

MỞ RỘNG MÔI TRƯỜNG ĐỊNH LÝ

WOLFGANG MAY `<may@informatik.uni-freiburg.de>`
ANDREAS SCHLECHTE `<ntheorem@andreas-schlechte.de>`

BIÊN DỊCH: KYANH `<kyanh@o2.pl>`

Bản dịch số **873** (2005/12/8)

cho **ntheorem** bản **1.24** (2004/09/20)

Tóm tắt nội dung

`ntherem.sty` là gói thể hiện các môi trường (tựa) định lý. Bên cạnh các tính năng giúp thay đổi cách thể hiện môi trường (tựa) định lý, gói còn giúp giải quyết vài vấn đề liên quan: đặt dấu kết thúc (`endmarks`), tạo bảng liệt kê các định lý.

Trái với các cách tiếp cận trước đây, gói giải quyết vấn đề đặt dấu kết thúc (`endmarks`) cho các môi trường tựa định lý (`theorem`, `definition`, `example`, `proof`) một cách tự động, chính xác, ngay cả đối với môi trường kết thúc bởi môi trường `displaymath` hoặc môi trường danh sách (thậm chí các môi trường này có thể lồng nhau – `nested`); nhờ đó giải quyết được hoàn toàn các trục trặc khi dùng gói `amsmath`. Nguyên lý làm việc của gói giống như cách `LATEX` điều khiển việc đặt nhãn, bằng cách sử dụng các tập tin `.aux`.

Gói cung cấp lệnh để tạo danh sách các môi trường tựa định lý, tương tự như khi liệt kê các hình vẽ bằng `\listoffigures`.

Sau khi biên dịch tài liệu vài lần (số lần tùy thuộc vào sự phức tạp của các tham khảo chéo; thường thì ba lần là đủ), các dấu kết thúc (`endmarks`) sẽ được đặt đúng chỗ, và danh sách các định lý sẽ được tạo ra.

Do gói `ntheorem.sty` sử dụng lệnh `\newtheorem` của `LATEX` chuẩn, các tài liệu cũ có thể chuyển qua dùng gói mà không cần thay đổi nội dung. Ngoài ra, gói còn tương thích với các tài liệu dùng gói `theorem.sty` của Frank Mitterbach.

License & Copyright Information

This is part of 'ntheorem-doc-vn' bundle.

Copyright (C) 2005 kyanh <kyanh at o2 dot pl>

This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version.

The latest version of this license is in
<http://www.latex-project.org/lppl.txt>
and version 1.3 or later is part of all distributions of LaTeX version 2003/12/01 or later.

This work has the LPPL maintenance status "maintained".

Current maintainer of this work is kyanh <kyanh at o2 dot pl>.

List of files of this bundle can be found in FILELIST.
These files ***MUST*** be distributed as a whole.

FILELIST

```
./FILELIST
./README
./COPYING
./ntheorem-doc-vn.pdf
./test.tex
./test.pdf

./src/TODO
./src/README.src
./src/Makefile
./src/ntheorem-doc-vn.tex
./src/example.tex
./src/license.tex
./src/intro.tex
./src/test.tex
./src/user-interface.tex
./src/help.tex
./src/thanks.tex
./src/endmarks.tex
./src/interference.tex
./src/preamble.tex
./src/title-abstract.tex
```

1	Acknowledgements	4
2	Giới thiệu. Ghi chú	4
3	Sử dụng gói	4
3.1	Nạp gói	4
3.2	Định nghĩa THM mới	6
3.3	Thay đổi kiểu dáng	6
3.3.1	Các tham số chung	7
3.3.2	Cho từng THM cụ thể	7
3.3.3	Lựa chọn font	8
3.3.4	Các kiểu đã định nghĩa	8
3.3.5	Thiết lập mặc định	9
3.3.6	Các THM chuẩn	9
3.3.7	THM chuẩn cá nhân	9
3.4	Danh sách THM	10
3.4.1	Thay đổi kiểu danh sách THM	10
3.4.2	Ghi thông tin vào danh sách THM	10
3.5	Định nghĩa kiểu (danh sách) THM	11
3.5.1	Định nghĩa kiểu THM	11
3.5.2	Định nghĩa kiểu danh sách THM	12
3.6	Dấu kết thúc	12
3.7	Tham khảo mở rộng	13
3.8	Linh tinh	14
4	Ảnh hưởng đến các gói khác	14
4.1	Ảnh hưởng đến tùy chọn lớp tài liệu	14
4.2	Với gói amslatex	14
4.2.1	Với gói amsmath	14
4.2.2	Với gói amsthm	15
4.3	Với gói babel	15
4.4	Với gói hyperref	15
5	Vấn đề thường gặp	15
5.1	Giới hạn của gói ntheorem	15
5.2	Các BUG đã biết	17
5.3	Câu hỏi mở	17
6	Ví dụ	18
6.1	Tham khảo mở rộng	23
6.2	Danh sách THM	24

1 Acknowledgements

This place is dedicated to all those, who helped us developing our separate styles and this combined package. Thanks to (listed in alphabetical order):

Donald Arseneau, Giovanni Dore, Oliver Karch, Frank Mittelbach, Gerd Neugebauer, Heiko Oberdiek, Boris Piwinger, Bernd Raichle, Rainer Schöpf, Didier Verna.

2 Giới thiệu. Ghi chú

Một môi trường tựa định lý, ta gọi tắt là (môi trường) THM, được minh họa như ở Hình 1 trang liền sau. Với mỗi THM, các tên **định lý**, **hệ quả**, **bổ đề**, **tiên đề**, **định nghĩa**, ví dụ, **ghi chú**, **chứng minh**,... được gọi là *tên của* THM. Phần **header** của THM bao gồm tên của THM, chỉ số của THM, tên tùy chọn (hay tên riêng) của THM. Phần nội dung của THM còn gọi là *thân* (body) của THM. Để ý rằng, tên của THM khác với tên của môi trường THM.¹ Trong ví dụ sau,

```
\newtheorem{foobar}{Menh de}
```

ta có THM foobar, với tên là Menh de, nhưng tên của môi trường tương ứng là foobar. Đôi khi, ta sẽ gọi *môi trường* THM thay cho *tên của môi trường* THM.

Cách thể hiện THM có thể được thay đổi nhờ các tham số về kiểu chữ cho **header**, cho **thân** THM, các bố trí **header**, khoảng trắng thụt đầu dòng, cách đánh số,... Để thỏa mãn các yêu cầu thay đổi này, gói `theorem.sty` của Fran Mittelbach đã được viết và trở thành gói chuẩn của L^AT_EX từ rất lâu.

Tuy nhiên, các tính năng khác nhưng dấu kết thúc **endmarks**, danh sách các **theorem** vẫn chưa được hỗ trợ bởi gói chuẩn đó. Giải quyết vấn đề này, có hai mở rộng của gói `theorem.sty` được phát triển: một gói chuyên về điều khiển **endmarks**, gói `thmmarks.sty`, và một gói chuyên về liệt kê danh sách THM, gói `newthm.sty`. Sau đó, Frank Mittelbach đề nghị kết hợp các hỗ trợ của hai gói này vào cùng một gói mới (sẽ là chuẩn). Đó chính là gói `ntheorem.sty` ;)

3 Sử dụng gói

3.1 Nạp gói

Gói `ntheorem.sty` có thể nạp như sau

```
\usepackage[options]{ntheorem}
```

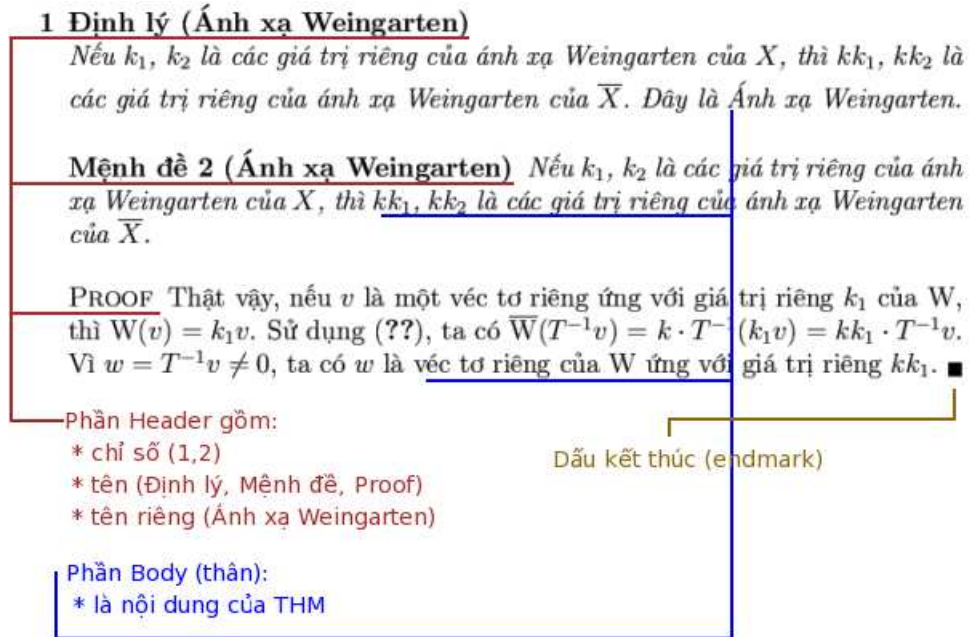
với *options* là danh sách các tùy chọn và các yêu cầu đặc biệt.

