

언어학자를 위한
T_EX · L_AT_EX · 한글L_AT_EX

류병래

Ver 0.91 [2000년 11월 18일]

to appear

in

CNU Occasional Papers and Utilities in Linguistic Analysis
(COPULA)

충남대학교 언어학과

한국언어정보학회

차례

제 1 장	TEX, L^AT_EX 그리고 한글L^AT_EX	11
제 1 절	TEX란 무엇인가	12
제 2 절	L ^A T _E X란 무엇인가?	13
제 3 절	한글L ^A T _E X가 나오기 까지	13
제 4 절	L ^A T _E X가 언어학자에게 좋은 22가지 이유	15
제 2 장	“언어학자를 위한 L^AT_EX” CD에서 설치하기	19
제 1 절	Setup 실행하기	19
제 2 절	MikTEX 설치하기	20
제 3 절	Ghostscript 설치하기	21
제 4 절	GSView 설치하기	22
제 5 절	WinEdt 설치하기	22
제 6 절	Acrobat Reader 설치하기	25
제 7 절	H ^A L ^A T _E X 설치하기	25
제 8 절	설정 변경하기	28
제 3 장	한글L^AT_EX로 만드는 문서의 구조	33
제 1 절	L ^A T _E X는 문서를 어떻게 조판하는가	33
제 2 절	논리적인 문서 디자인과 시각적인 문서 디자인	35
제 3 절	TEX의 기초	36
3.1	기본 명령어	36
3.2	문서형식	38
제 4 장	한글L^AT_EX로 예문과 예문주석 달기	43
제 1 절	꾸러미 소개	44

제 2 절	꾸러미 설치하기: <code>gb4e.sty</code> & <code>covington.sty</code> . . .	45
제 3 절	꾸러미 사용하기 (1): <code>gb4e.sty</code>	46
3.1	초보적인 사용법에서 전문가적 사용법까지 . .	46
3.2	예문 주석과 번역문의 정렬	50
3.3	예문에 문법성 표시하기	56
3.4	Varia: bar 기입하기, labelled bracketing 등 . .	56
3.5	예문에 이동한 선을 그리기 및 기타 사항 . . .	57
제 4 절	꾸러미 사용하기 (2): <code>covington.sty</code>	58
4.1	기본적인 예문 번호 매기기	58
4.2	주석문 정렬하기	60
4.3	Phrase structure rules 및 feature structures 그 리기	61
4.4	Accents	62
제 5 장	Attribute-Value-Matrices (AVM) 그리기	65
제 1 절	꾸러미 소개: <code>avm.sty</code>	65
제 2 절	꾸러미 설치하기: <code>avm.sty</code>	66
제 3 절	꾸러미 이용하기	66
3.1	꾸러미 환경설정 및 기초 사항	66
3.2	Sag & Wasow (1999)식으로 유형이 처리된 AVM 만들기	68
3.3	Pollard & Sag (1994)식으로 유형이 처리된 AVM 만들기	70
3.4	옵션을 이용해서 AVM을 더 간단하게 만들기 .	73
3.5	LFG에서 사용하는 기능등식과 기능기술 AVM 만들기	74
제 6 장	L^AT_EX로 수형도 그리기	77
제 1 절	수형도를 그리기 위한 꾸러미	77
제 2 절	<code>parsetree.sty</code> 를 이용한 수형도 그리기	79
2.1	기본적인 사용법	79
2.2	교점이 복잡한 수형도 그리기	82
2.3	AVM이 교점인 수형도 그리기	88
2.4	수형도에서 이동선을 그리기	92

제 3 절	<code>rtrees.sty</code> 를 이용한 수형도 그리기	94
3.1	기본적인 명령어	95
3.2	Type Hierarchy 그리기	95
3.3	일반적인 수형도 그리기	96
3.4	<code>rtrees.sty</code> 의 기타 기능	99
제 4 절	<code>tree-dvips.sty</code> 를 이용한 수형도 그리기	105
제 7 장	의미론 학자를 위한 한글\LaTeX의 기호와 수식	107
제 1 절	수학 기호	107
제 2 절	그리스 문자	109
제 3 절	특수 기호	110
제 4 절	각국 언어 특수 문자	112
제 5 절	다양한 괄호 등 기호 문자	114
제 6 절	Theorem과 Proof	117
제 8 장	한글\LaTeX로 담화표상구조 (DRS) 만들기	121
제 1 절	<code>covington.sty</code> 를 이용한 담화표상구조 (DRS) 그리기	121
제 2 절	Kamp and Reyle (1993)의 담화표상구조: <code>pstricks.sty</code>	126
제 9 장	한글\LaTeX로 국제음성문자 쓰기	129
제 1 절	꾸러미 소개: <code>ipa.sty</code>	129
제 2 절	WSUIPA 국제음성문자의 배열	130
제 3 절	음성문자 입력 예	136
제 10 장	한글\LaTeX로 그림 삽입하기	137
제 1 절	꾸러미 소개: <code>graphics.sty</code> 와 <code>graphicx.sty</code>	137
제 2 절	그림 삽입하기: 기본 명령	138
제 3 절	그림 삽입하기: 여러 가지 가능성	139
제 11 장	한글\LaTeX로 참고문헌을 자동으로 만들기	143
제 1 절	개념도	143
제 2 절	참고문헌을 자동생성을 위한 설정	145
제 3 절	<code>natbib.sty</code> 를 사용해서 본문에 인용구 달기	147
제 4 절	참고문헌 서지정보가 담긴 데이터베이스 만들기	149
제 5 절	언어와 정보 참고문헌 포맷: <code>ksli-natbib.bst</code>	152

제 6 절	BibTeX를 사용하지 않고 수동으로 참고문헌 목록 만들기	172
제 12 장	한글\LaTeX로 자동으로 색인 만들기	175
제 1 절	개념도	175
제 2 절	색인을 생성하기 위한 문서의 구조	177
제 3 절	본문에 색인 표제어 표시하기	180
제 4 절	색인 위치 지정	181
제 13 장	한글\LaTeX로 유인물 만들기	183
제 1 절	특수문자 꾸러미: pifont.sty	183
제 2 절	구성 코드	184
제 1 절	Introduction	187
제 2 절	2쪽을 1쪽으로 축소하기: pstops	188
부록 A	한글\LaTeX시스템 및 관련 지원 프로그램 설치하기	189
제 1 절	MikTeX 설치하기	189
1.1	MikTeX란 무엇인가	189
1.2	MikTeX 설치 방법	190
제 2 절	Ghostscript 설치하기	194
2.1	Ghostscript란 무엇인가	194
2.2	Ghostscript 설치방법	195
제 3 절	GSView 설치하기	197
3.1	GSView란 무엇인가	197
3.2	GSView 설치 방법	199
제 4 절	Winedt 설치하기	201
4.1	Winedt란 무엇인가	201
4.2	Winedt 설치 방법	202
제 5 절	Acrobat Reader 설치하기	208
5.1	Acrobat Reader란 무엇인가	208
5.2	Acrobat Reader 설치 방법	208
제 6 절	Graphic Utilities 설치하기	208
6.1	\LaTeX 와 Graphic Applications	208
6.2	Graphic Utilities 설치 방법	209

부록 B	L^AT_EX에 한글을 설치하기 - H^IA^TE_X	211
제 1 절	L ^A T _E X에서 한글 사용	211
1.1	H ^I A ^T E _X -0.991	212
1.2	hL ^A TeXp	212
1.3	CJK 패키지	213
제 2 절	H ^I A ^T E _X 설치하기	213
2.1	글꼴 설치	213
2.2	H ^I A ^T E _X -0.991을 위한 MiK _T E _X 1.20e 세팅 변경	216

변경/추가된 항목에 대한 정보

2000년 10월 20일 배포한 책자와 CD는 편의상 버전을 0.90이라고 불렀고, 이번 2000년 11월 18일 배포한 책자와 CD에는 버전을 0.91이라고 붙여 구분하였습니다. 0.91으로 바뀌면서 변경되거나 추가된 사항에 관한 정보를 여기에 간략하게 적습니다.

책: Ver 0.90에서 Ver 0.91로

- 각 개별 프로그램을 설치하는 방법을 부록 (부록 A) 으로 옮겨, “언어학자를 위한 L^AT_EX” CD에서 `setup.exe`를 이용해 설치하는 법과 분리해서 배열함.¹
- 한글L^AT_EX설치하는 방법을 부록 (부록 B) 으로 옮겨, “언어학자를 위한 L^AT_EX” CD에서 `setup.exe`를 이용해 설치하는 법과 분리해서 배열함.
- 수형도 그리는 방법에 관한 장을 보완
- 의미론 학자를 위한 기호 및 수식 넣는 방법을 담화 표상구조 (DRS) 그리는 방법과 장을 분리해서 보완 기술
- 담화 표상구조 (DRS) 그리는 방법을 의미론 학자를 위한 기호 및 수식 넣는 방법과 장을 분리해서 보완 기술

¹이런 충고를 해준 김종복 선생님께 감사드립니다.

- 참고문헌 을 작성하는 방법 (BibTeX) 에 관한 장을 추가
- 색인 만드는 방법 (MakeIndex) 에 관한 장을 추가
- 발표용 유인물 (Handout) 만드는 방법에 관한 장을 추가
- 국제음성문자 (IPA) 설치 및 사용법에 관한 장을 추가
- H_AT_EX와 관련된 꾸러미들을 `c:\texmf` 아래가 아닌 `c:\localtexmf` 아래에 두도록 함으로써, T_EX 자체의 업데이트와는 상관이 없도록 함. 이런 원칙은 다른 꾸러미의 설치에도 마찬가지로 적용함.
- H_AT_EX를 설치할 때, H_AT_EX를 위한 설정부분은 `c:\localtexmf`로 가도록 하고, H_AT_EX를 위해 MikT_EX의 설정을 변경하는 경우는 `c:\texmf` 아래의 해당 파일을 바꾸도록 함.
- `ftp://free.kasit.ac.kr/ChoF/` 사이트의 관리자를 조진오님께서 조진환 님으로 정정
- To do list를 제거함.
- 출처에 대한 정보를 추가 [계속].

CD: Ver 0.90에서 Ver 0.91로

- 기본적으로 설정을 완료한 상태로 실행 파일로 넣어 둔 `hlatex-0991-4ling.exe`를 `c:\texmf`가 아닌 상위 폴더인 `c:`에 풀도록 함.
이렇게 함으로써 H_AT_EX와 관련된 꾸러미들은 자동적으로 `c:\localtexmf\tex\latex` 아래로 가도록 하고, T_EX 전체와 관련된 설정은 `c:\texmf\tex\latex` 아래로 가도록 함.
- GSView 3.4버전 (2000년 7월 29일)이 가장 최신의 버전이고 Aladdin Free Software License에 의해 상업용이 아닌 한 여전히 자유롭게 사용할 수 있으나, Registration Reminder가 자꾸 떠 성가신 면이 있어서 이전 버전인 GSView 2.9로 교체함.

제 1 장

TEX, LATEX 그리고 한글LATEX

TEX는 매우 강력하고 안정적인 고품질의 “조판 시스템” (typesetting system)입니다. 그러나 TEX시스템은 초보자가 직접 쓰기에는 복잡하므로, 보통 LATEX라는 패키지를 통해서 씁니다. 거기에 한글을 쓰려면 다시 그 위에 “한글 LATEX” 를 설치해야 합니다. 이 장은 이들의 관계를 간략하게 설명하고, 문서가 조판되는 과정을 알기 쉽게 때로는 의도적으로 단순화시켜 기술한 것입니다.¹

¹이 장의 내용은 한국과 외국에 있는 TEX 혹은 LATEX에 관한 내용을 실고 있는 다양한 인터넷 사이트의 내용을 참고해 작성한 것이다. 한국에 있는 대표적인 사이트를 몇 개만 들면 다음과 같다.

- (1) 조진환 박사의 ChoF's TEXArchive (<http://free.kaist.ac.kr/ChoF/>);
- (2) 사회과학도를 위한 (La)TEX/PDF (<http://www.doeun.pe.kr/uselatex/index.html>);
- (3) 초보자를 위한 TeX 가이드 (http://heartkorea.com/tex/tex_main.htm);
- (4) 한글 TEX홈페이지 (<http://knot.kaist.ac.kr/htex/>);
- (5) 한글 LATEX홈페이지 (<ftp://ftp.linguistik.uni-erlangen.de/pub/HLaTeX/>);
- (6) TEX 배움마당 (<http://math.kangwon.ac.kr/~yhpark/tex/texframe.htm>);
- (7) 장대훈의 TEX World (<http://user.chollian.net/~patient/>).

제 1 절 $\text{T}_\text{E}\text{X}$ 란 무엇인가

$\text{T}_\text{E}\text{X}$ 는 1980년경 미국의 Stanford 대학교의 수학자이자 전산학자인 Donald E. Knuth 교수 (高德納, Professor Emeritus of The Art of Computer Programming)가 만든 조판 시스템입니다 [Knuth (1986d), Knuth (1986c), Knuth (1986b), Knuth (1986a)]. ‘알고리즘의 신’이라고 불리는 Donald E. Knuth 교수는 매우 복잡한 수식이 있는 어려운 책을 많이 낸 바 있는데, 자신의 책이 출판된 후, 글자와 문장의 조판, 특히 수학식의 조판이 엉망인 것을 보고 수작업으로 조판하는 것의 한계를 개탄하면서 이를 컴퓨터로 해결하고자 $\text{T}_\text{E}\text{X}$ 를 직접 만들 결심을 하게 됩니다. Donald E. Knuth 교수는 1970년대 후반 이 프로젝트에 6개월 정도를 투자할 생각이었으나, 실제로 $\text{T}_\text{E}\text{X}$ 가 완성된 모습을 갖추기까지는 10년이 걸리고 말았습니다. Donald E. Knuth 교수는 현상금을 걸고 버그를 잡았는데, 이런 그의 노력으로 $\text{T}_\text{E}\text{X}$ 는 버그가 없는 프로그램으로 유명합니다.

$\text{T}_\text{E}\text{X}$ 는 공식적으로는 가운데 철자인 E 가 내려가 있어서 $\text{T}_\text{E}\text{X}$ 라고 표기합니다. $\text{T}_\text{E}\text{X}$ 는 영어에서는 ‘Technology’를 의미하는 약어로 쓰이지만, 그 어원은 그리스어인 $\tau\epsilon\chi$ 에서 찾을 수 있습니다. 그리스어로 $\tau\epsilon\chi$ 는 기술 (technology) 과 예술 (art) 을 동시에 의미합니다. 그리스어의 단어 끝의 χ 발음은 ‘ch’ 이므로 $\text{T}_\text{E}\text{X}$ ’는 엄격하게 얘기해서 ‘텍’이라고 발음해서도 안되고, ‘테크’라고 발음해서도 안되며, [te:x] (‘테흐’에 가깝게 발음)라고 발음해야 합니다.

$\text{T}_\text{E}\text{X}$ 는 $\text{T}_\text{E}\text{X}$ 를 위해서 새로 설계된 폰트 기술언어인 METAFONT와 함께 개발되었습니다. 표준 $\text{T}_\text{E}\text{X}$ 에서는 METAFONT로부터 pk파일이라는 폰트파일을 만들어서 디스플레이와 프린팅에 이용합니다. 지금도 이 방법은 쓰이고 있지만 최근 와서 폰트 문제는 Type 1이라 불리는 Adobe사의 Postscript 폰트를 이용할 수 있게 되어서 좀 복잡해지고 그만큼 기능이 향상되었습니다.

$\text{T}_\text{E}\text{X}$ 는 소스까지 포함하여 모두 공개된 공개 소프트웨어이며 Unix 외에 Linux, DOS, Windows, Macintosh 등 거의 모든 컴퓨터 운용체제(Operating System)에서 동작합니다.

제 2 절 \LaTeX 란 무엇인가?

\TeX 가 강력하고 편리한 조판도구라는 점이 입증된 데다가 버그가 거의 없는 프로그램이라는 것이 알려지면서 이 도구를 자신의 작업에 이용하고자 하는 일반 사용자들도 늘어났습니다. 그러나 안타깝게도 \TeX 은 그 자체가 일종의 컴퓨터 프로그램 언어였기 때문에, 그것을 배워서 사용하려면 상당한 시간과 노력을 기울여야 했기에 일반인들이 쓰기에는 어려운 점이 많았습니다. 이런 어려움을 극복하기 위해 1980년대 초 Leslie Lamport는 \TeX 위에서 동작하는 좀 더 간편한 시스템을 만들게 되었으니, 이 시스템이 바로 \LaTeX 입니다 [Lamport (1987)].

Leslie Lamport가 만든 \LaTeX 는 사용자가 문서의 논리적 구조만 입력하면 그 외의 것은 프로그램이 내부적으로 알아서 처리하여 주는 매크로 모음입니다. \LaTeX 는 \TeX 를 기본 조판엔진으로 사용하기 때문에 \TeX 의 강력함은 그대로 갖고 있고, 여기에 문서 구조화의 방법과 스타일화된 레이아웃으로 누구라도 쉽게 \TeX 문서를 작성할 수 있도록 한 것입니다. 1991년에는 \LaTeX 의 글꼴단점을 보완하여 $\text{\LaTeX}2\epsilon$ 가 개발되었고 AMSTeX 과 \LaTeX 의 장점들을 통합시킨 AMSLaTeX 가 개발되어 오늘에 이르렀다.

현재 \TeX 의 버전은 윈도우를 따라서 3.14159이고, \LaTeX 는 \LaTeX -2.09를 거쳐서 현재는 $\text{\LaTeX}2\epsilon$ 가 쓰이고 있는데, 다른 한편 \LaTeX -3이 국제 전문가 그룹에 의해서 개발되고 있는 중입니다.

제 3 절 한글 \LaTeX 가 나오기 까지

\TeX 나 \LaTeX 에서 처음부터 바로 한글을 쓸 수 있었던 것은 아니고, 그중안 여러 사람들의 끊임없는 노력과 시간의 투자로 이제는 한글을 쓸 수 있게 되었습니다 [Cha (1996), Eun (2000)].

역사적으로 좀 더 거슬러 올라 가면, 1990년 여름부터 현재 KAIST의 전신인 KIST 수학과 고기형 교수와 진교택 교수 팀에 의해 한글을 \TeX 에서 사용하기 위한 작업이 진행되었는데, 여기서는 Macintosh 글꼴에 기반한 한글처리용 \TeX 의 개발이 주된 작업이었습니다. 이 프로그램에서는 \LaTeX 으로 한글을 쓸 수 있도록 모든 명령을 만들고 한글 글자체를 준비하여 hlatex을 보급하였습니다 (최우형, 백운

주) . 여기에는 기존의 $L_{A}T_{E}X$ 에 한글 글자체를 추가하는 `hfont.tex`, 한글 문서 작성 환경을 정의하는 `harticle.sty`과 `hreport.sty`, `hbook.sty` 그리고 한글 KS 완성형 부호 체계를 $T_{E}X$ 이 이해를 할 수 있는 $T_{E}X$ 모듬 명령으로 변환해 주는 전처리기(preprocessor) `htex`로 구성되어 있습니다. KAIST에서는 1993년에 이르러 전처리기를 사용하지 않는 한글 $T_{E}X$ 가 개발되었고, 이 한글 $T_{E}X$ 는 1994년 봄부터 대한수학회를 통해 회원에게 배포되기 시작했습니다.

최우형/백운주의 한글 패키지를 기반으로 은광희님이 한자의 사용을 추가한 `jhtex`을 만들었습니다. 여기에서는 일본의 Kanji 글자체를 `hlatex`에 덧붙쳐 사용할 수 있도록 하였고 문서 모양새도 좀 더 한글화 될 수 있도록 하였으며 글자체가 늘어남으로써 불가피해진 NFSS2의 도입이 이루어졌습니다. 이 두 프로그램은 좀 더 낫은, 하나의 프로그램으로 일반에게 보급될 수 있고자 $H_{L}A_{T}_{E}X$ 으로 통합되어 오늘에 이르게 되었습니다. ($H_{L}A_{T}_{E}X$ 0.92e판). $H_{L}A_{T}_{E}X$ 0.93판부터는 문화관광부에서 공개한 글자체와 Postscript type I 글자체, 그 외에도 비트맵 그림(bitmap graphic)에서 추출된 외각선 글자체가 METAFONT 소스로 보급됨으로써 $H_{L}A_{T}_{E}X$ 의 사용이 일반화되었습니다. 글자의 배치도 이전의 KAIST식인 조합 완성 혼합식을 포기하고 일반적인 한국 표준 완성형에 근거하게 되었습니다.

지금까지 나온 한글 $L_{A}T_{E}X$ 를 정리하면 다음과 같습니다.

- **한 $T_{E}X$ Ver 1.5:** (주) 한글과 컴퓨터에서 Windows에서 사용할 수 있게 만들어 상업용으로 출시한 제품; 현재 더 이상 지원하지 않음.
- **원레이텍 Ver 1.0:** (주) LG 소프트에서 Windows에서 사용할 수 있게 만들어 출시한 제품; 현재 더 이상 지원하지 않으나 공개 ftp 자료실에 가면 구할 수 있음.
- **Yah $T_{E}X$ Ver 0.6:** DOS 환경에서 돌아 가는 `emTEX3.1415`에 한글을 사용할 수 있도록 만든 ‘Yet Another Hangul $T_{E}X$ ’; 천리안 Math 동우회나 하이텔 (13. 공개자료실) 등에 가면 구할 수 있음.
- **hLaTeXp:** KAIST 수학과 차제춘 님이 만든 한글 $L_{A}T_{E}X$; 96/02/12 배포판 이후 업데이트가 안되고 있는 것으로 보임.

- **Werner Lemberg의 한글 \TeX :** 1994년 독일의 Werner Lemberg에 의해 만들어진 한글 \TeX ; 이 한글 \TeX 역시 k-시리즈 한글 글꼴에 바탕을 두고 있음; 국내에는 거의 알려지지 않은 한글 \TeX 임.
- **HLaTeX:** 1995년 독일 튀빙엔 대학교의 은광희 님에 의해 만들어진 한글 \LaTeX ; 글꼴 소스와 함께 배포된 현재 사용자가 많은 한글 \LaTeX ; 현재 0.991 버전이 배포되고 있음.

현재 한글 \LaTeX 는 은광희 님이 만드신 H \LaTeX (버전 0.991)와 차재춘 님이 만드신 hLaTeXp의 두 가지가 가장 많이 퍼져 있습니다. 둘은 각기 장단점이 있으나 어느 쪽도 잘 쓸 수 있습니다. 그러나 두 가지를 한 PC에서 사용하는 것은 추천하고 싶지 않은 방법입니다. 이 실천 가이드에서는 은광희 님의 H \LaTeX (ver 0.991)에 대해서만 기술합니다.

제 4 절 \LaTeX 가 언어학자에게 좋은 22가지 이유

\LaTeX 는 한번 빠지면 \LaTeX 예찬론자 되고 마는 단점이 있지만, 일반 사용자에게, 특히 언어학자들에게 다음과 같은 아주 유용한 특징들을 가지고 있다.²

² \LaTeX 가 다른 워드프로세서 보다 좋은 점은 이미 많이 알려져 있는 사항으로 <http://www.doeun.pe.kr/uselatex/prosetcons.html>에 잘 정리되어 있고 (원래는 <http://lark.cc.ukans.edu/mooreb/nomsg.html>에서 인용했다고 하더군요), 특히 언어학자들에게 좋은 점은 일반적인 장점과 구분이 어려운 점도 있어서 무순으로 나열하기로 합니다. 그러나 일반적으로 많이 알려진 워드프로세서인 “아래아 한글”이나 “Microsoft Word”가 \LaTeX 보다는 선호하는 사람들이 생각하는 일반적인 워드프로세서의 장점들도 있습니다. 공정한 비교를 위해 그 장점들을 나열하면 다음과 같습니다.

