

ΛT_EX-Lambda에서 한글 표현

—Hangul-ucs에 이르기까지—

노바디

2006년 5월 22일

차례

1	이 문서에 관하여	2
2	기본 개념	2
2.1	한글 입력 인코딩	2
2.2	ΛT _E X과 Lambda	3
2.3	한글 ΛT _E X 시스템 개요	3
3	EUC-KR ΛT _E X 한글 패키지	4
3.1	hΛT _E Xp	4
3.2	HΛT _E X	9
4	UTF-8 ΛT _E X 한글 패키지	14
4.1	CJK-ΛT _E X	14
4.2	ttfucs	18
4.3	Hangul-ucs	22
5	CP949 Lambda 한글 패키지	23
5.1	hangul-k-cp949	23
6	UTF-8 Lambda 한글 패키지	27
6.1	HΛT _E X/Lambda	27
6.2	DHHangul	32
7	기타	37
7.1	frktex	37
8	결어	41

1 이 문서에 관하여

L^AT_EX에서 한글을 표현하려는¹⁾ 노력이 시작된 지 10여년, 이제는 다양한 한글 L^AT_EX 시스템을 가지게 되었고 그 가운데 일부는 매우 안정적으로 동작하여 일반적인 조판과 출판 목적에 활용할 수 있을 정도가 되었다.

이 글은 2006년 5월 22일 현재 사용 가능한 L^AT_EX과 Lambda에서의 한글 구현 패키지를 일별하고 각 시스템의 특징을 확인하여 현재까지 이루어놓은 성과를 간략하게나마 알아보고 앞으로 한글 L^AT_EX 시스템의 발전 방향에 대한 약간의 시사를 얻고자 하는 것이다.

한글 L^AT_EX 시스템의 개발과 발전에 헌신하신 모든 분께 감사드린다.

2 기본 개념

2.1 한글 입력 인코딩

한글 L^AT_EX 시스템 개발의 초창기부터 문제가 되었던 것은 한글 입력 인코딩이었다. 아직도 혼란의 주요 요인 중 하나가 되고 있는 이 문제를 간략히 정리해 둔다.

EUC-KR 초창기 한글 L^AT_EX 시스템은 대부분 KS C 5601—후에 KS X 1001—완성형 부호화 체계를 기본 한글 입력 인코딩으로 채용하였는데, 이것을 EUC-KR 입력 인코딩이라고 부르기로 하겠다. 이 인코딩으로는 한글 2350자, 한자 4888자 외에 문장부호·약물 등 기호문자와 외국어 기호 문자 등을 표현할 수 있다.

CP949 이른바 “통합 한글 코드(UHC)” 또는 윈도의 “확장완성형”이라 불리는 입력 인코딩으로서 EUC-KR의 수퍼셋이다. EUC-KR 완성형 범위의 한글 문자는 완전히 호환되며, 현대 한글 11,172자를 모두 표현할 수 있다. L^AT_EX의 관점에서 말하자면 CP949 입력된 .tex 파일을 처리하려 할 때 이것을 8비트씩 쪼개어서 처리해야 하는데, EUC-KR 범위에 들지 않는 글자 중에는 T_EX의 control code로 쓰이는 코드 일부와 겹치는 부분이 있어 곤란한 문제가 생긴다. 이 문제를 피해가는 방법은 8비트 단위로 처리하는 T_EX으로는 현실적으로 어려움이 많고 16비트 단위로 처리하는 Omega/Lambda에 의하여야 한다. 즉, 원리상 CP949 입력 인코딩의 처리는 L^AT_EX으로는 불가능하거나 고통스런 트릭이 요구된다. 문제는 이것이, Windows 운영체제의 한글 표현 방법이라는 것이다.²⁾ L^AT_EX에서는 CP949를 무시하거나 EUC-KR만을 한정적으로 지원하거나, 유니코드 인코딩을 채택하는 수밖에 없기 때문에 윈도 운영체제 사용자와는 불화의 요인을 항상 안고 있다 하겠다.

UTF-8 유니코드 부호체계를 인코딩하는 방법은 여러 가지가 있을 수 있겠지만 UTF-8 인코딩만을 문제삼기로 하겠다. 현재 L^AT_EX이나 Lambda가 지원하는 유니코드 부호체계는 BMP 영역, 즉 UCS2이다. 이 부호체계를 사용하면 한글 표현의 제약이 없을 뿐 아니라, 폰트가 지원하기만 한다면 중세국어나 다국어도 하나의 문서에 표현할 수 있기 때문에 사실상 가장 바람직한 한글

1) 이 글에서 한글 ‘표현’이라고 하는 것은 한글 문자의 식자 자체를 가리키는 말이다. higher level typography의 구현은 문자 자체의 식자와는 구별되는 또다른 문제로서, 한글 서식과 타이포그래피의 구현이 가능한 패키지는 별도로 그 사실을 적시하였다.

2) 물론 Windows도 내부적으로는 유니코드를 사용할 것이다. 그러나 사용자 인터페이스 수준에서는 일관되게 CP949

표 1: 한글 \LaTeX -Lambda 시스템

입력	패키지	저자	Year	코멘트
EUC-KR	CJKLaTeX	W. Lemberg	2006	CJK 환경 [KS] 옵션
EUC-KR	HLaTeX	은광희	2005	한글 표시/서식
EUC-KR	hLaTeXp	차재춘	1995	.fmt 만듦. 한글 표시/서식
CP949	hangul-k	DohyunKim & Karnes	2005	Lambda
UTF-8	HLaTeX	은광희	2005	Lambda
UTF-8	Omega-CJK	ChoF & H. Okumura	2004	Lambda
UTF-8	DHHangul	DohyunKim	2005	Lambda, UHC, UTF-16
UTF-8	CJKLaTeX	W. Lemberg	2006	CJK 환경 [UTF8] 옵션
UTF-8	Hangul-ucs	DohyunKim & Karnes	2006	*
UTF-8	ttfucs	O. Cheong	2005	트루타입 글꼴 사용 한글 표시
로마자 입력	frktex	Fukui Rei	1994	원래 pTeX에서 쓰던 것임

부호화 체계라고 생각한다. \LaTeX 이 UTF-8 인코딩을 선호하는 이유는 영문자 영역이 종래 ASCII 코드와 호환되므로 편리하기 때문이다. 한글 영역은 3바이트로 부호화되기 때문에 원본 파일의 크기가 조금 늘어나는 경향은 있으나, 이것이 큰 문제가 되지는 않는다.

2.2 \LaTeX 과 Lambda

Lambda는 Omega의 \LaTeX 판이다. 즉 \LaTeX 시스템을 Omega에서 사용하도록 하는 것이다. Omega는 John Plaice와 Yannis Haralambous에 의하여 개발된 16비트 확장 \TeX 으로서, 16비트 수준의 유니코드 일부를 다룰 수 있다.³⁾ 다국어 처리의 획기적인 진전으로 기대를 한몸에 모았지만 안타깝게도 2002년 이후 개발이 중단되어 있다.⁴⁾

한글 처리에 있어서도 2002년경 Omega와 Lambda는 사실상 유일한 대안, 또는 한글 \TeX 의 미래로까지 인식되었던 적이 있다. 그러나 Omega의 발전을 기다리기 난망해진 지금 Omega/Lambda 한글 구현에 대한 관심이 조금 뒷전으로 밀려난 느낌이 든다. 그렇기는 하여도 현실적으로 중세 한글과 같은 문제는 아직까지 Omega/Lambda 이외의 마땅한 방법이 없는 실정이다.

2.3 한글 \LaTeX 시스템 개요

한글 \LaTeX 의 기원 또는 여명에 해당하는 초창기 매크로 패키지나 한글 시스템들에 대해서 별도로 다루지는 않는다. 우리가 보기에 $\text{\LaTeX}2_{\epsilon}$ 에서 한글을 구현하는 데 성공한 최초의 안정적 시스템은 h \LaTeX p라고 생각할 수밖에 없다. 그 이전의 전처리기를 이용한 한글 처리 방식은 진정한 의미에서 한글 \LaTeX 이라고 보기 어렵다는 생각이다.

현재 사용가능한 한글 \LaTeX 시스템을 요약하면 표 1과 같다. 이 가운데 주요한 몇 가지 패키지에서 한글을 사용하는 방법을 예시하고 각 시스템의 특징을 요약해 보자.

를 채택하고 있다.

3) 유니코드 자체는 16비트가 아니다. 그러나 Omega가 다루는 유니코드는 16비트 영역으로 제한된다.

4) 다만 Aleph라고 불리는 새로운 Omega(e-Omega)가 이를 이어받았으나 기대만큼의 발전을 보이지는 못하고 있다.

3 EUC-KR \LaTeX 한글 패키지

3.1 $\hbox{\LaTeX}$

Preamble $\hbox{\LaTeX}$ 시스템은 preamble에 아무런 설정을 하지 않고도 한글을 쓸 수 있다. 그 이유는 `hlatex.fmt`라는 포맷 파일을 만들어두고 한글 처리에 관한 매크로를 여기에 미리 포함시켜 두고 있기 때문이다. 한편, 한글 문서 서식(자동조사 등)을 자동화해주는 `hangul.sty`이라는 패키지가 별도로 있다.

```
\documentclass{article}
\usepackage{hangul}
```

컴파일 `hlatex.fmt` 포맷 파일을 불러주어야 한다. 파일 이름이 `foo.tex`이라 할 때,

```
#> tex "&hlatex" foo
```

특징

- EUC-KR 입력된 한글을 처리한다.
- 한글 label, 자동조사, 한글화된 서식을 제공한다.
- 에러 메시지의 한글 표시를 가능하게 하는 유틸리티를 제공한다.
- 비교적 다양한 한글 글꼴을 제공한다.