Các tùy chọn được cho nhờ *options* như sau:

standard
noconfig

```
standard noconfig
```

¹Trong thực tế, thường tên của môi trường THM và tên của THM có sự tương ứng 1-1, ví dụ `menhde` ứng với `Mệnh đề`, `dinhly` ứng với `Định lý`. Do đó, trong đa số trường hợp, sự phân biệt này không có ý nghĩa quan trọng... — kyanh



Hình 1: Môi trường THM

xem Mục 3.3.6 trang 9 và Mục 3.3.7 trang 9. Với một trong hai tùy chọn `standard` và `noconfig`, bạn có thể lựa chọn việc sử dụng hoặc không tập hợp các môi trường THM đã được định nghĩa sẵn.

- `amsthm` `amsthm`
 tùy chọn `amsthm` khi được dùng sẽ bảo đảm tính tương thích với các môi trường THM cung cấp bởi gói `amsthm`. Xem Mục 4.2 trang 14.
- `thmmarks` `thmmarks`
 tùy chọn `thmmarks` đồng ý để gói `ntheorem.sty` tự động đặt dấu kết thúc (`endmarks`) (xem Mục 3.3); khi dùng với gói `amsthm`, tùy chọn `thmmarks` phải được dùng kèm với tùy chọn `amsmath`. Xem thêm ở Mục 4.2.
- `thref` `thmref`
 tùy chọn `thref` cho phép mở rộng khả năng tham khảo chéo. Xem Mục 6.1 trang 23; khi dùng với gói `amsthm`, tùy chọn này phải đi kèm với tùy chọn `amsmath`. Xem thêm ở Mục 4.2.
- `hyperref` `hyperref`
 tùy chọn `hyperref` bảo đảm tương thích với gói `hyperref`. Xem Mục 4.4 trang 15.

Dưới đây là một ví dụ:

```
\usepackage{hyperref}
\usepackage[hyperref,thmmarks,noconfig]{ntheorem}
```

Với cách nạp gói như trên, bạn sẽ phải tự định nghĩa các môi trường THM, các dấu kết thúc sẽ được định vị tự động. Vì ta dùng gói `hyperref`, ta phải bảo đảm tính tương thích nhờ tùy chọn `hyperref`.

3.2 Định nghĩa THM mới

`\newtheorem` `\newtheorem`

Cú pháp của lệnh hoàn toàn giống như của lệnh chuẩn `\newtheorem`. Lệnh sẽ định nghĩa một THM mới. Có hai tham số bắt buộc là tên của môi trường và tên của THM. Tham số bổ sung chỉ ra cách đánh số môi trường.

```
\newtheorem{vidu}{Ví dụ}
```

Định nghĩa môi trường `vidu`, với tên là `Ví dụ` (như vậy, bạn sẽ có `Ví dụ 1`, `Ví dụ 2`, ...). Môi trường này sử dụng bộ đếm riêng `vidu`, và bạn có thể thay đổi giá trị bộ đếm này, chẳng hạn `\setcounter{vidu}0`.

```
\newtheorem{vidu2}[vidu]{Ví dụ khác}
```

Định nghĩa môi trường `vidu2`, với tên là `Ví dụ khác`. Môi trường này sẽ sử dụng cùng bộ đếm của môi trường `vidu` trong ví dụ trước.

```
\newtheorem{baitap}{Bài tập}[section]
```

Định nghĩa môi trường `baitap` (với tên là `Bài tập`), sử dụng bộ đếm thay đổi theo mục (`section`). Nếu bạn đang ở Mục số 5 chẳng hạn, bạn sẽ có `Bài tập 5.1`, `Bài tập 5.2`, ... Mỗi khi chuyển qua mục mới, bộ đếm sẽ được đặt về không, nghĩa là bạn sẽ có `Bài tập 6.1`, `Bài tập 6.2`, ..., `Bài tập 7.1`, `Bài tập 7.2`, ...

Khi gọi lệnh `\newtheorem` để tạo môi trường $\langle name \rangle$, thực ra sẽ có hai môi trường được tạo ra, là $\langle name \rangle$ và $\langle name* \rangle$. Điểm khác biệt duy nhất giữa hai môi trường này, cũng giống như sự khác biệt duy nhất giữa hai lệnh `\section` và `\section*`, là môi trường $\langle name* \rangle$ sẽ không đưa THM vào trong danh sách liệt kê các THM. Trong các ví dụ ở trên, bạn sẽ có chẳng hạn hai môi trường `baitap` và `baitap*`. Xem thêm Mục 3.4 trang 10.

`\renewtheorem` `\renewtheorem`

Định nghĩa lại môi trường đã có. Cách dùng tương tự như của `\newtheorem`. Bộ đếm sẽ được khởi tạo lại.

3.3 Thay đổi kiểu dáng

Với các môi trường tự định lý, bạn có thể thay đổi vài tham số (tùy chọn) trước khi gọi lệnh `\newtheorem` để tinh chỉnh cách thể hiện môi trường như ý bạn; các cài đặt nhờ tham số đó sẽ có tác dụng mỗi khi bạn sử dụng môi trường.

3.3.1 Các tham số chung

`\theorempreskipamount` `\theorempreskipamount` `\theorempreskipamount`
`\theorempostskipamount`

Các tham số bổ sung này ảnh hưởng đến khoảng cách theo chiều đứng – trên (`\theorempreskipamount`) và dưới (`\theorempostskipamount`) môi trường THM. Hai tham số này ảnh hưởng đến mọi môi trường THM và có thể điều chỉnh nhờ các lệnh thông thường điều khiển biến độ dài. Chúng là các chiều dài dạng *rubber*, vì thế có thể chứa các phần với dấu cộng hoặc trừ.

3.3.2 Cho từng THM cụ thể

Cách thể hiện của mỗi THM có thể tinh chỉnh nhờ các lệnh điều khiển sau đây.

`\theoremstyle` `\theoremstyle{<style>}`

Xác định kiểu dáng của THM. Các kiểu được cung cấp với `\ntheorem` bao gồm cả kiểu có trong gói `theorem.sty`. Xem liệt kê các kiểu ở Mục 3.3.4 trang sau. Ở Mục 3.5.1 trang 11 có nói về cách định nghĩa kiểu mới.

`\theoremheaderfont` `\theoremheaderfont{<fontcmds>}`

Dùng `<fontcmds>` để xác định font cho phần header của THM

Không như `theorem.sty`, lệnh `\theoremheaderfont` cho phép đổi font cho từng kiểu THM.

`\theorembodyfont` `\theorembodyfont{<fontcmds>}`

Xác định font cho phần thân (nội dung) THM.

`\theoremseparator` `\theoremseparator{<sep>}`

Dùng `<sep>` để ngăn cách phần header và phần thân của THM. Thường thì `<sep>` là dấu hai chấm (`:`) hoặc chấm (`.`).

`\theoremindent` `\theoremindent{<dimen>}`

dùng để xác định indent (khoảng cách so với lề bên trái).

! Ở đây, `<dimen>` là kích thước thật sự. Nếu bạn dùng kiểu *rubber* với các dấu plus hoặc minus trong phần `<dimen>`, bạn sẽ gặp lỗi.

