},\mathsf{D})$ in the following manner:

$$\text{FORM} \ni \varphi := p \mid i \mid \neg\psi \mid \psi \wedge \chi \mid \Diamond\psi \mid @_i\psi \mid \mathsf{E}\psi \mid \mathsf{D}\psi,$$

where $p \in \text{PROP}$, $i \in \text{NOM}$ and $\psi, \chi \in \text{FORM}$.

Other connectives and operators are defined in a standard way. Both E and D have dual operators. @ is self-dual. We abbreviate $\neg\mathsf{E}\neg$ as A.

Semantics

A model $\mathfrak{M}$ for hybrid logic $\mathcal{H}(@,\mathsf{E},\mathsf{D})$ is a triple $\langle W, R, V\rangle$ where:

$W \neq \emptyset$ is called a *domain*,
$R \subseteq W^2$ is called an *accessibility relation*,
$V : \text{PROP} \cup \text{NOM} \longrightarrow \mathcal{P}(W)$ such that for each $i \in \text{NOM}$ $V(i)$ is a singleton set; V is called a *valuation function*.

Relation $\models$ (*forcing*) is defined inductively:

$$\begin{array}{lcl}
\mathfrak{M}, w \models p & \Leftrightarrow & w \in V(p), \quad p \in \text{PROP};\\
\mathfrak{M}, w \models i & \Leftrightarrow & \{w\} = V(i), \quad i \in \text{NOM};\\
\mathfrak{M}, w \models \neg\varphi & \Leftrightarrow & \mathfrak{M}, w \not\models \varphi;\\
\mathfrak{M}, w \models \varphi \wedge \psi & \Leftrightarrow & \mathfrak{M}, w \models \varphi \mathbin{\dot\wedge} \mathfrak{M}, w \models \psi;\\
\mathfrak{M}, w \models \Diamond\varphi & \Leftrightarrow & \exists z \in W(wRz \mathbin{\dot\wedge} \mathfrak{M}, z \models \varphi);\\
\mathfrak{M}, w \models @_i\varphi & \Leftrightarrow & \{z\} = v(i) \mathbin{\dot\wedge} \mathfrak{M}, z \models \varphi;\\
\mathfrak{M}, w \models \mathsf{E}\varphi & \Leftrightarrow & \exists z \in W(\mathfrak{M}, z \models \varphi);\\
\mathfrak{M}, w \models \mathsf{D}\varphi & \Leftrightarrow & \exists z \in W(z \mathbin{\dot\neq} w \mathbin{\dot\wedge} \mathfrak{M}, z \models \varphi),
\end{array}$$

where $\Leftrightarrow, \exists, \dot{\wedge}, \models, \doteq$ are meta-language symbols. Henceforth, we will call the expressions containing these meta-language symbols the *domain expressions*.

3 Synthesising tableau calculus for the logic $\mathcal{H}(@, \mathsf{E}, \mathsf{D})$

Tableau calculi

Two main types of tableau calculi for hybrid logics are present in the literature, namely the *prefixed* and the *internalised* calculi. The prefixed calculi consist in introducing another sort of expressions, namely *prefixes*. They serve as labels for worlds, which, unlike nominals, are of meta-linguistic provenience. Another type of meta-language expressions occurring in prefixed tableaux are the *accessibility expressions*. The equality between two prefixes is expressed implicitly by imposing on them the satisfaction of the same nominal. Apparently, prefixed calculi are less complex than internalised calculi. Besides, basic hybrid logic $\mathcal{H}$ is not supplied with sufficient expressive power to internalise its own semantics. It therefore requires the domain expressions occurring in the calculus. The most widely known prefixed tableau calculi for hybrid logics come from Tzakova [12], Bolander and Braüner (who improved Tzakova's calculus to the terminating version) [5], Kaminski and Smolka [8]. The tableau calculus for hybrid logics obtained than the synthesised framework from [10] is also subsumed under the prefixed calculi class.

Internalised calculi for hybrid logics take advantage of the high expressive power of these logics which allows encoding the domain expressions within the language. Although internalisation of the logic allows dispensing with certain rules present in prefixed tableau calculi, it also jeopardises termination of the calculus by, e.g., using pure axioms (not including other formulas but nominals) to characterise frame conditions (cf. [3]).

In this section we present an internalised tableau calculus covering hybrid logics with the satisfaction operators, the universal modality and the difference modality. It resembles Blackburn's calculus from [2] modified by Bolander and Braüner in [5] and by Blackburn and Bolander in [3]. However, certain rules have been added (e.g. the rules for D).

Encoding the domain expressions

In [2] Blackburn made an observation that the language of hybrid logic with @ operators is sufficiently rich to express semantics within itself. As we mentioned in Section 2, there are three types of the domain expressions: satisfaction statements ($\mathfrak{M}, w \models \varphi$), accessibility statements (wRv) and equality statements ($w \doteq v$). Hybrid equivalents of the forgoing expressions are shown below.

$$\begin{array}{ll}
\mathcal{H}(\mathfrak{M}, w \models \varphi) = @_{i_w}\varphi & \mathcal{H}(\mathfrak{M}, w \not\models \varphi) = @_{i_w}\neg\varphi \\
\mathcal{H}(R(w, v)) = @_{i_w}\Diamond j_v & \mathcal{H}(\neg R(w, v)) = @_{i_w}\neg\Diamond j_v \\
\mathcal{H}(w \doteq v) = @_{i_w} j_v & \mathcal{H}(w \dot{\neq} v) = @_{i_x}\neg j_y
\end{array}$$

Both E and D operators allow mimicking @ operators: $@_i\varphi := \mathsf{E}(i \wedge \varphi)$ and $@_i\varphi := (i \wedge \varphi) \vee \mathsf{D}(i \wedge \varphi)$. Therefore, in the calculus we use the notation $i : \varphi$, rather than $@_i\varphi$, to keep its universal character. This *colon* notation will stand for one of the forgoing expressions, depending on a considered logic, except for the fact that whenever a logic includes @ operators, $i : \varphi$ means $@_i\varphi$.

Rules for the connectives:

$$(\neg)\ \frac{i:\neg j}{j:j} \qquad (\neg\neg)\ \frac{i:\neg\neg\varphi}{i:\varphi} \qquad (\wedge)\ \frac{i:\varphi\wedge\psi}{i:\varphi, i:\psi} \qquad (\neg\wedge)\ \frac{i:\neg(\varphi\wedge\psi)}{i:\neg\varphi \mid i:\neg\psi}$$

$$(\Diamond)^*\ \frac{i:\Diamond\varphi}{i:\Diamond j, j:\varphi} \qquad (\neg\Diamond)\ \frac{i:\neg\Diamond\varphi, i:\Diamond j}{j:\neg\varphi} \qquad (@)\ \frac{i:@_j\varphi}{j:\varphi} \qquad (\neg @)\ \frac{i:\neg @_j\varphi}{j:\neg\varphi}$$

$$(\mathsf{E})^*\ \frac{i:\mathsf{E}\varphi}{j:\varphi} \qquad (\neg\mathsf{E})\ \frac{i:\neg\mathsf{E}\varphi, j:j}{j:\neg\varphi} \qquad (\mathsf{D})^*\ \frac{i:\mathsf{D}\varphi}{i:\neg j, j:\varphi} \qquad (\neg\mathsf{D})\ \frac{i:\neg\mathsf{D}\varphi, j:j}{i:j \mid j:\neg\varphi}$$

Equality rules:

$$(\mathsf{ref})\ \frac{i:\varphi}{i:i} \qquad (\mathsf{sub})\ \frac{i:j, i:\varphi}{j:\varphi}$$

Closure rule and unrestricted blocking rule:

$$(\bot)\ \frac{i:\varphi, i:\neg\varphi}{\bot} \qquad (\mathsf{ub})\ \frac{i:i, j:j}{i:j \mid i:\neg j}$$

* Nominals in the conclusions are fresh on the branch.