코멘트

- 배포되는 글꼴이 pk bitmap font이다. 따라서 주어진 해상도(1200dpi, 600dpi, 300dpi)의 프린터 이외에는 사용할 수 없고 pdf와 같은 온라인 문서를 만들면 그다지 아름답지 못하다.⁵⁾
- `hyperref` 패키지 등을 이용하여 하이퍼링크를 구현하는 등 pdf 제작에는 한계가 있다.
- 영문 글꼴이 오직 Computer Modern만을 사용할 수 있다. 요즘의 다양한 글꼴에 대한 요구를 충족시켜주지 못한다.
- 한글 글꼴도 새로운 폰트를 추가할 수 없다.
- 1995년 이후 개선되지 않았기 때문에 최근의 패키지들과 충돌이 있을는지 알 수 없다.

예제 문서의 소스

다음 문서는 EUC-KR 인코딩으로 작성된 것이다. 출력은 600dpi 폰트를 사용하였는데, pdf로 보면 좀 그래도 600dpi 프린터에서 인쇄한 결과는 나쁘지 않다.

5) 비록 글꼴이 동일하지는 않으나 샘플로소프트의 TeXplus 뷰어를 설치하여 이것으로 $\hbox{\LaTeX}$ 문서를 읽으면 윈도 트루타입으로 치환되어 나타나기 때문에 이로부터 윈도 트루타입 윤곽선 글꼴을 이용하는 pdf를 제작하는 것이 불가능하지는 않다. 그러나 윈도 시스템에만 특정되어 있다는 점과, 원래 작성한 문서의 폰트와는 다른 폰트가 사용된다는 것은 생각해볼 점이다.

```

\documentclass{article}
\usepackage{hangulp}

\title{무한 합의 계산}
\author{케이스 데블린}
\date{}

\begin{document}

\maketitle

\begin{abstract}
이 문서는 ‘‘한글’’을 표현하는 다양한 방법을 테스트하기 위한 것이다.
텍스트는 \cite{millen}에서 가져왔으며,
h\LaTeX{p}의 서식 옵션 스타일인 \texttt{[hangul]}만을
부르고 작성하였다.
\end{abstract}

\section{수학자들은}
수학자들은 제타( $\zeta$ ) 함수에 있는 것과 같은 무한 합을 어떻게
계산하는 것일까? 더 기초적으로 이렇게 물을 수도 있다: 무한한 덧셈이
어떻게 유한한 결과를 산출할 수 있을까?

무한 합(무한급수) 속에 있는 항들이 자연수의 제곱들인 경우를
생각해보자.
\[
1+4+9+16+25+\cdots
\]
덧셈을 통해서 얻어지는 값들을 차례로 나열하면 다음과 같다.
 $1+4=5$ ,  $5+9=14$ ,  $14+16=30$ ,  $30+25=55$  등등. 이 부분 합들이
빠르게 커지므로, 전체 무한 합은 무한한 결과를 산출할 것이다.

이번에는 모든 자연수들의 제곱의 역수를 항으로 가진 무한 합을 생각해 보자.
\[
\frac{1}{1}+\frac{1}{4}+\frac{1}{9}+\frac{1}{16}+\frac{1}{25}+\cdots
\]
즉,
\[
1+0.25+0.1111+0.0625+0.04+\cdots
\]
이 무한 합의 부분 합들은 차례대로,  $1$ ,  $1.25$ ,  $1.36111$ ,  $1.42361$ ,
 $1.44361$  등이다. 부분 합들은 증가하지만, 증가량은 점점 작아진다. 따라서 무한
합이 어떤 유한 값을 가질 가능성이 있어 보인다.

\section{컴파일 방법}

```

h\LaTeX{p}는 다음과 같이 컴파일해야 한다.

```
\begin{verbatim}
```

```
#> tex "&hlatex" foo
```

```
\end{verbatim}
```

물론 \texttt{hlatex.fmt}라는 format 파일이 미리 작성되어 있어야 한다.

```
\begin{thebibliography}{0}
```

```
\bibitem{millen} 케이스 데블린. 전대호 옮김. 『수학의 밀레니엄 문제들』.  
도서출판 까치. 2004.
```

```
\end{thebibliography}
```

```
\end{document}
```

무한 합 계산

케이스 데블린

요약문

이 문서는 “한글”을 표현하는 다양한 방법을 테스트하기 위한 것이다. 텍스트는 [1]에서 가져왔으며, \LaTeX 의 서식 옵션 스타일인 `[hangu1]`만을 부르고 작성하였다.

1 수학자들은

수학자들은 제타(ζ) 함수에 있는 것과 같은 무한 합을 어떻게 계산하는 것일까? 더 기초적으로 이렇게 물을 수도 있다: 무한한 덧셈이 어떻게 유한한 결과를 산출할 수 있을까?

무한 합(무한급수) 속에 있는 항들이 자연수의 제곱들인 경우를 생각해 보자.

$$1 + 4 + 9 + 16 + 25 + \cdots$$

덧셈을 통해서 얻어지는 값들을 차례로 나열하면 다음과 같다. $1 + 4 = 5$, $5 + 9 = 14$, $14 + 16 = 30$, $30 + 25 = 55$ 등등. 이 부분 합들이 빠르게 커지므로, 전체 무한 합은 무한한 결과를 산출할 것이다.

이번에는 모든 자연수들의 제곱의 역수를 항으로 가진 무한 합을 생각해 보자.

$$\frac{1}{1} + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \cdots$$

즉,

$$1 + 0.25 + 0.11111 + 0.0625 + 0.04 + \cdots$$

이 무한 합의 부분 합들은 차례대로, 1, 1.25, 1.36111, 1.42361, 1.44361 등이다. 부분 합들은 증가하지만, 증가량은 점점 작아진다. 따라서 무한 합이 어떤 유한 값을 가질 가능성이 있어 보인다.

2 컴파일 방법

\LaTeX 는 다음과 같이 컴파일해야 한다.

```
#> tex "&hlatex" foo
```

물론 `hlatex.fmt`라는 format 파일이 미리 작성되어 있어야 한다.

참고문헌

[1] 케이스 데블린, 전대호 옮김, 『수학의 밀레니엄 문제들』, 도서출판 까치, 2004.

3.2 H_AT_EX

Preamble 두 종류의 스타일 패키지를 제공한다. `hfont` 패키지는 영문 문서에 한글을 “표시”만 할 수 있게 하는 것으로 영문 패키지와의 충돌을 최소화할 수 있으나 자동조사 등 한글화 기능이 동작하지 않고 `hangul` 패키지는 한글 서식과 한글 문서화 기능을 포함하고 있는 것이다.

```
\documentclass{article}
\usepackage{hangul} % 또는 \usepackage{hfont}
```

컴파일 `latex` 또는 `pdflatex`으로 컴파일할 수 있다.

```
#> pdflatex foo
```

특징

- 한글 서식 관련 풍부한 기능을 제공한다. 예를 들면 각주 스타일의 설정, 장절 항번의 한글화, 임의 자간 설정, 한글 non french spacing 구현, 페이지 양식의 한글화 등.⁶⁾
- 근 10여년간 가장 많은 일반 사용자를 가진 패키지이다.
- 유일한 공개 한글 Type 1 글꼴인 UHC 글꼴을 채택하고 있다. 그 결과 `dvips`/`pdflatex` 등으로도 윤곽선 글꼴을 포함하는 pdf를 제작할 수 있다.
- 많은 사람들의 노력으로 H_AT_EX의 기능을 여러 가지로 확장해두었다.
- 한글 label, 한글 글꼴 설정 매크로, 자동조사 등 한글화된 명령을 제공한다.
- 새로운 폰트를 추가하는 것이 가능하고 영문 글꼴과 한글 글꼴이 완전히 분리되어 있어 영문 글꼴의 선택 폭도 자유롭다.
- `hmakeindex`, `hbibtex`과 같은 색인 및 참고문헌 만들기의 한글화 유틸리티를 제공한다. `hind.ist`나 `halpha.bst`와 같은 스타일도 함께 제공된다.
- 다양한 글꼴이 제공되며, UHC 글꼴은 은글꼴의 바탕이 되었다.

코멘트

- `hyperref`과의 충돌 문제 때문에 `hangul` 패키지에서 하이퍼링크나 pdf 북마크 등을 만들 수가 없다.⁷⁾
- UHC 글꼴이 가끔 고해상도 Postscript 프린터에서 이상한 출력을 보일 때가 있다. 해결방법은 있지만 초보자들이 당황하게 만드는 요인이기도 하다.
- UHC 글꼴의 특성 때문에 pdf의 검색 추출 가능성이 제한된다.
- `AMS-LATEX` 등과 함께 쓰기가 어렵다.

6) 이 대부분이 H_AT_EX 1.0.1에서 구현된 것들이다.

7) `khyper`라는 패키지가 제공되기는 하지만 `hyperref`의 기능을 모두 쓸 수 있게 하는 것은 아니고 컴파일이 이상 없이 진행되도록 하는 정도의 기능만을 제공한다.