`\theoremnumbering` `\theoremnumbering{<style>}`

Kiểu đánh số cho THM. Các giá trị có thể là: `arabic` (default), `alph`, `Alph`, `roman`, `Roman`, `greek`, `Greek` và `fnsymbol`.

Rõ ràng, nếu môi trường THM sử dụng bộ đếm từ môi trường XYZ khác, thì kiểu đánh số của môi trường THM sẽ thừa hưởng từ XYZ.

`\theoremsymbol` `\theoremsymbol{<thing>}`

Lệnh này chỉ các tác dụng khi gói `ntheorem.sty` được nạp với tùy chọn

thmarks. Ở đây, $\langle thing \rangle$ sẽ được dùng như `endmark`, tức dấu kết thúc cho THM. Nếu không muốn dùng `endmark` cho riêng môi trường THM nào, dùng lệnh `\theoremsymbol{}`.

Nhờ các lệnh điều khiển trên, bạn có thể linh hoạt tạo ra các THM như ý, mà không phải nhọc công và phải quan tâm nhiều đến yếu tố kỹ thuật.

`\theoremclass` `\theoremclass{\theorem-type}`

With the command `\theoremclass{\theorem-type}` (where $\langle theorem-type \rangle$ must be an already defined theorem type), these parameters can be set to the values which were used when `\newtheorem` was called for $\langle theorem-type \rangle$.

Với `\theoremclass{LaTeX}`, kiểu dáng chuẩn của L^AT_EX cho các THM sẽ được dùng.

3.3.3 Lựa chọn font

Xét về mặt cấu trúc, mỗi THM là một phần đặc biệt của tài liệu, trong đó, phần `header` được thiết kế để dễ dàng phân biệt với phần còn lại của môi trường. Vì vậy, lệnh `\theoremheaderfont` thừa hưởng các đặc trưng của `\theorembodyfont`, và đến lượt mình, `\theorembodyfont` thừa hưởng các thuộc tính của phần tài liệu bên ngoài THM đang xét.

Ví dụ: nếu `\theorembodyfont` là `\itshape` và `\theoremheaderfont` là `\bfseries`, thì phần `header` thực tế có kiểu ***đậm và nghiêng***.

Nếu điều này làm bạn không vừa ý, cụ thể là bạn muốn phần `header` chỉ được in đậm, có thể làm như sau:

`\theoremheaderfont{\normalfont\bfseries}`

3.3.4 Các kiểu đã định nghĩa

Các kiểu dáng định lý sau đã có sẵn (như trong gói `theorem.sty`):

<code>plain</code>	Như kiểu dáng của L ^A T _E X chuẩn, ngoại trừ tham số bổ sung <code>\theorem...skipamount</code> được dùng.
<code>break</code>	Phần <code>header</code> ngăn cách với phần thân THM bởi dòng mới ² .
<code>change</code>	Chỉ số và tên THM hoán đổi vị trí. Tuy nhiên, phần <code>header</code> sẽ theo sau ngay bởi phần thân THM (so sánh với kiểu
<code>changebreak</code>	Là sự kết hợp hai kiểu <code>change</code> và <code>break</code> .
<code>margin</code>	Chỉ số được bố trí ở lề trái, không ngắt dòng sau phần <code>header</code> .
<code>marginbreak</code>	Như <code>margin</code> , nhưng ngắt dòng sau phần <code>header</code> .
<code>nonumberplain</code>	Như <code>plain</code> , nhưng không đánh số (dùng cho chứng minh,...)
<code>nonumberbreak</code>	Tổ hợp <code>break</code> và <code>nonumberplain</code> .
<code>empty</code>	Phần <code>header</code> chỉ gồm tên riêng (nếu có), còn chỉ số và tên của THM được bỏ qua.

²thực ra là một dấu ngắt dòng

3.3.5 Thiết lập mặc định

Khi không có tùy chọn nào được chỉ ra, nghĩa là gói `ntheorem.sty` được nạp đơn giản nhờ `\usepackage{ntheorem}`, các thiết lập sau sẽ được dùng:

```
\theoremstyle{plain}
\theoremheaderfont{\normalfont\bfseries}
\theorembodyfont{\itshape}
\theoremseparator{}
\theoremindent0cm
\theoremnumbering{arabic}
\theoremsymbol{}
```

Vì vậy, bằng cách dùng `\newtheorem{...}{...}`, bạn thu được cách thể hiện giống hệt trong L^AT_EX chuẩn.

3.3.6 Các THM chuẩn

Các THM chuẩn (được định nghĩa sẵn) có trong tập tin `ntheorem.std`, được nạp nhờ tùy chọn `standard` của gói. Khi dùng tùy chọn này, các gói phụ thuộc `amssymb` và `latexsym` được nạp tự động. Danh sách các THM như sau đây, là các THM hay dùng nhất trong các tài liệu tiếng Anh và tiếng Đức:

Theorems: `Theorem`, `Lemma`, `Proposition`, `Corollary`, `Satz`, `Korollar`,

Definitions: `Definition`,

Examples: `Example`, `Beispiel`,

Remarks: `Anmerkung`, `Bemerkung`, `Remark`,

Proofs: `Proof` and `Beweis`.

Với các THM thuộc họ ‘Theorems’, kiểu THM được dùng là `plain`, với font của phần thân THM là `\itshape`, của phần header là `\bfseries`, dấu kết thúc môi trường là `\ensuremath{_\Box}`.

Với họ ‘Definitions’, ‘Remarks’ và ‘Examples’, các thiết lập cũng tương tự, ngoại trừ phần `body` có font là `\upshape`.

Riêng họ ‘Proofs’ thì có vài điểm khác biệt nhỏ: họ này sử dụng kiểu `nonumberplain`, font cho phần thân là `\upshape`, cho phần header là `\scshape`, còn dấu kết thúc là `\ensuremath{_\blacksquare}`.

Để biết thêm chi tiết, bạn có thể đọc trực tiếp mã nguồn các định nghĩa trong tập tin `ntheorem.std`.

3.3.7 THM chuẩn cá nhân

Rõ ràng, mỗi người dùng có thể muốn có riêng mình các THM chuẩn, tương tự như các THM được cung cấp trong `ntheorem.std`. Tuy nhiên, không nên thay đổi tập tin `ntheorem.std`. Có cách khác, hay hơn, là dùng tập tin `ntheorem.cfg`. Nội dung của `ntheorem.cfg` cũng tương tự như `ntheorem.std`, nơi đặt định nghĩa các môi trường THM hay dùng nhất (chuẩn) của người dùng.

Nơi để tập tin `ntheorem.cfg` là bất kỳ đâu mà T_EX có thể tìm thấy. Điều đáng lưu ý là, tập tin này được nạp tự động, trừ khi bạn chỉ ra tùy chọn `noconfig` khi nạp

gói `ntheorem`; và khi tập tin `ntheorem.cfg` được dùng, thì tập tin `ntheorem.std` sẽ bị bỏ qua, kể cả khi bạn dùng tùy chọn `standard` cho gói.

3.4 Danh sách THM

`\listtheorems` Tương tự, như lệnh \LaTeX chuẩn `\listoffigures`, danh sách các THM được định nghĩa với `\newtheorem` có thể được liệt kê nhờ lệnh `\listtheorems`:

```
\listtheorems{<list>}
```

Đối số của lệnh là `<list>`, danh sách các THM cần được liệt kê, trong đó các THM cách nhau bởi dấu phẩy:

```
\listtheorems{theorem,thm,dinhly}
```

Trong ví dụ trên, các THM sẽ được liệt kê gồm `theorem`, `thm`, `dinhly` — ở đây, `theorem`, `thm` và `dinhly` tương ứng với các môi trường `\begin{theorem}...`, `\begin{thm}...`, `\begin{dinhly}...` — là các (môi trường) THM được tạo nhờ `\newtheorem`. Để ý rằng, các THM sao, ví dụ `\begin{dinhly*}...`, sẽ không có trong danh sách trên. Điều này cũng tương tự như các xử lý của `\section` và `\section*` trong lệnh `\tableofcontents`.