- Word는 처음 사용할 때 \LaTeX 보다 쉽게 시작할 수 있다. 그렇지만 작성하는 문서가 복잡해지면 복잡해질수록 Word가 돌아가는 방식을 이해하는 것도 그만큼 어려워진다.
- Word는 WYSIWYG라서 더 직관적이고 성격이 급한 사람들에게는 장점이다. 그러나 엄격하게 얘기하면, 그것은 조판 시스템의 설계상의 차이이지 단순하게 장점이라고 말할 수 없다. 최종 조판 결과도 좋으면서 WYSIWYG 방식이

- $L_{A}T_{E}X$ 는 완전히 무료이며 전문가들에 의해 (6개월 마다) 꾸준히 업그레이드가 되고 있다.
- $L_{A}T_{E}X$ 는 예문의 주석 (gloss)을 단어와 주석을 완벽하게 1:1로 대응시켜 아주 쉽고도 완벽하게 이상적인 예문 처리를 해 준다.
- $L_{A}T_{E}X$ 를 사용하면 복잡한 의미론의 수식이나 HPSG나 LFG 등에 혼한 AVLM을 통일적이고 완벽하게 자동으로 처리해 준다.
- $L_{A}T_{E}X$ 에는 없는 문자, 없는 수식 기호가 없다. 그리스 문자를 비롯하여 국제음성기호, 고대의 실험문자까지 $L_{A}T_{E}X$ 에서 구현하지 못할 문자는 없다.
- $L_{A}T_{E}X$ 는 차례, 예문 번호, 표번호, 참고문헌 오류, 색인 작성 등을 자동으로 생성해 주기 때문에 긴 논문이나 저서, 장기간에 걸친 논문작성 혹은 대학원생이 없는 교수님들의 색인작성 등을 원천적으로 쉽고도 오류없이 만들어 준다.
- $L_{A}T_{E}X$ 는 특정부분을 삭제하거나 첨가하여도 오류가 나지 않고 완벽한 문서를 만들어 주기 때문에, 장기간에 걸친 논문작성이나 첨삭이 잦은 신중한 저술활동에 상상하지 못할 정도로 유용하다.
- $L_{A}T_{E}X$ 는 주석처리 기능이 잘 발달되어 있어서 원 문서에는 있으나, 조판되지 않아야 할 개인적인 메모 등을 문서에 그대로 두어도 되기 때문에 문서의 내용 관리가 쉽다.

라면 분명 점점일 것이다. 그러나 Word는 그렇지 않다.

- Word 사용자가 $L_{A}T_{E}X$ 사용자보다 많다. 그러나 Word 사용자가 많다고 해서 Word로 만든 책이 더 팔리는 것은 아니다.
- Word 문서는 많은 경우 “Debug”해야 할 필요가 없다. 그러나 Word도 여러 원천에서 뽑아와 한 문서로 붙일 때에 알 수 없는 에러가 발생하기도 한다.
- 다른 사람과 공동작업을 하는 경우 Word 문서가 더 나올 때가 있다.
- Word로 작업할 때 “사용 설명서”를 옆에 둘 필요가 없다. 그러나 Word도 잘 사용하려면 $L_{A}T_{E}X$ 처럼 “사용 설명서”를 필요로 한다.

- 인터넷 시대에 PostScript 형식의 문서나 PDF (Portable Data Format) 형식의 문서가 표준이 되어 가고 있다. L^AT_EX는 이런 형식의 문서를 아주 쉽게 만들어 주며, 따라서 L^AT_EX는 국제적인 추세이다.
- L^AT_EX로 만든 문서는 폰트에 관계없이 전세계 어느 나라에서든 원래 만든 글자체로 완벽하게 볼 수 있다. 아래아 한글이나 MS Word는 한글 폰트가 있어야 (그리고 특수한 글자체를 사용한 문서라면 그 글자체가 있어야) 제대로 볼 수 있는 불편함에 비추어 L^AT_EX의 커다란 장점이 아닐 수 없다.
- L^AT_EX파일은 특정 기업의 전유물이 아닌 보다 널리 받아들여지는 포맷으로 되어 있어, 컴퓨터 기종에 관계없이 100% 호환되는 세계 유일의 공개 소프트웨어이다.
- L^AT_EX의 입력문서는 단순한 텍스트 문서이기 때문에 L^AT_EX는 플랫폼 독립적이다.
- L^AT_EX는 세계적으로 공인된 고품위 서체를 제공한다.
- L^AT_EX는 플랫폼 독립적이면서도 플랫폼에 따라서 출력결과가 달라지지 않는다.
- L^AT_EX문서는 하드 디스크를 크게 차지하지 않는다.
- 애플리케이션에 의해 문서가 손상될 가능성이 거의 없다.
- 찾아보기, 목차, 타이틀페이지를 자동적으로 생성한다.
- 프린트하거나 저장할 때 프로그램이 죽는 일이 없다.
- L^AT_EX파일을 만들려면 어떤 에디터 독립적이다. 텍스트 문서를 만들 수 있는 에디터이기만 하면 L^AT_EX원판 파일을 만들 수 있다.
- L^AT_EX파일을 생성하는 프로그램을 만들 수도 있다.
- 언제 어디서나 일관된 레이아웃을 유지한다.
- 페이지, 그림, 단락 번호를 체계적으로 유지할 수 있다. 따라서 긴 글을 쓰거나 책을 집필하는 경우 매우 유용하다.

- $L_A T_E X$ 매크로 바이러스가 없다.

제 2 장

“언어학자를 위한 L^AT_EX” CD에서 설치하기

이 장에서는 “언어학자를 위한 L^AT_EX” CD에서 T_EX시스템 및 관련 지원 프로그램을 사용자의 PC에 설치하는 방법을 기술하고 있습니다. T_EX시스템은 OS에 따라 그리고 같은 OS 내에서도 여러 애플리케이션을 선택할 수 있지만, 이 글에서는 Windows용 무료 T_EX배포 프로그램인 MikT_EX Package와 관련 지원 프로그램으로 Ghostscript, GSView, Winedt & Acrobat Reader 등을 설치하는 방법을 설명하기로 합니다.

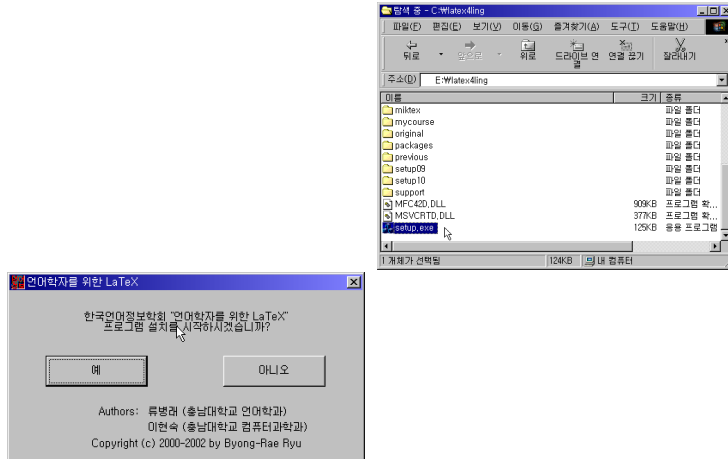
제 1 절 Setup 실행하기

“언어학자를 위한 L^AT_EX” CD에는 L^AT_EX를 Microsoft Windows 95/98/2000/ME/NT 운영체제를 가지고 있는 PC에 손쉽게 설치할 수 있도록 도와주는 실행파일인 `setup.exe`가 들어 있습니다. 그 CD를 CD-ROM 드라이브에 넣고 묻는 말에 답하기만 하면 자동으로 한글L^AT_EX를 완벽하게 설치해 줍니다. 다음은 그 절차를 간단히 설명한 것입니다.

이어지는 절에서는 각 개별 프로그램을 독자적으로 설치할 때의 설명이 들어 있습니다. 따라서 자동 실행 파일의 질문에 답하기 어려운 경우에는 해당 부분을 참조하기 바랍니다. 주의해야 할 사항은 하

드 디스크에 250 MB 정도의 여유 공간이 있어야 한다는 점입니다. 하드에 여유가 없는 사용자는 한글 폰트를 최소로 설치할 것을 권합니다.

1. “언어학자를 위한 L^AT_EX” CD를 CD-ROM 드라이브에 넣고 내용을 보면 아래의 왼쪽 그림과 같이 `setup.exe`가 있는데, 이 실행파일을 실행시킨다.
2. `setup.exe`를 실행하면, ‘한국언어정보학회 ‘언어학자를 위한 L^AT_EX’ 프로그램 설치를 시작하시겠습니까?’라는 질문이 나오는데 “예”를 클릭한다.



제 2 절 MikT_EX 설치하기

3. 그 다음에는 “MikT_EX를 설치하시겠습니까?”라는 질문이 나오는데 “예”를 클릭한다. 그러면 CD에서 MikT_EX 설치 파일을 읽어 오는데 몇 초 걸리고 다음 절에서 설명하는 MikT_EX 설치방법에 나오는 메시지가 계속 이어지는데, 대부분 디폴트 값을 선택하면 된다.
4. MikT_EX가 다 설치되기도 전에 “MikT_EX 프로그램 설치가 완료되었으면 확인을 누르세요”라는 메시지 창이 뜨는데, 그대로 두었다가 MikT_EX가 설치되고 난 다음에 “확인”을 클릭한

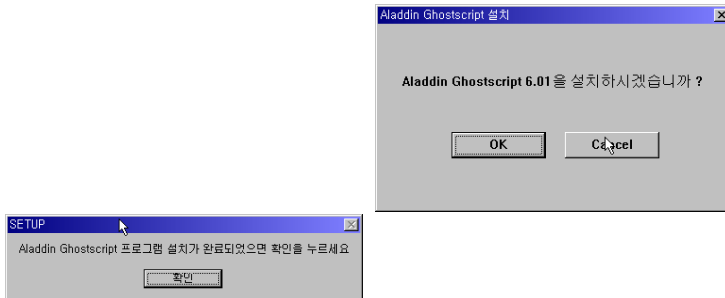
다. 주의할 것은 설치가 완료되지도 않았는데 확인을 누르면 설치하지 않는 것으로 `setup.exe`가 간주하기 때문에 반드시 다 설치한 다음에 “확인”을 눌러야 한다는 것이다.



5. MikTeX가 다 설치되고 난 다음 “MikTeX 프로그램 설치가 완료되었으면 확인을 누르세요”라는 질문에 “확인”을 클릭하면, 그 다음에는 “Aladdin Ghostscript 6.01을 설치하시겠습니까?”라는 질문이 나오는데 “예”를 클릭하고 묻는 말에 답한다.

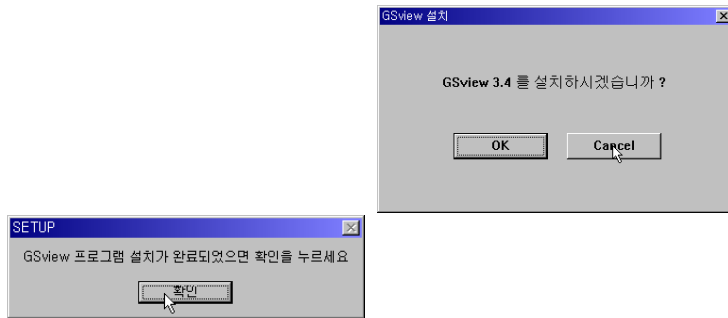
제 3 절 Ghostscript 설치하기

6. “Aladdin Ghostscript 프로그램 설치가 완료되었으면 확인을 누르세요”라는 메시지 창이 뜨는데, 그대로 두었다가 Aladdin Ghostscript가 설치되고 난 다음에 “확인”을 클릭한다.



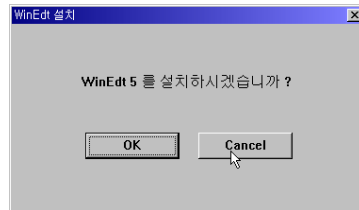
제 4 절 GSView 설치하기

7. Aladdin Ghostscript가 다 설치되고 난 다음 “GSView 3.4를 설치하시겠습니까?”라는 질문이 나오는데 “예”를 클릭하고 묻는 말에 답한다.
8. GSView가 설치되고 난 다음 “GSView 프로그램 설치가 완료되었으면 확인을 누르세요”라는 메시지 창에 있는 “확인”을 클릭한다.



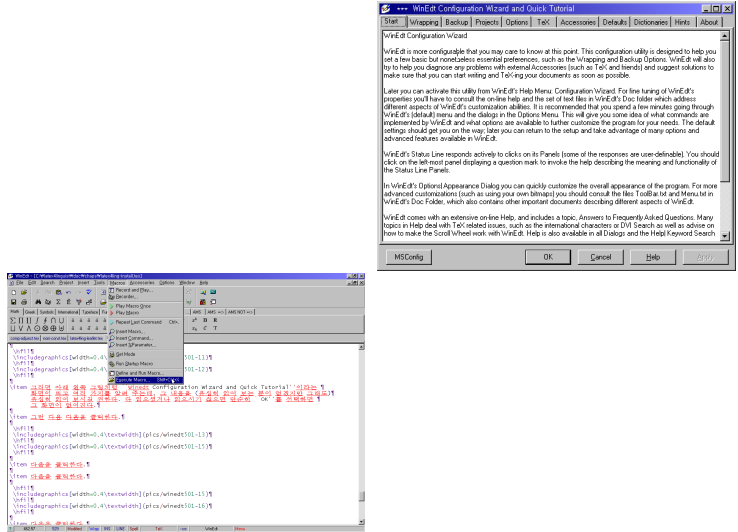
제 5 절 WinEdt 설치하기

9. GSView가 다 설치되고 난 다음 “WinEdt 5를 설치하시겠습니까?”라는 질문이 나오는데 “예”를 클릭하고 묻는 말에 답한다.



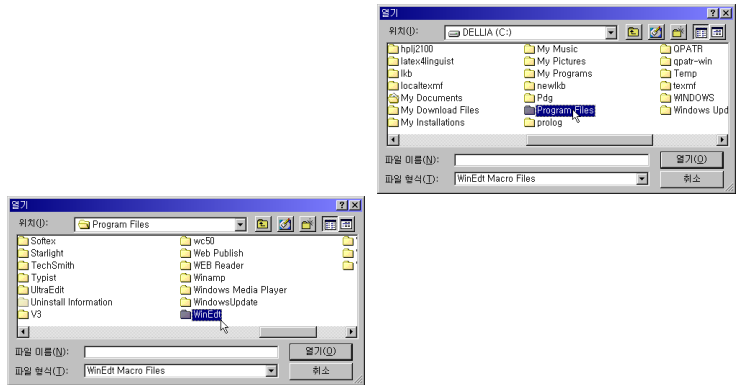
10. 맨 마지막 질문에 “Launch the program files?”라는 질문을 체크하고 “Finish”를 누르면, 아래 왼쪽 그림처럼 “Winedt Configuration Wizard and Quick Tutorial”이라는 화면이 뜨고 여러 가지를 알려 주는데, 그 내용을 (유심히 읽어 보는 분이 없겠지만 그래도) 유심히 읽어 보시길 권한다. 다 읽으셨거나 읽으시기 싫으면 단순히 “OK”를 선택하면 그 화면이 없어진다.

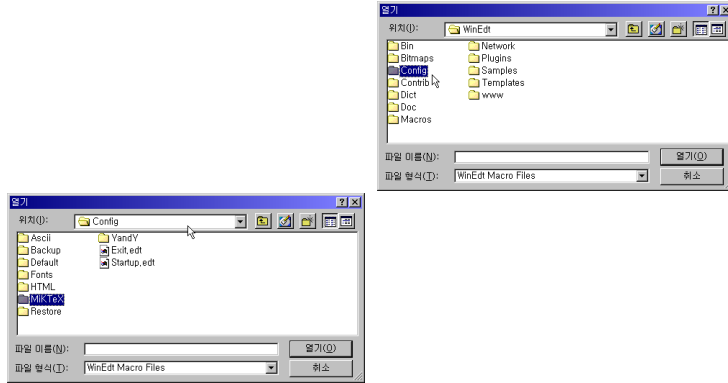
11. 그런 다음 아래 오른쪽 그림처럼 [Macros] > [Execute Macro ...]라는 메뉴를 선택한다.



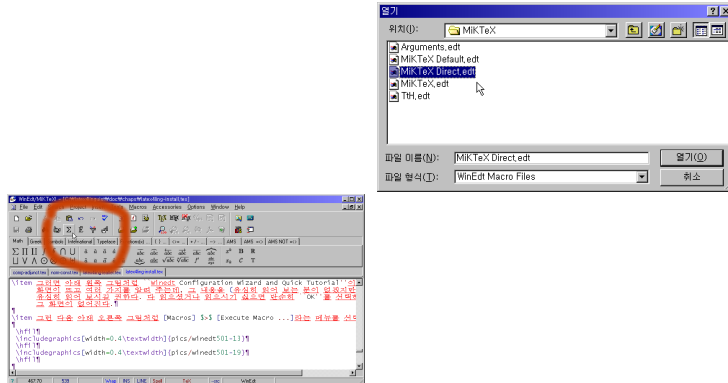
12. 그러면 아래 왼쪽 그림처럼 “Winedt Macro files”를 찾는 “열기” 화면이 나타나는데 이 때 Winedt가 설치되어 있는 폴더를 찾아 아래 오른쪽 그림처럼 “Config” 폴더를 찾아 다시 그 아래에 있는 “MikTeX”라는 폴더를 찾아 그 폴더를 연다. 일반적인 경우라면 다음과 같은 경로 아래에 있다:

c:\Program Files\Winedt\Config\MikTeX

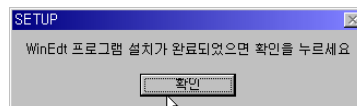




13. 그 폴더를 열면 아래 오른쪽 그림과 같이 5개의 .edt 파일이 들어 있는데, 그 중에서 그림처럼 “MikTeX Direct.edt”를 선택한다.
14. 그런 다음 아래 오른쪽 그림처럼 Pull-down 메뉴 아래에 있는 아이콘들 중에서 Σ 를 선택한다. 그러면 여러 가지를 특수기호를 선택만 하면 바로 입력해 주는 모드로 설정이 된다. 이렇게 하면 WinEdt의 설치가 완벽하게 끝난다.

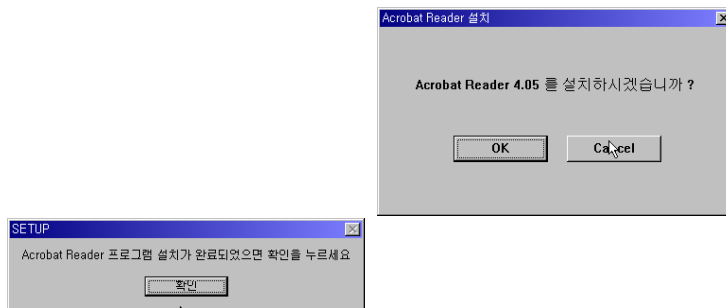


15. WinEdt 5가 설치되고 난 다음 “WinEdt 프로그램 설치가 완료되었으면 확인을 누르세요”라는 메시지 창에 있는 “확인”을 클릭한다.



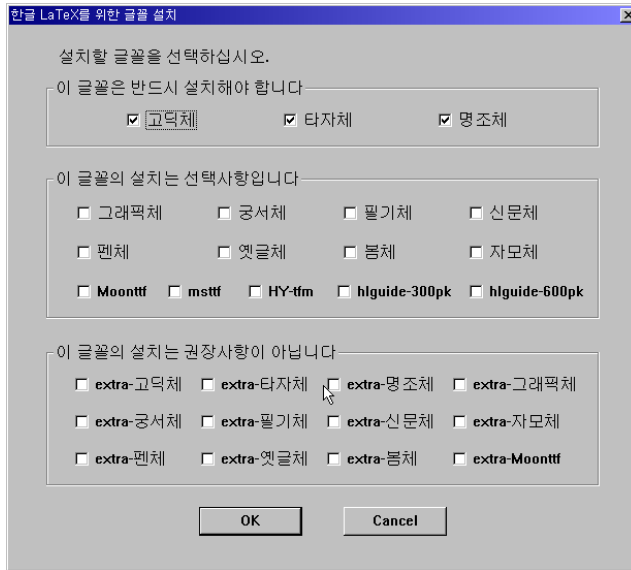
제 6 절 Acrobat Reader 설치하기

16. WinEdt 5가 다 설치되고 난 다음 “Acrobat Reader 4.05를 설치하시겠습니까?”라는 질문이 나오는데 설치하려면 “예”를 클릭하고 묻는 말에 답하고, 설치하지 않으려면 “아니오”를 클릭하고 다음 단계로 넘어간다.
17. Acrobat Reader가 설치되고 난 다음 “Acrobat Reader 프로그램 설치가 완료되었으면 확인을 누르세요”라는 메시지 창에 있는 “확인”을 클릭한다.

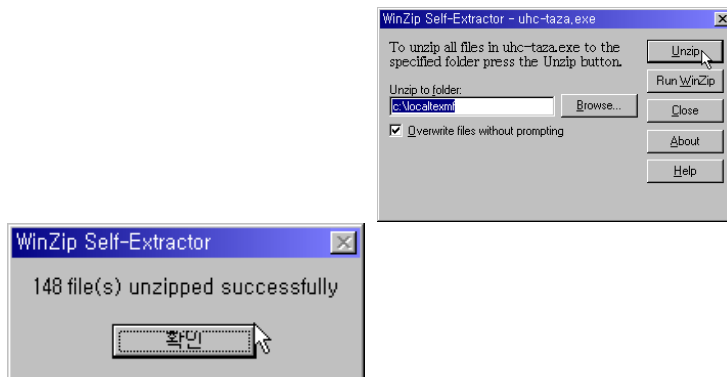


제 7 절 H_AT_EX 설치하기

18. 다음 단계는 아래 그림과 같이 H_AT_EX를 위한 글꼴 설치 화면이 나타나는데, 기본적으로 설치할 것은 이미 체크가 되어 있기 때문에 최소 설치를 위한 사람은 바로 “예”를 클릭하고, 더 설치하고자 하는 글꼴이 있으면 하드에 여유가 많다는 전제하에 원하는 글꼴을 체크하고 “예”를 클릭한다.



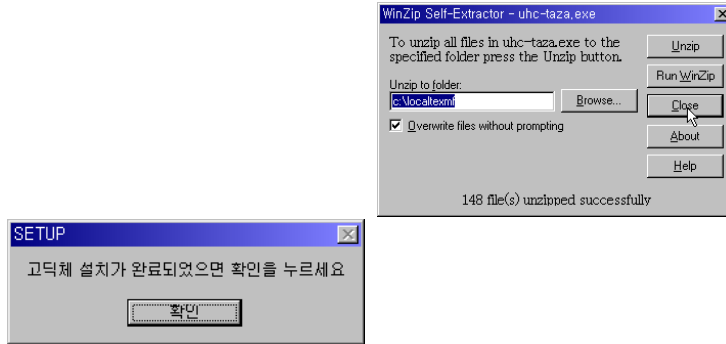
19. 글꼴들은 이미 실행 파일로 변환되어 있기 때문에 위에서 체크된 글꼴들은 순서에 상관없이 모두 설치되게 된다. 이 때 각 글꼴마다 다음과 같은 단계를 거친다. 우선 왼쪽 그림처럼 Winzip self Extractor가 실행되고 화면이 뜨는데, 이 때 Unzip을 클릭하면 그 글꼴이 설치되고 오른쪽과 같은 화면이 뜨는데 “확인”을 누른다.



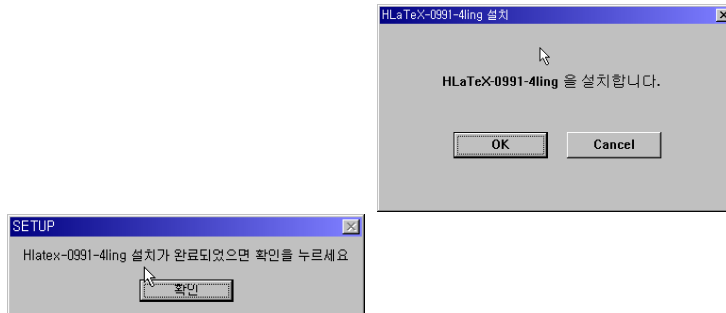
20. 그러면 다음 왼쪽 그림처럼 Winzip self Extractor 화면으로 돌아 가는데, 이 때 Close를 클릭한다.

21. 그러면 “XX체 설치가 완료되었으면 확인을 누르세요”라는 메

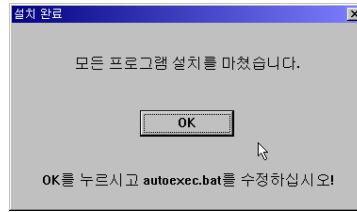
시지가 뜨는데, 설치가 되었으면 “확인”을 선택하여 다음 글꼴을 동일한 방법으로 설치한다. 선택한 글꼴이 많으면 이 과정을 선택한 글꼴의 수 만큼 반복하게 된다.



22. 글꼴 설치가 끝나 마지막 “XX체 설치가 완료되었으면 확인을 누르세요”라는 메시지에 “확인”을 클릭하면, 반드시 설치해야 하는 $\text{H}\text{L}\text{A}\text{T}\text{E}\text{X}$ -0991-4Ling을 설치하는 화면으로 넘어간다. 당연히 “ $\text{H}\text{L}\text{A}\text{T}\text{E}\text{X}$ -0991-4Ling을 설치합니다”에 아래 왼쪽 그림처럼 “예”를 선택한다.



23. 글꼴 설치하는 방법과 같이 이 실행파일도 Winzip self Extractor로 만들어져 있기 때문에 위에서와 같은 절차를 밟아 설치를 마치고 “ $\text{H}\text{L}\text{A}\text{T}\text{E}\text{X}$ -0991-4Ling 설치가 완료되었으면 확인을 누르세요”라는 메시지에 “확인”을 선택한다.