KTUG에서 이루어진 \LaTeX 의 확장

KTUG에서는 2002년에서 2003년 사이 \LaTeX 의 기능을 보충하거나 확장하는 개발이 이루어졌다. 그 중 중요한 것만 지적하면 다음과 같다.

hangul-k 김도현 & 김강수의 이 스타일은 \LaTeX 의 `hangul` 스타일을 거의 그대로 채택하되, 자동조사 기능을 새로 구현하여 `hyperref`과의 충돌을 제거한 것이다. 따라서 `hyperref`의 모든 기능을 자유롭게 사용하면서 `hangul`의 기능도 함께 사용할 수 있게 한 \LaTeX 확장 패키지이다. 이 스타일의 제작 경험은 그대로 `Hangul-ucs`로 이어졌다.

DVIPDFMx와 트루타입 글꼴 pdf 제작에 있어서 한글 텍스트의 검색 및 치환 가능성, 그리고 pdf 책갈피 제작 문제를 트루타입 글꼴 활용 기술과 조진환의 `DVIPDFMx`를 통하여 해결하였다. 트루타입 글꼴은 마침 박원규의 은글꼴이 공개되어 일반적인 목적에 사용할 수 있게 되었다.

한글 타이포그래피 김강수의 `hlatex-interword`, 도이기의 `hfncychap`와 같은 스타일을 통하여 한글의 자간, 행간, 장절 서식 등을 보완하였다.

예제 문서의 소스

다음 문서는 EUC-KR 인코딩으로 작성된 것이다. 다른 추가 패키지나 폰트를 쓰지 않고 `hangul` 패키지에 기본 폰트(UHC 명조)만을 사용하였다.

```
\documentclass{article}
\usepackage{hangul}

\title{무한 합의 계산}
\author{케이스 데블린}
\date{}

\begin{document}

\maketitle

\begin{abstract}
이 문서는 ‘‘한글’’을 표현하는 다양한 방법을 테스트하기 위한 것이다.
텍스트는 \cite{millen}에서 가져왔으며,
H $\text{\LaTeX}$ 의 서식 옵션 스타일인 \texttt{[hangul]}만을
부르고 작성하였다.
\end{abstract}

\section{수학자들은}
수학자들은 제타( $\zeta$ ) 함수에 있는 것과 같은 무한 합을 어떻게
계산하는 것일까? 더 기초적으로 이렇게 물을 수도 있다: 무한한 덧셈이
어떻게 유한한 결과를 산출할 수 있을까?

무한 합(무한급수) 속에 있는 항들이 자연수의 제곱들인 경우를
```

생각해보자.

\[

$1+4+9+16+25+\cdots$

\]

덧셈을 통해서 얻어지는 값들을 차례로 나열하면 다음과 같다.

$\$1+4=5\$$, $\$5+9=14\$$, $\$14+16=30\$$, $\$30+25=55\$$ 등등. 이 부분 합들이 빠르게 커지므로, 전체 무한 합은 무한한 결과를 산출할 것이다.

이번에는 모든 자연수들의 제곱의 역수를 항으로 가진 무한 합을 생각해 보자.

\[

$\frac{1}{1}+\frac{1}{4}+\frac{1}{9}+\frac{1}{16}+\frac{1}{25}+\cdots$

\]

즉,

\[

$1+0.25+0.11111+0.0625+0.04+\cdots$

\]

이 무한 합이 부분 합들은 차례대로, $\$1\$$, $\$1.25\$$, $\$1.36111\$$, $\$1.42361\$$, $\$1.44361\$$ 등이다. 부분 합들은 증가하지만, 증가량은 점점 작아진다. 따라서 무한 합이 어떤 유한 값을 가질 가능성이 있어 보인다.

\begin{thebibliography}{0}

\bibitem{millen} 케이스 데블린. 전대호 옮김. 『수학의 밀레니엄 문제들』. 도서출판 까치. 2004.

\end{thebibliography}

\end{document}

무한 합의 계산

케이스 데블린

요약

이 문서는 “한글”을 표현하는 다양한 방법을 테스트하기 위한 것이다. 텍스트는 [1] 에서 가져왔으며, \LaTeX 의 서식 옵션 스타일인 [hangul] 만을 부르고 작성하였다.

제 1 절 수학자들은

수학자들은 제타(ζ) 함수에 있는 것과 같은 무한 합을 어떻게 계산하는 것일까? 더 기초적으로 이렇게 물을 수도 있다: 무한한 덧셈이 어떻게 유한한 결과를 산출할 수 있을까?

무한 합(무한급수) 속에 있는 항들이 자연수의 제곱들인 경우를 생각해 보자.

$$1 + 4 + 9 + 16 + 25 + \cdots$$

덧셈을 통해서 얻어지는 값들을 차례로 나열하면 다음과 같다. $1 + 4 = 5$, $5 + 9 = 14$, $14 + 16 = 30$, $30 + 25 = 55$ 등등. 이 부분 합들이 빠르게 커지므로, 전체 무한 합은 무한한 결과를 산출할 것이다.

이번에는 모든 자연수들의 제곱의 역수를 항으로 가진 무한 합을 생각해 보자.

$$\frac{1}{1} + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \cdots$$

즉,

$$1 + 0.25 + 0.11111 + 0.0625 + 0.04 + \cdots$$

이 무한 합의 부분 합들은 차례대로, 1, 1.25, 1.36111, 1.42361, 1.44361 등이다. 부분 합들은 증가하지만, 증가량은 점점 작아진다. 따라서 무한 합이 어떤 유한 값을 가질 가능성이 있어 보인다.

참고 문헌

- [1] 케이스 데블린, 전대호 옮김. 『수학의 밀레니엄 문제들』. 도서출판 까치.
2004.

4 UTF-8 \LaTeX 한글 패키지

4.1 CJK- \LaTeX

Preamble, Environment CJK- \LaTeX 에서 한글을 쓰려면 두 가지 조치를 취하여야 한다. 하나는 CJK 패키지를 로드하는 것이고 다른 하나는 한글이 들어간 부분을 CJK 환경 안에 두는 것이다.

```
\documentclass{article}
\usepackage{CJK}

\begin{document}
\begin{CJK}[HL]{KS}{-}
% 한글은 여기에 옴.
\end{CJK}
\end{document}
```

컴파일 특별한 것은 없다. 다만 dvips/ps2pdf나 pdf \LaTeX 을 사용하려면 이들 드라이버가 처리할 수 있는 폰트를 이용해야 한다. 일반적으로 latex/dvipdfmx를 쓰는 것이 가장 안전하다.

```
#> latex foo
#> dvipdfmx foo
```

특징

- 이 패키지는 한글만을 위한 것이 아니라 중국어, 일본어, 베트남어, 타이어 등을 모두 지원하고 있다.
- 한글에 대해서도 EUC-KR과 UTF-8 등의 문자 인코딩을 다양하게 지원한다. 일본어에 대해서도 마찬가지이다.
- 대부분의 \TeX implementation에 포함되어 있고 비교적 간편하게 한글을 식자할 수 있다.
- 확장성이 뛰어나서 필요하다면 자신에게 알맞은 폰트를 적용하는 것이 쉽다.
- 한국어를 모어로 하지 않는 외국인이 한글 문서를 식자하거나 한글을 문서의 일부에 표시할 때 주로 사용한다.

코멘트

- 한글 서식이나 타이포그래피를 구현하려면 상당한 보충적 코딩이 필요하다.
- 자동조사 등 한국어에 특정한 기능이 충분히 구현되어 있지 않다.
- 기본으로 함께 제공되는 폰트가 빈약하여 자신에게 알맞은 폰트 세트를 별도로 조성하여야 하는 경우가 많다.

- 이유는 확실치 않으나⁸⁾ 이 시스템으로 한글 문서를 작성하는 한국어 모어 사용자를 별로 보지 못하였다.

예제 문서의 소스

이 예제에서는 UTF8 환경을 사용하였고, 폰트는 은글꼴(은바탕)을 채택하였다. Hangul-ucs에서 설치해주는 은글꼴을 CJK- \LaTeX 에서 약간의 설정만 추가하면 그대로 사용할 수 있기 때문이다. CJK- \LaTeX 의 UTF8 환경은 설치 후 바로 사용할 수가 없고 cyberbit이든 ms-batang이든 폰트를 별도로 조성해주어야 한다. 아래 예제가 컴파일되려면 Hangul-ucs의 은글꼴 폰트 설정이 되어 있는 상태에서 C70unbt.fdx라는 font definition 파일을 하나 작성해주면 된다. 자세한 사항은 KTUG-Faq:CJKLaTeX 페이지를 참고하라.

```
\documentclass{article}
\usepackage{CJK}

\begin{document}
\begin{CJK}{UTF8}{unbt}

\title{무한 합의 계산}
\author{케이 스 데블린}
\date{}

\maketitle

\begin{abstract}
이 문서는 ‘‘한글’’을 표현하는 다양한 방법을 테스트하기 위한 것이다.
텍스트는 \cite{millen}에서 가져왔으며,
CJK- $\LaTeX$  의 \textsf{UTF8} 환경으로 작성하였다.
\end{abstract}

\section{수학자들은}
수학자들은 제타( $\zeta$ ) 함수에 있는 것과 같은 무한 합을 어떻게
계산하는 것일까? 더 기초적으로 이렇게 물을 수도 있다: 무한한 덧셈이
어떻게 유한한 결과를 산출할 수 있을까?

무한 합(무한급수) 속에 있는 항들이 자연수의 제곱들인 경우를
생각해보자.
\[
1+4+9+16+25+\cdots
\]
덧셈을 통해서 얻어지는 값들을 차례로 나열하면 다음과 같다.
 $1+4=5$ ,  $5+9=14$ ,  $14+16=30$ ,  $30+25=55$  등등. 이 부분 합들이
빠르게 커지므로, 전체 무한 합은 무한한 결과를 산출할 것이다.
```

8) 아마도 H \LaTeX 이라는 충분히 만족스러운 한글 \LaTeX 이 있었기 때문이 아닐까 싶다.