Nếu trong danh sách `<list>` có THM nào đó chưa được định nghĩa (có thể do bạn gõ sai chính tả), danh sách THM vẫn được tạo ra, nhưng có thể trong kết quả xuất bạn không thấy gì cả! Gói sẽ cảnh báo bạn điều này.

3.4.1 Thay đổi kiểu danh sách THM

`\theoremlisttype` Danh sách các THM có thể thay đổi cách thể hiện. Điều này có thể làm được nhờ chọn kiểu cho danh sách:

```
\theoremlisttype{<type>}
```

Các giá trị có thể của `<type>` như sau đây. Bạn có thể xem thêm ở Mục 6 trang 18 về ví dụ:

`all` Liệt kê các THM với các thông tin gồm chỉ số THM, tên riêng (nếu có) của THM và số trang. Đây là kiểu mặc định.

`allname` Tương tự `all`, nhưng có kèm theo tên THM.

`opt` Tương tự `all`, nhưng chỉ các THM có tên riêng mới được liệt kê.

`optname` Tương tự `opt`, nhưng kèm theo tên THM.

3.4.2 Ghi thông tin vào danh sách THM

Tương tự hai lệnh `\addcontentsline` và `\addtocontents`, bạn có thể ghi thêm các thông tin bổ sung vào danh sách THM.³

`\addtheoremline` Tương tự lệnh `\addtocontentsline`:

³Không thể dùng lệnh `\addcontentsline` và `\addtocontents` một cách trực tiếp để ghi thông tin vào tập tin `.thm`, lý do là, so với các phần tử của bảng Mục lục, phần tử của danh sách THM phức tạp hơn nhiều.

```
\addtheoremline{<name>}{<text>}
```

với $\langle name \rangle$ là THM (đã được định nghĩa) và $\langle text \rangle$ là phần nội dung bạn muốn sẽ xuất hiện trong danh sách. Ví dụ

```
\addtheoremline{Example}{Extra Entry with number}
```

sẽ sinh ra trong danh sách THM một phần tử với các thông tin sau:

- Nhãn của môi trường “Example”,
- Chỉ số hiện tại của “Example”,
- Số trang nơi lệnh trên xuất hiện
- Đoạn `Extra Entry with number` như là tên riêng của THM.

Và như thế, kết quả ở trên giống hệt như khi bạn dùng

```
\begin{Example}[Extra Entry with number]  
...  
\end{Example}
```

ngoại trừ rằng không có nội dung nào của THM xuất hiện, và chỉ số THM vẫn giữ nguyên, không tăng.

`\addtheoremline*` Bạn cũng có thể dùng

```
\addtheoremline*{Example}{Extra Entry}
```

với kết quả tương tự khi dùng `\addtheoremline`, nhưng thông tin xuất hiện trong danh sách THM sẽ không có chỉ số THM.

`\addtotheoremfile` Có vài trường hợp, ví dụ khi danh sách THM quá dài, bạn có thể muốn thêm vài lệnh điều khiển, hoặc nội dung nào đó vào danh sách. Điều này có thể đạt được nhờ

```
\addtotheoremfile[<name>]{<text>}
```

với $\langle name \rangle$ là THM nào đó, $\langle text \rangle$ là mã lệnh hay đoạn văn,... bạn muốn thêm vào danh sách THM $\langle name \rangle$. Tham số $\langle name \rangle$ có thể bỏ qua, và khi đó, $\langle text \rangle$ sẽ được chèn vào mọi danh sách.

3.5 Định nghĩa kiểu (danh sách) THM

Chỉ đọc phần này nếu bạn có khả năng lập trình với L^AT_EX.

3.5.1 Định nghĩa kiểu THM

`\newtheoremstyle` Kiểu THM có thể định nghĩa như sau:

```
\newtheoremstyle{<name>}{<head>}{<opt-head>}
```

Sau lệnh trên, kiểu $\langle name \rangle$ sẽ hợp lệ, và bạn có thể dùng $\langle name \rangle$ làm đối số của lệnh `\theoremstyle`.

Ở đây, $\langle head \rangle$ là nhóm các lệnh điều khiển, phải sử dụng hai tham số, `##1` (chứa từ khóa) và `##2` (chứa chỉ số). Phần $\langle opt-head \rangle$ phải dùng tham số thứ ba `##3`, chứa phần tham số bổ sung.

Vì L^AT_EX tạo các THM nhờ `\trivlist`, cả hai phần khai báo $\langle head \rangle$ và $\langle opt-head \rangle$ phải có dạng `\item[... \theorem@headerfont ...]`, ở đó phần `...` được thiết lập tùy ý bởi người dùng. Nếu nhóm lệnh nào đó sinh ra các nội dung (output) đăng sau `\item[...]`, bạn cần cẩn thận với các khoảng trắng.

Nếu phần khai báo có sử dụng tới các lệnh `@`, và nếu `\newtheoremstyle` sẽ được dùng trong tập tin `.tex`, bạn cần đặt lệnh `@` vào cặp `\makeatletter` và `\makeatother`.

Để biết thêm chi tiết, bạn xem tài liệu về mã nguồn của `ntheorem.sty`.

`\renewtheoremstyle` Kiểu THM có thể định nghĩa lại, nhờ `\renewtheoremstyle`. Các dùng cũng tương tự như `\newtheoremstyle`.

3.5.2 Định nghĩa kiểu danh sách THM

`\newtheoremlisttype` Kiểu danh sách THM có thể tinh chỉnh nhờ

```
\newtheoremlisttype{<name>}{<start>}{<line>}{<end>}
```

Tham số đầu tiên $\langle name \rangle$ là tên của kiểu danh sách, sẽ được dùng cho làm đối số cho lệnh `\theoremlisttype`. Tham số $\langle start \rangle$ (tương ứng, $\langle end \rangle$) là chuỗi lệnh sẽ được thi hành mỗi khi bắt đầu (tương ứng, kết thúc) danh sách. (Trong các kiểu chuẩn, cả hai phần này đều rỗng.) Tham số $\langle line \rangle$ là chuỗi lệnh được thực thi cho mỗi phần tử của danh sách; trong chuỗi lệnh này có thể sử dụng bốn tham số: `##1` sẽ đại diện cho tên THM, `##2` cho chỉ số, `##3` cho tên riêng và `##4` cho số trang.

CHÚ Ý: Các kiểu do người dùng định nghĩa có thể đụng độ với gói `hyperref`.

`\renewtheoremlisttype` Dùng để định nghĩa lại kiểu đã có.

3.6 Dấu kết thúc

Các dấu kết thúc sẽ tự động đặt vào cuối thân THM khi dùng tùy chọn `thmmarks` cho gói `ntheorem`. Vì sự tự động đó, một vài lệnh dưới đây dùng để xử lý các dấu kết thúc trong vài trường hợp rất đặc biệt.

`\qed` `\qed` `\qedsymbol`
`\qedsymbol`

Trong một môi trường đơn lẻ,⁴ bạn có thể muốn thay dấu kết thúc chuẩn bởi dấu khác, theo ý bạn. Khi đó, hãy dùng `\qed`, nếu trước đó bạn đã định nghĩa dấu kết thúc bằng `\qedsymbol{<something>}` (trong kiểu chuẩn ứng với tùy chọn `standard` của gói, `\qedsymbol` được định nghĩa là ký hiệu dùng cho môi trường `proof`, các hệ quả đơn giản không đi kèm với chứng minh tường minh).

⁴Điều này có nghĩa, bạn muốn thay đổi dấu kết thúc cho một THM cụ thể nào đó mà thôi!

Ngoài ra, với một môi trường THM đơn lẻ mà khi định nghĩa bạn đã không chỉ ra dấu kết thúc cho THM đó, bạn vẫn có thể đặt dấu kết thúc nhờ `\qedsymbol` và `\qed` như vừa nói trên.

`\NoEndMark`
`\TheoremSymbol`

`\NoEndMark` `\TheoremSymbol`

Trong một số trường hợp khác, bạn có thể muốn đặt dấu kết thúc vào vị trí đúng như bạn chỉ ra, ví dụ bên trong môi trường `figure` hay `minipage`. Khi đó, tính năng tự động đặt dấu kết thúc cho THM hiện tại có thể tắt nhờ `\NoEndMark`. Sau khi gọi lệnh đó cho THM `<name>`, bạn có thể đặt dấu kết thúc vào bất kỳ đâu, bằng cách gọi lệnh `\<name>Symbol` — chẳng hạn `\dinhlySymbol`. Đối với tài liệu này, việc gọi `\TheoremSymbol` sẽ cho \diamond .