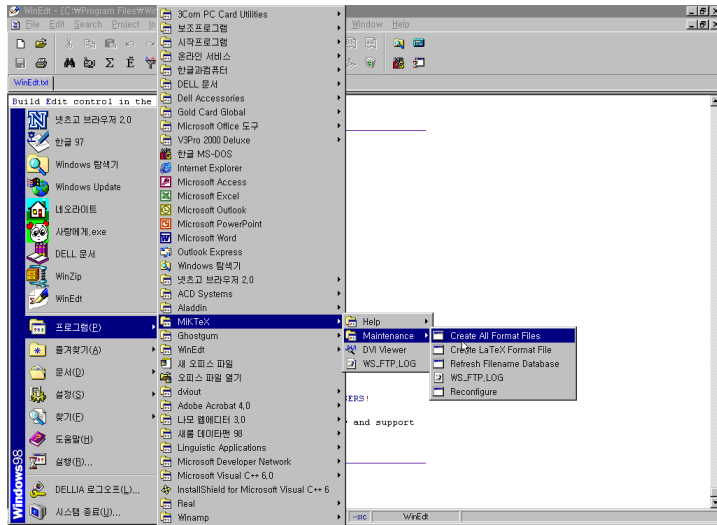


제 8 절 설정 변경하기

24. 모든 프로그램의 설치가 끝나면 아래 화면이 나타나는데 이때 “예”를 선택한다. 이 때 새로 설치된 프로그램들의 경로를 `autoexec.bat`에 추가해 주어야 한다. `autoexec.bat`를 텍스트 에디터로 열어 아래에 있는 내용이 없으면 그 내용을 `autoexec.bat`를 텍스트 에디터로 열어 이미 들어 있는 설정 아래에 다음과 같이 추가한다.

```
REM \begin{MiKTeX}
SET PATH=%PATH%;C:\TEXMF\MIKTEX\BIN;
REM \end{MiKTeX}
SET PATH=%PATH%;C:\ALADDIN\GS6.01;C:\GHOSTGUM\GSVIEW
SET PATH=C:\PROGRA~1\WINEDT
```

25. 이렇게 하면 일단은 기본적으로 필요한 프로그램이 모두 설치 되었다고 볼 수 있는데, Mik_TE_X를 사용하기까지는 다음 오른쪽 그림처럼
[시작] > [프로그램] > [Mik_TE_X] > [Maintenance]를 가서
“Creat All Format Files”를 선택하고 Format File이 만들어지고 나면, “Refresh Filename Database”를 실행시킨다.



26. WinEdt는 $\text{T}_{\text{E}}\text{X}$ 가 컴파일 과정에서 에러를 만났을 때, 도스창에서 e라고만 치면 에러난 곳이 어느 곳인지 알려주는 기능이 있습니다. 이 기능은 아주 유용한 기능이므로 다음과 같이 Mik $\text{T}_{\text{E}}\text{X}$ 의 초기화 파일 (`c:\texmf\miktex\config\miktex.ini`)을 수정해서 사용하시기를 추천합니다.

↳ 텍스트 에디터를 사용해서 `miktex.ini` 파일을 열고 이 파일의 마지막 부분에

```
;; Editor=notepad "%f"
;; Editor=winedt "[Open('%f');SelLine(%1,7)]"
;; Editor=emacs +%1 "%f"
Editor=notepad "%f"
```

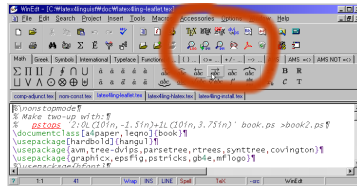
라고 되어 있으면, 다음과 같이 고치고 저장한다.

```
;; Editor=notepad "%f"
Editor=winedt "[Open('%f');SelLine(%1,7)]"
;; Editor=emacs +%1 "%f"
```

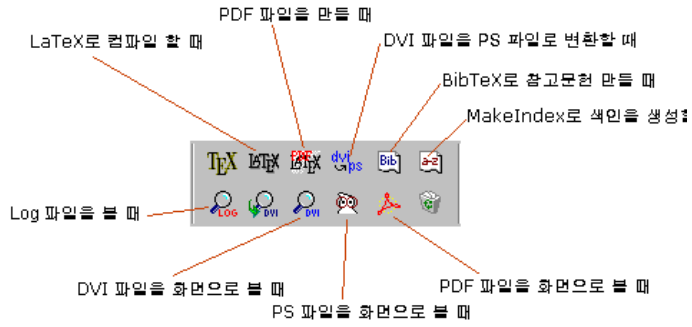
```
;; Editor=notepad "%f"
```

이 명령은 Mik_TE_X와 연동하는 에디터로 Notepad가 아닌 Winedt를 사용하겠다는 뜻을 Mik_TE_X에 전하는 역할을 한다.

27. 마지막으로 테스트를 해 보시길 권하고 싶은데, 이미 만들어진 문서를 찾아 L^AT_EX로 컴파일 시키는 것이 좋다. 우선 바탕 화면에 있는 WinEdt를 실행시키고, `c:\texmf\doc\ksli` 아래에 있는 `latex4ling-sample.tex`를 열고 메뉴 아이콘 중에서 LaTeX를 누른 후 도스창이 없어지면 DVI 아이콘을 눌러 `latex4ling-sample.dvi`를 본다. 잘 보이면 일단은 성공적으로 설치되었다고 안심해도 됩니다. 축하합니다.
28. Winedt에서는 아래에 있는 그림처럼 아이콘만 누르면 명령이 자동으로 수행됩니다.



29. 관련이 있는 아이콘을 조금 확대하면 다음과 같습니다. 그리고 각 아이콘이 무슨 명령을 수행하도록 프로그래밍 되었는지 아래 그림을 보면 알 수 있습니다.



제 3 장

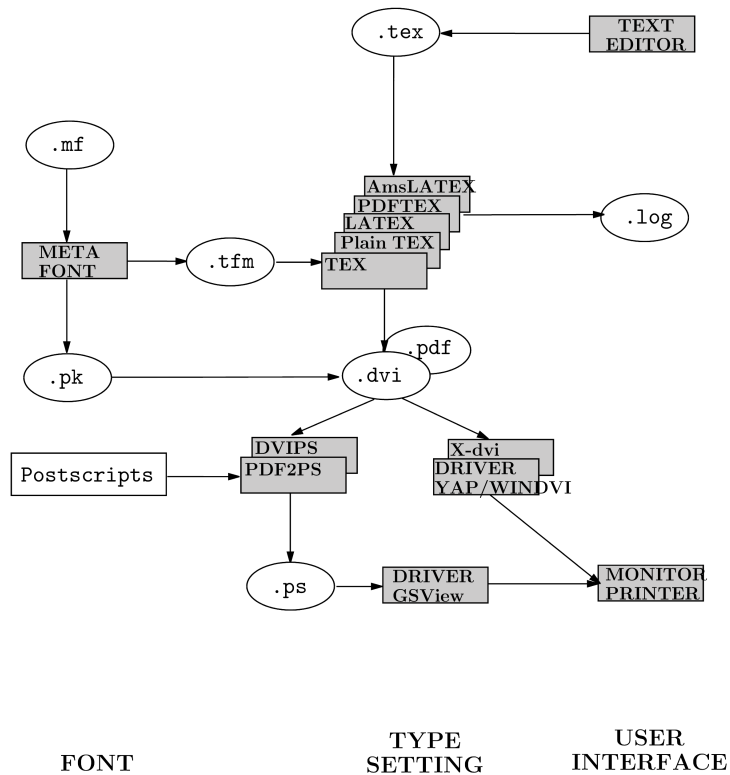
한글 \LaTeX 로 만드는 문서의 구조

제 1 절 \LaTeX 는 문서를 어떻게 조판하는가

\TeX 로 문서를 만드는 과정은 다음과 같습니다.

1. 적당한 편집기로 \TeX 소스화일 (*.tex)을 만듭니다.
 2. \TeX 소스 화일을 컴파일합니다. 그러면 *.dvi 화일이 나옵니다.
 3. *.dvi 화일은 dvi viewer 로 화면에서 보거나 프린트할 수 있습니다. 대부분 출력 장치의 표준이 ps (포스트스크립트) 이므로 보통 dvips 등으로 ps 화일로 변환한 후 출력하거나, *.pdf로 변환해서 출력합니다.
- \TeX 프로그램을 사용하여 input file을 처리하면 여러 형태의 output device (프린터, 화면 등) 에 적용할 수 있는 명령들을 포함하는 DEVICE INDEPENDENT 파일이라 불리는 *.dvi 라는 이진파일과
 - 처리내용의 정보를 담고 있는 로그파일인 *.log 파일,

- 자동으로 매기는 장, 절, 정리, 수식 번호나 cross reference 등의 정보가 담긴 *.aux 파일이 생긴다.
- DVI 파일을 dviwin이나 Yap과 같은 스크린 드라이버나 프린터 드라이버를 사용하여 화면이나 프린터로 조판된 결과를 본다. UNIX계열의 시스템에서는 Postscript파일로 전환한 후 출력하게된다.
- 위에서 설명한 절차들을 간략하게 그림으로 나타내면 다음과 같다.



제 2 절 논리적인 문서 디자인과 시각적인 문서 디자인

문서의 디자인에는 제목, 소제목, 차례, 각 항목, 본문, 인용문 등의 논리적 요소와 글자의 크기나 모양, 배치 등 시각적 요소가 있습니다.

오늘날 DTP (Desk Top Publishing) 나 워드프로세서의 문서 디자인 방법의 대세는 직접 문서의 모양을 보면서 편집하는 WYSIWYG (What You See Is What You Get) 입니다.

이에 대하여 시각적 개념이 아닌 논리적인 개념으로 문서를 디자인하는 방법이 있습니다. 이와 같이 하면 내용과 논리적인 구성에 집중하여 글을 쓸 수 있으며 문서가 논리적인 구조로 정리되므로 독자들도 읽기 쉽게 됩니다.

WYSIWYG 에서는 마우스로 대상을 선택하여 직접 명령을 내리는 식으로 디자인을 합니다. 논리적인 개념의 디자인에서 쓰는 방법은 보통 마크 업 (Mark Up) 이라는 방법이며 텍스트 화일에 필요한 디자인 요소들도 같이 적어 놓는 것입니다. 이는 마치 집을 지을 때 현장에서 직접 손으로 지휘하는 것과 현장에는 나오지 않으나 설계도를 그려주는 것에 비유할 수 있습니다.

워드프로세서는 문서를 만드는 것을 도와주는 프로그램입니다. 워드프로세서는 발전을 거듭한 결과 이제는 다양한 글자모양 (폰트) 과 그림이나 표 등 문자 이외의 요소들도 넣을 수 있게 되어 거의 DTP 와 같이 책을 만들 수 있는 수준에까지 이르렀습니다. 그러나 이러한 편집 작업은 여전히 상당 부분 수작업으로 되어 있습니다. 사실 그러한 세세한 편집이야말로 컴퓨터에게 맡겨야 할 일입니다.

출판사에 원고를 맡기면서 책의 모양새를 그려주면 책이 출판되어 나오듯이 문서의 내용과 편집 방침을 컴퓨터에 넣고 처리하여 알맞은 형식으로 정리된 문서를 얻는 방법은 몇 가지가 있습니다. 그중에서도 대표적이고 사용하기 쉬우며 널리 쓰이는 것이 TEX 입니다.

일반적인 워드프로세서나 탁상출판시스템 (DTP) 프로그램들은 화면에서 직접 문서가 만들어지는 모양을 보면서 작업합니다. 작업 환경은 마치 커다란 종이 위에 직접 펜으로 글을 쓰고 그림이나 표

를 오려 붙이는 것과 비슷합니다. \TeX 로 하는 작업은 상당히 간접적입니다. 모든 작업은 텍스트파일 (메모장 등으로 만드는 글자와 개행기호 (줄바꿈기호) 만이 있는 화일) 로 하며 그림이나 표도 텍스트 화일 내에 그림이나 표를 넣는 명령을 씌우므로써 합니다. 이는 마치 타자기로 원고지만을 친 다음 나머지 조판 작업은 전문가에게 맡기는 것과 비슷합니다.

\TeX 의 이러한 면은 어떻게 보면 어렵게 보입니다. 그러나 WYSIWYG (What You See Is What You Get) 을 내세우는 DTP 작업도 막상 마우스를 잡고 써 보면 논리적이지 못한 경우가 매우 많습니다. \TeX 는 글자와 도형의 크기와 위치를 65536 분의 1 인치 (약 0.0004 mm) 단위로 매우 섬세하게 조작하며 그 작업 또한 매우 논리적으로 만들어져 있습니다. 또한 문서작성을 도와주는 도구 - \TeX 매크로 - 의 수는 WYSIWYG 프로그램의 도구에 비할 바 없이 많습니다.

이러한 \TeX 가 가장 위력을 발휘하는 것은 수식을 많이 포함하는 수학이나 이공계 문서입니다. 또한 긴 글을 쓸 때에도 위력을 발휘합니다. 400 페이지 가량 되는 책이 있고 각 면마다 2,3 개의 그림과 도표가 있다면 이를 워드프로세서로 만드는 것은 대단한 시간낭비일 것입니다. 다 만들고 나서 각 소단원의 제목의 글자모양과 크기, 위치를 일제히 바꾸고자 하면, 그리고 이러한 제목이 400 개 정도 있다고 하면 이는 매우 어려운 일입니다. 그러나 \TeX 에서는 2,3 줄을 수정해서 제목의 정의를 바꾸면 해결됩니다.

또한 \TeX 소스 화일은 텍스트화일이므로 어떠한 컴퓨터에서도 읽고 쓸 수 있고 자신의 입맛에 맞는 에디터 (편집기) 를 쓸 수 있으며 출력결과물인 dvi 화일 또한 이름 그대로 장치독립적 (device-independent) 이므로 컴퓨터의 기종이나 OS 를 가리지 않습니다.

제 3 절 \TeX 의 기초

3.1 기본 명령어

- \TeX 에서는 두 종류의 글자가 있다. 하나는 출력물에 실제로 나타나는 글자이고 다른 하나는 \TeX 컴파일러에 지시하는 명령어로서의 의미를 갖는 것이다.

일반적인 글자(알파벳, 한글, 숫자 등)은 그대로 $\text{T}_{\text{E}}\text{X}$ 에서 인식하며 쉼표 뒤에는 짧은 여백, 마침표 뒤에는 긴 여백을 알아서 만들어 준다. 명령어는 \backslash (back slash)으로 시작하는 단어들이다. 명령어에서는 대문자와 소문자를 구별하여 쓴다. 그러므로 $\backslash\text{TeX}$ 과 $\backslash\text{tex}$ 은 다른 명령이 된다.

$\text{T}_{\text{E}}\text{X}$ 컴파일러에게 특별한 의미를 갖는 글자가 다음과 같이 몇 개 있다.

기호	설명
\backslash	$\text{T}_{\text{E}}\text{X}$ 의 명령을 나타내는 시작기호
$\$$	일반 텍스트모드와 수학모드를 구별한다
$\#$	매크로를 쓸 때 변수를 구별
$\&$	표를 그릴 때 열을 구별하는 기호
\sim	새 행에 의해 갈라지지 않는 여백
$\^$	위 첨자 표시할 때 쓴다
$_$	아래 첨자 표시할 때 쓴다
$\%$	주석의 시작을 표시
$\{$	덩어리의 시작
$\}$	덩어리의 끝
$\@$	다이아그램을 작성할 때 사용
$\backslash\backslash$	행바꿈 명령
\prime	왼쪽 홑 따옴표 (‘)
\rangle	오른쪽 홑 따옴표 (’)
\llcorner	왼쪽 겹따옴표 (“)
\lrcorner	오른쪽 겹따옴표 (”)
$-$	하이픈 (-)
$--$	중간 하이픈 (-)
$---$	긴 연결선 (—)
$\$ \cdot \$$	중간점 (·)
$\$ \dots \$$	긴 중간점 (...)

- 위 기호들을 글자 그대로 입력하려면 $\$ \backslash \$$, $\backslash \$$, $\backslash \#$ 으로 \backslash 을 글자 앞에 붙이면 된다.
- 입력시에 단어사이의 탭이나 띄우기(space) 또는 줄바꿈은 무

시하며 자동으로 가지런히 정렬된 상태로 나오게 된다. 그러므로 한 줄의 길이 등에 전혀 신경을 쓰지 않아도 된다. 예를 들면

```
This is          a
                book.
This  is a      book.
```

은 다 같이

This is a book.

으로 정상적인 결과로 출력된다.

- 문장을 바꿀 때는 완전히 빈 줄을 한 줄 이상 삽입하거나 또는 명령을 쓴다. 각 문장은 시작할 때 정해진 양의 공간만큼 들여 쓰기로 시작한다.

\TeX 의 기본 정신의 하나는 문서의 조판은 가능한 최소의 노력만으로 되도록 하는데 있다. 각 문단의 모양새나 줄넘김 등은 \TeX 가 알아서 최상의 품질로 계산한다. 그러므로 특별한 경우가 아닌 이상 강제로 줄바꿈이나 들여쓰기 등을 한다거나 하는 것은 자제되어야 한다.

3.2 문서형식

- 모든 \LaTeX 문서는 다음의 구조를 갖는다.

```

\documentclass[선택항목]{문서양식}
\usepackage[선택항목]{패키지}
...
... Preamble (초기화 영역)
...
\begin{document}
...
... (문서내용)
...
\end{document}

```

- 여기서 `documentclass`에서 사용하는 선택항목은 다음과 같다. 선택항목이 하나 이상 나올 때는 단순히 콤마를 사용하여 나열하면 된다.

영역	선택항목, 설명
글꼴크기	10pt (default), 11pt, 12pt
단편집	onecolumn (default), twocolumn
면	oneside (default), twoside
용지방향	portrait (default), landscape
용지크기	letterpaper, legalpaper, executivepaper a4paper, a5paper, b5paper
수식번호위치	leqno

- 문서형식은 다음과 같다. 문서형식은 하나만 택한다

양식	설명
book	책을 작성할 때
report	장으로 나누어진 큰 보고서
article	수십페이지 정도의 간단한 문서
letter	편지 작성할 때
slides	슬라이드 만들 때
amsart	AMSLaTeX을 사용한 article양식
amsbook	AMSLaTeX을 사용한 book양식

- `\usepackage`는 패키지를 하나씩만 담아서 여러 행에 걸쳐 있어도 상관없다. 더구나 패키지마다 선택항목이 다른 경우에는

이런 방식이 필수적이다.

패키지 문서는 새로운 명령이나 환경을 정의하거나 문서 클래스의 정의된 요소들을 재정의, 또는 교정해 주는 것이다. 여러 가지 특수한 목적을 가진 패키지 문서들을 사용함으로써 다양한 형식의 문서를 쉽게 작성할 수 있다. 잘 알려진 패키지로는 `latexsym`, `amsmath`, `makeidx`, `doc`, `exscale`, `fontenc`, `newlfont`, `oldlfont` 등이 있다.

- 초기화 영역(Preamble)이란 문서 전체에 영향을 미치는 사항에 관한 정의(macro)를 내리는 곳이다.
- 다음은 $\text{H}^{\text{L}}\text{A}^{\text{T}}\text{E}^{\text{X}}$ 에서 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 를 사용하여 언어학 논문을 편집하기 위한 가장 간단하고 표준적인 원본 문서의 예이다.


```

\documentclass[11pt,a4paper]{article}
\usepackage[hardbold]{hangul}
\usepackage{avm,tree-dvips,amsmath}
\usepackage{chicago,parsetree,graphicx,pifont, gb4e}
\title{언어학 논문 견본}
\author{류병태\
        {\small 충남대학교 언어학과}\
        {\tt\small http://www.cnu.ac.kr/~ryu/}\
        {\tt\small ryu@cnu.ac.kr}}
\date{2000년 10월 3일}
\begin{document}
\maketitle
\begin{abstract}
이 문장은 언어학 논문 견본의 요약문 첫 문장입니다.
많이 이용해 주시고 사랑해 주세요.
(충남대학교 언어학과)
\end{abstract}
\section{머리말}
이 문장은 언어학 논문 견본의 머리말 첫 문단
첫 문장입니다. 이 문장은 언어학 논문 견본의
머리말 첫 문단 두번째 문장입니다.

문단은 그냥 빈 줄 하나만 있으면 된다는 것을 아셨죠?
빈 줄이 여럿 있어도 하나의 문단 나누기로 간주 합니다.

\section{언어학이 좋은 이유}
이 문장은 언어학이 좋은 이유를 기술하고 있습니다.

\subsection{언어학이 몸에 좋은 이유}
이 문장은 언어학이 몸에 좋은 이유를 기술하고
있습니다.

\subsection{언어학이 머리에 좋은 이유}
이 문장은 언어학이 머리에 좋은 이유를 기술하고
있습니다.

\section{언어학이 매력적이고 도전적인 이유}
이 문장은 언어학이 매력적이고 도전적인 이유를
기술하고 있습니다.

\section{결론}

```

언어학 논문 견본

류병래
충남대학교 언어학과

0000년 00월 0일

요약

이 문장은 언어학 논문 견본의 요약문 첫 문장입니다(많이 이용해
주시고 사랑해 주세요((충남대학교 언어학과)

제 1 절 머리말

이 문장은 언어학 논문 견본의 머리말 첫 문단 첫 문장입니다. 이 문장은 언
어학 논문 견본의 머리말 첫 문단 두번째 문장입니다.

문단은 그냥 빈 줄 하나만 있으면 된다는 것을 아셨죠. 빈 줄이 여럿 있
어도 하나의 문단 나누기로 간주합니다.

제 1 절 언어학이 좋은 이유

이 문장은 언어학이 좋은 이유를 기술하고 있습니다.

... 언어학이 몸에 좋은 이유

이 문장은 언어학이 몸에 좋은 이유를 기술하고 있습니다.

... 언어학이 머리에 좋은 이유

이 문장은 언어학이 머리에 좋은 이유를 기술하고 있습니다.

제 1 절 언어학이 매력적이고 도전적인 이유

이 문장은 언어학이 매력적이고 도전적인 이유를 기술하고 있습니다.

제 1 절 결론

이 문장은 언어학 논문 견본의 결론입니다.

제 4 장

한글 \LaTeX 로 예문과 예문주석 달기

예문을 쓰지 않는 언어학 논문은 극히 드물다. 그러나 일반 워드프로세서를 사용한 예문 작업은 적어도 언어학자에게는 다음과 같은 네 가지의 아쉬움 혹은 어려움을 안겨 주었다.

- 일반적인 워드프로세서에서 예문을 출판된 책에서처럼 여백을 지켜 완벽하게 그리기란 쉬운 일이 아니다.
- 또한 영어로 된 논문에서 영어로 주석을 다는 경우 한국어 예문과 영어 주석을 단어 단위로 정렬을 하기가 쉽지 않음을 우리는 흔하게 경험한다.
- 더구나 논문의 완성 후 새로운 예문을 추가하거나 삭제해야 하는 경우 예문의 일련 번호를 실수없이 부여하는 작업은 많은 시간을 요하기도 하지만 논문이나 책이 긴 경우에는 실수없이 하는 경우가 없다고 해도 과언이 아니다.
- 마지막으로 “X-바 이론”을 소개하거나 다룰 때, 바의 수를 자유 자재로 넣기란 역시 쉬운 일이 아니었다.

\LaTeX 는 위에 제시한 네 가지 예문과 관련된 어려움은 물론이고 기타 예문과 관련되어 언어학자들이 가지고 있는 어려움을 한꺼번에

다 해결해 준다. 이 장은 \LaTeX 의 예문과 관련된 뛰어난 장점을 설명하고 어떻게 사용하는지 예를 통해 설명한다.

무엇보다도 새로운 \LaTeX 기법을 배우는 가장 좋은 방법은 그냥 예제를 따라서 하면 된다는 저자의 경험에 비추어 이 장에서는 가능하면 대표적인 좋은 예들을 제시하면서 예문과 예문에 주석을 완벽하게 다는 방법을 설명하려고 한다.

제 1 절 꾸러미 소개

언어학자들이 가장 많이 사용하는 예문 작성용 꾸러미로는 `gb4e.sty`과 `covington.sty`가 알려져 있다. 이 절에서는 이 두 꾸러미를 그 두 꾸러미에 대한 사용자 설명서를 중심으로 설치방법과 사용법을 기술하기로 한다.

예문의 번호 붙이기에 관한 한 두 꾸러미가 사용하는 방법은 같다. 왜냐 하면, `gb4e.sty`에서 사용하는 명령은 `covington.sty`의 것과 같고, 다시 `covington.sty`은 M. der Groot가 이 만든 `Midnight gloss` 혹은 `gloss.tex`라는 \TeX 매크로를 사용하기 때문이다. 그러나 꾸러미의 포괄 범위가 동일한 것도 아니고 각기 장단점을 가지고 있기 때문에 사용자의 구미에 따라 하나 혹은 (사용 범위를 다르게 할 경우) 두 꾸러미를 동시에 사용해도 된다고 생각한다.