이번에는 모든 자연수들의 제곱의 역수를 항으로 가진 무한 합을 생각해 보자.

```
\[
\frac{1}{1}+\frac{1}{4}+\frac{1}{9}+\frac{1}{16}+\frac{1}{25}+\cdots
```

```
\]
즉,
```

```
\[
1+0.25+0.11111+0.0625+0.04+\cdots
```

```
\]
이 무한 합을의 부분 합들은 차례대로, $1$, $1.25$, $1.36111$, $1.42361$,
$1.44361$ 등이다. 부분 합들은 증가하지만, 증가량은 점점 작아진다. 따라서 무한
합이 어떤 유한 값을 가질 가능성이 있어 보인다.
```

`\section{이 시스템의 특징}`

CJK-`\LaTeX` 은 비단 한글 UTF-8 문서뿐 아니라 EUC-KR 문서까지 잘 처리하며,
일본어와 중국어 등도 비교적 훌륭하게 처리한다.

```
\begin{thebibliography}{0}
```

```
\bibitem{millen} 케이스 데블린. 전대호 옮김. 『수학의 밀레니엄 문제들』.
도서출판 까치. 2004.
```

```
\end{thebibliography}
```

```
\end{CJK}
```

```
\end{document}
```


무한 합의 계산

케이스 데블린

Abstract

이 문서는 “한글”을 표현하는 다양한 방법을 테스트하기 위한 것이다. 텍스트는 [1]에서 가져왔으며, CJK- \LaTeX 의 UTF8 환경으로 작성하였다.

1 수학자들은

수학자들은 제타(ζ) 함수에 있는 것과 같은 무한 합을 어떻게 계산하는 것일까? 더 기초적으로 이렇게 물을 수도 있다: 무한한 덧셈이 어떻게 유한한 결과를 산출할 수 있을까?

무한 합(무한급수) 속에 있는 항들이 자연수의 제곱들인 경우를 생각해 보자.

$$1 + 4 + 9 + 16 + 25 + \dots$$

덧셈을 통해서 얻어지는 값들을 차례로 나열하면 다음과 같다. $1 + 4 = 5$, $5 + 9 = 14$, $14 + 16 = 30$, $30 + 25 = 55$ 등등. 이 부분 합들이 빠르게 커지므로, 전체 무한 합은 무한한 결과를 산출할 것이다.

이번에는 모든 자연수들의 제곱의 역수를 항으로 가진 무한 합을 생각해 보자.

$$\frac{1}{1} + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \dots$$

즉,

$$1 + 0.25 + 0.11111 + 0.0625 + 0.04 + \dots$$

이 무한 합의 부분 합들은 차례대로, 1, 1.25, 1.36111, 1.42361, 1.44361 등이다. 부분 합들은 증가하지만, 증가량은 점점 작아진다. 따라서 무한 합이 어떤 유한 값을 가질 가능성이 있어 보인다.

2 이 시스템의 특징

CJK- \LaTeX 은 비단 한글 UTF-8 문서뿐 아니라 EUC-KR 문서까지 잘 처리하며, 일본어와 중국어 등도 비교적 훌륭하게 처리한다.

References

- [1] 케이스 데블린. 전대호 옮김. 『수학의 밀레니엄 문제들』. 도서출판 까치. 2004.

4.2 ttfucs

LaTeX에서 UTF-8 인코딩 입력을 처리하는 데는 babel 패키지의 utf8.def와 Dominique Unruh의 unicode 패키지를 이용하는 방법이 있다. 이 가운데 요즘 거의 표준으로 받아들여지고 있는 것은 unicode 패키지 또는 그 일부인 utf8x.def를 이용하는 것이다. unicode 패키지가 비록 개발이 중단되었다고는 하나 UCS2 범위의 문자를 충분히 처리할 수 있을 정도의 능력을 이미 갖추고 있다.

ttfucs는 unicode 패키지의 utf8x.def를 이용하면서 유니코드 인코딩의 트루타입 폰트를 직접 조판에 이용할 수 있게 해준다. 따라서 한글 트루타입 폰트를 가지고 있다면 이것을 이용하여 한글 문자를 식자할 수 있게 되는 것이다.

Preamble utf8x.def를 부르는 과정, 트루타입 폰트를 설정하는 과정으로 이루어진다. 다음 보기는 은바탕 트루타입을 설정하는 예이다.

```
\documentclass{article}
\usepackage[utf8x]{inputenc}
\usepackage{ttfucs}
\pdfmapfile{+unttf-pdftex-dhucs.map}

\DeclareTruetypeFont{unbt}{ounbtm}
\DeclareTruetypeFontShape{unbt}{m}{n}{ounbtm}
\DeclareTruetypeFontShape{unbt}{bx}{n}{ounbtb}
\DeclareTruetypeFontShape{unbt}{m}{it}{ounbtmo}
\DeclareTruetypeFontShape{unbt}{m}{sl}{ounbtmo}
```

컴파일 이 패키지는 오직 pdfLaTeX만을 지원하기 때문에 latex과 dvipdfmx를 이용한 루트는 사용할 수 없다. 물론 dvips/ps2pdf도 안된다.

특징

- 단순하고 사용이 쉽기 때문에 복잡하지 않은 문서를 작성하는 데 적당하다.
- pdfLaTeX에서 트루타입을 사용해야 하는 경우 적당하다.

코멘트

- 한글화 패키지라고는 할 수 없다. 다만 한글 트루타입을 이용할 수 있을 뿐이다.
- 새로운 폰트를 사용하게 만드는 데 약간의 추가적인 노력이 필요하다.

예제 파일의 소스

다음 예제는 은바탕을 이용하는 것이다. 새로운 폰트를 사용하게 하려면 ttf2tfm을 이용하여 tfm 파일과 enc, map 파일을 만들면 된다.

```

\documentclass{article}
\usepackage[utf8x]{inputenc}
\usepackage{tffucs}
\pdfmapfile{+unttf-pdftex-dhucs.map}

\DeclareTruetypeFont{unbt}{ounbtm}
\DeclareTruetypeFontShape{unbt}{m}{n}{ounbtm}
\DeclareTruetypeFontShape{unbt}{bx}{n}{ounbtb}
\DeclareTruetypeFontShape{unbt}{m}{it}{ounbtmo}
\DeclareTruetypeFontShape{unbt}{m}{sl}{ounbtmo}

\title{무한 합의 계산}
\author{케이스 데블린}
\date{}

\begin{document}

\maketitle

\begin{abstract}
이 문서는 ‘‘한글’’을 표현하는 다양한 방법을 테스트하기 위한 것이다.
텍스트는 \cite{millen}에서 가져왔으며,
\txttt{tffucs} 패키지로 은바탕 글꼴을 이용하여
작성하였다.
\end{abstract}

\section{수학자들은}
수학자들은 제타( $\zeta$ ) 함수에 있는 것과 같은 무한 합을 어떻게
계산하는 것일까? 더 기초적으로 이렇게 물을 수도 있다: 무한한 덧셈이
어떻게 유한한 결과를 산출할 수 있을까?

무한 합(무한급수) 속에 있는 항들이 자연수의 제곱들인 경우를
생각해보자.
\[
1+4+9+16+25+\cdots
\]
덧셈을 통해서 얻어지는 값들을 차례로 나열하면 다음과 같다.
 $1+4=5$ ,  $5+9=14$ ,  $14+16=30$ ,  $30+25=55$  등등. 이 부분 합들이
빠르게 커지므로, 전체 무한 합은 무한한 결과를 산출할 것이다.

이번에는 모든 자연수들의 제곱의 역수를 항으로 가진 무한 합을 생각해 보자.
\[
\frac{1}{1}+\frac{1}{4}+\frac{1}{9}+\frac{1}{16}+\frac{1}{25}+\cdots
\]
즉,

```

```

\[
1+0.25+0.1111+0.0625+0.04+\cdots
\]
이 무한 합을의 부분 합들은 차례대로,  $1$ ,  $1.25$ ,  $1.36111$ ,  $1.42361$ ,
 $1.44361$  등이다. 부분 합들은 증가하지만, 증가량은 점점 작아진다. 따라서 무한
합이 어떤 유한 값을 가질 가능성이 있어 보인다.

\begin{thebibliography}{0}
\bibitem{millen} 케이스 데블린. 전대호 옮김. 『수학의 밀레니엄 문제들』.
도서출판 까치. 2004.
\end{thebibliography}

\end{document}

```

무한 합의 계산

케이스 데블린

Abstract

이 문서는 “한글”을 표현하는 다양한 방법을 테스트하기 위한 것이다. 텍스트는 [1]에서 가져왔으며, `ttfucs` 패키지로 은바탕 글꼴을 이용하여 작성하였다.

1 수학자들은

수학자들은 제타(ζ) 함수에 있는 것과 같은 무한 합을 어떻게 계산하는 것일까? 더 기초적으로 이렇게 물을 수도 있다: 무한한 덧셈이 어떻게 유한한 결과를 산출할 수 있을까?

무한 합(무한급수) 속에 있는 항들이 자연수의 제곱들인 경우를 생각해보자.

$$1 + 4 + 9 + 16 + 25 + \cdots$$

덧셈을 통해서 얻어지는 값들을 차례로 나열하면 다음과 같다. $1 + 4 = 5$, $5 + 9 = 14$, $14 + 16 = 30$, $30 + 25 = 55$ 등등. 이 부분 합들이 빠르게 커지므로, 전체 무한 합은 무한한 결과를 산출할 것이다.