Chú ý rằng, dòng cuối cùng trước khi kết thúc môi trường THM, ví dụ trước `\end{dinhly}`, không được là dòng trắng; bởi nếu không, dấu kết thúc sẽ bị bỏ qua. Xem thêm ví dụ 3 về điều này ở Mục 6 trang 18.

3.7 Tham khảo mở rộng

Tính năng tham khảo mở rộng có thể kích hoạt nhờ tùy chọn `thref` khi nạp lớp `ntheorem`.

Thường thì khi soạn tài liệu, ta có thể thay đổi⁵ từ mệnh đề sang định lý, từ định lý sang hệ quả, từ bổ đề sang chú ý, Khi sự thay đổi đó xảy ra, các tham khảo chéo cần phải phải đổi theo. Ví dụ, nếu trước đây bạn dùng “xem Mệnh đề~`\ref{completeness}`”, thì bây giờ bạn có thể phải đổi thành “xem Bổ đề~`\ref{completeness}`”. Rõ ràng, công việc đó quả là tỉ mỉ! Gói `ntheorem` cung cấp tính năng tham khảo chéo mở rộng, giúp bạn giải quyết vấn đề này. Lệnh

`\label{<label>}[<type>]`

sẽ gán kiểu `<type>` với nhãn `<label>`. Việc này được thực hiện tự động với các môi trường THM:

`\begin{Theorem}[<name>]\label{<label>}`

tương đương với

`\begin{Theorem}[<name>]\label{<label>}[Theorem]`

`\thref`

`\thref`

`\thref{<label>}`

Bây giờ, lệnh gọi như trên sẽ sinh ra, chẳng hạn “Định lý 42”. Để ý rằng, cần phải biên dịch tài liệu ít nhất hai lần sau khi có sự thay đổi về nhãn, để các tham khảo chéo được chính xác.

Tùy chọn `thmref` ảnh hưởng tới gói `babel`, do đó, khi tùy chọn này được dùng, gói `ntheorem` cần phải nạp sau gói `babel`. Tùy chọn cũng ảnh hưởng tới gói `amsmath`. Xem Mục 4.2 trang sau.

⁵Thay đổi ở đây, không có nghĩa là thay đổi nội dung, mà là thay đổi về tên gọi.

3.8 Linh tinh

Bên trong môi trường THM `\langle env \rangle`, tên riêng của THM có thể lấy được nhờ `\langle env \rangle name`. Chẳng hạn `\dinhlyname` sẽ cho tên riêng của THM `dinhly`.

4 Ảnh hưởng đến các gói khác

Vì `ntheorem` xây dựng lại hoàn toàn cách xử lý THM, nên nó không tương thích với mọi gói có liên quan đến cách xử lý THM.

Ngoài ra, thuật toán đặt dấu kết thúc `thmmarks` đòi hỏi phải thay đổi vài môi trường (xem tài liệu về mã nguồn của `ntheorem`). Vì vậy, những môi trường được thay đổi, xây dựng lại bởi các lớp tài liệu hay các gói sẽ không chịu ảnh hưởng bởi thuật toán cung cấp bởi gói `ntheorem`.

Tùy chọn `thref` sẽ định nghĩa lại lệnh `\label` và sẽ xử lý các nhãn trong khi đọc tập tin `.aux`. Chính vì thế, việc dùng tùy chọn này sẽ gây ra sự không tương thích với mọi gói có thay đổi lệnh `\label` hoặc `\newlabel`. Với gói `babel`, sự không tương thích được giải quyết chỉ khi gói `ntheorem` được nạp sau gói `babel`.

4.1 Ảnh hưởng đến tùy chọn lớp tài liệu

Gói `ntheorem` cũng đung độ với các tùy chọn `leqno` và `fleqn` cho lớp tài liệu. Nếu một trong hai tùy chọn đó được chỉ ra khi nạp lớp, gói `ntheorem` (với tùy chọn `thmmarks`) sẽ phát hiện được.

Nếu các tùy chọn đó không được chỉ ra khi gọi lớp, mà khi nạp gói `amsmath` (xem mục tiếp theo), thì tùy đó phải không được chỉ ra khi nạp gói `ntheorem`, bởi vì chính mọi môi trường của gói `amsmath` sẽ nhận ra tùy chọn ấy.

4.2 Với gói `amslatex`

Gói `ntheorem` ảnh hưởng đến gói `amsmath` và `amsthm`.

Chú ý rằng, gói `amstex.sty` của $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2.09$ đã cũ và bạn nên thay thế bởi hai gói `amsmath` và `amstext` của $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$. Gói `ntheorem` các phiên bản từ 1.18 về trước tương thích với `amsmath-1.x`, và các phiên bản từ 1.19 tương thích với `amsmath-2.x` (hy vọng vậy ;)

Tác giả `ntheorem` hy vọng có ai đó dùng và hiểu gói `amsmath` có thể tham gia đội phát triển và bảo dưỡng `ntheorem` để bảo đảm sự tương thích này.

4.2.1 Với gói `amsmath`

Sự tương thích với gói `amsmath` (ở các điểm: đặt dấu kết thúc, xử lý nhãn trong môi trường toán) được bảo đảm nhờ tùy chọn `amsmath` khi nạp lớp `ntheorem`.

- `\usepackage[thmmarks]{ntheorem}` cần phải được thay bởi `\usepackage[amsmath,thmmarks]{ntheorem}`), và tương tự
- `\usepackage[thref]{ntheorem}` cần thay bởi `\usepackage[amsmath,thref]{ntheorem}`).

Cũng cần lưu ý rằng, gói `amsmath` cần phải nạp trước gói `ntheorem`, để đảm bảo rằng các định nghĩa của `amsmath` sẽ được `ntheorem` xử lý lại.

4.2.2 Với gói `amsthm`

Gói `amsthm` xung đột với gói `theorem` (về kiểu THM). Thay vì dùng gói `amsthm`, bạn hãy dùng tùy chọn `amsthm` khi nạp gói `ntheorem`.

Tùy chọn đó cung cấp các kiểu THM là `plain`, `definition`, `remark` và `proof` đúng như của gói `amsthm`.

Lệnh `\newtheorem*` được cung cấp bởi `ntheorem` ngay cả khi bạn không dùng tùy chọn `amsthm`. Chú ý rằng, `\newtheorem*` luôn tương ứng với bản không đánh số của kiểu THM hiện tại; do đó, khi dùng `\newtheorem*` thì kiểu THM là kiểu đã có.

Lệnh `\newtheoremstyle` và `\swapnumbers` của gói `amsthm` không được `ntheorem` xây dựng lại. Vì vậy, bạn phải định nghĩa các lệnh này (như gói `amsthm` định nghĩa), cộng thêm các thay đổi nhờ dùng các lệnh `\theoremheaderfont` và `\theorembodyfont`.

Tóm lại, bạn không nên dùng gói `amsthm`, vì các tính năng cung cấp bởi `ntheorem` trực quan và thân thiện hơn.

4.3 Với gói `babel`

Khi dùng tùy chọn `thref`, gói `babel` phải được nạp trước gói `babel`.

4.4 Với gói `hyperref`

Vì gói `hyperref` định nghĩa lại lệnh `\contentsline` của `LATEX`, nên gói này sẽ trực trặc với `ntheorem` phiên bản 1.17 về trước. Từ bản 1.17 của `ntheorem`, có thêm tùy chọn `hyperref` bảo đảm sự tương thích: trong danh sách THM bạn sẽ có liên kết đến các THM tương ứng.

Chú ý rằng, nếu bạn định nghĩa (lại) kiểu danh sách THM như ở Mục 3.5.2 trang 12), kiểu mới đó sẽ không làm việc tốt với `hyperref`.