Fig. 1. Rules for the calculus $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$

Tableau calculus

Figure 1 presents the rules of the tableau calculus $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$ for the logic $\mathcal{H}(@,\mathsf{E},\mathsf{D})$.

Boolean rules are straightforward and require no additional comments. $(\Diamond)$, (E) and (D) are rules introducing new labels, which was marked as the side-condition for them. In the case of $(\neg\mathsf{E})$ and $(\neg\mathsf{D})$ the standard side-condition of former occurrence of a label on a branch was replaced by introducing an explicit premiss stating that a particular nominal has appeared as a label on a branch. The rule (ref) is a reflexivity rule that introduces to a branch the explicit information that a nominal occurred as a label within a branch. (sub) expresses the substitutability of two nominals as labels, provided that one of them is labelled by the other. The $(\bot)$ rule is self-evident. The (ub) rule is a variant of the *analytical cut rule* applied to nominals. Intuitively, if two labels appear on a branch, they either label two distinct worlds or the same world. Thus, (ub) allows comparing any pair of labels that appeared on a branch. As it will turn out before long, this possibility is essential for termination of the whole calculus. The rule $(\neg\mathsf{D})$ deserves a wider comment. In [3] Blackburn and Bolander notice that the $(\neg\mathsf{D})$ rule of the form

$$\frac{i:\neg\mathsf{D}\varphi, i:\neg j}{j:\neg\varphi}$$

breaks the completeness of the whole calculus. In [8] Kaminski and Smolka formulate $(\neg\mathsf{D})$ correctly but they do not explain why no such modification like from [3] can be applied to $(\neg\mathsf{D})$. In [10] Schmidt and Tishkovsky introduce the $(\dagger)$ condition which decides whether the refinement can be applied to a rule:

Theorem 1 ([10]). *Let $\mathcal{T}$ be a tableau calculus. Let β be the rule of the form $\frac{X_0}{X_1|\cdots|X_n}$ and let $\mathcal{T}^R$ be a refined version of $\mathcal{T}$. Suppose that $\mathcal{B}$ is an arbitrary open and fully-expanded branch in a $\mathcal{T}^R$-tableau.Let $F = \{\varphi_1,\ldots,\varphi_l\}$ be a set of all $\mathbf{K}(\mathsf{E}_n)$-formulas from $\mathcal{B}$ reflected in $\mathfrak{M}(\mathcal{B})$. Then the refined rule $\frac{X_0,\neg X_{i_1},\ldots,\neg X_{i_j}}{X_{i_{j+1}}|\cdots|X_{i_n}}$ is admissible (i.e. the* *resulting calculus $\mathcal{T}^R$ is still complete) satisfies the following condition is satisfied:*

$$\begin{array}{l} \textit{If } X_0(\varphi_{i_1}, \ldots, \varphi_{i_k}) \in \mathcal{B} \\ \textit{then } \mathfrak{M}(\mathcal{B}) \models X_m(\varphi_{i_1}, \ldots, \varphi_{i_k}), \textit{ for some } m \in \{1, \ldots, n\}. \end{array} \qquad (\dagger)$$

($\dagger$) holds for ($\neg\Diamond$) in most modal and description logics but, as it turns out, it fails for ($\neg$D).

Before we provide a proper method of constructing a tableau, we need to introduce several preliminary definitions.

Definition 1. *We call a branch of a $\mathcal{T}_{\mathcal{H}(@,E,D)}$ tableau* closed *if the closure rule was applied on it. If a branch is not closed, it is* open. *An open branch is* fully expanded *if no other rules are applicable on it.*

Definition 2. *Let* NOM($\mathcal{B}$) *be a set of nominals occurring as labels on a fully expanded branch $\mathcal{B}$ of a $\mathcal{T}_{\mathcal{H}(@,E,D)}$ tableau for a given input formula. We introduce the $\leftrightsquigarrow_{\mathcal{B}}$ relation over* NOM($\mathcal{B}$) *which we define in the following way:*

$$i \leftrightsquigarrow_{\mathcal{B}} j \textit{ iff } i : j \in \mathcal{B}.$$

Proposition 1. *$\leftrightsquigarrow_{\mathcal{B}}$ is the equivalence relation.*

Proof. Reflexivity is ensured by the (ref) rule. For *symmetry* assume that $i : j$ is on $\mathcal{B}$. By (ref) we obtain $i : i$ and after applying (sub) to these two premises we obtain $j : i$. For *transitivity* suppose that $i : j$ and $j : k$ are on $\mathcal{B}$. By symmetry we have that $j : i$ is also on $\mathcal{B}$. We therefore take $j : i$ and $j : k$ as premises of (sub) and obtain $i : k$.

Definition 3. *A rule of the $\mathcal{T}_{\mathcal{H}(@,E,D)}$ is applied* eagerly *in a tableau iff whenever it is applicable, it is applied.*

Definition 4. *Let $\prec_{\mathcal{B}}$ be an ordering on* NOM($\mathcal{B}$) *defined as follows:*

$$i \prec_{\mathcal{B}} j \textit{ iff } i : i \textit{ occurred on } \mathcal{B} \textit{ earlier than } j : j.$$

Note that $\prec_{\mathcal{B}}$ is well-founded and linear since no rule introduces more than 1 labelling nominal as a conclusion.

Definition 5. *To each $\mathcal{T}_{\mathcal{H}(@,E,D)}$ rule we affix the* priority number. *It indicates what the order of application of particular rules should be. The lower the number is, the sooner the rule should be applied. We have: (ref), ($\neg$): 1, (ub): 2, (sub):3, ($\neg\neg$), ($\wedge$), ($\neg\wedge$): 4, ($\neg\Diamond$), ($\neg E$), ($\neg D$): 5, ($\Diamond$), (E), (D): 6.*

Now we are ready to provide the tableau construction algorithm. As usual, we do it inductively.

Definition 6 (Tableau construction algorithm). ***Basic step****: For a given input formula φ put $i : \neg\varphi$ at the initial node. i is a nominal not occurring in φ.*
Inductive step*: Suppose that you performed n steps of a derivation. In the n+1th step apply the rules of $\mathcal{T}_{\mathcal{H}(@,E,D)}$* eagerly *respecting the priority ordering given in Definition 5 and fulfilling the following conditions:*

(c1) if the application of a rule results in a formula that is already present on a branch, do not perform this application;

(c2) rules of priority 5 and 6 can only by applied to labels that are the least elements (with respect to $\prec_{\mathcal{B}}$*) of the equivalence class (with respect to* $\leftrightsquigarrow_{\mathcal{B}}$*);*
(c3) the ($\Diamond$) *must not be applied to formulas of the form* $i : \Diamond j$*. We call them the* accessibility formulas*;*
(c4) apply the ($\bot$) *rule whenever it is possible.*

If after the $n + 1th$ *step of derivation:*

(a) all tableau branches are closed, stop and return: theorem*,*
(b) there are open branches in a tableau and no further rules are applicable (respecting conditions (c1)-(c4)), stop and return: non-theorem*;*
(c) there are open branches in a tableau and further rules are applicable (respecting conditions (c1)-(c4)), proceed to the $n + 2th$ *step.*

We will explain the way the (ub) rule works more carefully in Section 5.

4 Soundness and Completeness of $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$

In the current section we state and prove soundness and completeness of the forgoing calculus. First, we formulate the following

Definition 7. *We call a tableau calculus* $\mathcal{T}$ *sound if and only if for each satisfiable input formula* φ *each tableau* $\mathcal{T}(\varphi)$ *is open, i.e., there exists a fully expanded branch on which no closure rule was applied. A tableau calculus is called complete if and only if for each unsatisfiable input formula* φ *there exists a closed tableau, i.e. a tableau where a closure rule was applied on each branch.*

For soundness it amounts to proving that particular rules preserve satisfiability. For completeness we take the contrapositive of the condition given in Definition 7 and demonstrate that if there exists an open, fully expanded branch $\mathcal{B}$ in a tableau for φ then there exists a model for φ.