이번에는 모든 자연수들의 제곱의 역수를 항으로 가진 무한 합을 생각해 보자.

$$\frac{1}{1} + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \cdots$$

즉,

$$1 + 0.25 + 0.11111 + 0.0625 + 0.04 + \cdots$$

이 무한 합의 부분 합들은 차례대로, 1, 1.25, 1.36111, 1.42361, 1.44361 등이다. 부분 합들은 증가하지만, 증가량은 점점 작아진다. 따라서 무한 합이 어떤 유한 값을 가질 가능성이 있어 보인다.

References

- [1] 케이스 데블린. 전대호 옮김. 『수학의 밀레니엄 문제들』. 도서출판 까치. 2004.

4.3 Hangul-ucs

unicode 패키지를 이용한다는 점에서는 `ttfucs`와 같지만, `Hangul-ucs`는 지금까지 논의된 거의 대부분의 한글화 관련 이슈를 집대성한 매크로 패키지이다. 특히 \LaTeX 의 고질적인 문제였던 한글과 영문을 섞어쓰는 경우의 자간 분리, 그리고 pdf 제작에 특화된 기능들 이외에도, 현재 한글 문서 작성 상의 요구에 대부분 대응하고 있다고 생각한다.

\LaTeX 의 계승

- \LaTeX 의 `hangul.sty`가 가지고 있던 타이포그래피 정의를 물려받았다.
- 한글 방점과 같은 \LaTeX 의 기능을 동일하거나 유사하게 구현하고 있다. 즉, \LaTeX 에 있는 기능은 `Hangul-ucs`에도 있다.
- `hangul-k`에서 시작된 \LaTeX 개선 경험이 반영되어 있다.
- \LaTeX 의 간편한 사용 방식을 물려받았다.

타이포그래피의 강화

- \LaTeX 에 한글 타이포그래피를 구현한 경험이 흡수되었다.
- 괄호 앞에서 행나눔, 괄호 안 영문 단어의 하이픈처리와 같은 기초적 타이포그래피 기능을 구현하였다.
- 자간, 행간, 장절항변 설정 등 한글 타이포그래피를 구현하고 있다.
- 폰트 활용 가능성의 폭을 대폭 넓혔다.

호환성의 강화

- 영문 패키지와의 충돌을 최소한으로 줄였다.
- 특히 $\mathcal{AMS}\text{-}\text{\LaTeX}$ 과 완벽하게 호환된다.
- `beamer`, `(new)prosper` 등의 패키지와의 충돌이 없다.
- 특히, `memhangul` 패키지를 통하여 `memoir` 클래스 문서를 작성할 수 있다.

pdf 제작에 특화

- 한글 북마크가 완전하게 구현된다.
- `DVIPDFMx`는 물론이고 `pdf \LaTeX` 에서도 한글 텍스트의 검색과 추출을 가능하게 하였다.
- `type 1`, 트루타입 폰트를 이용하여 pdf를 제작할 수 있다.
- 하이퍼링크와 같은 기본적인 pdf 제작 기능을 완벽하게 지원한다.

예제 `Hangul-ucs`에 대해서 별도의 예는 들지 않는다. 이 문서 자체가 예제이다.

5 CP949 Lambda 한글 패키지

5.1 hangul-k-cp949

hangul-k-cp949는 KTUG에서 이루어진 CP949 식자를 위한 테스트의 결과물 중 하나이다. 원래 이것은 은광희, 신정식 님 등이 Lambda와 \LaTeX 을 이용한 한글 식자를 테스트하는 과정에서 (부산물로) 나온 것이었는데 \LaTeX 이 이 가운데 CP949 지원 부분을 제외하고 UTF-8만을 채택하게 되었기 때문에 폐기된 부분을 hangul-k 패키지가 흡수하여 유지하고 있는 것이라고 할 수 있다.

Preamble 다음과 같이 하여 UTF-8 문서를 작성하면 된다.

```
\documentclass{article}
\usepackage{hangul-k}
```

컴파일 Lambda로 컴파일한다.

```
#> lambda foo
```

특징

- 현대 한글 11,172자를 모두 표현할 수 있다.
- \LaTeX /Lambda와 동일하다 입력 인코딩이 CP949이다.
- 영문 글꼴은 한글 글꼴의 영문 부분을 이용하여 식자한다. 즉 한글 글꼴과 영문 글꼴이 분리되어 있지 않다.
- Lambda 시스템의 특징 때문에 pdf 제작했을 때 하이퍼링크나 (트루타입 폰트를 이용한 경우) 검색 추출은 문제없지만 책갈피는 만들 수 없다.
- pdfLambda가 없기 때문에 `lambda` → `dvipdfmx` 루트로 pdf를 만든다.

예제 문서의 소스

```
\documentclass{article}
\usepackage{hangul-k}

\title{무한 합의 계산}
\author{케이스 데블린}
\date{}

\begin{document}

\maketitle
```

```

\begin{abstract}
이 문서는 ‘‘한글’’을 표현하는 다양한 방법을 테스트하기 위한 것이다.
텍스트는 \cite{millen}에서 가져왔으며,
hangul-k 스타일을 써서 CP949 인코딩의 문서를
작성하였다.
\end{abstract}

\section{수학자들은}
수학자들은 제타( $\zeta$ ) 함수에 있는 것과 같은 무한 합을 어떻게
계산하는 것일까? 더 기초적으로 이렇게 물을 수도 있다: 무한한 덧셈이
어떻게 유한한 결과를 산출할 수 있을까?

무한 합(무한급수) 속에 있는 항들이 자연수의 제곱들인 경우를
생각해보자.
\[
1+4+9+16+25+\cdots
\]
덧셈을 통해서 얻어지는 값들을 차례로 나열하면 다음과 같다.
 $1+4=5$ ,  $5+9=14$ ,  $14+16=30$ ,  $30+25=55$  등등. 이 부분 합들이
빠르게 커지므로, 전체 무한 합은 무한한 결과를 산출할 것이다.

이번에는 모든 자연수들의 제곱의 역수를 항으로 가진 무한 합을 생각해 보자.
\[
\frac{1}{1}+\frac{1}{4}+\frac{1}{9}+\frac{1}{16}+\frac{1}{25}+\cdots
\]
즉,
\[
1+0.25+0.1111+0.0625+0.04+\cdots
\]
이 무한 합의 부분 합들은 차례대로,  $1$ ,  $1.25$ ,  $1.36111$ ,  $1.42361$ ,
 $1.44361$  등이다. 부분 합들은 증가하지만, 증가량은 점점 작아진다. 따라서 무한
합이 어떤 유한 값을 가질 가능성이 있어 보인다.

\section{이 시스템의 특징}

Lambda를 이용하는 시스템이기 때문에 현대 한글을 전부 표현할 수 있다.
‘‘이나 ‘’과 같은 완성형 범위 밖의 문자가 가능하다. 특히 CP949
인코딩으로 입력할 수 있다는 점이 특징이다. 그밖에는 H\LaTeX/Lambda의
경우와 같다.

\begin{thebibliography}{0}
\bibitem{millen} 케이스 데블린. 전대호 옮김. 『수학의 밀레니엄 문제들』.
도서출판 까치. 2004.
\end{thebibliography}

\end{document}

```


무한 합의 계산

케이스 데블린

요약

이 문서는 “한글”을 표현하는 다양한 방법을 테스트하기 위한 것이다. 텍스트는 [1]에서 가져왔으며, hangul-k 스타일을 써서 CP949 인코딩의 문서를 작성하였다.

제 1 절 수학자들은

수학자들은 제타(ζ) 함수에 있는 것과 같은 무한 합을 어떻게 계산하는 것일까? 더 기초적으로 이렇게 물을 수도 있다: 무한한 덧셈이 어떻게 유한한 결과를 산출할 수 있을까?

무한 합(무한급수) 속에 있는 항들이 자연수의 제곱들인 경우를 생각해 보자.

$$1 + 4 + 9 + 16 + 25 + \dots$$

덧셈을 통해서 얻어지는 값들을 차례로 나열하면 다음과 같다. $1 + 4 = 5$, $5 + 9 = 14$, $14 + 16 = 30$, $30 + 25 = 55$ 등등. 이 부분 합들이 빠르게 커지므로, 전체 무한 합은 무한한 결과를 산출할 것이다.

이번에는 모든 자연수들의 제곱의 역수를 항으로 가진 무한 합을 생각해 보자.

$$\frac{1}{1} + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \dots$$

즉,

$$1 + 0.25 + 0.11111 + 0.0625 + 0.04 + \dots$$

이 무한 합의 부분 합들은 차례대로, 1, 1.25, 1.36111, 1.42361, 1.44361 등이다. 부분 합들은 증가하지만, 증가량은 점점 작아진다. 따라서 무한 합이 어떤 유한 값을 가질 가능성이 있어 보인다.

제 2 절 이 시스템의 특징

Lambda를 이용하는 시스템이기 때문에 현대 한글을 전부 표현할 수 있다. ‘떡’이나 ‘행’과 같은 완성형 범위 밖의 문자가 가능하다. 특히 CP949 인코딩으로 입력할 수 있다는 점이 특징이다. 그밖에는 \LaTeX /Lambda의 경우와 같다.

참고 문헌

- [1] 케이스 데블린, 전대호 옮김. 『수학의 밀레니엄 문제들』. 도서출판 까치. 2004.