- `gb4e.sty`은 “Government-Binding style” 꾸러미의 4e 버전으로서, 저자는 독일 Tübingen 대학교 언어학과의 연구원 Hans-Peter Kolb (`kolb@sfs.nphil.uni-tuebingen.de`)와 네덜란드의 Craig Thiersch 교수 (`thiersch@kub.nl`) 이다. 저자들은 M. Covington, M. der Groot, J. Frampton and L. Holt 등의 아이디어를 쓰기 좋게 통합시킨 것이라 밝히고 있다. `gb4e.sty`는 Michael A. Covington의 `cgloss4e.sty`을 요구하기 때문에 `gb4e.sty`를 사용하려면 `cgloss4e.sty`도 함께 인스톨되어 있어야 한다.
- `gb4e.sty`은 예문 번호를 (2)a, (3)a 식으로 기입하기에 편하고 Linguistic Inquiry 스타일의 언어학 논문에서 사용하는 포맷의 예문 번호를 부여한다. 다음은 전형적인 예이다.

(1) a. 이 문장은 첫번째 예문입니다.

b. 이 문장은 두번째 예문입니다.

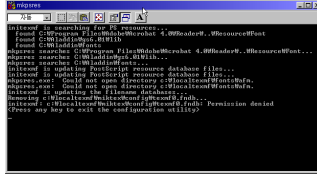
- `covington.sty`은 Michael A. Covington (Artificial Intelligence Programs, The University of Georgia, `mcovingt@aisun1.ai.uga.edu`)가 만든 꾸러미이다. `covington.sty`은 Multiple accents (e.g., \acute{e})를 지원한다.
- `covington.sty`은 예문 번호를 (5.1) (a), (5.2) (a) 식으로 장 번호와 함께 기입하기에 편하고 Chomsky의 Knowledge of Language에서 처럼 예문번호를 오른쪽에 넣을 수도 있도록 해 준다. 다음은 전형적인 예이다.

(4.1) (a) This is the first sentence.

(b) This is the second sentence.

제 2 절 꾸러미 설치하기: gb4e.sty & covington.sty

1. `gb4e.sty`을 설치하기 위해서는, `c:\localtexmf\tex\latex\`아래에 `gb4e`라는 디렉토리를 하나 만들고, 언어학자를 위한 L^AT_EX CD의 `packages` 아래에 있는 `gb4e.sty`와 `cgloss4e.sty`를 그 아래에 복사한다.
2. `covington.sty`을 설치하기 위해서는, `c:\localtexmf\tex\latex\`아래에 `covington`이라는 디렉토리를 하나 만들고, 언어학자를 위한 L^AT_EX CD의 `packages` 아래에 있는 `covington.sty`를 그 아래에 복사한다.
3. 어느 꾸러미를 설치하든지 MikTeX가 각 꾸러미 파일을 `c:\localtexmf\tex\latex\` 아래에 있다는 것을 알아야 하기 때문에 꾸러미를 해당 폴더에 복사하고 나서는 반드시 시작>프로그램>MikTeX>Maintenance>Refresh File Name Database를 실행시킨다. 그러면 여러분은 다음과 같은 도스창을 보게 될 것이고, File Name Database가 새롭게 작성되고 나면 스페이스 키를 눌러 주면 그 창은 없어진다.



제 3 절 꾸러미 사용하기 (1): gb4e.sty

3.1 초보적인 사용법에서 전문가적 사용법까지

- `\begin{document}`가 시작하기 전인 초기화 영역(Preamble)에 다음과 같이 이 꾸러미를 사용한다고 선언해 두어야 한다.

```
\usepackage{gb4e}
```

- `gb4e.sty`를 이용해서 예문을 넣기 위해서는 `\begin{exe}`와 `\end{exe}` 라는 환경을 적고 그 사이에 `\ex` 라고 적은 다음에 예문을 적는다. 주의할 것은 글쓰는 사람이 예문 번호를 지정하는 것이 아니라 \TeX 가 알아서 예문 번호를 부여한다는 것이다. [구체적인 예문 번호를 본문에서 다시 인용해야 할 경우가 있는데 그 경우는 `\label{ex1}`과 (`\ref{ex1}`) 같은 쌍을 이용하는데 아래에서 자세히 설명한다. 이 예문을 다음에 이용하기 위해 `\label{ex1}`을 기입해 둔다.] 다음은 그 예이다.

```
\begin{exe}
\ex\label{ex1} 이것은 예문이다.
\end{exe}
```

(2) 이것은 예문이다.

- `gb4e.sty`를 이용해서 여러 하위 예문이 있는 예문군을 넣기 위해서는

```
\begin{exe}\ex
\begin{xlist}
\end{xlist}
\end{exe}
```

이라는 환경을 적고 예문은 그 사이에 `\ex` 라고 적고 이어서 적는다. 다음은 그 예이다.

```
\begin{exe}\ex\label{ex2}
\begin{xlist}
\ex 이것은 첫번째 예문이다.
\ex 이것은 두번째 예문이다.
\end{xlist}
\end{exe}
```

- (3) a. 이것은 첫번째 예문이다.
b. 이것은 두번째 예문이다.

- `gb4e.sty`를 이용해서 상위 예문을 적고 그 아래에 여러 하위 예문이 있는 예문군을 넣기 위해서는 `\begin{exe}` 다음에 `\ex` 라고 적고 상위 예문을 적고, 하위 예문들은 `\begin{xlist}` 다음에 `\ex` 라고 적은 다음 예문을 적으면 된다. 주의할 사항은 `\begin{exe}`나 `\begin{xlist}` 로 시작한 환경은 내포된 순서에 따라 `\end{xlist}`나 `\end{exe}`로 완결되도록 해야 한다. 그렇지 않으면 에러 메시지를 받게 된다. 다음은 그 예이다.

```
\begin{exe}
\ex\label{ex3} 다음은 두 개의 한국어 예문이다.
\begin{xlist}
\ex 이것은 첫번째 예문이다.
\ex 이것은 두번째 예문이다.
\end{xlist}
\end{exe}
```

- (4) 다음은 두 개의 한국어 예문이다.

- a. 이것은 첫번째 예문이다.
 - b. 이것은 두번째 예문이다.
- 하위 예문은 계속 반복되어 내포될 수 있다. 이런 경우를 위해서는 계속해서 `\begin{xlist}`와 `\end{xlist}` 환경을 쓰면 된다. 예문 번호는 \TeX 가 알아서 자동으로 메겨 주기 때문에 신경 쓸 필요가 없다. 그러나 특정한 형태의 번호를 쓰고 싶으면 `\begin{xlist}` 대신에 다음 표에 나오는 환경 중 하나를 선택해서 예를 들어 `\begin{xlisti}` 등과 같이 쓰면 된다.

환경	기본적인 번호의 형태
<code>\begin{exe} ... \end{exe}</code>	(1) arabic (Toplevel)
<code>\begin{xlist} ... \end{xlist}</code>	depends on level of embedding
<code>\begin{xlista} ... \end{xlista}</code>	a. alphabetical
<code>\begin{xlisti} ... \end{xlisti}</code>	ii. roman
<code>\begin{xlistn} ... \end{xlistn}</code>	3. arabic
<code>\begin{xlistI} ... \end{xlistI}</code>	IV. Roman
<code>\begin{xlistA} ... \end{xlistA}</code>	E. Alphabetical

- **[cross-reference]** 구체적인 예문 번호를 본문에서 다시 인용해야 할 경우가 있는데 그 경우는 `\label{ex1}`과 `(\ref{ex1})` 같은 쌍을 이용한다. `\label{ex1}`은 번호를 받을 곳을 지정하는 역할을 하고, `(\ref{ex1})`는 실제 문서에서 부여된 번호를 가져오는 역할을 한다.

본문에서는 `(\ref{ex1})`와 같이 부여한 번호를 가져와 \TeX 가 그 부여된 번호를 본문에 자동으로 기입해 준다. 주의할 사항은 `\label{ex1}` 등에서 { }안에 나오는 식별기호는 각 예문 번호를 구분하기 위해서 들어간 것이므로 예문마다 다르게 할 당해야 한다는 것이다. 위에서는 `\label{ex1}`, `\label{ex2}`, `\label{ex3}`으로 구분하였다.

본문에 `(\ref{ex3})`이라고 치면 (4)처럼 상위예문과 하위 예문이 모두 나온 예문의 번호를 가져 와서 \TeX 가 자동으로 번호를 쓰고, `(\ref{ex1})`이라고 치면 (2)처럼 처음 예문의 번호를 가져 온다. 번호가 제대로 할당되게 하려면 컴파일을 두번

이상해야 한다. 방금 쓴 글을 코드와 함께 제시하면 다음과 같다.

본문에 `\verb+(\ref{ex3})+`이라고 치면 `(\ref{ex3})`처럼 상위예문과 하위 예문이 모두 나온 예문의 번호를 가져 와서 `\TeX`가 자동으로 번호를 쓰고, `\verb+(\ref{ex1})+`이라고 치면 `(\ref{ex1})`처럼 처음 예문의 번호를 가져 온다.

본문에 `(\ref{ex3})`이라고 치면 (4)처럼 상위예문과 하위 예문이 모두 나온 예문의 번호를 가져 와서 `\TeX`가 자동으로 번호를 쓰고, `(\ref{ex1})`이라고 치면 (2)처럼 처음 예문의 번호를 가져 온다.

- **[original number 가져오기]** 고전이라고 하는 문헌의 예문을 예문 번호 그대로 가져 와야 하는 경우도 있다. 이런 경우를 위해서는 `\exi{}`라는 명령을 쓰고 `{ }` 안에 그 원본의 번호를 기입하면 된다. 예를 들어, Chomsky (1981:264, ex.29f) 나오는 (29)과 (30)번의 예문을 원본의 예문 번호와 함께 쓰고 싶으면 다음과 같이 한다. 이 경우 `\TeX`는 자동으로 번호를 바꾸지 않는다.

```
\ldots let us now contemplate Chomsky (1981:264, ex.29f):
\begin{exe}
\exi{(29)} \al\lb{INFL} AGR]\lb{VP} V\ldots]
\exi{(30)}
\begin{xlist}[(ii)]
\exi{(i)} \be\lb{VP} V-AGR\ldots]
\exi{(ii)} \be\lb{VP}\lb{VP} V-AGR\ldots] NP]
\end{xlist}
\end{exe}
```

...let us now contemplate Chomsky (1981:264, ex.29f):

(29) $\alpha_{[\text{INFL AGR}]_{[\text{VP V} \dots]}}$

(30) (i) $\beta_{[\text{VP V-AGR} \dots]}$

(ii) $\beta_{[\text{VP} [\text{VP V-AGR} \dots] \text{NP}]}$

- **[prime number 적기]** 언어학 논문을 쓰다 보면 가끔 이미 쓴 번호에 프라임(')만을 붙여 써야 하는 경우도 있다. `gb4e.sty`는 이런 경우를 위해 단순한 `\ex` 대신 `\exp{}` 라는 명령을 제공하고 있다. 사용법은 `\exp{<label>}`의 `{ }` 안에 프라임을 제외한 번호를 공유하는 예문의 `\label{}`안에 나온 식별기호를 쓰면 된다. 다음은 사용 코드와 조판 후의 결과이다.

```
\ldots which contrasts sharply with
\begin{exe}
\exp{here}[??]{Here is two.}
\end{exe}
```

... which contrasts sharply with

(3') ?? Here is two

- 의도적으로 **number없이 예문적기** 언어학 논문을 쓰다 보면 가끔 예문번호를 의도적으로 적지 않아야 하는 경우도 있다. `gb4e.sty`는 이런 경우를 위해 단순한 `\ex` 대신 `\sn` 이라는 명령을 제공하고 있다. 다음은 사용 코드와 조판 후의 결과이다.

```
\ldots here we have an example
\begin{exe}
\sn[\\%]{Colourless green ideas sleep furiously.}
\end{exe}
of a sentence too famous to be numbered\ldots
```

... here we have an example

% Colourless green ideas sleep furiously.

of a sentence too famous to be numbered...

3.2 예문 주석과 번역문의 정렬

- 예문 주석을 위해서는 `\ex` 다음, 예문이 시작되기 전 `\gll`이 선언되어야 하고 정렬이 되어야 하는 각 문장이 끝나는 지점은

\\로 끝나는 지점이 명확히 지정되어 있어야 한다. 정렬은 빈 칸이 나오는 지점을 기준으로 T_EX가 알아서 자동으로 정렬해 준다. 다음은 코드의 예이고, 그 코드를 컴파일하면 다음과 같은 조판결과를 얻는다.

```
\begin{exe}\ex
  \begin{xlist}
    \ex\gll  철수는 영희를 사랑한다.\\
      Chelswu-{\sc top} Yenghuy-{\sc acc}
      love-{\sc pres}-{\sc decl}\\
    \ex\gll  영희는 철수를 사랑하지 않는다.\\
      Yenghuy-{\sc top} Chelswu-{\sc acc} love-{\sc suf}
      {do not}-{\sc pres}-{\sc decl}\\
  \end{xlist}
\end{exe}
```

- (5) a. 철수는 영희를 사랑한다.
Chelswu-TOP Yenghuy-ACC love-PRES-DECL
- b. 영희는 철수를 사랑하지
Yenghuy-TOP Chelswu-ACC love-SUF
않는다.
do not-PRES-DECL

- 혹시 세 줄에 걸친 주석을 달아야 할 경우라면 \gll 대신 l을 하나 덧붙여 \glll이라는 명령을 쓰면 된다. 사용법은 기본적으로 \gll과 동일하다.

```

\begin{exe}\ex
  \begin{xlist}
    \ex\glll 철수는 영희를 사랑한다.\
      Chelswu-{\nun} Yenghuy-{\lul}
      love-{\n}-{\ta}\
      Chelswu-{\sc top} Yenghuy-{\sc acc}
      love-{\sc pres}-{\sc decl}\
    \ex\glll 영희는 철수를 사랑하지 않는다.\
      Yenghuy-{\nun} Chelswu-{\lul} love-{\ci}
      {do not}-{\n}-{\ta}\
      Yenghuy-{\sc top} Chelswu-{\sc acc}
      love-{\sc suf} {do not}-{\sc pres}-{\sc decl}\
  \end{xlist}
\end{exe}

```

- (6) a. 철수는 영희를 사랑한다.
 Chelswu-nun Yenghuy-lul love-n-ta
 Chelswu-TOP Yenghuy-ACC love-PRES-DECL
- b. 영희는 철수를 사랑하지
 Yenghuy-nun Chelswu-lul love-ci
 Yenghuy-TOP Chelswu-ACC love-SUF
 않는다.
 do not-n-ta
 do not-PRES-DECL

- 예문에 번역문을 달아야 하는 경우 $\backslash\text{glt}$ 라는 명령을 번역문의 앞에 쓰면 된다. $\backslash\text{glt}$ 는 $\backslash\backslash$ 를 필요로 하지 않는다.

```

\begin{exe}\ex
\begin{xlist}
\ex\glll 철수는 영희를 사랑한다.\
  Chelswu-{\nun} Yenghuy-{\lul} love-{\n}-{\ta}\
  Chelswu-{\sc top} Yenghuy-{\sc acc}
  love-{\sc pres}-{\sc decl}\
  \glt ‘Chelswu loves Yenghuy.’
\ex\glll 영희는 철수를 사랑하지 않는다.\
  Yenghuy-{\nun} Chelswu-{\lul} love-{\ci}
  {do not}-{\n}-{\ta}\
  Yenghuy-{\sc top} Chelswu-{\sc acc} love-{\sc suf}
  {do not}-{\sc pres}-{\sc decl}\
  \glt ‘Yenghuy does not love Chelswu.’
\end{xlist}
\end{exe}

```

- (7) a. 철수는 영희를 사랑한다.
 Chelswu-nun Yenghuy-lul love-n-ta
 Chelswu-TOP Yenghuy-ACC love-PRES-DECL
 ‘Chelswu loves Yenghuy.’
- b. 영희는 철수를 사랑하지
 Yenghuy-nun Chelswu-lul love-ci
 Yenghuy-TOP Chelswu-ACC love-SUF
 않는다.
 do not-n-ta
 do not-PRES-DECL
 ‘Yenghuy does not love Chelswu.’

- 예문이 이탤릭체로 되어 있는 있는 것을 싫어 하는 사용자나 그런 예문을 허용하지 않는 저널에 논문을 투고할 때에는 초기화 영역 (Preamble) 혹은 `\begin{exe}` 앞에 다음과 같은 명령을 추가하면 된다. 이를 다시 되돌리고 싶으면 이후에 필요한 곳에 `\let\eachwordone=\it`라고 적으면 예문이 다시 이탤릭체로 조판된다. 다음은 이탤릭체를 해제한 명령의 예이고 이어지는 예문은 그 명령에 따른 조판결과이다.

```

\let\eachwordone=\rm
\begin{exe}\ex
\begin{xlist}
\ex\glll 철수는 영희를 사랑한다.\
  Chelswu-\{nun\} Yenghuy-\{lul\} love-\{n\}-\{ta\}\
  Chelswu-\{\sc top\} Yenghuy-\{\sc acc\}
  love-\{\sc pres\}-\{\sc decl\}\
  \glt ‘Chelswu loves Yenghuy.’
\ex\glll 영희는 철수를 사랑하지 않는다.\
  Yenghuy-\{nun\} Chelswu-\{lul\} love-\{ci\}
  \{do not\}-\{n\}-\{ta\}\
  Yenghuy-\{\sc top\} Chelswu-\{\sc acc\} love-\{\sc suf\}
  \{do not\}-\{\sc pres\}-\{\sc decl\}\
  \glt ‘Yenghuy does not love Chelswu.’
\end{xlist}
\end{exe}

```

- (8) a. 철수는 영희를 사랑한다.
 Chelswu-nun Yenghuy-lul love-n-ta
 Chelswu-TOP Yenghuy-ACC love-PRES-DECL
 ‘Chelswu loves Yenghuy.’
- b. 영희는 철수를 사랑하지
 Yenghuy-nun Chelswu-lul love-ci
 Yenghuy-TOP Chelswu-ACC love-SUF
 않는다.
 do not-n-ta
 do not-PRES-DECL
 ‘Yenghuy does not love Chelswu.’

그러나 다음 예문처럼 여러 행에 걸친 예문이 주석과 함께 나올 경우에는 글꼴을 달리하는 것이 시각적으로 읽기에 쉽다는 점을 이탤릭체로 된 예문을 해제할 경우 염두에 둘 필요가 있다. 다음의 두 예문을 비교해 보기 바란다.

- (9) *Wenn jemand in die Wüste zieht und lebt dort*
 If someone in the desert draws and lives there

von Heuschrecken oder sich im Wald verirrt hat
 from grasshoppers or self in the woods errored has
und nährt sich von Wurzeln und Beeren, ...
 and nourishes self from roots and berries ...
 ‘if one retreats to the desert and lives there from grasshoppers or gets lost in the woods and lives off of roots and berries’

- (10) Wenn jemand in die Wüste zieht und lebt dort
 If someone in the desert draws and lives there
von Heuschrecken oder sich im Wald verirrt hat
 from grasshoppers or self in the woods errored has
und nährt sich von Wurzeln und Beeren, ...
 and nourishes self from roots and berries ...
 ‘if one retreats to the desert and lives there from grasshoppers or gets lost in the woods and lives off of roots and berries’

- [원문과 주석문의 빈 칸이 일치하지 않는 경우의 주석문 정렬하기] 원문은 한 단어인데 주석문은 두 단어 이상으로 주석을 달아야 하는 경우나 원문에는 한 단어로 나와 있으나 주석문에는 특별히 대응되는 요소가 없는 경우는 자칫 전체 주석문의 정렬을 한 두 칸 씩 뒤로 밀리게 할 수 있다. 이런 경우는 글을 쓰는 사람이 알아서 { }을 써서 그 안에 나오는 단어들끼리 한 단어와 같이 취급되도록 하거나, 빈 { }을 주석문에 삽입해서 인쇄는 안 되어도 \TeX 가 그곳에 무엇이 있어 한 단어인 양 알도록 속임수(??)를 써야 한다. 다음은 네덜란드어를 영어로 주석문을 단 경우의 예이다.

```
\begin{exe}\ex
  \gll Dit is een voorbeeldje in het Nederlands.\
      This is a {little example} in {} Dutch.\
  \glt ‘This is a little example in Dutch.’
\end{exe}
```

- (11) Dit is een voorbeeldje in het Nederlands.
 This is a little example in Dutch.

‘This is a little example in Dutch.’

3.3 예문에 문법성 표시하기

- 예문에 문법성에 관한 표시를 할 때는 `\ex` 다음에 바로 이어서 `[]`를 쓰고 그 안에 `[*]` 혹은 `[??]` 라고 기입하면 된다. 다음은 이 꾸러미의 저자들이 제시하는 예이다.

```
\begin{exe}
\ex I invited Fred for dinner.
\ex[?]{I invited for dinner Fred.}
\ex[*]{I for invited Fred dinner.}
\ex \gll Den Fritz$_1$ habe ich $e_1$ zum Essen eingeladen.\\
      the Fred have I {} {to the} eating invited.\\
\glt I invited Fred for dinner.
\end{exe}
```

- (12) I invited Fred for dinner.
- (13) ? I invited for dinner Fred.
- (14) * I for invited Fred dinner.
- (15) *Den Fritz₁ habe ich e₁ zum Essen eingeladen.*
 the Fred have I to the eating invited.
 ‘I invited Fred for dinner.’

3.4 Varia: bar기입하기, labelled bracketing 등

`gb4e.sty`은 언어학 논문 작성에 유용한 몇 가지의 기능을 더 가지고 있는데 이 절에서 그들을 소개하기로 한다.

- [X-bar 스타일의 bar기입하기]: `gb4e.sty`은 `\obar{X}`, `\ibar{X}`, `\iibar{X}`, `\mbar{X}` 명령어를 제공한다. 이들 명령은 조판되었을 때 디폴트로 X^0 \bar{X} $\bar{\bar{X}}$ X^{\max} 와 같이 나타난다. 만약 초기화 영역에 `\primebars`라는 명령을 주면, 동일한 명령이 X^0 X' XP X^{\max} 처럼 조판되어 나온다.

명령어	디폴트 형태	\primebars 형태
<code>\obar{X}</code>	X^0	X^0
<code>\ibar{X}</code>	\bar{X}	X'
<code>\iibar{X}</code>	$\overline{\bar{X}}$	XP
<code>\mbar{X}</code>	X^{\max}	X^{\max}

- [Labelled bracketing을 다는 방법]: `gb4e.sty`은 `\lb{X}`와 `\rb{X}`명령어를 왼쪽과 오른쪽의 labelled bracketing을 위해 제공한다. 사용 코드와 조판 결과는 다음과 같다.

```
\lb{NP}[1]\lb{D}[1] the\lb{\ibar{N}} example]
```

```
[NP1[D1 the][ $\bar{N}$  example]]
```

- [그리스 문자 지원]: 그리스 문자는 수학모드에서 지원하는데, `gb4e.sty`은 일반 텍스트 모드에서도 그리스 문자를 사용할 수 있도록 해준다.

<code>\al</code>	α
<code>\be</code>	β
<code>\ga</code>	γ
<code>\de</code>	δ
<code>\th</code>	θ

- `gb4e.sty`은 또 윗첨자와 아래 첨자가 수학모드에서만 자동하는 것을 보완해 일반 텍스트 모드에서도 `_`와 `^`가 작동하도록 환경을 개선하였다.

3.5 예문에 이동한 선을 그리기 및 기타 사항

`gb4e.sty`은 Lex Holt가 개발한 `\atop-`, `\atcenter` 매크로를 사용할 수 있게 해 주고, J. Frampton이 개발한 이동 경로의 예시를 위한 화살표를 잘 지원한다. 다음은 이동 화살표의 코드와 조판 결과를 제시하고 있다. 더 자세한 것은 해당 꾸러미의 사용법을 참고하기 바란다.

```

\begin{exe}
\ex \atcenter{
\arrowalign{\fillright\pd&\link3&\fillleft{\vrule\hfil}\cr
Did&\ &John&\ &t_{\mathrm{Agr}}&\ &appear &t&\ to be likely
&t&\ to win.\cr
&&\fillright\pu&\link3&\centr{\vrule\spacer\pu}&\lf
&\fillleft\vrule\cr}
}
\end{exe}

```

(16) Did John t_{Agr} appear t to be likely t to win.

제 4 절 꾸러미 사용하기 (2): covington.sty

4.1 기본적인 예문 번호 매기기

- covington.sty을 이용해서 예문의 번호를 일괄적으로 매기려면 `\begin{example}`로 시작하고 `\end{example}`로 끝나는 환경에 그냥 예문을 적기만 하면 된다. 다음은 예문 만드는 코드와 조판 결과를 예시한 것이다.

```

\begin{example}
This is a sentence.
\end{example}

```

(4.2) This is a sentence.