Theorem 2. $\mathcal{T}_{\mathcal{H}(@,E,D)}$ *is sound.*

Proof. By easy verification of all the rules.

Suppose that $\mathcal{B}$ is an open, fully expanded branch in a $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$ tableau for φ. We define a model $\mathfrak{M}(\mathcal{B}) = \langle W, R, V\rangle$ derived from $\mathcal{B}$ in the following way:

$W = \{[i]_{\leftrightsquigarrow_{\mathcal{B}}} \mid i : i \in \mathcal{B}\}$;
$R = \{([i]_{\leftrightsquigarrow_{\mathcal{B}}}, [j]_{\leftrightsquigarrow_{\mathcal{B}}}) \mid i : \Diamond j :\in \mathcal{B}\}$;
$V = \{(i, [i]_{\leftrightsquigarrow_{\mathcal{B}}}) \mid i : i \in \mathcal{B}\} \cup \{(p, U) \mid p \in \textsc{prop}$, p occurred in $\mathcal{B}$ and $U = \{[i]_{\leftrightsquigarrow_{\mathcal{B}}} \mid i : p \in \mathcal{B}\}\}$.

Lemma 1. *Suppose that* $\mathcal{B}$ *is an open, fully expanded branch in a* $\mathcal{T}_{\mathcal{H}(@,E,D)}$ *tableau for* φ*. Then if* $i : \psi \in \mathcal{B}$ *then* $\mathfrak{M}(\mathcal{B}), [i]_{\leftrightsquigarrow_{\mathcal{B}}} \models \psi$*.*

Proof. By induction on the complexity of ψ. Since all cases save $\psi = \mathsf{D}\chi$ and $\psi = \neg\mathsf{D}\chi$ are covered by proofs given in [3] and [5], we only consider missing cases.

Case: $\psi = \mathsf{D}\chi$. We have $i : \mathsf{D}\chi \in \mathcal{B}$. After applying (D) we obtain $i : \neg j \in \mathcal{B}$ and $j : \chi \in \mathcal{B}$. By the inductive hypothesis we have that $\mathfrak{M}(\mathcal{B}), [j]_{\leftrightsquigarrow_{\mathcal{B}}} \models \chi$. It suffices to show that $[i]_{\leftrightsquigarrow_{\mathcal{B}}}$ and $[j]_{\leftrightsquigarrow_{\mathcal{B}}}$ are distinct. Suppose that they are the same equivalence class. But then, by Def. 2, $i : j \in \mathcal{B}$, which contradicts the fact that $\mathcal{B}$ is open.

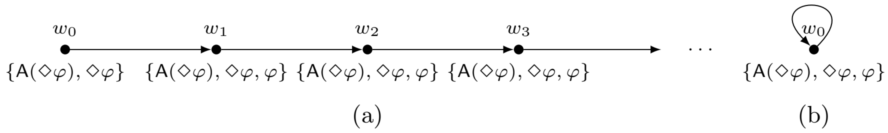


Fig. 2. (a) and (b) present, respectively, an infinite and a finite (minimal) model for the formula $\mathsf{A}(\Diamond\varphi)$. Both of them can be obtained from a tableau if the (ub) rule is involved, since it allows merging worlds in an arbitrary way, provided that the consistency is preserved.

Case: $\psi = \neg\mathsf{D}\chi$. We have $i : \neg\mathsf{D}\chi \in \mathcal{B}$. If no labels different than i occurred on $\mathcal{B}$, it means that $W = \{[i]_{\leftrightsquigarrow_{\mathcal{B}}}\}$ and therefore $\mathfrak{M}(\mathcal{B}), [i]_{\leftrightsquigarrow_{\mathcal{B}}} \models \neg\mathsf{D}\chi$ trivially holds. Suppose that $\mathcal{L}$ is a set of labels different than i, which occurred in $\mathcal{B}$. Pick an arbitrary label j from $\mathcal{L}$. After applying $(\neg\mathsf{D})$ to $\neg\mathsf{D}\chi$ we obtain that either $i : j \in \mathcal{B}$ or $j : \neg\chi \in \mathcal{B}$. If the former is the case, by the inductive hypothesis we obtain that $[i]_{\leftrightsquigarrow_{\mathcal{B}}} = [j]_{\leftrightsquigarrow_{\mathcal{B}}}$. If the latter holds, then by the inductive hypothesis, $\mathfrak{M}(\mathcal{B}), [j]_{\leftrightsquigarrow_{\mathcal{B}}} \models \neg\chi$. Both cases are subsumed by $\mathfrak{M}(\mathcal{B}), [i]_{\leftrightsquigarrow_{\mathcal{B}}} \models \neg\mathsf{D}\chi$. Since j was picked arbitrarily, we obtain the conclusion.

Theorem 3. *$\mathcal{T}_{\mathcal{H}(@,E,D)}$ is complete.*

Proof. By Definition 7 and Lemma 1.

5 Termination of $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$

Exploiting the (ub) rule and the conditions (c1)-(c4) we show that $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$ is terminating for the logic $\mathcal{H}(@,\mathsf{E},\mathsf{D})$, provided that it has the *finite model property* for a certain class of frames.

First, we make a remark that will be useful afterwards (cf. [11]).

Remark 1. For each $[i]_{\leftrightsquigarrow_{\mathcal{B}}}$ the number of applications of the rules introducing a new label, namely $(\Diamond)$, (E), (D), to members of $[i]_{\leftrightsquigarrow_{\mathcal{B}}}$ is finite.

Proof. Indeed, if the (ub) is eagerly applied and the conditions (c2) and (c3) are fulfilled, it ensures that no superfluous application of $(\Diamond)$, (E), (D) is performed, since they are only applied to one member of $[i]_{\leftrightsquigarrow_{\mathcal{B}}}$ and are not applied to accessibility formulas (otherwise it would lead to an infinite derivation that could not be subjected to blocking). Since the input formula φ is assumed to be finite, therefore for each i that occurred in $\mathcal{B}$ $[i]_{\leftrightsquigarrow_{\mathcal{B}}}$ the number of $(\Diamond)$, (E), (D) applications is finite.

Corollary 1. *For each $\mathcal{T}_{\mathcal{H}(@,E,D)}$ tableau branch $\mathcal{B}$ is finite iff W of $\mathfrak{M}(\mathcal{B})$ is finite.*

Now we are ready to state the lemma that is essential for termination of $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$. However, before we do this, we explain informally how the (ub) rule works. Our tableau calculus by default handles all distinct nominals that were introduced to a branch as labelling distinct worlds. It leads to a situation where a satisfiable formula having a simple model generates an infinite tableau (see Fig. 2). The (ub) rule compares all labels that occurred in a branch and its left conclusion merges each pair unless it leads to the inconsistency. As a consequence, if a formula has a model $\mathfrak{M}$ of a certain cardinality, it will be reflected by a finite, fully expanded open branch of a $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$ tableau. The reason is that the left conclusion of the (ub) rule decreases the cardinality of a model whenever possible, so a model of the cardinality not-greater than the cardinality of $\mathfrak{M}$ will eventually be obtainable from one of the branches of a tableau. The formal argument is presented in the following lemma.

Lemma 2. *Suppose that a finite model $\mathfrak{N} = \langle W', R', V' \rangle$ satisfies a formula φ. Then there exists an open branch $\mathcal{B}$ in a $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$ tableau and $\mathfrak{M}(\mathcal{B}) = \langle W, R, V \rangle$ such that $\mathsf{Card}(W) \leq \mathsf{Card}(W')$.*

Proof. We proceed by induction on the number of steps in the derivation. During the derivation we construct a branch $\mathcal{B}$ in such a way that $\mathfrak{M}(\mathcal{B})$ is partially isomorphic to $\mathfrak{N}$ (cf. [11]).