5 Vấn đề thường gặp

5.1 Giới hạn của gói `ntheorem`

- `ntheorem` dùng tập tin `.aux` để lưu thông tin về dấu kết thúc, do đó, cần biên dịch tài liệu ít nhất hai lần để các dấu kết thúc được đặt đúng vị trí.
- Cũng do sử dụng tập tin `.aux` để lưu thông tin về danh sách trong tập tin `.thm`, cần thêm ít nhất hai lần biên dịch nữa. Việc di chuyển các THM có thể cần tới 5 lần biên dịch để có danh sách THM đúng đắn.
- Gói xử lý phần tên riêng của THM (tham số bổ sung khi gọi môi trường) theo vài cách khác nhau trong danh sách, do đó, đã sao chép nguyên xi, không triển khai (`expand`) phần tham số bổ sung đó vào tập tin `.thm`. Hệ quả là, nếu bạn dùng chẳng hạn lệnh `\thesection` bên trong phần tên riêng thì

kết quả sẽ không như ý. Bạn không nên dùng bất kỳ lệnh gì trong phần tên riêng THM!

- Nếu các THM lồng nhau kết thúc ở cùng thời điểm, `ntheorem` chỉ lập dấu kết thúc cho môi trường sâu nhất (level cao nhất) Chẳng hạn

```
\begin{Lemma}
Some text.
\begin{Proof}
The Proof
\end{Proof}
\end{Lemma}
```

sẽ đưa tới kết quả sau

α Bổ đề:
Some text.

CHỨNG MINH The Proof ■

Bạn có thể vượt qua nhược điểm này, bằng cách thêm một nội dung ẩn sau khi kết thúc THM bên trong, khi đó, THM bên ngoài sẽ có dấu kết thúc như ý. Quan sát ví dụ sau đây:

```
\begin{Lemma}
Some text.
\begin{Proof}
The Proof
\end{Proof}~
\end{Lemma}
```

kết quả là

β Bổ đề:
Some text.

CHỨNG MINH The Proof ■



- Sử dụng tùy chọn `fleqn` khi gọi lớp tài liệu có thể sinh ra rắc rối. Lý do là tùy chọn `fleqn` điều khiển các phương trình không phải bởi `$$` mà bởi danh sách (thử kiểm tra xem điều gì xảy ra nếu bạn dùng

```
\begin{theorem} \[ displaymath \] \end{theorem}
```

trong \LaTeX chuẩn: nội dung `displaymath` sẽ không được đặt ở dòng riêng. Cũng như vậy, với công thức dài, chỉ số phương trình và dấu kết thúc có thể sẽ gần công thức hơn so với bình thường.

- Một cách tự nhiên, `ntheorem` không làm việc cùng với các kiểu (gói) liên quan đến

1. xử lý môi trường THM tự định lý,

2. xử lý dấu kết thúc (ví dụ `\[...]`, `eqnarray`,...)

- `ntheorem` không tương thích với gói `theorem` của Frank Mittelbach, là một gói dùng để biểu diễn THM rất phổ biến.

Gói `ntheorem` không thể dùng chung với gói `theorem`, nhưng có thể dùng thay cho gói `theorem`.

5.2 Các BUG đã biết

- Khi kết thúc môi trường THM ngay sau nội dung, dấu kết thúc sẽ bị bỏ qua. Ví dụ

```
\begin{Lemma} Lemma\end{Lemma}
```

sẽ sinh ra

```
 $\gamma$  Bổ đề:  
Lemma
```

Vì vậy, ít nhất một khoảng trắng hoặc dấu ngắt dòng cần phải có trước `\end{...}`. Ngoài ra, trước `\end{...}` không được là dòng trắng.

- Với kiểu THM `break`, nếu việc ngắt dòng sau `header` làm cho phần nội dung tiếp theo “xấu xí” (theo cách hiểu của `LATEX`), thì việc ngắt dòng đó sẽ bị bỏ qua.

5.3 Câu hỏi mở

Các câu hỏi này liên quan chủ yếu đến việc phát triển gói `ntheorem` — nghĩa là không có ích lắm với người dùng bình thường. Nếu ai đó có câu trả lời cho một trong các câu hỏi dưới đây, vui lòng thông báo cho tác giả gói `ntheorem`; tác giả của câu trả lời sẽ có mặt trong `Acknowledgements`.

- Với các phương trình (biểu diễn nhờ môi trường `equation`), dấu kết thúc sẽ được đặt sau chỉ số phương trình (được canh giữa theo chiều đứng). Hiện tại, vẫn chưa có thuật toán để canh giữa chỉ số phương trình đồng thời đặt dấu kết thúc ở bên dưới (việc này đòi hỏi phải biết chiều cao của nội dung phương trình)
- Thuật toán đặt dấu kết thúc dựa trên kết quả của việc kiểm tra xem liệu phần nội dung bình thường đã hết chưa (khi gặp kết thúc môi trường `\end{...}`). Việc kiểm tra này có câu trả lời *một phần* (`partially`) nhờ `\ifhmode`: trong một dòng, `LATEX` luôn ở chế độ `\hmode`. Nhưng sau các biểu thức toán, `LATEX` cũng ở chế độ `\hmode`. Vì thế, phải kiểm tra thêm về `\lastkip`: sau biểu thức toán thì `\lastskip=0`. Trong hầu hết trường hợp, khi nội dung vừa ghi xong vào một dòng, thì `\lastskip ≠ 0`. Nhưng điều này không luôn chắc chắn: nếu mã nguồn có dạng `...text\label{bla}`, thì (sau đó) `\lastskip=0`. Và khi đó, dấu kết thúc bị bỏ qua.

Câu hỏi đặt ra là, làm thế nào để xác định thời điểm `LATEX` vừa kết thúc việc biểu diễn công thức toán?

- Trong vấn đề trên về nhãn: kiểu `break` sẽ gắng ngắt dòng sau phần `header` bằng cách dùng `\hfill\penalty-8000` đằng sau `item` của `\trivlist`. Vì vậy, `TeX` sẽ chuyển vào chế độ `horizontal`. The label places a “whatsit” somewhere ... and, it seems that the “whatsit” makes `TeX` think that there is a line of text.⁶

6 Ví dụ

Các thiết lập được dùng như sau đây. Chú ý rằng, các thiết lập tuân theo nguyên tắc thừa kế. Ví dụ, thiết lập `font` không được khởi tạo về trạng thái bình thường khi ta khai báo môi trường `Lemma`, vì thế `Lemma` sẽ thừa kế các thiết lập của môi trường `Theorem`.

Một số ví dụ có dùng lệnh `\color` được cung cấp bởi gói `xcolor.sty`.

Định lý:

```
\theoremstyle{marginbreak}
\theoremheaderfont{\normalfont\bfseries}\theorembodyfont{\slshape}
\theoremsymbol{\ensuremath{\diamondsuit}}
\theoremseparator{:}
\newtheorem{Theorem}{Theorem}
```

Bổ đề:

```
\theoremstyle{changebreak}
\theoremsymbol{\ensuremath{\heartsuit}}
\theoremindent0.5cm
\theoremnumbering{greek}
\newtheorem{Lemma}{Lemma}
```

Hệ quả:

```
\theoremindent0cm
\theoremsymbol{\ensuremath{\spadesuit}}
\theoremnumbering{arabic}
\newtheorem{Corollary}[Theorem]{Corollary}
```

Ví dụ:

```
\theoremstyle{change}
\theorembodyfont{\upshape}
\theoremsymbol{\ensuremath{\ast}}
\theoremseparator{}
\newtheorem{Example}{Example}
```

Định nghĩa

```
\theoremstyle{plain}
\theoremsymbol{\ensuremath{\clubsuit}}
\theoremseparator{.}
\newtheorem{Definition}{Definition}
```

⁶kyanh: “whatsit”, đó là cái gì vậy?

Chúng minh:

```
\theoremheaderfont{\sc}\theorembodyfont{\upshape}  
\theoremstyle{nonumberplain}  
\theoremseparator{  
\theoremsymbol{\rule{1ex}{1ex}}  
\newtheorem{Proof}{Proof}
```

1 Ví dụ (Ví dụ đơn giản) Một chiều đi trên con đường này, Hoa điệp vàng trái dưới chân tôi,...

Các ví dụ tiếp theo minh họa cho việc đặt các dấu kết thúc ở các biểu thức toán, phương trình đơn lẻ và các dãy phương trình. *

1 Định lý (Định lý dài):

Ví dụ về môi trường *Theorem*, lồng bên trong nó là các môi trường khác như *Example*, *Lemma*,... Khi đang ở ngay trong môi trường *Theorem*, chữ có màu đen; khi ở trong các môi trường sâu hơn, màu của môi trường sẽ đổi khác để dễ phân biệt.