6 UTF-8 Lambda 한글 패키지

6.1 H_ΛTeX/Lambda

H_ΛTeX은 L_ΛTeX 패키지인 동시에 Lambda 패키지이기도 하다. L_ΛTeX 대신 Lambda로 컴파일하려면 문서를 UTF-8로 작성하기만 하면 된다. 이 시스템을 편의상 H_ΛTeX/Lambda라고 부르기로 하자. 실제로 별도의 설치도 불필요하고 H_ΛTeX이 설치된 상태 그대로 이용하면 되는 것이다.

Preamble 문서의 작성 방법은 H_ΛTeX의 경우와 대동소이하다. 다만 저장할 때 인코딩을 UTF-8로 한다.

컴파일 latex이 아닌 lambda로 컴파일하면 된다.

```
#> lambda foo
```

특징

Lambda를 사용함으로써 얻는 이점은 다음과 같다.

- EUC-KR 문자 범위에 구애되지 않고 모든 현대 한글 음절문자를 다 표현할 수 있다.
- 번거로운 자동조사 관련 에러를 만나지 않을 수 있다.
- 한글 매크로의 정의와 사용이 용이하다.

코멘트

그러나, Lambda 자신의 한계와 H_ΛTeX/Lambda 설계상의 선택으로 인해 다음과 같은 불편한 점이 생긴다.

- 단일 폰트 접근을 사용하고 있기 때문에 영문자와 한글이 모두 같은 글꼴로 처리된다. 이것은 영문자 선택의 폭을 좁힐 뿐 아니라, 한글 폰트의 영문 글자가 과연 영문자 타이포그래피를 충족할 만큼 고품위인가에 의문이 있어 결과적으로 얻어진 판면이 만족스럽지 못한 경우가 있다.
- 기본 글꼴로 표현할 수 없는 한자나 외국어 문자를 식자하기가 상대적으로 어렵다.
- pdf 제작에 있어서 한글 책갈피가 구현되지 않는다.
- 이따금 Lambda 프로그램이 이해할 수 없는 동작을 하거나 예고없이 죽어버리는 경우가 있다. 즉 T_EX만큼 안정성이 높지 않다.
- Lambda와 호환되지 않는 L_ΛTeX 패키지들이 많다.
- 결정적으로, Omega/Lambda가 개발이 중단되었다.⁹⁾

예제 문서의 소스

9) T_EX과 같이 bug-free를 선언하고 더이상 개선할 것이 없어서 중단된 것이 아니다.

```

\documentclass{article}
\usepackage{hangul}

\title{무한 합의 계산}
\author{케이스 데블린}
\date{}

\begin{document}

\maketitle

\begin{abstract}
이 문서는 ‘‘한글’’을 표현하는 다양한 방법을 테스트하기 위한 것이다.
텍스트는 \cite{millen}에서 가져왔으며,
H\LaTeX\ 문서로서 서식 스타일 \texttt{[hangul]}만을
부르고 작성하였다.
\end{abstract}

\section{수학자들은}
수학자들은 제타( $\zeta$ ) 함수에 있는 것과 같은 무한 합을 어떻게
계산하는 것일까? 더 기초적으로 이렇게 물을 수도 있다: 무한한 덧셈이
어떻게 유한한 결과를 산출할 수 있을까?

무한 합(무한급수) 속에 있는 항들이 자연수의 제곱들인 경우를
생각해보자.
\[
1+4+9+16+25+\cdots
\]
덧셈을 통해서 얻어지는 값들을 차례로 나열하면 다음과 같다.
 $1+4=5$ ,  $5+9=14$ ,  $14+16=30$ ,  $30+25=55$  등등. 이 부분 합들이
빠르게 커지므로, 전체 무한 합은 무한한 결과를 산출할 것이다.

이번에는 모든 자연수들의 제곱의 역수를 항으로 가진 무한 합을 생각해 보자.
\[
\frac{1}{1}+\frac{1}{4}+\frac{1}{9}+\frac{1}{16}+\frac{1}{25}+\cdots
\]
즉,
\[
1+0.25+0.1111+0.0625+0.04+\cdots
\]
이 무한 합의 부분 합들은 차례대로,  $1$ ,  $1.25$ ,  $1.36111$ ,  $1.42361$ ,
 $1.44361$  등이다. 부분 합들은 증가하지만, 증가량은 점점 작아진다. 따라서 무한
합이 어떤 유한 값을 가질 가능성이 있어 보인다.

\section{이 시스템의 특징}

```

Lambda를 이용하는 시스템이기 때문에 현대 한글을 전부 표현할 수 있다.
,,이나 ,,과 같은 완성형 범위 밖의 문자가 가능하다. 특히 CP949
인코딩으로 입력할 수 있다는 점이 특징이다.

```
\begin{thebibliography}{0}  
\bibitem{millen} 케이스 데블린. 전대호 옮김. 『수학의 밀레니엄 문제들』.  
도서출판 까치. 2004.  
\end{thebibliography}  
  
\end{document}
```

무한 합 계산

케이스 데블린

요약

이 문서는 “한글”을 표현하는 다양한 방법을 테스트하기 위한 것이다. 텍스트는 [1]에서 가져왔으며, HEX 문서로서 서식 스타일 [hangu1]만을 부르고 작성하였다.

제 1 절 수학자들은

수학자들은 제타(ζ) 함수에 있는 것과 같은 무한 합을 어떻게 계산하는 것일까? 더 기초적으로 이렇게 물을 수도 있다: 무한한 덧셈이 어떻게 유한한 결과를 산출할 수 있을까?

무한 합(무한급수) 속에 있는 항들이 자연수의 제곱들이나 제곱들의 역수인 경우를 생각해 보자.

$$1 + 4 + 9 + 16 + 25 + \cdots$$

덧셈을 통해서 얻어지는 값들을 차례로 나열하면 다음과 같다. $1 + 4 = 5$, $5 + 9 = 14$, $14 + 16 = 30$, $30 + 25 = 55$ 등등. 이 부분 합들이 빠르게 커지므로, 전체 무한 합은 무한한 결과를 산출할 것이다.

이번에는 모든 자연수들의 제곱의 역수를 항으로 가진 무한 합을 생각해 보자.

$$\frac{1}{1} + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \cdots$$

즉,

$$1 + 0.25 + 0.11111 + 0.0625 + 0.04 + \cdots$$

이 무한 합의 부분 합들은 차례대로, 1, 1.25, 1.36111, 1.42361, 1.44361 등이다. 부분 합들은 증가하지만, 증가량은 점점 작아진다. 따라서 무한 합이 어떤 유한 값을 가질 가능성이 있어 보인다.

제 2 절 이 시스템의 특징

Lambda를 이용하는 시스템이기 때문에 현대 한글을 전부 표현할 수 있다. '떡'이나 '행'과 같은 완성형 범위 밖의 문자가 가능하다. 특히 CP949 인코딩으로 입력할 수 있다는 점이 특징이다.

참고 문헌

- [1] 케이스 데블린, 전대호 옮김. 『수학의 밀레니엄 문제들』. 도서출판 까치. 2004.

6.2 DHHangul

이 Lambda 패키지는 hangul-k와 동시에 개발된 것이다. hangul-k를 작성하면서 느끼게 된 8비트 문자처리의 한계 때문에 만약 Omega를 사용한다면 어떤 결과를 얻을 수 있겠는지 테스트해본 것이고, 그 결과가 상당히 좋아서 특히 중세 한글 문헌을 식자하기 위한 여러 가지 시도를 하게 되었다. 현재 상태로도 중세 한글 문헌이나 복잡한 한자가 쓰이는 고문헌의 식자에 유용하게 활용할 수 있다. 특히 중세 한글 문헌의 처리를 위하여 한양 PUA 영역 코드를 첫가끝으로 변환하는 유틸리티를 포함하고 있으며, 자소조합 또는 한양 글꼴의 완성형 중세 문자를 이용한 식자가 모두 가능한 유일한 Lambda 패키지이다. 이 패키지에서 발전한 기술의 일부는 Hangul-ucs에 계승되었다.¹⁰⁾ 사실상 Lambda에서 한글 사용의 완결편.

Preamble 중세 문자의 처리에 어떤 방식을 사용할 것인지 지정하는 작업이 필요하다. 또한 영역별로 다른 폰트를 지정할 수 있기 때문에 원한다면 각각의 폰트를 설정하는 작업도 해주어야 한다.

```
\documentclass{article}
\usepackage[ohanyang,engname]{dhucs}
\usepackage[UT1]{fontenc}
\usepackage{omega}
\Hserrifonts{unbt}{unbt}{obatang}{hybt} % 한글/한자/자모옛한글/pua옛한글
\Hsansfonts{ungt}{ungt}{obatang}{hygt}
\Hmonofonts{untz}{ungt}{obatang}{hygt}
```

컴파일 lambda로 컴파일한다. pdf 제작은 dvipdfmx.

```
#> lambda foo
```

특징

- UTF-8뿐 아니라 UTF-16, UHC, CP949 등 가능한 모든 한글 인코딩을 다 지원한다.
- 영문 글꼴과 한글 글꼴의 분리뿐 아니라 현대 한글 영역, 한자 영역, 옛한글 영역에 각각 다른 글꼴을 할당할 수 있어서 부족한 글꼴 문제를 여러 글꼴의 결합으로 해결할 수 있게 하였다.
- 첫가끝 옛한글, 한양 PUA 완성형 옛한글, GSUB 옛한글 등을 모두 식자에 활용할 수 있다. 폰트가 없을 뿐.
- 색인 작성을 위한 스크립트가 제공된다. 또한 한양 PUA 옛한글 코드를 첫가끝 한글 코드로 변환하는 스크립트도 제공된다.
- 한자에도 자동조사가 작동한다.
- 영문 글꼴을 자유롭게 선택하거나 설정할 수 있다.