- covington.sty을 이용해서 복잡한 예문의 번호를 일괄적으로 매기려면 `\begin{example}`로 시작하고 `\end{example}`로 끝나는 환경에 `\begin{itemize}`로 시작하고 `\end{itemize}`로 끝나는 환경에 `\item[(a)]`와 같이 예문의 하위 번호를 기입하고

예문을 쓰면 된다. 다음은 복잡한 예문에 번호를 부여하는 코드와 조판 결과를 예시한 것이다.

```
\begin{example}
\begin{itemize}
\item[(a)] This is the first sentence.
\item[(b)] This is the second sentence.
\end{itemize}
\end{example}
```

(4.3) (a) This is the first sentence.

(b) This is the second sentence.

- 예문이 연속으로 나올 경우 `\begin{example}`를 매번 반복하지 않고

`\begin{examples}`라고 한 번만 명령을 주고 예문들은 `\item`을 쓰고 계속해서 그 다음에 기입하면 된다. 다음에 이런 경우의 코드와 조판 결과를 예시한다.

```
\begin{examples}
\item This is the first sentence.
\item This is the second sentence.
\end{examples}
```

(4.4) This is the first sentence.

(4.5) This is the second sentence.

- [오른쪽에 번호 나오게 하기]: `covington.sty`을 사용하면 예문의 오른쪽에 예문의 번호가 부여되게 할 수 있다. 이는 `covington.sty`에 들어 있는 `\exampleno`라는 명령을 이용한 것인데, 예문 번호가 나올 자리를 지정하고 그 자리에 `\exampleno`를 쓰면 된다. 다음 예는 맨 오른쪽에 예문의 번호가 나오도록 한 코드이고 그 조판 결과는 그 아래에서 볼 수 있는 바와 같다.

```
\begin{flushleft}
This is a sentence. \hfill (\exampleno)\label{ex4}
\end{flushleft}
```

Here's what you get:

This is a sentence. (4.6)

- **[cross-reference]**: `covington.sty`에서도 cross-reference를 잘 지원한다. 사용법은 `gb4e.sty`에서의 사용법과 같다. 여기에 그 예를 반복해서 적어둔다.

본문에 `\verb+(\ref{ex4})+`라고 치면 `(\ref{ex4})`처럼 바로 위의 예문의 번호를 가져 와서 \TeX 가 자동으로 번호를 쓴다.

본문에 `(\ref{ex4})`라고 치면 (4.6)처럼 바로 위의 예문의 번호를 가져 와서 \TeX 가 자동으로 번호를 쓴다.

4.2 주석문 정렬하기

- 이미 설명한 바와 같이 `covington.sty`에서 사용하는 주석 정렬 방법도 결국은 Marcel R. van der Groot의 `gloss.tex`를 사용하고 있어서 `gb4e.sty`에서 설명한 방법과 기본적으로는 동일하다. 그러나 알 수 없는 이유로 `covington.sty`에서는 주석문이 다 끝나면 예문마다 `\glend`라는 명령을 한 번 더 주어 `flushleft` 환경을 종료하게 되어 있다. 다음은 그 예이다.

```
\begin{example}
\gll Dit is een Nederlands voorbeeld. \\
      This is a Dutch example. \\
\glt 'This is an example in Dutch.'
\glend
\end{example}
```

- (4.7) *Dit is een Nederlands voorbeeld.*
 This is a Dutch example.
 ‘This is an example in Dutch.’

```
\glll Hoc est aliud exemplum.\
      n.sg.nom 3.sg n.sg.nom n.sg.nom\
      This is another example.\
\glt  ‘This is another example.’
\glend
```

Hoc est aliud exemplum.
 n.sg.nom 3.sg n.sg.nom n.sg.nom
 This is another example.

‘This is another example.’

4.3 Phrase structure rules 및 feature structures 그리기

covington.sty에서는 Phrase structure rules 및 feature structures를 그리는 명령을 내장하고 있다. \psr, \fs, \lfs 등의 명령어를 사용하면 된다.

- **psr 명령:** 우선 구구조규칙은 단순히

```
\psr{S}{NP~VP}
```

라고만 치면 다음과 같은 모양의 구구조 규칙이 생성된다.

$$S \rightarrow NP VP$$

- **fs 명령:** covington.sty에서는 또 feature structures를 쉽게 그릴 수 있는 매크로를 내장하고 있는데 그 사용법에 대한 예시는 다음과 같다.

```
\fs{\rm CASE: \it nom \
      \rm PERSON: \it 3rd}
```

라고 치면

$$\left[\begin{array}{l} \text{CASE : } \textit{nom} \\ \text{PERSON : } \textit{3rd} \end{array} \right]$$

가 생성된다.

- **lfs 명령:** feature structure 위에 범주를 기입해야 하는 경우 `covington.sty`이 쉬운 해결책을 제시해 준다. 다음과 같이 하면 된다.

```
\lfs{\rm N}{\rm CASE: \it nom \\  
          \rm PERSON: \it 3rd}
```

N

$$\left[\begin{array}{l} \text{CASE : } \textit{nom} \\ \text{PERSON : } \textit{3rd} \end{array} \right]$$

- **lfs와 psr 명령:** feature structure로 된 구구조 규칙을 만드는 데에도 `covington.sty`은 커다란 도움을 준다. 단순히 다음과 같이 하면 된다.

```
\psr{\lfs{\rm S}{\rm TENSE: \it pres}}  
  {\lfs{\rm NP}{\rm CASE: \it nom \\  
            \rm NUMVER: \it sing}}  
  {\lfs{\rm VP}{\rm TENSE: \it pres \\  
            \rm NUMBER: \it sing} }
```

$$\begin{array}{ccc} \text{S} & \rightarrow & \text{NP} \qquad \text{VP} \\ \left[\text{TENSE : } \textit{pres} \right] & & \left[\text{CASE : } \textit{nom} \right] \quad \left[\text{TENSE : } \textit{pres} \right] \\ & & \left[\text{NUMBER : } \textit{sing} \right] \quad \left[\text{NUMBER : } \textit{sing} \right] \end{array}$$

4.4 Accents

- \LaTeX 는 국제 특수 문자를 잘 지원한다. 일반적인 포맷으로 다음과 같은 명령을 쓰면 된다. 그러나 이중으로 앤센트가 들

어 간 문자는 L^AT_EX에서도 쉽게 구현할 수가 없었다 그러나 `covington.sty`은 이런 이중 액센트가 나오는 특수 문자를 지원한다.

- 다음과 같이 입력하면 L^AT_EX는 특수 문자를 조판해 준다.

```
\‘{x} \’{x} \^{}{x} \”{x} \~{}{x} \={x}
\H{x} \t{xx} \c{x} \d{x} \b{x}
```

̀́ ˆ ˘ ˙ ˚ ˛ ˜ ˝ ˘x ˙x ˚x

- L^AT_EX 는 외국문자도 쉽게 구현해 준다. 다음이 그 예이다.

```
\i \j \ae \AE \oe \OE \aa \AA \o \O \l \L \ss ‘ ! ‘
```

ı ĵ æ Æ œ Œ å Å ø Ø ł Ł ß ĺ ĵ

- L^AT_EX 가 일반적으로 지원하지 않던 문자 가운데 `covington.sty`이 지원하는 문자들이 많이 있다. 그 명령어들을 소개한다.

두 가지 액센트를 동시에 쓰기 위한 명령으로

`\twoacc[...|...]` 가 있다. 예를 들면 `\twoacc[\~|\={a}]`
= $\tilde{\acute{a}}$ 와 같다.

`\acm{...}` for acute over macron, e.g., `\acm{a}` = $\acute{\bar{a}}$

`\grm{...}` for grave over macron, e.g., `\grm{a}` = $\grave{\bar{a}}$

`\cim{...}` for circumflex over macron, e.g., `\cim{a}` = $\hat{\bar{a}}$

제 5 장

Attribute-Value- Matrices (AVM) 그리기

HPSG나 LFG의 Attribute-Value-Matrices (AVM)를 일반적인 워드프로세서에서 그리기란 여간 고통스러운 일이 아닐 뿐더러 많은 시간을 투자해야 하는 일임을 언어학을 전공하는 사람이라면, 특히 HPSG나 LFG를 문법이론으로 애용하는 사람들은 잘 알고 있으리라 생각한다.

L^AT_EX가 가지고 있는 장점 중의 하나가 이러한 어려움을 한 순간에 다 해결해 준다는 점이다. 무엇보다도 새로운 L^AT_EX기법을 배우는 가장 좋은 방법은 그냥 예제를 따라서 하면 된다는 저자의 경험에 비추어 이 장에서는 가능하면 대표적인 좋은 예들을 제시하면서 설명하려고 한다.

제 1 절 꾸러미 소개: avm.sty

- AVM을 그리는데 사용되는 가장 일반적인 꾸러미는 Stanford


```

\documentclass[12pt]{article}
\usepackage{hangul}
\usepackage{avm}
\begin{document}

...

\end{document}

```

- 언제나 AVM은 `\begin{avm}`으로 시작해서 `\end{avm}`으로 끝나는 환경 사이에 기입해야 한다. 이미 다 아시겠지만 `\begin{avm}`과 `\begin{avm}`은 본문 내용의 일부이기 때문에 `\begin{document}`와 `\end{document}` 사이에 와야 한다. 예를 들면, 다음과 같은 문서의 구조가 AVM이 나오는 문서 구조가 된다.

```

\documentclass[12pt]{article}
\usepackage{hangul}
\usepackage{avm}
\begin{document}

\begin{avm}
\[
  PHON & /사랑하/ \\
  SYNSEM$| $LOC$| $CAT$| $HEAD & \@1 {\em verb} \\
  DTRS$| $HEAD-DTR$| $SYNSEM$| $LOC$| $CAT$| $HEAD & \@1
\]
\end{avm}
\end{document}

```

- 위에서 `\begin{avm}`부터 `\begin{avm}`까지의 입력명령으로 우리는 다음과 같은 조판결과를 얻게 된다.

PHON	/사랑하/
SYNSEM LOC CAT HEAD	□ <i>verb</i>
DTRS HEAD-DTR SYNSEM LOC CAT HEAD	□

- 다음은 `avm.sty`를 사용하는 기본적인 명령어들이다.

<code>\[</code>	AVM의 왼쪽 대괄호를 그리는 명령; 항상 왼쪽 대괄호와 오른쪽 대괄호의 숫자가 맞아야 한다.
<code>\]</code>	AVM의 오른쪽 대괄호를 그리는 명령; 항상 오른쪽 대괄호와 왼쪽 대괄호의 숫자가 맞아야 한다.
<code>&</code>	자질과 지질 값으로 구성된 열들의 정렬을 위한 명령
<code>\\</code>	하나의 AVM 행이 끝나는 지점을 표시
<code>\$\$</code>	자질과 자질이 연속적으로 나오는 경로(PATH)의 일괄적 표시
<code>\@{}</code>	BOXED INTEGER를 위한 표시; { } 사이에 박스 안에 나올 문자를 기입하나, 단순 숫자는 <code>\@{1}</code> 대신 단순히 <code>\@1</code> 로 기입해도 됨
<code>\{</code>	AVM의 왼쪽 중괄호를 그리는 명령; 항상 왼쪽 중괄호와 오른쪽 중괄호의 숫자가 맞아야 한다.
<code>\}</code>	AVM의 오른쪽 중괄호를 그리는 명령; 항상 오른쪽 중괄호와 왼쪽 중괄호의 숫자가 맞아야 한다.
<code>\<</code>	AVM의 왼쪽 중각괄호를 그리는 명령; 항상 왼쪽 각괄호와 오른쪽 각괄호의 숫자가 맞아야 한다.
<code>\></code>	AVM의 오른쪽 각괄호를 그리는 명령; 항상 오른쪽 각괄호와 왼쪽 각괄호의 숫자가 맞아야 한다.

3.2 Sag & Wasow (1999)식으로 유형이 처리된 AVM 만들기

- 우선 다음과 같은 명령을 `.tex` 파일에 입력하고 컴파일시키면 다음과 같은 그 아래에 있는 예쁜 AVM을 얻는다.

```

\begin{avm}
\[[ \em feat-struct]\]
  SYN \[[ \em gramm-cat]\]
    HEAD & \em pos]\]
    COMPS & list(\em synsem-struct)\]
    SPR & list(\em synsem-struct)\]
    GAP & list(\em synsem-struct)
  \]\]
  SEM \[[ \em sem-struct]\]
    MODE & \{prop, ques, dir, ref, none}\]\]
    INDEX & \{\}\em index, none}\]\]
    RESTR & list(\em predication)
  \]\]
\]
\end{avm}

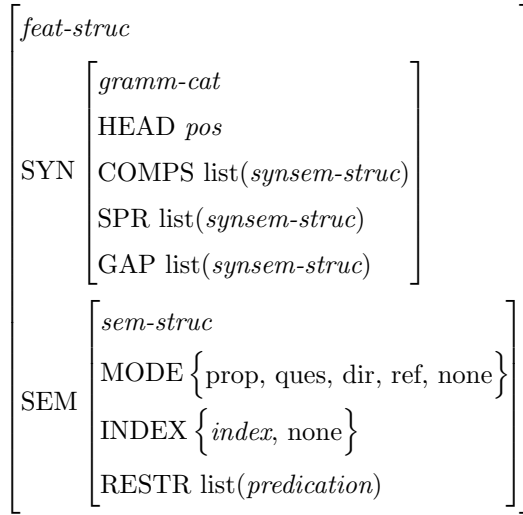
```

<i>feat-struct</i>	[<table style="border: none;"> <tr> <td style="border: none;"><i>gramm-cat</i></td> <td style="border: none;">]</td> </tr> <tr> <td style="border: none;">HEAD</td> <td style="border: none;"><i>pos</i></td> </tr> <tr> <td style="border: none;">SYN</td> <td style="border: none;">COMPS</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;">list(<i>synsem-struct</i>)</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;">SPR</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;">list(<i>synsem-struct</i>)</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;">GAP</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;">list(<i>synsem-struct</i>)</td> </tr> <tr> <td style="border: none;">]</td> <td style="border: none;">]</td> </tr> <tr> <td style="border: none;"><i>sem-struct</i></td> <td style="border: none;">[</td> </tr> <tr> <td style="border: none;">SEM</td> <td style="border: none;">MODE</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;">{prop, ques, dir, ref, none}</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;">INDEX</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;">{<i>index</i>, none}</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;">RESTR</td> </tr> <tr> <td style="border: none;"></td> <td style="border: none;">list(<i>predication</i>)</td> </tr> <tr> <td style="border: none;">]</td> <td style="border: none;">]</td> </tr> </table>	<i>gramm-cat</i>]	HEAD	<i>pos</i>	SYN	COMPS		list(<i>synsem-struct</i>)		SPR		list(<i>synsem-struct</i>)		GAP		list(<i>synsem-struct</i>)]]	<i>sem-struct</i>	[SEM	MODE		{prop, ques, dir, ref, none}		INDEX		{ <i>index</i> , none}		RESTR		list(<i>predication</i>)]]
<i>gramm-cat</i>]																																			
HEAD	<i>pos</i>																																			
SYN	COMPS																																			
	list(<i>synsem-struct</i>)																																			
	SPR																																			
	list(<i>synsem-struct</i>)																																			
	GAP																																			
	list(<i>synsem-struct</i>)																																			
]]																																			
<i>sem-struct</i>	[
SEM	MODE																																			
	{prop, ques, dir, ref, none}																																			
	INDEX																																			
	{ <i>index</i> , none}																																			
	RESTR																																			
	list(<i>predication</i>)																																			
]]																																			

- 이때 AVM 시작과 끝을 표시하는 \[와 \]표시는 프로그래밍을 할 때 의례 그렇듯이 예에서 볼 수 있는 것처럼 줄을 맞춰 나중에 알아 보기 쉽게 하는 것이 좋다.
- HEAD & \em pos} 등의 표기에서 중간에 들어 간 & 표시는 나중에 조판되었을 때 열들의 간격을 가지런히 맞추기 위한 목적

으로 들어가 있다. 조판 결과의 열들이 정렬될 필요가 없으면 자질과 자질값 사이에 &를 넣지 않는다.

- 만약 HEAD `{\em pos}` 처럼 중간에 들어 간 &를 빼면 그 출력 결과는 다음과 같다. 위의 그림과 비교하면 이해가 쉬울 것이다. 어느 것을 선택하는가는 사용자의 몫이다.



3.3 Pollard & Sag (1994)식으로 유형이 처리된 AVM 만들기

- 만약 Sag and Wasow (1999)에서와는 달리, 그러나 Pollard and Sag (1994)에서 처럼 `sort`를 AVM의 좌측 하단에 위치시키고 싶으면

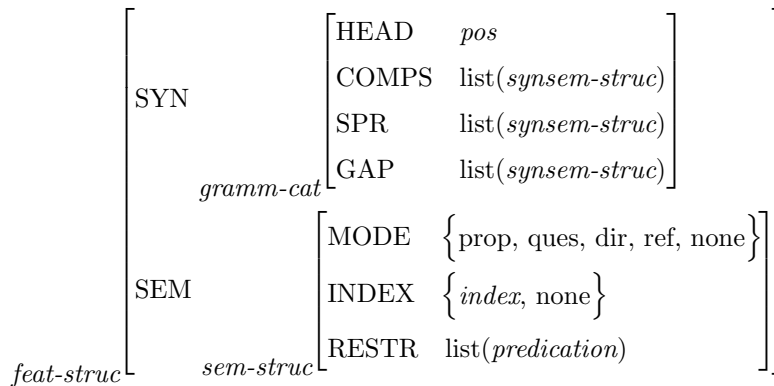
```
\sort{\em sort}{\[ ... \]}
```

와 같이 기입하면 된다. 맨 처음의 코드와 아래에 있는 코드를 비교하고 그 결과를 비교해 보면 이 차이를 잘 알 수 있다.

```

\begin{avm}
\sort{\em feat-struct}{\[ SYN
      \sort{\em gramm-cat}{\[
      HEAD & {\em pos}\
      COMPS & list({\em synsem-struct})\
      SPR & list({\em synsem-struct})\
      GAP & list({\em synsem-struct})
            \]}\
      SEM
      \sort{\em sem-struct}{\[
      MODE & {\prop, ques, dir, ref, none}\
      INDEX & {\em index}, none\
      RESTR & list({\em predication})
            \]}
    \]}
\end{avm}

```



- 위에서 Sort는 항상 Feature가 끝나는 곳에서 시작하고 있다는 것을 알 수 있다. 이런 방식은 일반적으로 장소를 많이 차지하는 단점이 있다. 따라서 위에서 처럼 Sag & Wasow (1999) 방식이 아닌 Pollard & Sag (1994) 방식을 고수하면서 장소를 아끼고 싶으면, Feature와 그 다음에 나오는 Sort를 겹치도록 명령을 넣으면 된다. 이때 사용하는 명령어가

```
\osort{\em sort}{\[ ... \]}
```

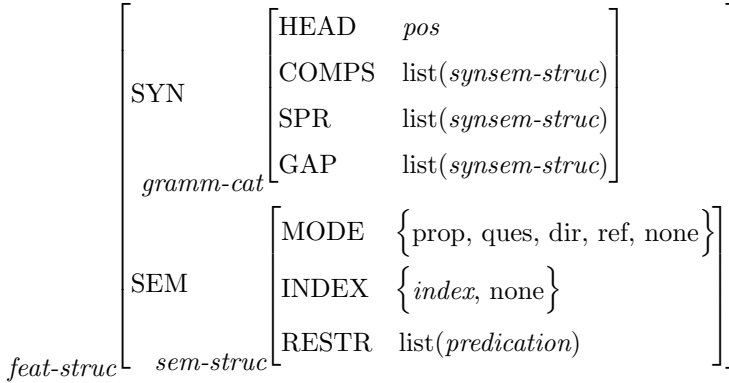
이다. 참고로 위에서는 단순히

```
\sort{\em sort}{\[ ... \]}
```

와 같이 기입했다는 점을 기억하면 그 차이를 쉽게 알 수 있을 것이다.

다음에 그 코드와 결과를 연이어 제시한다. 이 경우 Sort가 바로 앞의 Feature보다도 길 때에는 시각상으로 좋지 못한 결과를 얻는 수도 있으니 조심할 필요가 있다.

```
\begin{avm}
\osort{\em feat-struct}{\[ SYN
    \osort{\em gramm-cat}{\[
    HEAD & {\em pos}\
    COMPS & list({\em synsem-struct})\
    SPR & list({\em synsem-struct})\
    GAP & list({\em synsem-struct})
    \]}\
    SEM
    \osort{\em sem-struct}{\[
    MODE & \{prop, ques, dir, ref, none\}\
    INDEX & \{\}\{\em index}, none\}\
    RESTR & list({\em predication})
    \]}\
\]}
\end{avm}
```

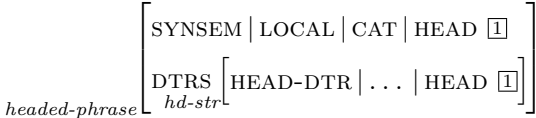
3.4 옵션을 이용해서 AVM을 더 간단하게 만들기

- AVM 그리는 방법 중에서 가장 추천할 만한 것은 L^AT_EX의 장점이기도 한 매크로를 이용해서 AVM을 더 쉽게 그리는 것이다. `avm.sty`에는 선택할 수 있는 옵션이 많은데 그 중에서 `\begin{document}` 앞의 Preamble에 혹은 `\begin{avm}` 바로 앞에 AVM의 유형, 자질값의 글꼴, Sort의 유무에 관한 옵션을 먼저 주고 사용하면 편리하다. 다음은 옵션을 부여한 예이다. 이렇게 옵션을 주고 다음과 같은 코드를 기입하고 컴파일하면 그 아래에 있는 것과 같은 예쁜 AVM이 생긴다.

```

\avmfont{\sc}
\avmoptions{sorted,active}
\avmvalfont{\rm}
\avmsortfont{\scriptsize\it}
\begin{avm}
[{\headed-phrase}\
    synsem|local|cat|head @{1} \
    dtrs [hd-str
        head-dtr|\ldots|head @{1}]
\end{avm}

```



3.5 LFG에서 사용하는 기능등식과 기능기술 AVM 만들기

- LFG의 기능기술 (functional descriptions)에 사용되는 AVM은 다음과 같이 만들면 된다.

```
\avmfont{\sc}
\avmoptions{sorted,active}
\avmvalfont{\rm}
\avmsortfont{\scriptsize\it}
\begin{avm}
[{} pred & ‘사랑하< \_ , \_ >’    \\
      tense & pres                    \\
      subj  & [{} pred & ‘철수’]    \\
      obj   & [{} pred & ‘영희’]]
\end{avm}
```

$$\left[\begin{array}{l} \text{PRED} \quad \text{‘사랑하}\langle -, - \rangle\text{’} \\ \text{TENSE} \quad \text{pres} \\ \text{SUBJ} \quad \left[\text{PRED} \quad \text{‘철수’} \right] \\ \text{OBJ} \quad \left[\text{PRED} \quad \text{‘영희’} \right] \end{array} \right]$$

- LFG에서는 HPSG에서 boxed integer로 처리하는 structure sharing을 그래프를 사용하는 기능등식 (functional equation)을 사용한다. 기능등식에 사용되는 곡선을 그리기 위해서는 또 다른 꾸러미인 `tree-dvips.sty`를 사용해야 한다. `tree-dvips.sty`를 사용하면 곡선이 .dvi 파일에서는 보이지 않고 .ps파일로 만들어야 최종적인 작업이 완성된다. 다음은 그 한 예이다.

```

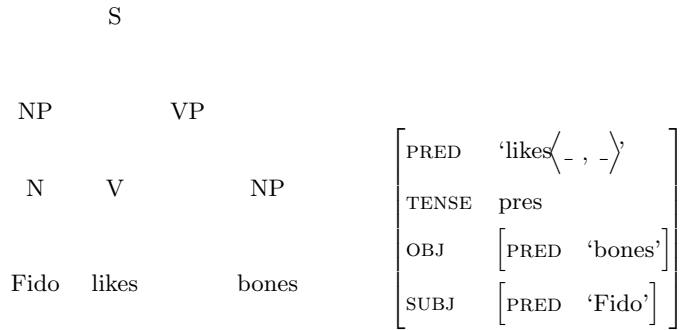
\begin{small}
\ vbox{
\ begin{tabular}{cccc}
& \ node{s}{S} \ \ [.3in]

\ node{np1}{NP} && \ node{vp}{VP} \ \ [.2in]

\ node{n}{N} & \ node{v}{V} && \ node{np2}{NP} \ \ [.3in]

\ node{fido}{Fido} & \ node{likes}{likes} && \ node{bones}{bones}
\ end{tabular}\ hspace*{.2in}
\ avmfont{\ sc}
\ avmoptions{sorted,active}
\ avmvalfont{\ rm}
\ avmsortfont{\ scriptsize\ it}
\ begin{avm}
\ node{f1}{[{} pred & 'likes< \_ , \_ >' \ \
tense & pres \ \
obj & [{} pred & 'bones'] \ \
subj & \ node{f2}{[{} pred & 'Fido']] } ]}
\ end{avm}}
\ nodeconnect[b]{s}[t]{np1}
\ nodeconnect[b]{s}[t]{vp}
\ nodeconnect[b]{np1}[t]{n}
\ nodeconnect[b]{vp}[t]{v}
\ nodeconnect[b]{vp}[t]{np2}
\ nodeconnect[b]{n}[t]{fido}
\ nodeconnect[b]{v}[t]{likes}
\ nodeconnect[b]{np2}[t]{bones}
\ anodecurve[r]{s}[l]{f1}{.5in}
\ anodecurve[r]{vp}[l]{f1}{.3in}
\ anodecurve[r]{v}[l]{f1}{.3in}
\ anodecurve[r]{np1}[b]{f2}{1in}
\ anodecurve[r]{n}[b]{f2}{1in}
\ end{small}

```



- `tree-dvips.sty` 꾸러미에 대해서는 그 꾸러미에 첨부되어 있는 사용자 설명서를 보기 바라며, 여기에서는 더 이상 자세히 설명하지 않기로 한다.