Ví dụ tiếp theo sẽ không xuất hiện trong danh sách *Example*, vì nó được dùng với dạng sao:

```
\begin{Example*}  
...  
\end{Example*}
```

2 Ví dụ (kết thúc với biểu thức toán) Hãy để ý vị trí dấu kết thúc ở dưới dòng biểu diễn công thức

$$f^{(n)}(z) = \frac{n!}{2\pi i} \int_{\partial D} \frac{f(\zeta)}{(\zeta - z)^{n+1}} d\zeta$$

*

Bây giờ, ta sẽ ghi vài thông tin bổ sung vào danh sách *THM (Example)*; thông tin được ghi sẽ không gồm chỉ số *THM*:

```
\addtheoremline*{Example}{Extra Entry}
```

δ Bổ đề (Biểu thức trong mảng (dãy) phương trình):

Bổ đề (*Lemma*) được thụt đầu dòng và đánh chỉ số với chữ số Hy Lạp.

Xét ví dụ sau đây, trông có vẻ tốt:

```
[\begin{array}{l}  
  a = \begin{array}{t}{l}  
    first\ line \\  
    second\ line  
  \end{array}%  
  \mbox{\color{green}{try to put this text in the lowest line}}  
\end{array}  
\]
```

Nhưng biểu diễn như trên mà không sử dụng các mẹo đặc biệt nào (thử để lựa chọn các vị trí [t], [c], [b]), kết quả có thể không như ý:

$a =$ *first line* try to put this text in the lowest line
second line



ε BỔ ĐỀ (Phương trình):

Với các phương trình có đánh số, dấu kết thúc sẽ được đặt sau chỉ số phương trình, và được canh giữa theo chiều đứng. Hiện tại, chưa có thuật toán để bố trí chỉ số phương trình canh giữa (theo chiều đứng), còn dấu kết thúc lại được đặt vào cuối phương trình — việc này đòi hỏi phải biết chiều cao của nội dung phương trình. Nếu ai đó biết, vui lòng báo cho tác giả của `ntheorem`.

$$\int_{\gamma} f(z) dz := \int_a^b f(\gamma(t))\gamma'(t) dt \tag{1}$$



Với kiểu `break`, nếu sử dụng việc đánh nhãn như sau

```
\begin{Lemma}[Breakstyle]\label{breakstyle}
```

ζ BỔ ĐỀ (Breakstyle):

thì như bạn thấy, xuất hiện các khoảng trắng thừa ...

Nếu dấu phần trăm (chú thích) được đặt ngay sau lệnh tạo nhãn (hoặc dùng lệnh `\ignorespaces`), ví dụ

```
\begin{Lemma}[Breakstyle]\label{breakstyle}%
```

thì khoảng trắng thừa sẽ mất.

Với các kiểu đã được định nghĩa bởi `ntheorem`, điều này chỉ xảy ra chỉ với kiểu `break`. Đây không phải là lỗi, mà chính là cách xử lý của `LATEX`.

Ví dụ với `eqnarray`:

$$f(z) = \frac{1}{2\pi i} \int_{\partial D} \frac{f(\zeta)}{\zeta - z} d\zeta \tag{2}$$

$$= \frac{1}{2\pi} \int_0^{2\pi} f(z_0 + re^{it}) dt \tag{3}$$



CHỨNG MINH (OF NOTHING)

$$f(z) = \frac{1}{2\pi i} \int_{\partial D} \frac{f(\zeta)}{\zeta - z} d\zeta$$

$$= \frac{1}{2\pi} \int_0^{2\pi} f(z_0 + re^{it}) dt$$



Đến đây là kết thúc Định lý 1.



Nếu có nhiều môi trường cùng bên trong một môi trường THM, môi trường cuối cùng mới có dấu kết thúc

Định nghĩa 1 (với danh sách).

$$\int_{\gamma} f(z) dz := \int_a^b f(\gamma(t))\gamma'(t) dt \quad (4)$$

- như vậy, bạn đã thấy dấu kết thúc làm việc thế nào với văn bản
- với biểu thức toán
- và với danh sách. ♣

2 Hệ quả (Q.E.D.):

Đây là hệ quả tầm thường, kết thúc bởi `\qedsymbol{\textrm{q.e.d}}` và `\qed`. q.e.d

3 Ví dụ

$$f^{(n)}(z) = \frac{n!}{2\pi i} \int_{\partial D} \frac{f(\zeta)}{(\zeta - z)^{n+1}} d\zeta$$

Nếu có văn bản theo sau môi trường, dấu kết thúc sẽ đặt sau văn bản đó. *

Ví dụ tiếp theo được cho bởi mã nguồn sau đây. Chú ý rằng, lệnh `~\hfill~` được chèn vào để ngăn cản L^AT_EX quản lý danh sách lồng nhau theo cách của chính L^AT_EX (môi trường `verbatim` là danh sách `\trivlist`). Có nghĩa là, việc này sẽ khiến L^AT_EX bắt đầu môi trường `verbatim` với một dòng mới.

```
\begin{Example}
~\hfill~
\begin{verbatim}
And, it also works for verbatim
... when the \end{verbatim} is in the
same line as the text ends. \end{verbatim}
~ this space is important !!
\end{Example}
```

4 Ví dụ (dùng ‘verbatim’)

```
And, it also works for verbatim
... when the end{verbatim} is in the
same line as the text ends.
```

Không được chừa dòng trắng trước `\end{theorem}`, bởi làm thế sẽ bỏ qua dấu kết thúc.

```
\begin{Theorem}
some text ... but no end mark

\end{Theorem}
```

3 Định lý:

some text ... but no end mark

Bây giờ là hệ quả sẽ xuất hiện với tên riêng hơi khác trong danh sách Corollary:

```
\begin{Corollary*}[title in text]\label{otherlabel}
...
\end{Corollary*}
\addtheoremline{Corollary}{title in list}
```

4 Hệ quả (title in text):

Thể hiện trong

môi trường

canh giữa.



5 Định lý (trích dẫn):

Trong môi trường trích dẫn quote, nội dung thường được thụt vào ở tất cả các dòng ở cả bên trái và bên phải. Tuy nhiên, dấu kết thúc sẽ không vẫn được đặt đúng vào lề bên phải.



Dưới đây là ví dụ về việc tắt/bật việc đặt dấu kết thúc tự động.

```
\begin{Theorem}[Manual End Mark]\label{somelabel}
a line of text with a manually set endmark \hfill\TheoremSymbol\\
some more text, but no automatic endmark set. \NoEndMark
\end{Theorem}
```

6 Định lý (Manual End Mark):

a line of text with a manually set endmark

some more text, but no automatic endmark set.



Để ý rằng, lệnh `\hfill` được chèn vào trước dấu kết thúc để đặt dấu đó vào lề bên phải.

5 Ví dụ (nhánh hơn) Dấu kết thúc được tự động đặt trở lại...



Nếu bạn không thích dùng chỉ số là chữ số Hy Lạp và cách thụt đầu dòng cho Lemma, bạn có thể định nghĩa lại:

```
\theoremstyle{changebreak}
\theoremheaderfont{\normalfont\bfseries}\theorembodyfont{\slshape}
\theoremsymbol{\ensuremath{\heartsuit}}
\theoremsymbol{\ensuremath{\diamondsuit}}
\theoremseparator{:}
\theoremindent0.5cm
\theoremnumbering{arabic}
\renewtheorem{Lemma}{Lemma}
```

7 Lemma:

Bây giờ là bổ đề khác, với cách đánh số dùng chữ số Ả Rập. Chú ý rằng chỉ số vẫn tiếp tục tăng. \diamond

Tham số bổ sung xác định tên riêng của THM có thể lấy được nhờ chẳng hạn `\Theoremname` hoặc `Exemplename`,...— điều này chỉ có thể làm bên trong môi trường THM hiện tại mà thôi.

```
\begin{Theorem}[\color{red}{some name}\normalcolor]
Obviously, we are in Theorem~\Theoremname.
\end{Theorem}
```

7 Định lý (some name):

*Obviously, we are in Theorem **some name**.* \diamond

Tính năng có thể dùng, chẳng hạn để sinh ra tự động các mã cho môi trường chú thích, `verbatim`,...:

```
\begin{exercise}[quicksort]
<the exercise text>
\begin{verbatimwrite}{solutions/\exercisename.c}
<C-code>
\end{verbatimwrite}
\verbatiminput{solutions/\exercisename.c}
\end{exercise}
```

Đoạn mã trên sẽ viết mã `<C-code>` vào tập tin `solutions/quicksort.c`, và sau đó nạp vào nhờ lệnh `\verbatiminput`.