¹⁰⁾ 예를 들면 한자 뒤에 자동조사를 붙이는 것 등.

코멘트

- Lambda 자체의 한계와 글꼴의 부족 때문에 그 성능을 충분히 실증할 기회를 갖지 못하였다.
- Aleph/Lamed와 충돌이 있다.
- 일부 TeX 시스템의 otp2ocp가 잘 작동하지 않아서 설치에 어려움을 겪는 경우가 있다.

예제 문서의 소스

```
\documentclass{article}
\usepackage[ohanyang]{dhhangul}
\usepackage{ohanyang}

\title{무한 합 의 계산}
\author{케이스 데블린}
\date{}

\begin{document}

\maketitle

\begin{abstract}
이 문서는 ‘‘한글’’을 표현하는 다양한 방법을 테스트하기 위한 것이다.
텍스트는 \cite{millen}에서 가져왔으며,
DHHangul에 서식 스타일 \texttt{ohanyang}을
부르고 작성하였다.
\end{abstract}

\section{수학자들은}
수학자들은 제타( $\zeta$ ) 함수에 있는 것과 같은 무한 합을 어떻게
계산하는 것일까? 더 기초적으로 이렇게 물을 수도 있다: 무한한 덧셈이
어떻게 유한한 결과를 산출할 수 있을까?

무한 합(무한급수) 속에 있는 항들이 자연수의 제곱들인 경우를
생각해보자.
\[
1+4+9+16+25+\cdots
\]
덧셈을 통해서 얻어지는 값들을 차례로 나열하면 다음과 같다.
 $1+4=5$ ,  $5+9=14$ ,  $14+16=30$ ,  $30+25=55$  등등. 이 부분 합들이
빠르게 커지므로, 전체 무한 합은 무한한 결과를 산출할 것이다.

이번에는 모든 자연수들의 제곱의 역수를 항으로 가진 무한 합을 생각해 보자.
\[
\frac{1}{1}+\frac{1}{4}+\frac{1}{9}+\frac{1}{16}+\frac{1}{25}+\cdots
\]
```

```

즉,
\[
1+0.25+0.1111+0.0625+0.04+\cdots
\]
이 무한 합을 부분 합들은 차례대로,  $1$ ,  $1.25$ ,  $1.36111$ ,  $1.42361$ ,
 $1.44361$  등이다. 부분 합들은 증가하지만, 증가량은 점점 작아진다. 따라서 무한
합이 어떤 유한 값을 가질 가능성이 있어 보인다.

\section{옛 한글의 식자}
\parindent=0pt
<여기 옛 한글 텍스트가 옴>


\begin{thebibliography}{0}
\bibitem{millen} 케이스 데블린. 전대호 옮김. 『수학의 밀레니엄 문제들』.
도서출판 까치. 2004.
\end{thebibliography}

\end{document}

```

무한 합 계산

케이스 테이블

요약

이 문서는 “한글”을 표현하는 다양한 방법을 테스트하기 위한 것이다.
텍스트는 [1]에서 가져왔으며, DHHangul에 서식 스타일 `ohanyang`을 부르고 작성하였다.

1 수학자들은

수학자들은 제타(ζ) 함수에 있는 것과 같은 무한 합을 어떻게 계산하는 것일까?
더 기초적으로 이렇게 물을 수도 있다: 무한한 덧셈이 어떻게 유한한 결과를 산출할 수 있을까?

무한 합(무한급수) 속에 있는 항들이 자연수의 제곱들인 경우를 생각해보자.

$$1 + 4 + 9 + 16 + 25 + \dots$$

덧셈을 통해서 얻어지는 값들을 차례로 나열하면 다음과 같다. $1 + 4 = 5$, $5 + 9 = 14$, $14 + 16 = 30$, $30 + 25 = 55$ 등등. 이 부분 합들이 빠르게 커지므로, 전체 무한 합은 무한한 결과를 산출할 것이다.

이번에는 모든 자연수들의 제곱의 역수를 항으로 가진 무한 합을 생각해 보자.

$$\frac{1}{1} + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \dots$$

즉,

$$1 + 0.25 + 0.11111 + 0.0625 + 0.04 + \dots$$

이 무한 합의 부분 합들은 차례대로, 1, 1.25, 1.36111, 1.42361, 1.44361 등이다.
부분 합들은 증가하지만, 증가량은 점점 작아진다. 따라서 무한 합이 어떤 유한 값을 가질 가능성이 있어 보인다.

2 옛한글의 식자

國國之징語:어音뎡이

國은 나라히라 之는 입겨지라 語는 말쌔미라

나랏말쌔미

異英乎龔中등國國히야

異는 다를 씨라 乎는 아모그에 訢는 겨체 쓰는 字ㅣ라 中國은 皇帝 겨신 나라

히니 우리나라 常談애 江南이라 訢는 나라

中등國國에 달아

與:영文문字종로 不龔相상流류通通홀씨

與는 이와 더와 訢는 겨체 쓰는 字ㅣ라 文은 글와리라 不은 아니 訢는 쁘디라

相은 서르 訢는 쁘디라 流通은 흘러 스믈출 씨라

文문字종와로 서르 스믈디 아니홀씨

참고문헌

- [1] 케이스 데블린, 전대호 옮김, 『수학의 밀레니엄 문제들』, 도서출판 까치, 2004.

7 기타

7.1 frktex

토쿄대 후쿠이 레이 교수가 작성한 $\text{k}\text{T}\text{E}\text{X}$ 은 일본어 TEX 시스템인 $\text{p}\text{T}\text{E}\text{X}$ 에서 한글을 쓸 수 있도록 한 작은 패키지이다. 이것을 $\text{p}\text{T}\text{E}\text{X}$ 이 아닌 상황, 즉 tEX 과 같은 곳에서도 사용할 수 있도록 조금 수정한 판을 $\text{frk}\text{T}\text{E}\text{X}$ 이라 부른다.

Preamble, 입력 이 패키지는 한글 입력 체계를 사용하지 않으며, 한국어의 로마자 대응을 입력 방식으로 하고 있다. 이 로마자 대응 체계는 우리나라의 로마자 표기법과는 상당히 다른 독특한 체계인데, 입력의 편의를 철저하게 고수해서 자모 하나에 하나의 문자를 대응시키고 있다. 일부는 예일 방식에서 왔고 일부는 편의를 위하여 특정 키를 강제 할당하고 있다.

```
\documentclass{article}
\usepackage{frktex}

\begin{document}
\hg{'annieqhasei'io}
\end{document}
```

컴파일 일반적인 $\text{L}\text{A}\text{T}\text{E}\text{X}$ 문서와 같이 컴파일한다. $\text{pdf}\text{L}\text{A}\text{T}\text{E}\text{X}$ 에서도 문제없다.¹¹⁾

```
#> latex foo
```

특징

- 두 개의 글꼴이 제공되는데 각각 명조와 고딕에 해당한다. 고딕 글꼴은 $\backslash\text{bf}$ 에서 불린다. 이 글꼴을 Computer Hangul 글꼴이라 한다.
- 글꼴은 매우 심플하고 자소 조합형식으로 이루어져 있다. 즉, 종성이 있는 글자와 없는 글자에 약간 다른 모음을 배치하고 다른 것은 자소를 결합하도록 하는 방식으로 한글을 구현한다.
- 한글의 음절 경계를 자동으로 판별하며, 현대 한글 음절문자는 물론이고 일부 옛한글도 식자할 수 있다.
- 한글 시스템을 설치하는 것이 너무나 번거로운 경우 정말 심플하게 한두 글자를 사용하는 데는 딱이다.
- 한글 문서를 인쇄나 출판할 정도의 품위는 아니다.

예제 문서의 소스

11) 필자가 $\text{pdf}\text{L}\text{A}\text{T}\text{E}\text{X}$ 에서 사용하기 위해서 Computer Hangul 글꼴을 type 1으로 변환한 바가 있다.

```

\documentclass{article}
\usepackage{frktex}

%무한 합 계산
\title{\hg{muhan hab'yi gieisan}}
%케이스 데블린
\author{\hg{kei'isy deibyrin}}
\date{}

\begin{document}

\maketitle

\begin{abstract}
%이 문서는 한글을 표현하는 다양한 방법을 테스트하기
\hg{'i munsenyn hangyr'yr piohienhanyn da'iaqhan baqbeb'yr teisytyhagi
% 위한 것이다.
'uihan ges'ida.}
% frktex을 이용하여 작성하였다.
\texttt{frktex}\hg{'yr 'i'ioqha'ie jagseqha'ieSda.}
\end{abstract}

\section{\hg{suhagjadyr'yn}}
%수학자들은 제타( $\zeta$ ) 함수에 있는 것과 같은 무한 합을 어떻게
%계산하는 것일까? 더 기초적으로 이렇게 물을 수도 있다: 무한한 덧셈이
%어떻게 유한한 결과를 산출할 수 있을까?
\hg{suhagjadyr'yn jeita}( $\zeta$ ) \hg{hamsu'ei 'iSnyn gesgoa gat'yn muhan
hab'yr 'eDehgei gieisanhanyn ges'irGa?}
\hg{de gicojeg'yro 'irehgei mur'yr sudo 'iSda:}
\hg{muhanhan desseim'i 'eDehgei 'iuhanhan giergoaryr sancurhar su
'iS'yrGa?}

%무한 합(무한급수) 속에 있는 항들이 자연수의 제곱들인 경우를
%생각해보자.
\hg{muhan hab(muhangybsu) sog'ei 'iSnyn haqdyr'i ja'iensu'yi
jeigobdyr'in gieq'uryr saiqgaghaiboja.}
\[
1+4+9+16+25+\cdots
\]
%덧셈을 통해서 얻어지는 값들을 차례로 나열하면 다음과 같다.
\hg{desseim'yr toqhaise 'ed'ejinyn gabsdyr'yr carieiro na'ierhamien
da'ymgoa gatda.}
 $1+4=5$ ,  $5+9=14$ ,  $14+16=30$ ,  $30+25=55$ 
%등등. 이 부분 합들이
%빠르게 커지므로, 전체 무한 합은 무한한 결과를 산출할 것이다.