Basic step: φ is satisfiable on $\mathfrak{N}$, so there must exist $w \in W'$ such that $\mathfrak{N}, w \models \varphi$. If also $\mathfrak{N}, w \models i$ such that i does not occur in φ, we put at the initial node of the derivation $i : \varphi$. If no such nominal holds in w, we conservatively extend $\mathfrak{N}$ by adding fresh nominal i to w and put at the initial node of the derivation $i : \varphi$.

Inductive step: Application of each tableau rule should be considered as a separate case. Nevertheless, only four rules seem to be essential for this proof, namely $(\Diamond)$, (E), (D) and (ub), i.e. rules that either introduce a new label to a branch or identify labels already present on a branch. We consider each of them.

Case: $(\Diamond)$. Suppose that a formula $\Diamond\psi$ occurred at the nth node of the derivation. It means that we associated the label i of this node with a world in W' that satisfies $\Diamond\psi$ and i. What follows, there must exist a world v such that wRv and $\mathfrak{N}, v \models \psi$. If v does not satisfy any nominal l that has not yet occurred on the branch either as a label or as a subformula, we conservatively extend $\mathfrak{N}$ by ascribing l to v. Applying $(\Diamond)$ to $\Diamond\psi$ we obtain $i : \Diamond j$ and $j : \psi$. We put l in place of j.

Case: (E). Suppose that a formula $\mathsf{E}\psi$ occurred at the nth node of the derivation. It means that we associated the label i of this node with a world in W' that satisfies $\mathsf{E}\psi$ and i. Therefore, there exists a world v such that $\mathfrak{N}, v \models \psi$. If v does not satisfy any nominal l that has not yet occurred on a branch either as a label or as a subformula, we conservatively extend $\mathfrak{N}$ by ascribing l to v. Applying (E) to $\mathsf{E}\psi$ we obtain $j : \psi$. We put l in place of j.

Case: (D). Suppose that a formula $\mathsf{D}\psi$ occurred at the nth node of the derivation. It means that we affixed the label i of this node to a world in W' that satisfies $\mathsf{D}\psi$ and i. Therefore, there exists a world v such that $\mathfrak{N}, v \models \psi \land \neg i$. If v does not satisfy any nominal l that has not yet occurred on a branch either as a label or as a subformula, we conservatively extend $\mathfrak{N}$ by ascribing l to v. Applying (D). to $\mathsf{D}\psi$ we obtain $j : \neg i$ and $j : \psi$. We put l in place of j.

Case: (ub). Suppose that during the derivation two labels i and j have been introduced to $\mathcal{B}$. By the inductive hypothesis we mapped these labels to worlds w and v of (the conservative extension of) W'. Either the world w satisfies $i \land j$ (which would mean that w and v are the same world) or it satisfies $i \land \neg j$ (which indicates that w and v are distinct). If the former is the case, we pick the left conclusion of (ub) and add it to $\mathcal{B}$, if the latter is the case, we choose the right conclusion of (ub) and add it to $\mathcal{B}$.

Since **B** is open, we can construct a model $\mathfrak{M}(\mathcal{B}) = \langle W, R, V \rangle$ out of it. Now we show that $\mathsf{Card}(W) \leq \mathsf{Card}(W')$ (we consider $\mathfrak{N}$ as already conservatively extended in progress of constructing $\mathcal{B}$). We set a function $f : W' \longrightarrow W$ as follows

$$f(w) = \begin{cases} [i]_{\leftrightsquigarrow_{\mathcal{B}}}, & \text{if there is } i : i \in \mathcal{B} \text{ such that } i \text{ was affixed} \\ & \text{to } w \text{ during the derivation} \\ \text{arbitrary element of } W, & \text{otherwise} \end{cases}$$

f is injective and if we cut it to these elements of W' to which we assigned a nominal during the derivation, it is also an isomorphism. That concludes the proof.

To conclude our considerations it is sufficient to prove that the logic $\mathcal{H}(@, \mathsf{E}, \mathsf{D})$ has the finite model property. The following proposition deals with it.

Proposition 2. *The logic* $\mathcal{H}(@, E, D)$ *has the* effective finite model property *with the bounding function* $\mu = 2^{\mathsf{Card}(\mathsf{Sub}(\varphi))+1}$*, where* $\mathsf{Sub}(\varphi)$ *is a set of all subformulas of a formula* ϕ*.*

Proof. We use the standard, filtration-based argument. Suppose that a formula φ is satisfied on a (possibly infinite) model $\mathfrak{M} = \langle W, R, V \rangle$. It means that there exists $w \in W$ such that $\mathfrak{M}, w \models \varphi$.We show that there exists a finite model $\mathfrak{M}'$ that satisfies φ and whose cardinality does not exceed $2^{\mathsf{Card}(\mathsf{Sub}(\varphi))+1}$.

First, we set the relation $\leftrightsquigarrow_\varphi$ on W in the following way:

$$w \leftrightsquigarrow_\varphi v \qquad \text{iff} \qquad \text{for all } \psi \in \mathsf{Sub}(\varphi)\ \mathfrak{M}, w \models \psi \text{ iff } \mathfrak{M}, v \models \psi.$$

It is straightforward that $\leftrightsquigarrow_\varphi$ is the equivalence relation

Now we are ready to construct our finite model that will satisfy ϕ. Let $\mathfrak{M}' = \langle W', R', V' \rangle$ such that:

$W' = W/_{\leftrightsquigarrow_\varphi} \uplus W/_{\leftrightsquigarrow_\varphi}$;
$R' = \{(|v|_{\leftrightsquigarrow_\varphi}, |u|_{\leftrightsquigarrow_\varphi}) : R(v, u)\}$;
$V'(p) = \{|v|_{\leftrightsquigarrow_\varphi} : v \in V(p)\}$ for all proposition letters in φ;
$V'(i) = \{|v|_{\leftrightsquigarrow_\varphi} : v \in V(i)\}$ for all nominals in φ.

We prove that $\mathfrak{M}'$ satisfies φ by induction on the complexity of subformulas of φ. Since the proof for the modal part of $\mathcal{H}(@, \mathsf{E}, \mathsf{D})$ is well known (cf. [4]) and the case of $\psi = i$ follows immediately from the definition of V', we confine ourselves to proving the cases of $@_i\chi$, $\mathsf{E}\chi$ and $\mathsf{D}\chi$.

Case: $\psi = @_i\chi$. Suppose that $\mathfrak{M}, v \models @_i\chi$. It means that χ holds at a world u at which also i holds. This world is transformed to a singleton equivalence class $\{u\}$ in W'. By the inductive hypothesis it follows that $\mathfrak{M}', \{u\} \models i$ and $\mathfrak{M}', \{u\} \models \chi$. Hence $\mathfrak{M}', |v| \models @_i\chi$.

Case: $\psi = \mathsf{E}\chi$. Suppose that $\mathfrak{M}, v \models \mathsf{E}\chi$. It means that there exists a world u at which χ holds. By the inductive hypothesis $\mathfrak{M}', |u| \models \chi$. Hence $\mathfrak{M}', |v| \models \mathsf{E}\chi$.

Case: $\psi = \mathsf{D}\chi$. Suppose that $\mathfrak{M}, v \models \mathsf{D}\chi$. It means that there exists a world u different than v, at which χ holds. By the inductive hypothesis $\mathfrak{M}', |u| \models \chi$. Two complementary cases might occur. If $|v| \neq |u|$, then we obtain $\mathfrak{M}', |v| \models \chi$. If, however, $|v| = |u|$, it means that χ is also satisfied by a copy of $|v|$ that we pasted to W' at the stage of the construction of $\mathfrak{M}'$. Since $|v|$ and its copy are distinct, we obtain $\mathfrak{M}', |v| \models \chi$.