제 6 장

L^AT_EX로 수형도 그리기

일반 워드프로세서에서 선을 그리는 것은 쉬운 일이다. 그러나 일반 워드프로세서에서 출판된 책에 나오는 것과 같은 사선을 정교하게 그리는 것은 그리 쉬운 일이 아니다. 더구나 언어학 논문에 가끔 등장하는 수형도를 정교하게 그리는 것은 여간 어려운 일이 아니다. L^AT_EX는 수형도를 비교적 쉽게 그릴 수 있는 조판기능을 제공한다. L^AT_EX에서 수형도를 그리는 목적으로 고안된 꾸러미들은 매우 많다. 그러나 이 장에서는 L^AT_EX에서 수형도를 그리는 목적으로 고안된 꾸러미를 3 가지를 소개하고 그들의 사용법을 예제와 함께 기술하기로 한다. 각 꾸러미마다 장단점이 있기 때문에 어느 꾸러미를 선택해서 문서를 만드느냐 하는 문제는 사용자가 선택해야 할 문제이다.

제 1 절 수형도를 그리기 위한 꾸러미

- **parsetree.sty**: 수형도를 L^AT_EX 상에서 그리기 위한 꾸러미들 중에서 가장 직관적으로 쉽고 명료한 꾸러미이다. *.dvi 파일에서도 사선이 보인다는 점은 다른 꾸러미에는 없는 장점으로 꼽힌다. 저자는 Eirik Hektoen이고 L^AT_EX CD에 수록된 꾸러미의 버전은 1994년 02월 24일날 발표한 것이다. 본문에서 소개

하고 있는 사용법은 영국 Essex 대학교의 Doug Arnold가 작성한 사용법 문서를 참고한 것이다.

- **rtree.sty**: Stanford대학교의 Rob Malouf가 HPSG에서 수형도를 잘 그리려는 목적으로 1996년 7월 22일 발표한 꾸러미이다. 언어학자를 위한 L^AT_EX CD에 수록된 꾸러미의 버전은 1997년 06월 16일날 발표한 것이다. 이 꾸러미는 수형도를 그리는데 `ps-tricks` 매크로를 사용한다. Sag을 비롯한 Stanford대학교의 대부분의 학자들은 수형도를 그리는데 이 꾸러미를 사용한다. 그러나 이 꾸러미의 단점은 다른 꾸러미와 마찬가지로 `.ps` 화일로 만들었을 경우에만 사선이 보인다는 것이다.
- **tree-dvips.sty**: 가장 강력하고 쓰임이 다양한 언어학자에게는 `must`라고 해도 과언이 아닌 꾸러미이다. Stanford 대학교의 Emma Pease가 저자이고, 사용자 설명서가 비교적 잘 되어 있다. 이 꾸러미는 단순히 수형도를 그리기 위한 목적으로만 만들어진 것이 아니고 다른 여러 분야에도 적용될 수 있기 때문에 매우 강력하며, 이런 이유로 다른 꾸러미와 함께 사용하는 것이 일반적이다.
- 그밖에 다음과 같은 꾸러미들이 있으나 여기서는 지면상의 제약으로 자세한 소개는 하지 않기로 한다.
 - Matijs van Zuijlen이 1998년 발표한 수형도를 그리기 위한 꾸러미인 `synttree.sty`;
 - 독일 Tübingen대학교 Wolfgang Sternefeld 박사 (`Wolfgang.Sternefeld@uni-tuebingen.de`)가 만든 `linguex.sty`와 그 꾸러미와 함께 쓰는 `ps-trees.sty`가 있고;
 - 일본에서 만든 `jpsgtree.sty`;
 - Greg Lee가 만든 `tree preprocessor`, Avery Andrews의 `trees.sty`;
 - Anthony Bloesch가 만든 `treetex.sty`;
 - Peter Vanroose (`peter@dit.lth.se`)가 만든 `trees.sty`;
 - Columbia 대학교의 David Eppstein (`Eppstein@cs.Columbia.edu`)이 만든 `trees.sty`;

– Avery DeLano Andrews III가 만든 `trees.sty`.

제 2 절 `parsetree.sty`를 이용한 수형도 그리기

2.1 기본적인 사용법

- 초기화 (Preamble)부분에 다음과 같이 선언하고 설명서를 따라 작성하면 된다.

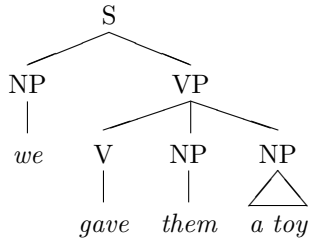
```
\usepackage{parsetree}
```

- `parsetree.sty`은 딸범주가 세 개 이하인 수형도를 그리기에 좋은 꾸러미이다. 기본적으로 수형도의 시작과 종료는 다음과 같이 괄호로 시작해서 괄호로 끝나게 되어 있고,

```
(  
)
```

수형도의 교점은 `.NP.`처럼 `.로` 둘러 쌓여 있으면 된다. 다음은 단순한 예이다.

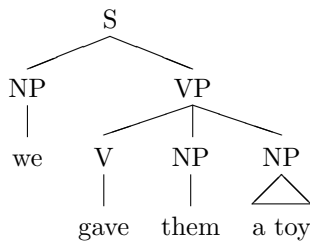
```
\begin{parsetree}
  ( .S.
    (.NP. 'we')
    (.VP.
      (.V. 'gave' )
      (.NP. 'them')
      (.NP. ~ 'a toy')
    )
  )
\end{parsetree}
```



- 수형도의 맨 아래 어휘 요소가 보통의 글꼴로 나오게 하려면 초기화 (Preamble)부분에 다음과 같이 명령을 주면 된다. 그 명령의 의미는 `parsetree`의 `leaf`의 글꼴을 로마자(`\rm`)로 설정하라는 것이다.

```
\ptleafont{\normalsize\rm}{11pt}{3pt}
```

위의 명령으로 어휘요소는 꾸러미에서 설정된 기본 글꼴인 이탤릭체에서 사용자의 설정인 로마꼴로 바뀌어 있음을 알게 된다. 위에 있는 수형도와 아래에 있는 수형도를 비교해 보면 알 수 있다.

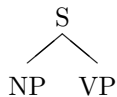


- 기본적인 용법은 다음 표와 같다.
 - 수형도는 `\begin{parsetree}`과 `\end{parsetree}` 사이에 와야 한다.
 - 모든 (하위) 수형도는 왼쪽 괄호 (())로 시작해서 오른쪽 괄호 (())로 끝나는데, 그 괄호 안에는 교점과 딸범주들이 기입된다.
 - 비종단교점은 시작과 끝을 점 (.)으로 구분해 준다: 예를 들면, `.S`처럼 기입해야 한다. 점으로 구분된 비종단기호 내부에서는 아무리 복잡한 수식이라도 기입할 수 있으나,

이 꾸러미에서 특수한 의미가 있는 괄호((,))나 점(.) 혹은 ~ 등은 쓰지 말아야 한다.

- 종단교점은 하나의 교점으로 이루어 지는 것이 원칙이고, 홀따옴표 안에 기입한다: 예를 들어, ‘baby’, ‘gave’ 처럼 기입하면 된다. 그러면 상위 교점에서 자동으로 수직으로 선이 만들어지면서 수형도가 그려 진다.
 - Dummy를 그릴 때는 앞에 ~을 넣으면 된다. ~이 쓰이면 수직선 대신, 삼각형이 그려진다.
- 수형도의 모양을 바꾸기 위한 명령은 다음과 같은 것들이 있다.
 - \pthorgap: 자매 범주 사이의 간격을 조정할 때 (default 12pt)
 - \ptvergap: 어머니와 딸 범주 사이의 간격을 조정할 때 (default 12pt)
 - \ptnodefont: 비종단기호의 글꼴 설정
 - \ptleafont: 종단기호의 글꼴 설정
 - 단순한 예 (1):

```
\begin{parsetree}
(.S.
  .NP.
  .VP.
)
\end{parsetree}
```



- 단순한 예 (2):

```
\begin{parsetree}
(.NP. 'Sam')
\end{parsetree}
```

```

NP
|
Sam

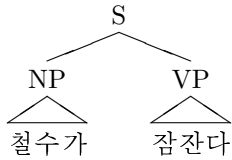
```

- 단순한 예 (3):

```

\pthagap{24pt}
\ptvergap{10pt}
\begin{parsetree}
(.S.
  (.NP. ~‘철수 가’)
  (.VP. ~‘잡잔다’)
)
\end{parsetree}

```



2.2 교점이 복잡한 수형도 그리기

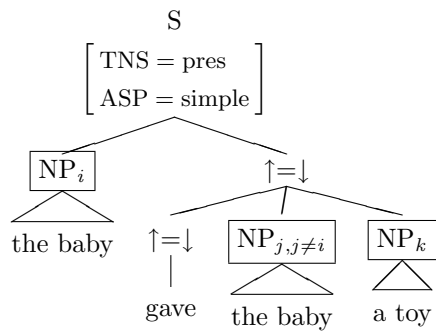
이미 위에서 설명했듯이 `parsetree.sty`를 이용해서 수형도를 그릴 때에는, 비종단 교점이 앞뒤로 `.`에 의해 구분되어 있어야 한다. 이를 이용해서 교점이 복잡한 수형도를 간단하게 그릴 수 있다.

- `covington.sty`에서 제공하는 명령어인 `lfs` 명령어를 이용해서 최상위 교점을 만들고 박스가 있는 NP 교점, 인덱스가 나와 있는 교점 등을 수형도에 나타낼 수 있다. 다음 명령어와 그 아래에 있는 수형도의 모습을 비교해 보기 바란다.

```

\begin{parsetree}
  ( .\lfs{\rm S}{\rm TNS=pres\
    \rm ASP=simple}.
    (\fbox{NP$_i$}. ~ .the baby.)
    ( .$\uparrow$=$\downarrow$.
      (.$\uparrow$=$\downarrow$. .gave. )
      (\fbox{NP$_{j, \{ }{j \neq i}}$}.
        ~ .the baby. )
      (\fbox{NP$_k$}. ~ .a toy.)
    )
  )
\end{parsetree}

```

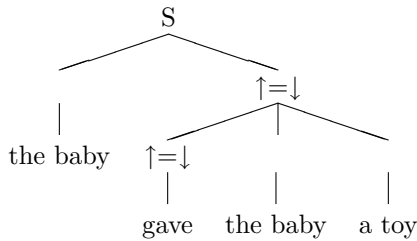


- 교점에 아무 것도 나오지 않게 하려면, 당연히 교점이 시작되는 .과 교점이 끝나는 . 사이에 아무 것도 넣지 않으면 된다. 다음이 그 예가 될 것이다.

```

\begin{parsetree}
  ( .S.
    ( . . .the baby.)
    ( . $\uparrow$ = $\downarrow$ .
      ( . $\uparrow$ = $\downarrow$ . .gave. )
      ( . .
        .the baby. )
      ( . . .a toy.)
    )
  )
\end{parsetree}

```

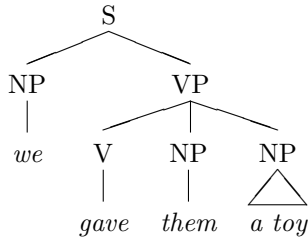


- 수형도를 그리는 명령에 `parsetree.sty`에서 명령어로 쓰이는 기호들인 `(,), ., ~` 등이 쓰일 수 없다. 그러나 부득이 이런 기호들을 수형도에 나타내려면 다음과 같이 하면 된다.
`parsetree.sty`에서 명령어로 쓰이는 기호들인 `(,), ., ~`는 내부적으로는 각각 `\ptbegtree`, `\ptendtree`, `\ptnode`, `\pttritrue`로 바꾸어 해석된다. 우리는 원본에 아예 이런 명령어들을 직접 기입함으로써 특수기호를 수형도에 나타낼 수 있다. 그 방법은 다음과 같다.

- `(`를 `\ptbeg`으로 바꾼다.
- `)`를 `\ptend`로 바꾼다.
- `.N.`와 같은 교점은 `\ptnode{N}`와 같이 바꾼다.
- ‘`↑=↓`’와 같은 어휘요소를 `\ptleaf{↑=↓}`와 같이 바꾼다.
- Dummy를 표현하기 위한 `~`은 `\pttritrue`로 바꾼다.

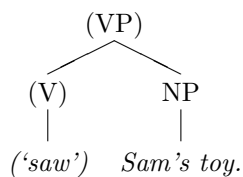
이와 같은 과정을 적은 코드는 다음처럼 될 것이다. 이 수형도는 결과적으로 이 장의 첫 번째 수형도와 같은 조판 결과를 출력으로 낸다.

```
\ptbegtrees
  \ptbeg \ptnode{S}
    \ptbeg \ptnode{NP} \ptleaf{we} \ptend
    \ptbeg \ptnode{VP}
      \ptbeg \ptnode{V} \ptleaf{gave} \ptend
      \ptbeg \ptnode{NP} \ptleaf{them} \ptend
      \ptbeg \ptnode{NP} \pttritrue \ptleaf{a toy} \ptend
    \ptend
  \ptend
\ptendtrees
```



이제 우리는 교점에 괄호가 나오거나, 이 나오는 수형도를 위의 내부 명령어를 직접 기입함으로써 그릴 수 있다. 다음은 그 아래 나오는 수형도를 그리기 위한 코드이다.

```
\ptbegtrees
  \ptbeg \ptnode{(VP)}
    \ptbeg \ptnode{(V)} \ptleaf{('saw)} \ptend
    \ptbeg \ptnode{NP} \ptleaf{Sam's~~~toy.} \ptend
  \ptend
\ptendtrees
```



- 마지막으로 좀 더 복잡한 수형도를 보면 `parsetree.sty`의 표현능력이 크다는 것을 알 수 있다. 편의상 `\npile`이라는 명령을 다음과 같이 정의하고 그 명령을 이용하여 코드를 작성하면 다음과 같은 수형도를 얻을 수 있다.

```

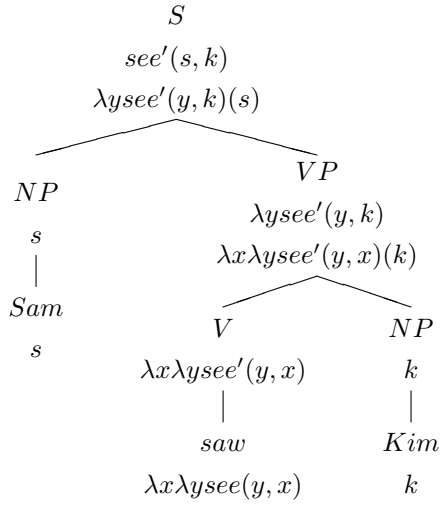
\newcommand{\npile}[1]{%
  \ptnode{          %
    \(\ \begin{array}{c}#1%
      \end{array}    %
  \)  }

```

```

\ptbegtree
\ptbeg
\npile{S\\
    see' (s, k)\\
    \lambda y see' ( y,k ) (s)}
\ptbeg
\npile{NP\\s}
    \npile{Sam\\s}
\ptend
\ptbeg
\npile{VP\\ \lambda y see' ( y,k )\\
    \lambda x \lambda y see' ( y,x ) (k)}
    \ptbeg
    \npile{V\\ \lambda x \lambda y see' ( y,x ) }
        \npile{saw\\ \lambda x \lambda y see ( y,x )}
    \ptend
    \ptbeg
    \npile{NP\\k}
        \npile{Kim\\k}
    \ptend
\ptend
\ptend
\ptendtree

```



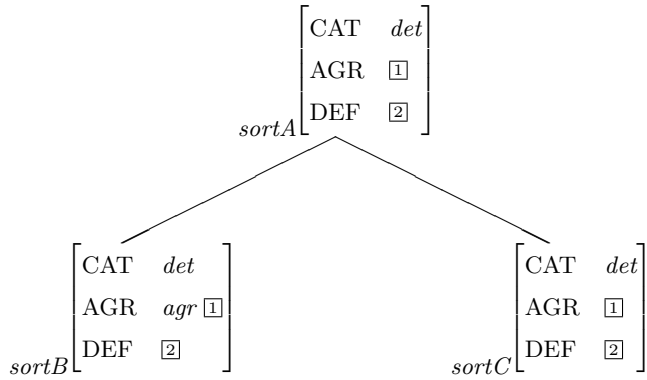
2.3 AVM이 교점인 수형도 그리기

- . 과 . 사이에 무엇이든 나와도 되기 때문에 그 사이에 `avm.sty`을 이용한 수형도를 넣으면 교점이 AVM인 수형도가 된다. 다음은 한 예이다.


```

\begin{parsetree}
  \pthorgap{75pt}
  \ptvergap{40pt}
  ( .\begin{small}
    \begin{avm}
      \sort{\em sortA}{\[CAT & {\em det} \\\
                          AGR & \@1 \\\
                          DEF & \@2 \\\]}
    \end{avm}
    \end{small}.
    .\begin{small}
      \begin{avm}
        \sort{\em sortB}{\[
          CAT & {\em det} \\\
          AGR & {\em agr} \@1 \\\
          DEF & \@2 \\\]}
        \end{avm}
        \end{small}.
        .\begin{small}
          \begin{avm}
            \sort{\em sortC}{\[
              CAT & {\em det} \\\
              AGR & \@1 \\\
              DEF & \@2 \\\]}
            \end{avm}
            \end{small}.
          )
        \end{parsetree}

```



- 이런 수형도는 교점이 끝나는 지점과 시작하는 지점을 찾는 것이 어렵다. 이런 경우 각 교점에 나오는 AVM을 명령어로 치환하면 전체적인 조망을 쉽게 할 수 있다. 그 방법은 $\text{\newcommand{X}{Y}}$ 라는 형태의 명령어를 쓰는 것이다. 그 명령의 의미는 Y전체를 X라고 간단하게 쓰겠다는 것이다. 다음 코드는 각각 아래에 있는 조판을 얻기 위한 것이다.

```

\newcommand{\fsA}{
\begin{small}
\begin{avm}
\sort{\em sortA}{\[CAT & {\em det} \\\
AGR & \textcircled{1} \\\
DEF & \textcircled{2} \\\
\end{avm}
\end{small}}

```

$$\text{sortA} \left[\begin{array}{ll} \text{CAT} & \text{det} \\ \text{AGR} & \text{\textcircled{1}} \\ \text{DEF} & \text{\textcircled{2}} \end{array} \right]$$

```

\newcommand{\fsB}{
\begin{small}
\begin{avm}
\sort{\em sortB}{\[
CAT & {\em det} \\\
AGR & {\em agr} \@1 \\\
DEF & \@2 \]}
\end{avm}
\end{small}}

```

$$\text{sortB} \left[\begin{array}{l} \text{CAT } \textit{det} \\ \text{AGR } \textit{agr} \textcircled{1} \\ \text{DEF } \textcircled{2} \end{array} \right]$$

```

\newcommand{\fsC}{
\begin{small}
\begin{avm}
\sort{\em sortC}{\[
CAT & {\em det} \\\
AGR & \@1 \\\
DEF & \@2 \]}
\end{avm}
\end{small}}

```

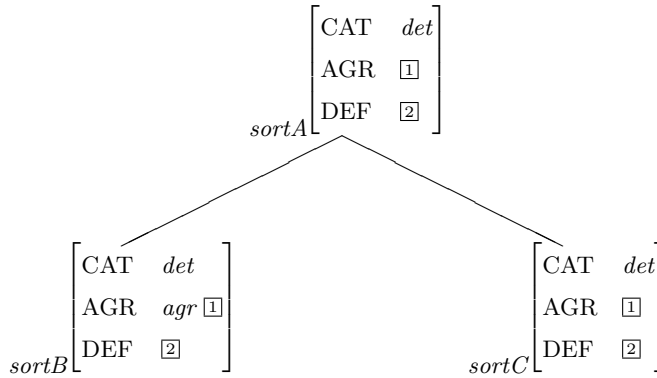
$$\text{sortC} \left[\begin{array}{l} \text{CAT } \textit{det} \\ \text{AGR } \textcircled{1} \\ \text{DEF } \textcircled{2} \end{array} \right]$$

- 이런 방법으로 일단 `\fsA`, `\fsB`, `\fsC` 등의 명령어가 정해지면 교점에 `\fsA`, `\fsB`, `\fsC` 라는 명령어만 넣어도 이미 정해 놓은 수형도가 교점에 나타나는 수형도를 얻을 수 있다. 이 경우 장점으로는 수형도의 전체적인 조망을 쉽게 할 수 있다는 것이다.

```

\begin{parsetree}
  \pthorgap{75pt}
  \ptvergap{40pt}
  ( .\fsA .
    .\fsB .
    .\fsC .
  )
\end{parsetree}

```



이런 방식으로 우리는 아무리 복잡한 수형도라 하더라도 간단한 수형도에 기반해서 혹은 간단한 교점들을 미리 정의해 놓고 쉽게 조망하면서 그릴 수가 있다.

2.4 수형도에서 이동선을 그리기

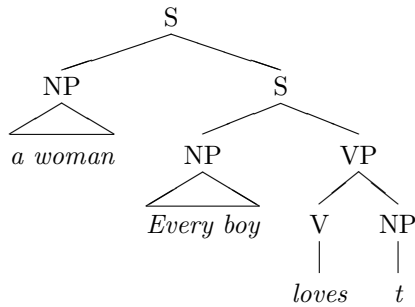
`parsetree.sty` 그 자체는 교점을 자유자재로 넘어 가는 사선을 그릴 수 없다. 그러나 `tree-dvips` 꾸러미를 함께 사용하면 다음과 같은 수형도를 그릴 수 있다.

- “양화사 상승” (“Quantifier Raising”)을 나타내는 수형도를 예로 들어 설명하면 다음과 같다. 여기에서 `\abarnodeconnect[-10pt]{2}{1}`와 `\node` 등은 `tree-dvips`의 명령어이기 때문에 `\usepackage{tree-dvips}`가 초기화 영역에 선언되어야 한다.

```

\begin{parsetree}
( .S.
  (.NP. ~ '\node{1}{\strut a woman}')
  (.S.
    (.NP. ~ 'Every boy')
    (.{VP}.
      (.V. '{loves}' )
      (.NP. '\node{2}{\strut t}'))
    )
  )
)
\abarnodeconnect[-10pt]{2}{1}
\end{parsetree}

```

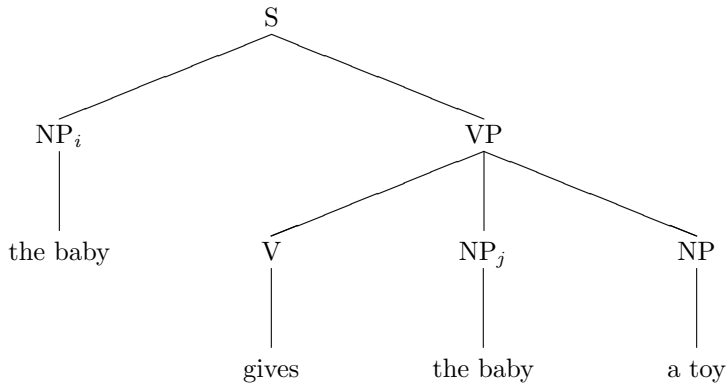


- 동일한 원리로 그려진 더 복잡한 수형도의 예를 코드와 함께 제시하면 다음과 같다.

```

\pthorgap{50pt}
\ptvergap{30pt}
\begin{parsetree}
  ( .\node{1}{S}.
    ( .\node{2}{NP$_i$}. .the baby.)
    ( .\node{3}{VP}.
      (. \node{4}{V}. .\node{5}{gives}. )
      (.NP$_j$ .the baby.)
      (.NP .a toy.)
    )
  )
\anodecurve[t1]{2}[1]{1}{20pt}
\anodecurve[r]{1}[t]{3}{20pt}
\anodecurve[b1]{3}[t]{4}{20pt}
\anodecurve[b1]{4}[1]{5}{20pt}
\end{parsetree}

```



제 3 절 `rtrees.sty`를 이용한 수형도 그리기

Stanford대학교의 Rob Malouf가 HPSG에서 수형도를 잘 그리려는 목적으로 1996년 7월 22일 발표한 꾸러미이다. 언어학자를 위한 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ CD에 수록된 꾸러미의 버전은 1997년 06월 16일날 발표한 것이다. 이 꾸러미는 수형도를 그리는데 `ps-tricks` 매크로를 사용한다. `Sag`을

비롯한 Stanford대학교의 대부분의 학자들은 수형도를 그리는데 이 꾸러미를 사용한다. 그러나 이 꾸러미의 단점은 다른 꾸러미와 마찬가지로 .ps 화일로 만들었을 경우에만 사선이 보인다는 것이다.