```

```

\hg{dyqdyq. 'i bubun habdyr'i Barygei kejimyro, jencei muhan hab'yn
muhanhan giergoaryr sancurhar ges'ida.}

%이번에는 모든 자연수들의 제곱의 역수를 항으로 가진 무한 합을 생각해 보자.
\hg{'iben'einyr modyn ja'iensudyr'yi jeigob'yi 'iegsuryr haq'yro gajin
muhan hab'yr saiqgaghaiboja.}
\[
\frac{1}{1}+\frac{1}{4}+\frac{1}{9}+\frac{1}{16}+\frac{1}{25}+\cdots
\]
%즉,
\hg{jygyr,}
\[
1+0.25+0.1111+0.0625+0.04+\cdots
\]
%이 무한 합은 부분 합들은 차례대로, $1$, $1.25$, $1.36111$, $1.42361$,
%$1.44361$ 등이다. 부분 합들은 증가하지만, 증가량은 점점 작아진다. 따라서 무한
%합이 어떤 유한 값을 가질 가능성이 있어 보인다.
\hg{'i muhan hab'yi bubun habdyr'yn carieidairo,}
$1$, $1.25$, $1.36111$, $1.42361$, $1.44361$
\hg{dyq'ida. bubun habdyr'yn jyqgahajiman, jyqgariaq'yn jemjem jag'ajinda.
Darase muhan hab'i 'eDen 'iuhan gabs'yr gajir ganyqseq'i 'iS'e bo'inda.}

\begin{thebibliography}{0}
\bibitem{millen}
%케이스 데블린. 전대호 옮김. 『수학의 밀레니엄 문제들』.
%도서출판 까치. 2004.
\hg{kei'isy deibyrrin. jendaiho 'ormgim.} \hg{{\bf suhag'yi mirreini'em
munjeidyr}}. \hg{dosecurpan Gaci.} 2004.
\end{thebibliography}

\end{document}

```

무한 합 계산

케이스 데블린

Abstract

이 문서는 한글을 표현하는 다양한 방법을 테스트하기 위한 것이다.
`frktex`을 이용하여 작성하였다.

1 수학자들은

수학자들은 제타(ζ) 함수에 있는 것과 같은 무한 합을 어떻게 계산하는 것일까? 더 기초적으로 이렇게 물을 수도 있다: 무한한 덧셈이 어떻게 유한한 결과를 산출할 수 있을까?

무한 합(무한급수) 속에 있는 항들이 자연수의 제곱들인 경우를 생각해보자.

$$1 + 4 + 9 + 16 + 25 + \dots$$

덧셈을 통해서 얻어지는 값들을 차례로 나열하면 다음과 같다. $1 + 4 = 5$, $5 + 9 = 14$, $14 + 16 = 30$, $30 + 25 = 55$ 등등. 이 부분 합들이 빠르게 커지므로, 전체 무한 합은 무한한 결과를 산출할 것이다.

이번에는 모든 자연수들의 제곱의 역수를 항으로 가진 무한 합을 생각해보자.

$$\frac{1}{1} + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \dots$$

즉,

$$1 + 0.25 + 0.11111 + 0.0625 + 0.04 + \dots$$

이 무한 합의 부분 합들은 차례대로, 1, 1.25, 1.36111, 1.42361, 1.44361 등이다. 부분 합들은 증가하지만, 증가량은 점점 작아진다. 따라서 무한 합이 어떤 유한 값을 가질 가능성이 있어 보인다.

References

[1] 케이스 데블린. 전대호 옮김. 수학의 밀레니엄 문제들. 도서출판 까치. 2004.

8 결어

이상에서 중요한 몇 가지 한글 \LaTeX 시스템들을 일별하여 보았다. 이것을 간략히 정리하면 표 2와 같다.

표 2: 주요 한글 \LaTeX -Lambda 패키지들

패키지	인코딩	처리	(기본) 글꼴	글꼴 확장	pdf 제작	pdf 북마크	서식
$\text{H}\text{\LaTeX}$	EUC-KR	latex	UHC	가능	가능	불가능	있음
$\text{h}\text{\LaTeX}\text{p}$	EUC-KR	latex	#신명조 등	불가능	불가능	불가능	있음
hangul-k	EUC-KR	latex	UHC	은글꼴	가능	가능	있음
CJK- \LaTeX	EUC-KR/UTF-8	latex	?	가능	가능	불가능	없음
ttfucs	UTF-8	pdflatex	?	가능	가능	가능	없음
$\text{H}\text{\LaTeX}$ /Lambda	UTF-8	lambda	UHC	가능	가능	불가능	있음
DHHangul	UTF8/CP949	lambda	은글꼴	가능	가능	불가능	있음
Hangul-ucs	UTF-8	latex	은글꼴	가능	가능	가능	있음

생각건대, 이 많은 한글 \LaTeX 패키지들의 족적은 가히 \LaTeX 에서 한글을 구현하는 데 얼마나 많은 사람들이 노력해왔는가를 보여주는 한 예라고 할 것이다. 그리하여 이제는 한글 문서를 \LaTeX 으로 식자하고 출판하는 데 거의 아무런 어려움을 겪지 않을 정도가 되었다고 생각한다.

한글 \LaTeX 패키지를 요약하면서 느낀 점은, 대략 다음과 같다.

- (1) 한글 코드 문제의 혼선은 아직도 심각하다. 애시당초 조합형과 완성형의 대립에서부터 시작된 코드 스캔들을 이렇게 복잡하게 만든 데는 적어도 두 가지 계기가 있는 것 같다. 하나는 유니코드에 자모영역, 음절영역 해가면서 통일되지 못한 코드 구현 체계를 모두 집어넣은 것, 다른 하나는 가장 영향력이 큰 운영체제인 MS 윈도가 유니코드 표준을 따르지 않은 점. 대체로 Linux는 음절영역 유니코드를 중심으로 한 한글 환경에 익숙하고 Mac OS X는 첫가끝 코드를 시스템이 사용하고 있지만 윈도만이 CP949라는 ‘이상한’ 코드를 채택함으로써 \LaTeX 사용자에게 고통을 가중시킨 점이 있다고 생각된다.
- (2) 자유 글꼴의 문제. 대부분의 한글 글꼴은 상업용으로 제작되어 판매되고 있다. \LaTeX 의 세계에서 요구하는 것은 그런 장식용 글꼴들이 아니라 단정하고 아름다운 본문글꼴인데, 이 분야에서 이루어지는 일들도 마음놓이지 않는 점이 너무 많다. 예컨대 국가가 제작해서 배포한 문화부 글꼴의 망신이나, 가장 보편적으로 사용되는 윈도 바탕체가 자유 글꼴이 아닌 점 등을 생각하면 글꼴 문제야말로 진정한 한글화를 위해서 반드시 넘어야 할 산이라고 생각된다. 다행히 은글꼴과 같은 좋은 글꼴이 한 벌 있기는 하나, 다양성이라는 측면에서, 그리고 한글의 자유로운 표현이라는 점에서 2% 부족한 것도 사실이다.
- (3) 이래저래 우리에게 좋은 에디터가 필요하다. 여전히 윈도 사용자를 괴롭히고 있는 CP949의 문제는 WinEdt을 대체할 만한 좋은 에디터를 보급하거나 WinEdt 자신이 UTF-8을 지원하도록 개선되어야 하지 않을까 하는 생각이 있다. WinEdt이 모든 것을 해결해준다면 얼마나 좋으랴. 다만 최근 EmEditor가 그런대로 쓸 만하다고 하니 윈도 사용자의 갈증이 얼마간 해결이 된 것인지.
- (4) 한글 \LaTeX 에 관한 한, KTUG의 기여가 상당하다는 것이다. 은광희 선생께서 KTUG에 project 페이지를 열면서, “ $\text{H}\text{\LaTeX}$ 이 있어야 할 곳이 KTUG이라고 생각한다”고 말씀하신 바 있듯이,

KTUG의 몇 년은 그동안 갈피를 잡기 어려웠던, 또는 정체되어 있던 한글 \LaTeX 발전에 상당한 영향을 미치고 논의의 가닥을 잡아, 마침내 생산적인 결과를 얻은 대표적 사례로 꼽힐 만하다고 본다.

아직도 해결해야 될 문제는 많이 남아 있는 것 같다. 그러나 이런 문제는 어떻게든 극복할 방법이 있는 지엽적인 것으로서, 한글을 “표현한다”는 애초의 목적은 충분히 달성된 것으로 보아도 좋을 것이다.

도구가 아무리 좋아도 내용이 빈약하면 무슨 소용이겠는가? 도구가 갖추어졌으면, 더 아름답고 쓸모있는 문서를 많이 생산해내어야 하지 않겠는가.