Observe that pasting a distinct copy of $W/_{\leftrightsquigarrow_\varphi}$ to W' is only necessary if D is involved. Therefore, in other cases the bounding function $\mu = 2^{\mathsf{Card}(\mathsf{Sub}(\varphi))}$.

Consequently, we obtain the following result:

Theorem 4. $\mathcal{T}_{\mathcal{H}(@,E,D)}$ *is terminating.*

Proof. Follows from Corollary 1, Lemma 2 and Proposition 2.

Obviously, the bounding function μ from Proposition 2 can be reduced (cf. [1, 9]), however, the main aim of this paper is not optimising the complexity of $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$. Besides, the filtration-based argument can be easily adapted for different types of frames. Thus, we formulate the following strategy-condition for performing the derivation in $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$:

(tm) Expand a branch of $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$-tableau until the number of equivalence classes of individuals in $\mathcal{B}$ exceeds the bound given by μ function. Then stop.

It turns our tableau calculus into a deterministic decision procedure.

6 Concluding remarks

In this paper we presented an internalised tableau-based decision procedure for the logic $\mathcal{H}(@,\mathsf{E},\mathsf{D})$. Tableau calculus $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$ was proven to be sound, complete and terminating. In the existing literature of the subject several approaches to systematic treatment of decision procedures for hybrid logics can be found. We recall two of them. In [3] and [5] Blackburn, Bolander and Braüner provide a terminating internalised tableau-based decision procedure for the logic $\mathcal{H}(@,\mathsf{E})$. However, their main concern is different from ours. Their attempts are focused on tailoring a suitable tableau calculus for each logic separately. Therefore, they introduce two different blocking mechanisms, namely *subset blocking* and *equality blocking* for the logics $\mathcal{H}(@)$ and $\mathcal{H}(@,\mathsf{E})$ and modify the notion of *urfather* subject to a particular logic. The resulting calculus is conceptually complex but seems to avoid any superfluous performances of the rules. In [7] Götzmann, Kaminski and Smolka describe *Spartacus*, which is a tableau prover for hybrid logics with @ operators and universal modality. Thanks to the application of advanced blocking and optimisation techniques, namely *pattern-based blocking* and *lazy branching* the system is very efficient in terms of complexity.

The decision procedure introduced in this paper presents the approach which is different from the aforementioned. It introduces (ub) as an explicit tableau rule which is sound and, together with the conditions (c1)-(c4), ensures termination of the whole calculus. (ub) allows a direct comparison of every pair of labels that occurred on a branch and, therefore, subsumes any other blocking mechanisms. (ub) is a generic rule which means that it generates every possible configuration of labels occurring on a branch. In comparison to [3] and [7] many of these configurations are superfluous. However, the huge advantage of this approach is conceptual simplicity which allows to avoid introducing complicated strategies of searching for a pair of labels that are liable to blocking mechanism. Additionally, for each satisfiable formula φ (ub) ensures that a minimal model for φ (in terms of a domain size) will be generated, which cannot be guaranteed by the systems of [3] and [7]. Moreover, $\mathcal{T}_{\mathcal{H}(@,\mathsf{E},\mathsf{D})}$ provides a uniform approach to all hybrid logics mentioned in the paper and covers the case of difference modality which is omitted in [3]. A possible direction of future work may be investigating whether the applications of the (ub) rule can be optimised.

Acknowledgements

The research reported in this paper is a part of the project financed from the funds supplied by the National Science Centre, Poland (decision no. DEC-2011/01/N/HS1/01979).

References

1. Areces C., Blackburn P., Marx M.: A road-map on complexity for hybrid logics. Proc. of the 8th Annual Conference of the European Association for Computer Science Logic (CSL 1999), Madrid, Spain, 307–321
2. Blackburn P.: Internalizing labelled deduction. J. Log. Comput. 10(1), 137—168 (2000)
3. Blackburn P., Bolander T.: Termination for Hybrid Tableaus. J. Log. Comput. 17(3), 517–554 (2007)
4. Blackburn P., Venema Y., de Rijke M.: Modal Logics. Cambridge University Press (2002)
5. Bolander T., Braüner T.: Tableau-based Decision Procedures for Hybrid Logic. J. Log. Comput. 16, 737–763 (2006)
6. Braüner T.: Hybrid Logic and its Proof-Theory Springer (2011)
7. Götzmann D., Kaminski M., Smolka G.: Spartacus: A Tableau Prover for Hybrid Logic. Electron. Notes Ther. Comput. Sci. 262, 127-139 (2010)
8. Kaminski M., Smolka G.: Terminating Tableau Systems for Hybrid Logic with Difference and Converse. J. Log. Lang. Inf. 18, 437–464 (2009)
9. Mundhenk M., Schneider T.: Undecidability of Multi-modal Hybrid Logics. Electron. Notes Ther. Comput. Sci. 174, 29–43 (2007)
10. Schmidt, R. A., Tishkovsky, D.: Automated Synthesis of Tableau Calculi. Log. Meth. Comput. Sci. 7(2), 1–32 (2011)
11. Schmidt R.A., Tishkovsky D.: Using Tableau to Decide Description Logics with Full Role Negation and Identity. http://www.mettel-prover.org/papers/ALBOid.pdf (2012)
12. Tzakova M.: Tableau Calculi for Hybrid Logics. LNCS 1617, 278–292 (1999)

A Frame-Based Semantics of Locative Alternation in LTAG

Yulia Zinova

Heinrich Heine University Düsseldorf
zinova@phil.uni-duesseldorf.de

Abstract. In this paper I present the analysis of locative alternation phenomena in Russian and English. This analysis follows the approach proposed in [13] and uses LTAG and Frame Semantics. The combination of a syntactic theory with an extended domain of locality and frames provides a powerful mechanism for argument linking. Metagrammar factorization allows to determine not only lexical, but also constructional meaning that is essential for locative alternation analysis.

1 Introduction

There is a number of formalisms that capture the idea that the meaning of a verb-based construction depends both on the lexical meaning of the verb and on the construction in which the verb is used ([8], [19]). The question is how exactly the components of the meaning are distributed and how they combine.

In [13] a combination of Lexicalized Tree Adjoining Grammars ([9]) and Frame Semantics is introduced. It is shown that the resulting framework is very flexible with respect to the factorisation and combination of lexical and constructional units on the syntax and semantics level and is also suitable for computational processing.

Though there already exist a number of different approaches to semantic construction using LTAG ([10], [5], [12]) and approaches that combine other syntactic formalisms with Frame Semantics ([3], [4]), the novel combination of an LTAG and Frame Semantics benefits from both extended domain of locality and underspecification allowed by frames.

In this paper I want to present the analysis of locative alternation that benefits of flexibility offered by the novel framework.

2 Tree Adjoining Grammar

Tree Adjoining Grammar (TAG, [9]) is a tree-rewriting grammar formalism. It consists of a finite set of *elementary trees* with labelled nodes with two operations on them: *substitution* and *adjunction*. All elementary trees are either *auxiliary trees* or *initial trees*. An *auxiliary tree* is a tree which has exactly one *foot node* - a leaf that is marked with an asterisk. Leaf nodes can be labelled with terminals and other nodes are labelled only with non-terminals. The derivation process starts from an initial tree and in the final *derived tree* all the leaves must be labelled by terminals.