Bây giờ, ta định nghĩa môi trường `KappaTheorem` sử dụng cùng kiểu như môi trường `Theorem`, được đánh số tiếp theo các Hệ quả (Corollary) (các `Theorem` cũng được đánh số theo Corollary). Để ý rằng, ta sẽ đưa ra phần header và phần dấu kết thúc khá phức tạp.

```
\theoremclass{Theorem}
\theoremsymbol{\ensuremath{a\atop b}}
\newtheorem{KappaTheorem}[Theorem]{\(\kappa\)-Theorem}
```

8 κ -Theorem (1st κ -Theorem):

Đây là định lý κ đầu tiên.

a
b

6.1 Tham khảo mở rộng

Lệnh `\label` chuẩn được mở rộng: bây giờ nó có thể nhận thêm tham số bổ sung dùng để xác định tên cấu trúc dùng để đánh nhãn – nhờ đó việc tham khảo chéo được linh hoạt hơn. Ví dụ, mục này được đánh nhãn như sau:

```
\subsection*{...}%
\label{sec-ExtRef}[Mu.c]
```

Như đã nói, với môi trường THM, phần tham số bổ sung của `\label` sẽ được tự động thêm vào. Do đó,

```
\begin{Theorem}[Manual End Mark]\label{somelabel}|
```

tương đương với (xem trang 22)

```
\begin{Theorem}[Manual End Mark]\label{somelabel}[Theorem] |
```

Lệnh `\thref{<label>}` sẽ sinh ra thông tin bổ sung, cho biết tên THM tương ứng với tham số bổ sung của `\label`. Ví dụ:

```
This is \thref{sec-ExtRef}
... \thref{somelabel}
... \thref{otherlabel}
... \thref{kappatheorem1}
```

sẽ cho ta

This is Mục 6.1. ... Định lý 6 ... Hệ quả 4 ... κ -Theorem 8

Phải cẩn thận: việc xử lý tham số bổ sung được tiến hành tự động chỉ cho các môi trường được định nghĩa nhờ `\newtheorem`, nghĩa là sẽ không có sự quản lý cho các Mục (`\section`), phương trình (`equation`) hay danh sách (`enumerate`). Việc gọi `\thref{<label>}` cho các nhãn chưa được thiết lập với phần tham số bổ sung sẽ sinh ra các kết quả khó tưởng tượng: nếu `<label>` không ở bên trong môi trường THM, lỗi sẽ sinh ra; ngược lại, tên THM hiện tại sẽ được dùng, ví dụ việc gọi `\thref{xxxlabel}` sẽ sinh ra chẳng hạn “Định lý `<number>`”!

Chú ý rằng không có hỗ trợ cho các tham khảo chéo phức, như “xem Định lý 5 và 7”.⁷

6.2 Danh sách THM

Với

```
\addtotheoremfile{Added into all theorem lists}
```

thì ở mọi danh sách, dòng “Added into all theorem lists” sẽ được chèn vào. Tuy nhiên, trong tài liệu này, ta không làm như thế, vì ta dùng định dạng khác của danh sách THM.

Để chèn chỉ danh sách các ví dụ (Example), có thể làm chẳng hạn

```
\addtotheoremfile[Example]{%
\color{blue}{Only concerning Example lists}}
```

Với

```
\theoremlisttype{all}
\listtheorems{Lemma}
```

⁷This would require plural-forms for different languages and handling of `\ref`-lists, probably splitting into different sublists for different environments. If someone is interested in programming this, please contact us; **it seems to be algorithmically easy, but tedious.**

mọi bỏ đề sẽ được liệt kê

α	16
β	16
γ	17
δ	Biểu thức trong mảng (dãy) phương trình	19
ε	Phương trình	20
ζ	Breakstyle	20
7	23

Như ta thấy trong kết quả trên, xuất hiện dòng chỉ toàn dấu chấm. Ta sử dụng kiểu danh sách `opt` để liệt kê chỉ các bỏ đề có tên riêng:

```
\theoremlisttype{opt}
\listtheorems{Example}
```

cho ta kết quả

0	Extra Entry with number	11
	Extra Entry	11
1	Ví dụ đơn giản	19
	Extra Entry	19
4	dùng ‘verbatim’	21
5	nhanh hơn	22

[Only concerning Example lists](#)

Như ta thấy, dòng chữ “[Only concerning Example lists](#)” xuất hiện ở cuối danh sách — ta đã thêm dòng này nhờ `\addtotheoremfile` ở ví dụ trên.

Bây giờ ta định nghĩa kiểu danh sách mới, sử dụng môi trường `tabular`:

```
\newtheoremlisttype{tab}%
{\begin{tabular*}{\linewidth}%
  {@{}lrl@{\extracolsep{\fill}}r@{}}%
  {##1&##2&##3&##4\\}%
{\end{tabular*}}
```

Sử dụng kiểu mới `tab` như sau:

```
\theoremlisttype{tab}
\listtheorems{Theorem,Lemma}
```

Các định lý và bỏ đề sẽ được liệt kê theo kiểu mới:

Bổ đề	α	16
Bổ đề	β	16
Bổ đề	γ	17
Định lý	1 Định lý dài	19
Bổ đề	δ Biểu thức trong mảng (dãy) phương trình	19
Bổ đề	ε Phương trình	20
Bổ đề	ζ Breakstyle	20
Định lý	3	22
Định lý	5 trích dẫn	22
Định lý	6 Manual End Mark	22
Lemma	7	23
Định lý	7 some name	23

Cũng có thể sử dụng môi trường tạo danh sách của L^AT_EX để tạo kiểu mới như sau đây:

```
\newtheoremstyle{list}%
  {\begin{trivlist}\item}
  {\item[##2 ##1:]\ ##3\dotfill ##4}%
  {\end{trivlist}}
\theoremstyle{list}
\listtheorems{Corollary}
```

Kết quả với kiểu mới bây giờ là

α Bổ đề:	16
β Bổ đề:	16
γ Bổ đề:	17
δ Bổ đề: Biểu thức trong mảng (dãy) phương trình	19
ε Bổ đề: Phương trình	20
ζ Bổ đề: Breakstyle	20
7 Lemma:	23

Trong ví dụ này, sau mỗi phần tử `\item` của danh sách, lệnh `_` được dùng thay cho `_`, vì trong trường hợp sau, lệnh `\dotfill` sẽ báo lỗi nếu phần tham số bổ sung `##3` không có.

Index

A	
<code>\addtheoremline</code>	10
<code>\addtheoremline*</code>	11
<code>\addtotheoremfile</code>	11
<code>amsthm</code> (option)	5
H	
<code>hyperref</code> (option)	5
L	
<code>\listtheorems</code>	10
N	
<code>\newtheorem</code>	6
<code>\newtheoremlisttype</code>	12
<code>\newtheoremstyle</code>	11
<code>noconfig</code> (option)	4
<code>\NoEndMark</code>	13
Q	
<code>\qed</code>	12
<code>\qedsymbol</code>	12
R	
<code>\renewtheorem</code>	6
<code>\renewtheoremlisttype</code>	12
<code>\renewtheoremstyle</code>	12
S	
<code>standard</code> (option)	4
T	
<code>\theorembodyfont</code>	7
<code>\theoremclass</code>	8
<code>\theoremheaderfont</code>	7
<code>\theoremindent</code>	7
<code>\theoremlisttype</code>	10
<code>\theoremnumbering</code>	7
<code>\theorempostskipamount</code>	7
<code>\theoremreskipamount</code>	7
<code>\theoremseparator</code>	7
<code>\theoremstyle</code>	7
<code>\TheoremSymbol</code>	13
<code>\theoremsymbol</code>	7
<code>thmmarks</code> (option)	5
<code>\thref</code>	13
<code>thref</code> (option)	5