3.1 기본적인 명령어

기본적인 명령어들은 다음과 같다.

- 기본적인 환경은 `\begin{tree}`로 시작해서 `\end{tree}`로 끝나야 한다.
- `\br{NODE}{DAUGHTERS...}`: 어머니 교점이 NODE이고 딸교점(들)이 DAUGHTERS인 수형도를 만든다. DAUGHTERS 자리에 나오는 것은 또 다시 `\br`로 시작되는 수형도이거나, `\lf`, `\rlf`, `\tlf`로 시작되는 어휘요소들이다.
- `\lf{NODE}`: 이탤릭체로 된 종단기호 (어휘요소)를 생성해 낸다.
- `\rlf{NODE}`: 로만 정체로 된 종단기호 (어휘요소)를 생성해 낸다.
- `\tlf{NODE}`: 삼각형으로 처리된 이탤릭체의 어휘요소들을 생성해 낸다.
- `\psset{treenodesize=0.055in}`: 교점들의 간격을 조정하는 명령어이다.

3.2 Type Hierarchy 그리기

단순한 이분지 유형위계 (Type Hierarchy)를 그리는 방법은 기본적으로 수형도를 그리는 방법과 동일하다. 다만, `\br{NODE}{DAUGHTERS...}`에서 NODE 자리에 DAUGHTERS 자리에 이탤릭체의 유형이 오면 된다.

```

\begin{tree}
\psset{treenodesize=0.055in}
\br{\emph{sign}}{\lf{\emph{word}}
  \br{\emph{phrase}}{\lf{\emph{non-headed}}
    \br{\emph{headed}}{\lf{\emph{head-adjunct}}
      \br{\emph{head-nexus}}{\lf{\emph{head-filler}}
        \lf{\emph{hd-only}}
      \br{\emph{hd-valence}}{\lf{\emph{head-subj}}
        \lf{\emph{head-spr}}
        \lf{\emph{head-comps}}}}}}}}
\end{tree}

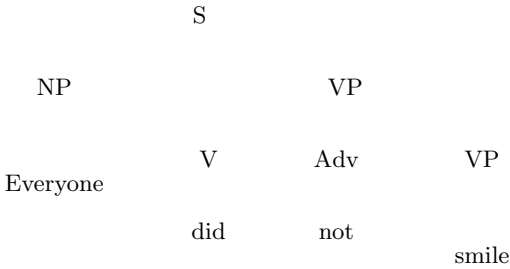
```

*sign**word**phrase**non-headed**headed**head-adjunct**head-nexus**head-filler* *hd-only**hd-valence**head-subj* *head-spr* *head-comps*

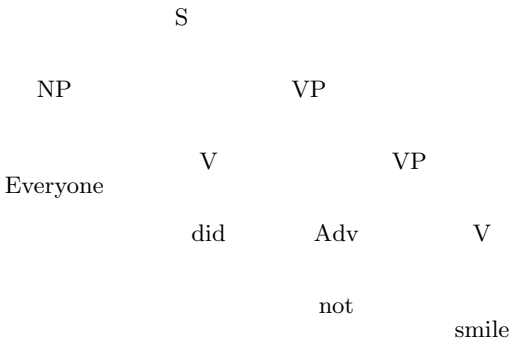
3.3 일반적인 수형도 그리기

`\br{NODE}{DAUGHTERS...}`에서 적절한 교점과 어휘요소들을 기입하기만 하면 되기 때문에 일반적인 수형도를 그리는 방법은 이제 더 이상 어려운 일이 아닐 것이다. `rtrees.sty` 사용시 주의할 것은 각 명령어들이 }로 닫히게 되어 있는데 그 숫자가 서로 일치하지 않으면 에러 메시지를 낸다는 것이다. Top-down 식으로 그리든 Bottom-up식으로 그리든 열려진 { 기호의 수 만큼 } 나오도록 조심할 필요가 있다.


```
\begin{tree}
\br{S}{\br{NP}{\tlf{Everyone}}
      \br{VP}{\br{V}{\lf{did}}
              \br{Adv}{\lf{not}}
              \br{VP}{\tlf{smile}}}}
\end{tree}
```



```
\begin{tree}
\br{S}{\br{NP}{\tlf{Everyone}}
      \br{VP}{\br{V}{\lf{did}}
              \br{VP}{\br{Adv}{\lf{not}}
                      \br{V}{\tlf{smile}}}}}
\end{tree}
```



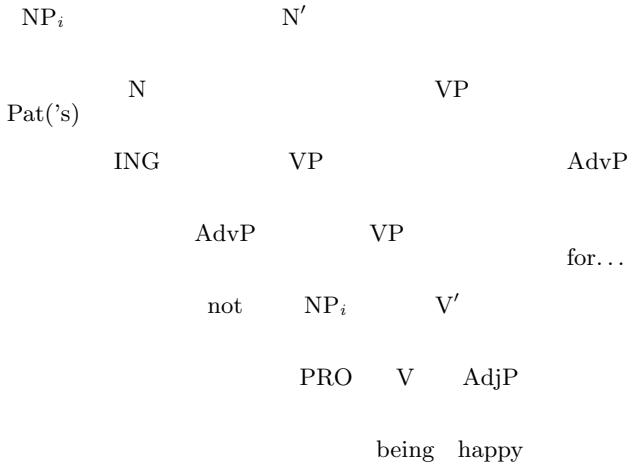
Malouf의 논문에 자주 등장하는 수형도를 보면서 `rtrees.sty`의 사용법을 익히기 바란다.

```

\begin{tree}
\psset{treeseq=.25in}
\br{NP}{\br{NP$_i$}{\tlf{Pat('s)}}
      \br{N$'$}{\br{N}{\lf{ING}}
        \br{VP}{\br{VP}{
          \br{AdvP}{\lf{not}}
          \br{VP}{\br{NP$_i$}{\lf{PRO}}
            \br{V$'$}{\br{V}{\lf{being}}
              \br{AdjP}{\lf{happy}}}}}}
          \br{AdvP}{\tlf{for\ldots}}}}}}
\end{tree}

```

NP

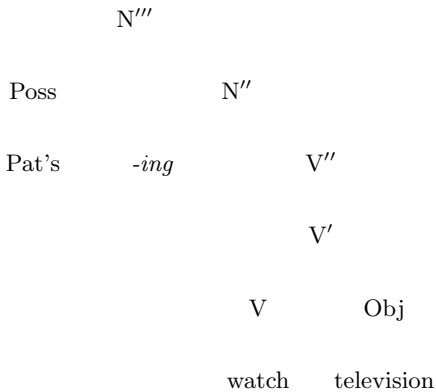


다음 수형도도 그의 박사학위 논문에 나오는 수형도이다. 교점에 '으로 바를 처리한 경우이다.

```

\begin{tree}
\br{N$'''}{\br{Poss}{\lf{Pat's}}
      \br{N$'''}{\lf{\emph{-ing}}
        \br{V$'''}{\br{V$'''}{
          \br{V}{\lf{watch}}
          \br{Obj}{\lf{television}}}}}}
\end{tree}

```



3.4 rtrees.sty의 기타 기능

- rlap의 사용:** 수형도는 상위 교점의 길이를 계산해서 그 교점의 하단 한가운데에 사선의 꼭지점이 놓이도록 하는데, 이 때 상위 교점이 너무나 길면 전체 수형도의 균형에 문제가 있는 경우가 있다. 이런 경우를 피하기 위하여 특정 부분을 `\rlap{\ }`으로 묶어, 그 명령으로 묶인 부분은 상위 교점의 길이를 계산할 때, 산입되지 않도록 해 줌으로써 균형잡힌 수형도를 그릴 수 있게 해 준다. 아래 수형도 예에서는 `\rlap{\ [=S]}`와 `\rlap{\ [=VP]}`가 노드의 전체 길이 계산에서 제외되었음을 알 수 있다. 다음과 같은 코드는 바로 다음에 나오는 수형도를 조판 결과로 내놓는다. 다음에 두 가지 예들 코드와 함께 제시한다.

```

\begin{tree}
\br{proj(-ing)\rlap{\ [=S]}}{\br{NP}{\lf{John}}
  \br{Infl}{\lf{-ing}}
  \br{proj(sing)\rlap{\ [=VP]}}{\br{V}{\lf{sing}}
    \br{NP}{\tlf{the aria}}}}
\end{tree}

```

proj(-ing) [=S]

NP Infl proj(sing) [=VP]

John -ing V NP

sing

the aria

```

\begin{tree}
\br{proj(sing+ing)\rlap{\ [=NP]}}{\br{NP}{\lf{John}}
  \br{proj(sing+ing)\rlap{\ [=NP]}}
    {\br{N}{\br{V}{\lf{sing}}
      \lf{-ing}}
    \br{NP}{\tlf{the aria}}}}
\end{tree}

```

proj(sing+ing) [=NP]

NP proj(sing+ing) [=NP]

John N NP

V -ing

the aria

sing

- **nccurve**의 사용: 수형도를 그리다 보면 설명의 필요성에 의해서 직선이 아닌 곡선으로 두 개 이상의 교점을 연결해야 할 경우가 있다. 이런 경우를 위해서는 `\nccurve{NODE1}{NODE1}`를 사용한다. 이 때 NODE1과 NODE2는 `\lf[NODE1]{N}`과 `\lf[NODE2]{painting}`처럼 []안에 교점의 이름을 설정해 두어야 한다. 다음 두 예들

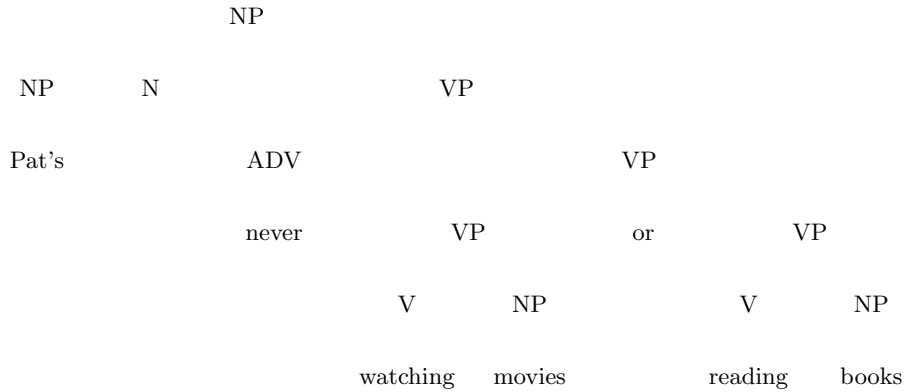
을 보면, 그 명령어의 사용법을 쉽게 알 수 있다.

```
\begin{tree}
\br{NP}{\br{NP}{\lf{Brown's}}
  \lf[up]{N}
  \br{VP}{\br{V}{\lf[down]{painting}}
    \br{NP}{\tlf{his daughter}}}}
\nccurve{up}{down}
\end{tree}
```

NP

NP	N	VP
Brown's	V	NP
	painting	his daughter

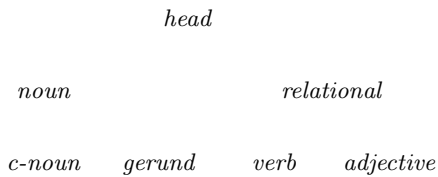
```
\begin{tree}
\br{NP}{\br{NP}{\lf{Pat's}}
  \lf[up]{N}
  \br{VP}{\br{ADV}{\lf{never}}
    \br{VP}{\br{VP}{\br{V}{\lf[down1]{watching}}
      \br{NP}{\lf{movies}}}
    \lf{or}
    \br{VP}{\br{V}{\lf[down2]{reading}}
      \br{NP}{\lf{books}}}}}}
\nccurve[ncurvA=1.5]{up}{down1}
\nccurve[ncurvA=.95,ncurvB=.55]{up}{down2}
\end{tree}
```



- **nctdiag**의 사용: 다음은 `nctdiag{NODE1}`을 사용해서 교점을 연결하는 예이다. 이 명령어는 “다중계승위계” (Multiple Inheritance Hierarchy)를 그릴 때 유용하게 쓰일 수 있다.

```

\begin{tree}
\psset{treenodesize=0.05in}
\br{\emph{head}}{\br{\emph{noun}}{\Tn\lf{\emph{c-noun}}
\TR[name=down]{\emph{gerund}}
\pstree{\TR[name=up]{\emph{relational}}
{\lf{\emph{verb}}\lf{\emph{adjective}}}}}
\ncdiag{up}{down}
\end{tree}
  
```



- **avmtree**의 사용: 다음은 `\begin{avmtree}`와 `\end{avmtree}`를 사용해서 교점에 AVM이 나오는 수형도를 그리는 방법이다.

```

\begin{avmtree}
\br{\[ \emph{noun-poss-cx}\[
  CAT \[ HEAD & \@1\[
    VAL & \[ SUBJ & \q<\ \q>\[
      COMPS & \q<\ \q>\[
        SPR & \q<\ \q>
      \]
    \]
  \]}
{\br{\@2\[ CAT\|HEAD & \[ \emph{noun}\[
  CASE \emph{gen}
  \]
  \]}
{\lf{Brown's}}
\br{\[ \emph{head-comp-cx}\[
  CAT \[ HEAD & \@1\[
    VAL & \[ SUBJ & \q<\@2\q>\[
      COMPS & \q<\ \q>\[
        SPR & \q<\@2\q>
      \]
    \]
  \]}
{\br{\[ CAT \[ HEAD & \@1\emph{gerund}\[
  VAL & \[ SUBJ & \q<\@2\q>\[
    COMPS & \q<\@3\q>\[
      SPR & \q<\@2\q>
    \]
  \]
  \]}
{\lf{painting}}
\br{\@3\[ CAT\|HEAD \emph{noun} \]}
{\lf{his daughter}}}}
\end{avmtree}

```

$$\left[\begin{array}{l} \textit{noun-poss-cx} \\ \text{CAT} \left[\begin{array}{l} \text{HEAD} \text{ ①} \\ \text{VAL} \left[\begin{array}{l} \text{SUBJ} \langle \rangle \\ \text{COMPS} \langle \rangle \\ \text{SPR} \langle \rangle \end{array} \right] \end{array} \right] \end{array} \right]$$

$$\text{②} \left[\text{CAT} \mid \text{HEAD} \left[\begin{array}{l} \textit{noun} \\ \text{CASE} \textit{gen} \end{array} \right] \right] \quad \left[\begin{array}{l} \textit{head-comp-cx} \\ \text{CAT} \left[\begin{array}{l} \text{HEAD} \text{ ①} \\ \text{VAL} \left[\begin{array}{l} \text{SUBJ} \langle \text{②} \rangle \\ \text{COMPS} \langle \rangle \\ \text{SPR} \langle \text{②} \rangle \end{array} \right] \end{array} \right] \end{array} \right]$$

$$\text{Brown's} \quad \left[\begin{array}{l} \text{CAT} \left[\begin{array}{l} \text{HEAD} \text{ ①} \textit{gerund} \\ \text{VAL} \left[\begin{array}{l} \text{SUBJ} \langle \text{②} \rangle \\ \text{COMPS} \langle \text{③} \rangle \\ \text{SPR} \langle \text{②} \rangle \end{array} \right] \end{array} \right] \end{array} \right] \quad \text{③} \left[\text{CAT} \mid \text{HEAD} \textit{noun} \right]$$

painting

his daughter

- 수형도의 일부를 점선으로 구분해 두기:

`\psovalbox[linestyle=dashed,boxsep=false]{NP}`와 같은 명령을 사용해서 다음 수형도에 나타난 바와 같이 교점이나 수형도의 일부가 점선으로 구분된 수형도를 만들 수 있다.


```

\begin{tree}
\br{\psovalbox[linestyle=dashed,boxsep=false]{NP}}
      {\br{S}{\br{VP}{\br{NP}{\lf{Subj}}
        \br{V$'}{\lf{V}}
          \br{NP}{\lf{Obj}}}}}}
\end{tree}

```

NP

S

VP

NP

V'

Subj

V

NP

Obj

제 4 절 tree-dvips.sty를 이용한 수형도 그리기

(to be written)

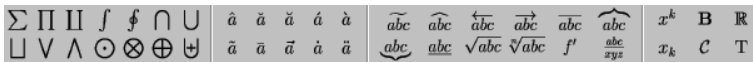
제 7 장

의미론 학자를 위한 한글 LATEX 의 기호와 수식

이 장에서는 의미론 학자들이 사용할 가능성이 있는 특수 기호들을 모아 정리하고 복잡한 수식의 예를 몇 개 예로 들어 수식을 작성하는 방법을 설명하고자 한다.

제 1 절 수학 기호

수학 기호가 형식의미론 논문이나 저서에 많이 등장한다. WinEdt를 사용하면, 해당 아이콘을 클릭하기만 하면 해당 명령어가 원본 파일에 자동으로 입력되기 때문에 편리하다. WinEdt의 아이콘을 소개하면 다음과 같다.



다음에 제시된 수학기호들을 제대로 사용하기 위해서는 초기화영역(Preamble)에 다음과 같이 `\usepackage{amsmath}`와 `\usepackage{amssymb}`으로 설정되어 있어야 한다.

```
\documentclass[leqno,b5paper]{book}
\usepackage[hardbold]{hangul}
\usepackage{amsmath,amssymb}
```

수학 기호는 아래의 표의 화살표 왼쪽과 같이 입력하면 화살표 오른쪽과 같이 조판되어 나온다.

<code>\$\$\sum\$</code>	→	Σ	<code>\$\$\prod\$</code>	→	Π
<code>\$\$\coprod\$</code>	→	\coprod	<code>\$\$\int\$</code>	→	\int
<code>\$\$\oint\$</code>	→	\oint	<code>\$\$\bigcap\$</code>	→	\bigcap
<code>\$\$\bigcup\$</code>	→	\bigcup	<code>\$\$\bigsqcup\$</code>	→	\bigsqcup
<code>\$\$\bigvee\$</code>	→	\bigvee	<code>\$\$\bigwedge\$</code>	→	\bigwedge
<code>\$\$\bigodot\$</code>	→	\bigodot	<code>\$\$\bigotimes\$</code>	→	\bigotimes
<code>\$\$\bigoplus\$</code>	→	\bigoplus	<code>\$\$\biguplus\$</code>	→	\biguplus
<code>\$\$\hat{a}\$</code>	→	\hat{a}	<code>\$\$\check{a}\$</code>	→	\check{a}
<code>\$\$\breve{a}\$</code>	→	\breve{a}	<code>\$\$\acute{a}\$</code>	→	\acute{a}
<code>\$\$\grave{a}\$</code>	→	\grave{a}	<code>\$\$\tilde{a}\$</code>	→	\tilde{a}

\bar{a}	\vec{a}	\vec{a}	\vec{a}
\dot{a}	\ddot{a}	\ddot{a}	\ddot{a}
\widetilde{abc}	\widehat{abc}	\widehat{abc}	\widehat{abc}
\overleftarrow{abc}	\overrightarrow{abc}	\overrightarrow{abc}	\overrightarrow{abc}
\overline{abc}	\overbrace{abc}	\overbrace{abc}	\overbrace{abc}
\underbrace{abc}	\underline{abc}	\underline{abc}	\underline{abc}
\sqrt{abc}	$\sqrt[n]{abc}$	$\sqrt[n]{abc}$	$\sqrt[n]{abc}$
f'	$\frac{abc}{xyz}$	$\frac{abc}{xyz}$	$\frac{abc}{xyz}$
x^k	\mathbf{B}	\mathbf{B}	\mathbf{B}
x_k	\mathcal{C}	\mathcal{C}	\mathcal{C}
T	T	T	T

제 2 절 그리스 문자

그리스 문자도 WinEdt를 사용하면 쉽게 입력할 수 있다. 해당 아이콘을 클릭하기만 하면 해당 명령어가 원본 파일에 자동으로 입력되기 때문이다. WinEdt의 아이콘을 소개하면 다음과 같다.

Γ	Δ	Θ	Λ	Ξ	Π	α	β	γ	δ	ϵ	ζ	η	θ	ϑ	ι	κ	λ	μ	ν	F	\square
Σ	Υ	Φ	Ψ	Ω	\S	ξ	\omicron	π	ϖ	ρ	ϱ	σ	ς	τ	υ	ϕ	φ	χ	ψ	ω	\lrcorner

위의 그리스 문자를 개별적으로 입력할 때는 아래의 표의 화살표 왼쪽과 같이 입력하면 된다. 그러면 화살표 오른쪽과 같이 조판되어 나온다.

$\backslash\Gamma$	\rightarrow	Γ	$\backslash\Delta$	\rightarrow	Δ
$\backslash\Theta$	\rightarrow	Θ	$\backslash\Lambda$	\rightarrow	Λ
\backslashXi	\rightarrow	Ξ	$\backslash\Pi$	\rightarrow	Π
$\backslash\Sigma$	\rightarrow	Σ	\backslashUpsilon	\rightarrow	Υ
\backslashPhi	\rightarrow	Φ	\backslashPsi	\rightarrow	Ψ
\backslashOmega	\rightarrow	Ω	$\backslash\text{\$}$	\rightarrow	$\text{\$}$
$\backslash\alpha$	\rightarrow	α	$\backslash\beta$	\rightarrow	β
$\backslash\gamma$	\rightarrow	γ	$\backslash\delta$	\rightarrow	δ
$\backslash\epsilon$	\rightarrow	ϵ	$\backslash\text{\varepsilon}$	\rightarrow	\varepsilon
$\backslash\zeta$	\rightarrow	ζ	$\backslash\eta$	\rightarrow	η

$\backslash\theta$	\rightarrow	θ	$\backslash\text{\vartheta}$	\rightarrow	\vartheta
$\backslash\iota$	\rightarrow	ι	$\backslash\text{\kappa}$	\rightarrow	\kappa
$\backslash\lambda$	\rightarrow	λ	$\backslash\text{\mu}$	\rightarrow	\mu
$\backslash\text{\nu}$	\rightarrow	\nu	$\backslash\text{\xi}$	\rightarrow	\xi
$\backslash\text{\o}$	\rightarrow	\o	$\backslash\text{\pi}$	\rightarrow	\pi
$\backslash\text{\varpi}$	\rightarrow	\varpi	$\backslash\text{\rho}$	\rightarrow	\rho
$\backslash\text{\sigma}$	\rightarrow	\sigma	$\backslash\text{\varsigma}$	\rightarrow	\varsigma
$\backslash\text{\tau}$	\rightarrow	\tau	$\backslash\text{\upsilon}$	\rightarrow	\upsilon
$\backslash\text{\phi}$	\rightarrow	\phi	$\backslash\text{\varphi}$	\rightarrow	\varphi
$\backslash\text{\chi}$	\rightarrow	\chi	$\backslash\text{\psi}$	\rightarrow	\psi
$\backslash\text{\omega}$	\rightarrow	\omega			

제 3 절 특수 기호

특수 기호도 WinEdt를 사용하면 쉽게 입력할 수 있다. 해당 아이콘

을 클릭하기만 하면 해당 명령어가 원본 파일에 자동으로 입력되기 때문이다. WinEdt의 아이콘을 소개하면 다음과 같다.



위의 특수 기호 문자를 개별적으로 입력할 때는 아래의 표의 화살표 왼쪽과 같이 입력하면 된다. 그러면 화살표 오른쪽과 같이 조판되어 나온다.