Substitution allows to replace a non-terminal leaf with a new tree and *adjunction* is used for replacing an internal node with an auxiliary tree. Adjunction to the node

R.K. Rendsvig and S. Katrenko (Eds.): *ESSLLI 2012 Student Session Proceedings*, CEUR Workshop Proceedings vol.: see http://ceur-ws.org/, 2012, pp. 202–210.

labelled X is allowed if the root and foot nodes of the adjoining auxiliary tree have the same label X. It is also possible to indicate nodes where adjunction is obligatory or not allowed with *OA* and *NA* subscripts respectively.
Figure 1 shows an example of a derivation: the initial tree for *Mary* substitutes into the subject slot of the elementary tree for *laughs*, and the *sometimes* auxiliary tree for the VP modifier adjoins to the VP node. *Feature-structure based TAG*, or FTAG, is a

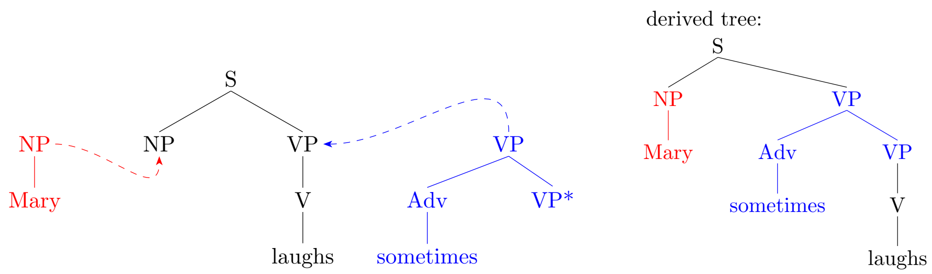


Fig. 1. Example of a TAG derivation

variant of TAG in which elementary trees are enriched with feature structures ([20]). In an FTAG each node has a top feature structure and all the nodes except substitution nodes have a bottom feature structure. Feature unification happens during the derivation process when adjunction and substitution take place. Due to the extended domain of locality, nodes within one elementary tree can share features, allowing to express constraints among dependent nodes easily.
For natural languages a specific version of TAG called *lexicalized TAG,* or LTAG is used. In an LTAG, each elementary tree must have at least one non-empty lexical item, called *lexical anchor*. The second important principle for a natural language TAG is that every elementary tree where the lexical anchor is a predicate must contain slots (leaves with non-terminal labels) for all arguments of this predicate, including the subject, and for nothing else (*theta-criterion for TAG*, [6]).
The facts that LTAGs have extended domains of locality and that elementary trees are lexicalized and contain slots for all the predicate's arguments, make them good candidates for combination with frame-based compositional semantics ([13]). In the approach proposed in [13], a single semantic representation (a semantic frame in this case) is linked to the entire elementary tree. When coupling an elementary tree with a semantic frame, syntactic arguments can be directly linked to their counterpart in the semantics. Described approach is similar to ones in [7] and [14], but uses different kind of semantic representation. Semantic composition is then modeled by unification which is a result of performing adjunction and substitution. Figure 2 provides a simple illustration of syntactic and semantic composition. In this example, substitutions trigger unifications between $\boxed{1}$ and $\boxed{3}$ and $\boxed{2}$ and $\boxed{4}$ which leads to correct insertion of argument frames into the frame of *loves*.
Linguistic generalizations in TAGs are captured by a metagrammar. There are two steps of factorization, which are important for this paper:

- *unanchored elementary trees* are specified separately from lexical anchors;

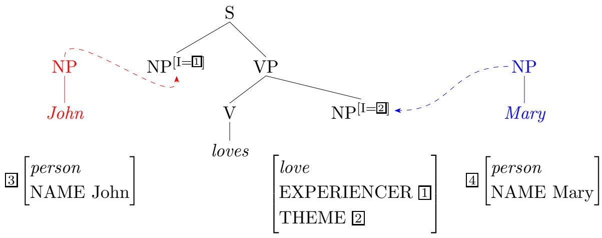


Fig. 2. Syntactic and semantic composition for *John loves Mary*

- trees are organized into *tree families* which represent different realizations of one subcategorization frame.

This allows to define a meaning for sets of unanchored elementary trees, i.e., a meaning of constructions.

3 The Data

3.1 Previous approaches

(1) and (2) show basic examples of locative alternation in English and Russian. Despite the fact that in English both constructions have a PP and it can be omitted without losing the specific construction meaning, let us call the first variant ((1a), (2a)) *prepositional phrase construction*, or *PPC* and the second variant ((1b), (2b)) - *instrumental case construction*, or *ICC* for convenience of referring to them.

(1) a. John$_{\boxed{1}}$ loaded the hay$_{\boxed{2}}$ into the truck$_{\boxed{3}}$. (PPC)
b. John$_{\boxed{1}}$ loaded the truck$_{\boxed{3}}$ with hay$_{\boxed{2}}$. (ICC)

(2) a. Ivan$_{\boxed{1}}$ zagruzil seno$_{\boxed{2}}$ v vagon$_{\boxed{3}}$.
Ivan loaded hay$_{acc,def}$ in wagon$_{gen,indef/def}$.
Ivan loaded the hay into a/the wagon.
b. Ivan$_{\boxed{1}}$ zagruzil vagon$_{\boxed{3}}$ senom$_{\boxed{2}}$.
Ivan loaded wagon$_{acc,def}$ hay$_{instr,indef}$.
Ivan loaded the wagon with hay.

PPCs are traditionally analyzed as having a change of location meaning and ICCs - as having a change of state meaning ([11], [16], [8]). An analysis for (1) following [11] is provided in (3). It demonstrates that there is a difference between the two constructions, but only the difference in the perspective is shown.

(3) a. X CAUSE [BECOME [hay BE ON truck]]
b. X CAUSE [BECOME [truck$_{\boxed{z}}$ BE [WITH [hay BE ON z]]]]

The analysis proposed in [16], which can be found under (4), provides more detailed information about the difference between PPCs and ICCs. (4a) tells us that the hay changes its location as a result of the loading event, while (4b) describes that the result is a change in the state of the wagon. One can notice that in (3) there is no explicit reference to the verb itself and the only component that is taken from the verb meaning is that the result of the loading is that the THEME is on the LOCATION in the end.

(4) a. [[x ACT] CAUSE [y BECOME P_{loc} z] [LOAD]$_{MANNER}$]
b. [[x ACT] CAUSE [z BECOME []$_{STATE}$ WITH-RESPECT-TO y] [LOAD]$_{MANNER}$]

Using frame semantics, one can assign two frames in 3 to the two different constructions. For the PPC, one has to remove the concrete verb *load* and replace it with *change_of_location* effect. So the first frame tells us that the activity of the Actor (X) causes the Theme (Y) to change its location to the Goal (Z). For the ICC's frame in order to introduce the Manner one can simply embed the caused change of location frame under the MANNER atribute. The second frame in this case would mean that the activity of the Actor (X) causes the Theme (Z) to change its state by means of changing the location of the third argument (Y) to Z.

$$\boxed{0}\begin{bmatrix} \textit{causation} & \\ \text{CAUSE} & \boxed{4}\begin{bmatrix} \textit{activity} \\ \text{ACTOR } \boxed{1} \end{bmatrix} \\ \text{EFFECT} & \boxed{5}\begin{bmatrix} \textit{change-of-loc} & \\ \text{THEME} & \boxed{2} \\ \text{GOAL} & \boxed{3} \end{bmatrix} \end{bmatrix} \quad \boxed{0}\begin{bmatrix} \textit{causation} & \\ \text{CAUSE} & \boxed{4}\begin{bmatrix} \textit{activity} \\ \text{ACTOR } \boxed{1} \end{bmatrix} \\ \text{EFFECT} & \boxed{5}\begin{bmatrix} \textit{change\_of\_state} & \\ \text{THEME } \boxed{3} & \\ \text{MANNER } \boxed{6} & \begin{bmatrix} \textit{causation} & \\ \text{CAUSE} & \boxed{7}\begin{bmatrix} \textit{activity} \\ \text{ACTOR } \boxed{1} \end{bmatrix} \\ \text{EFFECT} & \boxed{8}\begin{bmatrix} \textit{change\_of\_loc} \\ \text{THEME } \boxed{2} \\ \text{GOAL } \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Fig. 3. Frame semantics for PPC and ICC

3.2 Detailed Russian and English Data

The question that arises if one looks carefully at what the sentences in (1) and (2) mean is whether it is really the case that there is no change of state in PPC examples? In fact, any loading activity leads to both a change of location of the content and some change of state of the container (if it is specified), and the difference between two constructions is that

- different components of the effect become more salient;
- in the case of ICC initial and result states of the container are specified.

In order to understand how the meaning of verbs and constructions should be represented let us look at the whole range of the verbs allowing locative alternation that one can find in English and Russian. [18] provides the following classification for English:

Content-oriented classes:

(a) simultaneous forceful contact and motion of a mass against a surface (brush, drape, spread, etc.);
(b) vertical arrangement on a horizontal surface (heap, pile, stack);
(c) force is imparted to a mass, causing ballistic motion in a specified spatial distribution along a trajectory (inject, splash, spray, etc.);
(d) mass is caused to move in a widespread or non directed distribution (scatter, seed, sow, etc.).

Container-oriented classes:

(e) a mass is forced into a container against the limit of its capacity (crowd, jam, stuff, etc.);
(f) a mass of size, shape, or type defined by the intended use of a contained is put into the container, enabling it to accomplish its function (load, pack, stock).

From the description of verb classes that allow locative alternation in English one can see that the result state of the container in case of ICC is such that the action cannot be performed any longer. There is no result state common for all the cases, so it depends on the verb, i.e. on how the change of location happens. The easiest way to solve this would be to assume different construction meanings for different verb classes (e.g. one with the Effect of the Theme being full and the other one with the Effect of the Theme being covered), but let us first look at some Russian data.
In Russian a lot of verbs allow only one of the constructions, i.e. a change of construction requires a change of verb prefix (a list can be found in [1]). However, some of the verbs from the list remain the same in both prepositional and instrumental constructions. Such verbs can be organised in three groups: the first one is similar to the (f) group in English (see example (2)), the second one is similar to group (a) in English, like in (5), and the third class is like a combination of the first and the second: a mass is put into a container, enabling it to accomplish its function, or on a container, covering its surface (6).
With the vebs from the third group an interesting effect can be observed: while in the case of PPC example (6a) there is a preposition which tells us that the content goes *in* the container, in the case of ICC example (6b) two different readings are possible: the content can be put in the container or the content can cover the container. In both cases there is a clear result state: either the container is full or the container's surface is fully covered with content. This means that the verb *zasypat'* ('to fill/to cover') does not provide information about how the THEME is positioned at the GOAL. In case of PPC this information comes from the preposition used (both *v* ('in') and *na* ('on') are possible) and in ICC the ambiguity can be resolved only using world knowledge. So (6) demonstrates conclusively that there should be one construction accounting for different result states of the theme and allowing to get different interpretations of one verb due to underspecification of how the change of location process goes.

(5) a. On namazal maslo na hleb.
He distributed butter$_{acc,def}$ on bread$_{acc,indef}$
He distributed butter over a piece of bread.
b. On namazal hleb maslom.
He covered bread$_{acc,def}$ butter$_{instr,indef}$
He covered a piece of bread with butter.

(6) a. On zasypal sahar v banku.
He put suggar$_{acc,def}$ in can$_{acc,indef/def}$

He put sugar in a/the tin.

b. On zasypal banku saharom.
He covered/filled $\text{tin}_{acc,def}$ $\text{sugar}_{instr,indef}$
He covered/filled the tin with sugar.

4 Locative Alternation: The Analysis

4.1 Syntactic representation

In the previous section we were looking only at "full" examples, where both container and content are present. However, the constructions that are being discussed can be used when only the direct object of the verb is present; in this case, they will have the same difference in semantics. Using LTAG and metagrammar decomposition one can obtain the tree family in 4 for the PPC and tree family in 5 for the ICC (the second NP_{INSTR} stands for both NP in instrumental case in Russian and PP with preposition "with" in English).

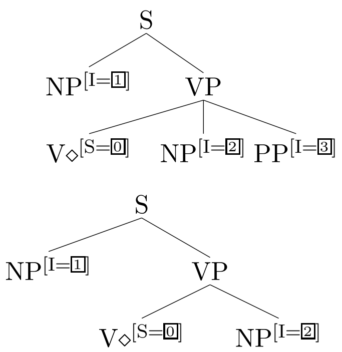


Fig. 4. Unanchored trees for the PPC

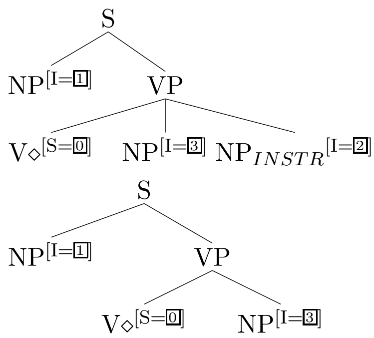


Fig. 5. Unanchored trees for the ICC

4.2 Scales and Proposed Frame Semantics

In the case of the PPC, the semantics of the whole phrase can be compositionaly derived from the semantics of the verb and its arguments, while in the case of the ICC there is a part of the meaning, that comes from the construction itself. The goal now is to provide such semantics for the ICC and verbs allowing locative alternation such that in combination they form the desired frame representation of the semantics of a sentence.

Following ideas in [17] where one can find a discussion of the representation of attributes, events, and results while implementing Fillmore's Frame Semantics ([2]) I introduce attributes of initial and result state and a scale which is determined by its type, start and end points. The change of state is either a decrease or an increase of the value on an ordered scale (discussion of analysis of scalar change can be found in [15]). The direction is given by the values of atributes ENDP and STARTP (end and start points), which replaces the LESSER atribute of ordering proposed in [17]. Some of the verbs specify a concrete initial or result state (INIT and RESULT respectively), but *load* does not have any initial or result state specified within its semantics, so it just determines the scale and two values on it. Summarizing the ideas, one obtains the following scheme:

- in the verb both change of location and change of state effects are specified;
- MANNER attribute is not needed because it is already described in the change of location subframe;
- change of state is described by scale, initial state, and result state;
- SCALE atribute can have a type such as "degree of fullness" that is a subtype of the type "scale" and thus replaces it during the unification;
- initial and result states are values on the scale;
- the ICC specifies that initial and result states are equal to the start and end points of the scale, respectively.

$$\boxed{0}\begin{bmatrix} \textit{causation} & \\ \text{EFFECT} & \begin{bmatrix} \textit{scalar\_change\_of\_state} & \\ \text{THEME} & \boxed{3} \\ \text{SCALE} & \boxed{7}\begin{bmatrix} \textit{scale} & \\ \text{STARTP} & \boxed{5} \\ \text{ENDP} & \boxed{6} \end{bmatrix} \\ \text{INIT} & \begin{bmatrix} \textit{state} & \\ \text{ENTITY} & \boxed{3} \\ \text{SCALE} & \boxed{7} \\ \text{VALUE} & \boxed{5} \end{bmatrix} \\ \text{RESULT} & \begin{bmatrix} \textit{state} & \\ \text{ENTITY} & \boxed{3} \\ \text{SCALE} & \boxed{7} \\ \text{VALUE} & \boxed{6} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Fig. 6. Frame for the ICC

$$\boxed{0}\begin{bmatrix} \textit{causation} & \\ \text{ACTOR}\ \boxed{1} & \\ \text{THEME}\ \boxed{2} & \\ \text{GOAL}\ \boxed{3} & \\ \text{CAUSE} & \begin{bmatrix} \textit{activity} \\ \text{ACTOR}\ \boxed{1} \end{bmatrix} \\ \text{EFFECT} & \left\{ \begin{bmatrix} \textit{change\_of\_location} & \\ \text{THEME} & \boxed{2} \\ \text{GOAL} & \boxed{3} \end{bmatrix}, \begin{bmatrix} \textit{scalar\_change\_of\_state} & \\ \text{THEME} & \boxed{3} \\ \text{SCALE} & \boxed{7}\begin{bmatrix} \textit{degree\_of\_fullness} \\ \text{STARTP EMPTY} \\ \text{ENDP FULL} \end{bmatrix} \\ \text{INIT} & \begin{bmatrix} \textit{state} & \\ \text{ENTITY} & \boxed{3} \\ \text{SCALE} & \boxed{7} \\ \text{VALUE} & \boxed{8} \end{bmatrix} \\ \text{RESULT} & \begin{bmatrix} \textit{state} & \\ \text{ENTITY} & \boxed{3} \\ \text{SCALE} & \boxed{7} \\ \text{VALUE} & \boxed{9} \end{bmatrix} \end{bmatrix} \right\} \end{bmatrix}$$

Fig. 7. Frame representation of the verb *load*

Figure 7 shows a lexical frame for the verb *load*. As one can see, when all the arguments are filled, the right meaning for the whole PPC follows automatically. The semantics of the ICC is a caused change of state meaning that gets further constrained when a specific lexical anchor is inserted. Figure 6 shows how the unanchored tree for the ICC is linked to its semantic frame. The correct argument linking happens because I features in the syntactic tree and the thematic roles in the semantic frame are identical. This is done in a local way (within the domain of an elementary tree) because of LTAG's extanded domain of locality. The S feature of the V node serves for unification of the lexical frame for the verb and the constructional frame. When a lexical anchor is inserted, this feature unifies with the S feature of the lexical item. The result of this unification for the ICC with a lexical anchor *load* is shown in Fig. 8.

$$
\boxed{0}\begin{bmatrix}
\multicolumn{2}{l}{\textit{causation}}\\
\text{ACTOR} & \boxed{1}\\
\text{THEME} & \boxed{2}\\
\text{GOAL} & \boxed{3}\\
\text{CAUSE} & \begin{bmatrix}\textit{activity}\\ \text{ACTOR}\ \boxed{1}\end{bmatrix}\\
\text{EFFECT} & \left\{ \begin{bmatrix}\multicolumn{2}{l}{\textit{change\_of\_location}}\\ \text{THEME} & \boxed{2}\\ \text{GOAL} & \boxed{3}\end{bmatrix}, \begin{bmatrix}\multicolumn{2}{l}{\textit{scalar\_change\_of\_state}}\\ \text{THEME} & \boxed{3}\\ \text{SCALE} & \boxed{7}\begin{bmatrix}\textit{degree\_of\_fullness}\\ \text{STARTP EMPTY}\\ \text{ENDP FULL}\end{bmatrix}\\ \text{INIT} & \begin{bmatrix}\multicolumn{2}{l}{\textit{state}}\\ \text{ENTITY} & \boxed{3}\\ \text{SCALE} & \boxed{7}\\ \text{VALUE} & \text{empty}\end{bmatrix}\\ \text{RESULT} & \begin{bmatrix}\multicolumn{2}{l}{\textit{state}}\\ \text{ENTITY} & \boxed{3}\\ \text{SCALE} & \boxed{7}\\ \text{VALUE} & \text{full}\end{bmatrix}\end{bmatrix} \right\}
\end{bmatrix}
$$

Fig. 8. Resulting frame for the ICC and the verb *load*

5 Conclusion

This is a case study for combining an LTAG with Frame Semantics, in which I have described a model for locative alternation in English and Russian. This analysis uses LTAG's mechanism of separation between unanchored elementary trees and lexical anchors to separate the contribution of the lexical meaning from the contribution of construction and follows the ideas expressed in [13]. Another advantage of combination of an LTAG with Frame Semantics is that LTAG's extended domain of locality allows direct linking of thematic roles of the arguments with corresponding syntatctic slots. As this framework is a new one, there are a lot of open questions and a wider range of semantic phenomena should be examined.

References

1. Dudchuk, P.: Locativnaja alternacija i struktura glagol'noj gruppy [Locative Alternation and the structure of VP]. diploma thesis, Moscow State University (2006)
2. Fillmore, C.J.: Frame semantics. In: of Korea, T.L.S. (ed.) Linguistics in the Morning Calm, pp. 111–137. Hanshin Publishing Co., Seoul (1982)
3. Frank, A.: Generalizations over corpus-induced frame assignment rules. In: Proceedings of the LREC 2004 Workshop on (2004)
4. Frank, A.: Question answering from structured knowledge sources. Journal of Applied Logic 5(1), 20 – 48 (2007)

5. Frank, A., van Genabith, J.: GlueTag. Linear logic based semantics for LTAG – and what it teaches us about LFG and LTAG. In: Butt, M., King, T.H. (eds.) Proceedings of the LFG01 Conference. Hong Kong (2001)
6. Frank, R.: Syntactic Locality and Tree Adjoining Grammar: Grammatical, Acquisition and Processing Perspectives. Ph.D. thesis, University of Pennsylvania (1992)
7. Gardent, C., Kallmeyer, L.: Semantic Construction in FTAG. In: Proceedings of EACL 2003. pp. 123–130. Budapest (2003)
8. Goldberg, A.E.: Constructions. A Construction Grammar Approach to Argument Structure. Cognitive Theory of Language and Culture, The University of Chicago Press, Chicago and London (1995)
9. Joshi, A.K., Schabes, Y.: Tree-Adjoning Grammars. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, pp. 69–123. Springer, Berlin (1997)
10. Joshi, A.K., Vijay-Shanker, K.: Compositional semantics with lexicalized Tree-Adjoining Grammar (LTAG): How much underspecification is necessary? In: Blunt, H.C., Thijsse, E.G.C. (eds.) Proceedings of the Third International Workshop on Computational Semantics (IWCS-3). pp. 131–145. Tilburg (1999)
11. Kageyama, T.: Denominal verbs and relative salience in lexical conceptual structure. In: Kageyama, T. (ed.) Verb semantics and syntactic structure, pp. 45–96. Kurioso Publishers, Tokyo (1997)
12. Kallmeyer, L., Joshi, A.K.: Factoring Predicate Argument and Scope Semantics: Underspecified Semantics with LTAG. Research on Language and Computation 1(1–2), 3–58 (2003)
13. Kallmeyer, L., Osswald, R.: A frame-based semantics of the dative alternation in lexicalized tree adjoining grammars (2012), submitted to Empirical Issues in Syntax and Semantics 9
14. Kallmeyer, L., Romero, M.: Scope and situation binding in LTAG using semantic unification. Research on Language and Computation 6(1), 3–52 (2008)
15. Kennedy, C., Levin, B.: Measure of change: The adjectival core of degree achievements. In: McNally, L., Kennedy, C. (eds.) Adjectives and adverbs. syntax, semantics, and discourse. Oxford University Press, Oxford (2008)
16. Levin, B., Rappaport Hovav, M.: Morphology and lexical semantics. In: Spencer, A., Zwicky, A.M. (eds.) Handbook of morphology, pp. 248–271. Blackwell Publishers, Oxford (1998)
17. Osswald, R., Van Valin, Jr., R.: Framenet, frame structure, and the syntax-semantics interface (2012), `http://www.phil-fak.uni-duesseldorf.de/fileadmin/Redaktion/Institute/Allgemeine_Sprachwissenschaft/Dokumente/Rainer_Osswald/FrameStructure-Osswald-VanValin-2012-02-20.pdf`, manuskript Heinrich-Heine Universität
18. Pinker, S.: Learnability and cognition: The aquisition of argument structure. MIT Press, Cambridge, MA (1989)
19. Van Valin, R.D., LaPolla, R.J.: Syntax. Cambridge University Press, Cambridge (1997)
20. Vijay-Shanker, K., Joshi, A.K.: Feature structures based tree adjoining grammar. In: Proceedings of COLING. pp. 714–719. Budapest (1988)