<code>\$\$\aleph\$</code>	→	\aleph	<code>\$\$\hbar\$</code>	→	\hbar
<code>\$\$\imath\$</code>	→	\imath	<code>\$\$\jmath\$</code>	→	\jmath
<code>\$\$\ell\$</code>	→	ℓ	<code>\$\$\wp\$</code>	→	\wp
<code>\$\$\Re\$</code>	→	\Re	<code>\$\$\Im\$</code>	→	\Im
<code>\$\$\prime\$</code>	→	$'$	<code>\$\$\emptyset\$</code>	→	\emptyset
<code>\$\$\angle\$</code>	→	\angle	<code>\$\$\infty\$</code>	→	∞
<code>\$\$\partial\$</code>	→	∂	<code>\$\$\nabla\$</code>	→	∇
<code>\$\$\triangle\$</code>	→	\triangle	<code>\$\$\forall\$</code>	→	\forall
<code>\$\$\exists\$</code>	→	\exists	<code>\$\$\neg\$</code>	→	\neg
<code>\$\$\surd\$</code>	→	\surd	<code>\$\$\top\$</code>	→	\top
<code>\$\$\bot\$</code>	→	\bot	<code>\$\$\backslash\$</code>	→	\backslash

$\backslash flat$	→	b	$\backslash natural$	→	♮
$\backslash sharp$	→	♯	$\backslash $	→	
$\backslash clubsuit$	→	♣	$\backslash diamondsuit$	→	◇
$\backslash heartsuit$	→	♥	$\backslash spadesuit$	→	♠
$\backslash dag$	→	†	$\backslash ddag$	→	‡
$\backslash S$	→	§	$\backslash P$	→	¶
$\backslash copyright$	→	©	$\backslash pounds$	→	£
$\backslash checkmark$	→	✓	$\backslash maltese$	→	✠
$\backslash circledR$	→	®	$\backslash yen$	→	¥
$\backslash ulcorner$	→	┌	$\backslash urcorner$	→	┐
$\backslash llcorner$	→	└	$\backslash lrcorner$	→	┘
$\backslash diamond$	→	◇	$\backslash mho$	→	Ω
$\backslash ldots$	→	...	$\backslash cdots$	→	⋯
$\backslash Box$	→	□	$\backslash cdot$	→	·
$\backslash vdots$	→	⋮	$\backslash ddots$	→	⋱

제 4 절 각국 언어 특수 문자

각국 언어 특수 문자도 WinEdt를 사용하면 쉽게 입력할 수 있다. 해당 아이콘을 클릭하기만 하면 해당 명령어가 원본 파일에 자동으로 입력되기 때문이다. WinEdt의 아이콘을 소개하면 다음과 같다.



위의 각국 언어 특수 문자를 개별적으로 입력할 때는 아래의 표의 화살표 왼쪽과 같이 입력하면 된다. 그러면 화살표 오른쪽과 같이 조판되어 나온다.

<code>\`{o}</code> → ò	<code>\' {o}</code> → ó
<code>\^{o}</code> → ô	<code>\" {o}</code> → ö
<code>\~{o}</code> → õ	<code>\={o}</code> → ȍ
<code>\.{o}</code> → ȏ	<code>\u{o}</code> → ů
<code>\v{o}</code> → ǒ	<code>\H{o}</code> → Ǫ
<code>\t{o}</code> → ȩ	<code>\c{o}</code> → ȣ
<code>\d{o}</code> → ȡ	<code>\b{o}</code> → Ƞ
<code>\oe</code> → œ	<code>\ae</code> → æ
<code>\aa</code> → å	<code>\o</code> → ø
<code>\l</code> → ł	<code>?'</code> → ĭ
<code>\OE</code> → Œ	<code>\AE</code> → Æ
<code>\AA</code> → Å	<code>\O</code> → Ø
<code>\L</code> → Ł	<code>!'</code> → ĳ

<code>\{a}</code> → ä	<code>\{e}</code> → ë
<code>\{i}</code> → ï	<code>\{o}</code> → ö
<code>\{u}</code> → ü	<code>\{y}</code> → ÿ
<code>\ss</code> → ß	<code>\{A}</code> → Ä
<code>\{E}</code> → Ę	<code>\{I}</code> → Ĩ
<code>\{O}</code> → Ö	<code>\{U}</code> → Ü
<code>\{Y}</code> → Ÿ	<code>\SS</code> → SS
<code>\c{c}</code> → ç	<code>\'e}</code> → è
<code>\'e}</code> → é	<code>\^e}</code> → ê
<code>\'o}</code> → ò	<code>\'o}</code> → ó
<code>\^o}</code> → ô	<code>\c{C}</code> → Ç
<code>\'E}</code> → È	<code>\'E}</code> → É
<code>\^E}</code> → Ê	<code>\'O}</code> → Ò
<code>\'O}</code> → Ó	<code>\^O}</code> → Ô

제 5 절 다양한 괄호 등 기호 문자

각국 언어 특수 문자도 WinEdt를 사용하면 쉽게 입력할 수 있다. 해당 아이콘을 클릭하기만 하면 해당 명령어가 원본 파일에 자동으로 입력되기 때문이다. WinEdt의 아이콘을 소개하면 다음과 같다.

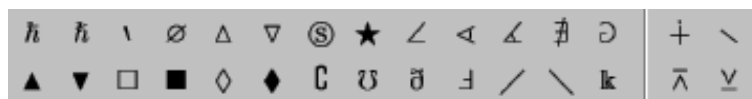
([{ [[< /	↑ ↓ ⇅	>= >= >=	<code>\left \begin</code>
)] }]] > \	↑ ↓ ⇅	>= >= >=	<code>\right \end</code>

위의 각국 언어 특수 문자를 개별적으로 입력할 때는 아래의 표의 화살표 왼쪽과 같이 입력하면 된다. 그러면 화살표 오른쪽과 같이 조판되어 나온다.

$\$(\$$ → ($\$)\$$ →)
$\$[\$$ → [$\$]\$$ →]
$\$\{\$$ → {	$\$\}\$$ → }
$\$\lfloor\$$ → ⌊	$\$\rfloor\$$ → ⌋
$\$\lceil\$$ → ⌈	$\$\rceil\$$ → ⌉
$\$\langle\$$ → <	$\$\rangle\$$ → >
$\$/\$$ → /	$\$\backslash\$$ → \
$\$ \$$ →	$\$\ \$$ →
$\$\uparrow\$$ → ↑	$\$\downarrow\$$ → ↓
$\$\updownarrow\$$ → ⇕	$\$\Updownarrow\$$ → ⇓
$\$\Downarrow\$$ → ⇩	$\$\Uparrow\$$ → ⇪

그 밖에 긴요하게 쓰이는 기호들이 WinEdt의 메뉴에 아이콘으로 나와 있는데, 지면 상의 이유로 자세한 입력 방법은 생략하기로 한다. 상황 의미론 (Situation Semantics)에서 쓰이는 Support 기호 등 다양한 기호가 지원되니 아래 그림을 참고해서 문서 작성에 이용하면 좋겠다는 바램이다. 다행히도 WinEdt에서는 아이콘만 클릭하면 입력 명령어를 바로 원본 파일에 기입해 주기 때문에 입력 명령어를 일일이 다 외우고 있지 않아도 된다.

● 수학 기호 1



● 수학 기호 2



- 화살표 2



- 등식 기호 1



- 등식 기호 2



제 6 절 Theorem과 Proof

형식 의미론이나 자질 논리 (Feature Logic) 등 비교적 복잡한 수식이 있는 논문을 쓰는 일은 아마도 L^AT_EX가 없으면 내용 보다는도 기호 그리는 일이 더 시간을 허비하는 일일 것이다. 다음에 두 가지 예를 들어 Theorem이나 Proof 등을 작성하는 방법을 소개한다.

- 다음과 같이 초기화 영역 (Preamble)에 기입한다.

```
\documentclass[11pt]{article}
\newtheorem{thm}{Theorem}
\newenvironment{pf}{\medskip\noindent{\bf Proof}\enspace}
{\hfill\newline\smallskip}
```

- 그런 다음 `\begin{document}` 아래에 다음과 같은 Theorem을 얻기 위해서는 원본 파일에 그 아래에 있는 명령어들을 넣으면 된다.

Theorem 1

$$\frac{\pi}{2} = \lim_{n \rightarrow \infty} \frac{2}{1} \frac{2}{3} \frac{4}{3} \frac{4}{5} \frac{6}{5} \frac{6}{7} \cdots \frac{2n}{2n-1} \frac{2n}{2n+1}$$

```

\begin{thm}
$$ \frac{\pi}{2} = \lim_{n \to \infty} \frac{2}{1} \frac{2}{3} \frac{4}{3} \frac{4}{5} \frac{6}{5} \frac{6}{7} \cdots \frac{2n}{2n-1} \frac{2n}{2n+1} $$
\end{thm}

```

- Proof도 마찬가지이다. 다음과 같은 조판 결과를 얻기 위해서는 그 아래에 있는 것과 같이 원판 파일에 적혀 있으면 된다.

Proof The proof is left to the reader (Follow the steps below).

1. Use the iterated integral formular for the powers of sine functions to prove:

$$\int_0^{\frac{\pi}{2}} \sin^{2n} x \, dx = \frac{2n-1}{2n} \cdot \frac{1}{2} \frac{\pi}{2},$$

2. For $n = 1, 2, 3, \dots$, use the fact that \sin^n is decreasing to prove the following inequality:

$$1 \leq \frac{\int_0^{\pi/2} \sin^{2n-1} x \, dx}{\int_0^{\pi/2} \sin^{2n+1} x \, dx} \leq 1 + \frac{1}{2n}$$

3. Find the limit of the Willis product by using the ratio of the integrals of $\sin^{2n} x$ and $\sin^{2n+1} x$ over the interval $[0, \pi/2]$.

```

\begin{pf} The proof is left to the reader
(Follow the steps below).

\begin{enumerate}

\item Use the iterated integral formular for the powers
of sine functions to prove:
\begin{eqnarray*}
\int_0^{\frac{\pi}{2}} \sin^{2n}x \, dx &=& \frac{2n-1}{2n} \cdot \\
&& \frac{1}{2} \frac{\pi}{2}, \\
\int_0^{\frac{\pi}{2}} \sin^{2n+1}x \, dx &=& \frac{2n}{2n+1} \cdot \\
&& \frac{1}{2} \frac{\pi}{2}.
\end{eqnarray*}

\item For  $n=1,2,3,\dots$ , use the fact that  $\sin^n x$  is
decreasing to prove the following inequality:
$$
1 \leq \frac{\int_0^{\pi/2} \sin^{2n-1}x \, dx}{\int_0^{\pi/2} \sin^{2n+1}x \, dx} \leq 1 + \frac{1}{2n}
$$

\item Find the limit of the Willis product by using the ratio
of the integrals of  $\sin^{2n}x$  and  $\sin^{2n+1}x$  over the
interval  $[0, \pi/2]$ .

\end{enumerate}
\end{pf}

```


제 8 장

한글 \LaTeX 로 담화표상구조 (DRS) 만들기

Hans Kamp가 제안한 담화표상이론 (Discourse Representation Theory; DRT)에서 사용하는 담화표상구조 (DRS)를 그리는 가장 쉬운 방법은 이미 앞에서 설명한 Michael Covington의 `covington.sty`에 정의된 명령어를 사용하는 것이다. 이 장에서는 이 방법을 포함한 담화표상구조 (DRS) 그리는 방법을 소개한다.¹

제 1 절 `covington.sty`를 이용한 담화표상구조 (DRS) 그리기

- `covington.sty`을 사용하려면 초기화영역에 `\usepackage{covington}`이라고 선언해 두어야 한다. 이 선언이 유효하려면 Mik \TeX 가 이 파일이 어디에 있는가 하는 정보를 Database에 가지고 있어야 한다.

그렇려면 `c:\localtexmf\tex\latex` 아래에 `covington.sty`을

¹이 장의 내용은 영국 Essex 대학교의 Doug Arnold가 정리한 내용을 기반으로 하고 있다. 그 사이트에는 이런 구조 뿐만 아니고 다른 언어학 관련 \LaTeX 에 정보가 많이 있다. 그 주소는 다음과 같다: <http://clwww.essex.ac.uk/latex4ling/>

복사해 놓고 MikTeX의 Filename Database를 새로 작성하도록 명령을 넣어야 한다.

- 담화표상구조를 그리는 기본 환경은 `\begin{flushleft}`로 시작하고 `\end{flushleft}`로 끝나는 환경이다. 이 환경에 다음에 소개하는 기본 명령어들을 넣으면 된다.
- 담화표상구조를 그리는 기본 명령어는 `\drs{ }{ }`이다. 이 명령어는 두 개의 논항을 필요로 하는데, 하나는 상자의 위에 나타나는 담화변항의 집합 (set of discourse variables)이고 다른 하나는 DRS 조건들 (set of conditions)이다. 다음은 가장 단순한 예이다.

```
\begin{flushleft}
\drs{x_1}{donkey(x_1)}
\end{flushleft}
```

x_1
$donkey(x_1)$

- 일반적으로 DRS의 조건부에는 여러 행으로 이뤄진 수식이 있는데, 이렇게 행을 바꿀 때에는 `\\`라는 명령어를 행이 바뀔 자리에 넣어두면 된다.

```
\begin{flushleft}
\drs{x_1,x_2}{man(x_1)\\
whisles(x_1)\\
dog(x_2)\\
follows(x_2,x_1)}
\end{flushleft}
```

제 1 절 COVINGTON.STY를 이용한 담화표상구조 (DRS) 그리기 123

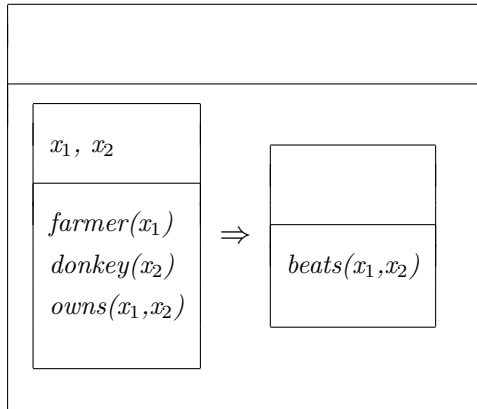
x_1, x_2
$man(x_1)$ $whistles(x_1)$ $dog(x_2)$ $follows(x_2, x_1)$

- 담화표상구조 (DRS)는 또 단독으로 나오지 않고 다른 DRS를 내포하고 있는 복잡한 구조를 가지고 있는 경우가 많은데, 이런 복잡한 담화표상구조 (DRS)를 그리기 위해서는 반복해서 `\drs{ }{ }`를 쓰면 된다.

```

\begin{flushleft}
\drs{~}
{\drs{x_1,~x_2}{farmer(x_1)\
donkey(x_2)\
owns(x_1,x_2)}~~
{\large $\rightarrow$}~
\drs{~}{beats(x_1,x_2)}
}
\end{flushleft}

```

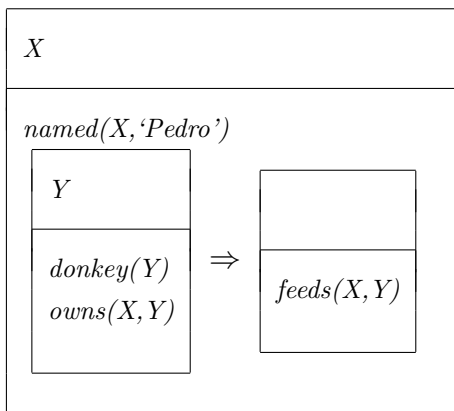


- 위에서 보는 바와 같이 담화변항 (discourse variable)을 상위의 DRS에서 계승하는 경우에는 $\backslash\text{drs}\{\sim\}\{\}$ 처럼 $\backslash\text{drs}$ 의 첫번째 논항 자리에 \sim 을 쓰면 된다.

```

\begin{flushleft}
\drs{X}
{named(X, 'Pedro') \\\
\drs{Y}\{donkey(Y)\\\
owns(X, Y)\}\sim\sim
{\large $\rightarrow$}\sim
\drs{\sim}\{feeds(X, Y)\}
}
\end{flushleft}

```



- 담화표상구조 (DRS)는 또한 예문을 동반하는 경우가 있는데, 이를 위해서는 특별히 $\backslash\text{sdrs}$ 라는 명령어를 쓰면 되는데, 이 $\backslash\text{sdrs}\{\}\{\}\{\}$ 는 세 개의 논항을 필요로 하는데 첫번째 논항 자리에 예문을 쓰고 두번째 논항에 담화변항, 세번째 위치에 조건들을 쓴다.

```

\begin{flushleft}
\sdrs{A donkey is green.}\{X}\{donkey(X)\\\
green(X)\}
\end{flushleft}

```

제 1 절 COVINGTON.STY를 이용한 담화표상구조 (DRS) 그리기 125

A donkey is green.

X
$donkey(X)$ $green(X)$

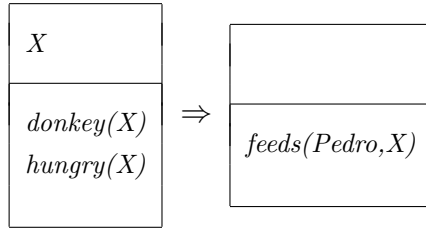
- 담화표상구조 (DRS)에는 부정기호 (\neg)를 쓰는 경우가 많다. `\negdrs`가 부정기호 (\neg)가 있는 담화표상구조 (negated DRS)를 그리는 명령어이다.

```
\begin{flushleft}
\negdrs{X}{donkey(X)\green(X)}
\end{flushleft}
```

X
\neg $donkey(X)$ $green(X)$

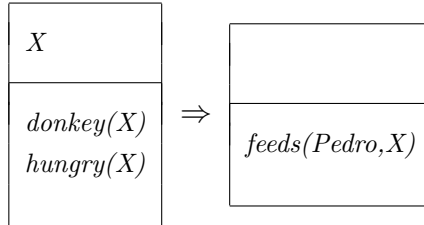
- 담화표상구조 (DRS)를 그리는 환경에서 화살표 등 수식에 등장하는 기호는 언제든지 필요하면 필요한 장소에 삽입할 수 있다. 그러나 \implies 는 특히 자주 쓰이기 때문에 하나의 독립된 명령어로 정의를 해 두었다. 그 명령어는 `\ifdrs`이고 총 4개의 논항을 요구하는데, 처음 두 개는 왼쪽 DRS를 위한 정보이고 마지막 두 개는 오른쪽 DRS를 위한 정보이다.

```
\begin{flushleft}
\ifdrs{X}{donkey(X)\hungry(X)}
{~}{feeds(Pedro,X)}
\end{flushleft}
```



- 간격 조정이 마음에 안들면 `\ifdrs` 명령어 대신 `\alifdrs`라는 명령어를 사용할 수 있다. `\ifdrs`를 쓰면 왼쪽 DRS가 약간 왼쪽으로 옮겨진 DRS를 얻을 수 있다.

```
\begin{flushleft}
\alifdrs{X}{donkey(X)\hungry(X)}
  {~}{feeds(Pedro,X)}
\end{flushleft}
```



제 2 절 Kamp and Reyle (1993)의 담화표상구조: `pstricks.sty`

Kamp and Reyle (1993)는 새로운 유형의 담화표상구조 (DRS)를 사용하고 있는데, 위에 적은 기존의 담화표상구조 (DRS)와는 어떻게 다른지 알아 보기 위하여 우선 예를 먼저 들고 작성하는 방법을 설명하겠다.

$x y$		u
farmer(x)	every	$u = y$
donkey(y)	x	x beats u
x owns y		

제 2 절 KAMP AND REYLE (1993)의 담화표상구조: PSTRICKS.STY127

위의 구조의 특징적인 점은 와 안에 수식이 온다는 점이다. 이런 DRS는 covington.sty에 정의되어 있는 명령어로는 불충분하다. 근본적인 이유는 를 구현하기 힘들다는 점이다. 이런 DRS를 그리기 위해 우리는 pstricks.sty을 사용한다.

- pstricks.sty을 사용하려면 초기화영역에 \usepackage{pstricks}이라고 선언해 두어야 한다. 이 선언이 유효하려면 MikTeX가 이 파일이 어디에 있는가 하는 정보를 Database에 가지고 있어야 한다 [MikTeX의 Refresh Filename Database 실행].
- 안에 수식이 들어 있는 담화표상구조를 그리는 방법은 \psframebox{ }의 { } 안에 수식을 넣으면 되는데, 이 경우 covington.sty의 방법과 다른 점은 dvips.exe를 실행시켜 *.dvi 파일을 *.ps로 바꾸고 난 다음에야 선이 보인다는 점이다.
- 안에 수식이 들어 있는 담화표상구조를 그리는 방법은 \psdiabox{ }의 { } 안에 수식을 넣으면 된다. 이후의 과정은 \psframebox{ }를 사용하는 방법과 동일하다.
- 이제 위의 구조를 생성하는데 사용한 코드를 제시하면 다음과 같다.

```

\psframebox{
  \psframebox{
    \begin{tabular}{c}
      $$ $y$\
      farmer$(x)$\
      donkey$(y)$\
      $$ owns $$
    \end{tabular}
  }
\psdiabox{
  \begin{tabular}{c}
    every\$$
  \end{tabular}
}
\psframebox{
  \begin{tabular}{c}
    $$\ $u = y$\ $$ beats $$
  \end{tabular}
}
}

```

- 의 크기는 [framesp=-3pt] 등의 옵션을 \psdiabox에 주면 된다. 다음은 각각 -1pt와 6pt를 값으로 준 경우의 예이다.

```
\psdiabox[framesep=-1pt]{ 철수 가 영희를 사랑한다 }
```

철수가 영희를 사랑한다

```
\psdiabox[framesep=6pt]{ 철수 가 영희를 사랑한다 }
```

철수가 영희를 사랑한다

제 9 장

한글 \LaTeX 로 국제음성문자 쓰기

국제 음성문자 (International Phonetic Alphabet; IPA)를 \LaTeX 에서 사용할 수 있도록 도와주는 꾸러미로 `wsuipa` (Washington State University International Phonetic Alphabet)와 `tipa` (\TeX International Phonetic Alphabet)가 비교적 널리 알려져 있으나, 두 꾸러미 중 `wsuipa`가 설치가 쉽고 다른 꾸러미와의 충돌이 적다는 경험에 비추어 `wsuipa`를 사용하도록 추천하고 싶다. 이 장에서는 이 꾸러미의 설치와 사용법을 설명한다.

제 1 절 꾸러미 소개: `ipa.sty`

- `wsuipa` (Washington State University International Phonetic Alphabet) 메뉴얼에 의하면 `wsuipa`는 다음과 같은 길을 걸어 온 것으로 되어 있다. 여기에서 소개하는 `wsuipa`는 1994년 7월 16일 배포된 $\text{\LaTeX}2_{\epsilon}$ 용 버전이다.

– adapted to \LaTeX by W. Antweiler, University of Köln,
05/28/90

- adapted by Andrej Brodnik (Andy) for NFSS, University of Waterloo, 11/01/93
- adapted by Andrej Brodnik and Denis Roegel for NFSS2, CRIN (Centre de Recherche en Informatique de Nancy), 16/07/94
- **wsuipa** (Washington State University International Phonetic Alphabet)를 \LaTeX 에서 사용하려면 글꼴과 관련된 다른 패키지와 마찬가지로 글꼴을 `c:\localtexmf\fonts\아래의` 정해진 디렉토리 아래에 폰트를 설치하는 일부러 시작해야 한다.
- “언어학자를 위한 \LaTeX ” Ver. 0.91 CD [2000년 11월 18일]에는 이를 자동으로 설치하도록 실행파일 안에 묶어 두었기 때문에 이 CD로 설치하는 사람은 글꼴의 설치에 대해서는 신경쓰지 않아도 된다.
- 그런 다음 여느 꾸러미와 마찬가지로 Mik \TeX 의 Filename Database를 실행시켜 미리 Mik \TeX 가 글꼴의 위치를 파악하고 있도록 해준다.
- 이 단계가 끝나면, 문서에 이 꾸러미를 사용하겠다고 초기화영역 (Preamble)에 다음과 같이 선언하고, 원본 파일에 **wsuipa**에서 해석할 수 있는 명령을 넣으면 된다.

```
\documentclass[a4paper, 11pt]{article}
\usepackage{ipa}
\begin{document}

\end{document}
```

제 2 절 WSUIPA 국제음성문자의 배열

wsuipa는 Pullum and Ladusaw (1986)의 내용을 가장 충실하게 따르고 있는 것으로 보인다. 다음 표는 **wsuipa**에서 정의되고 있는 명령과 Pullum and Ladusaw (1986)에서 명명한 IPA를 비교해 놓은 것이

다. L^AT_EX로 IPA를 쓰려면, 가운데 열의 명령을 원본 파일에 기입하고 컴파일하면 된다.

WSUIPA Characters			
Char Code	Char	IPAMACS Name	PULLUM & LADUSAW Name
'00	ɐ	\inva	turned a
'01	ɑ	\scripta	script a
'02	α	\nialpha	lowercase non-italic alpha
'03	ɒ	\invscripta	turned script a
'04	ʌ	\invv	inverted v
'05	ḅ	\crossb	crossed b
'06	Ḅ	\barb	barred b
'07	ḃ	\slashb	slashed b
'10	Ḇ	\hookb	hooktop b
'11	β	\nibeta	non-italic lowercase beta
'12	ç	\slashe	slashed c
'13	ç	\curlyc	curly-tail c
'14	Ĉ	\clickc	stretched c
'15	ċ	\crossd	crossed d
'16	Ċ	\bard	barred d
'17	ċ	\slashd	slashed d
'20	ċ	\hookd	hooktop d
'21	ḋ	\taild	right-tail d
'22	ḏ	\dz	d-yogh ligature
'23	ð	\eth	eth
'24	Ḑ	\scd	small capital D
'25	ə	\schwa	schwa
'26	ɶ	\er	right-hook schwa
'27	ɷ	\reve	reversed e
'30	ε	\niepsilon	non-italic greek epsilon
'31	ɷ	\reveysilon	reversed non-italic epsilon
'32	ɷ	\hookreveysilon	right-hook reversed non-italic epsilon
'33	ɷ	\closedreveysilon	closed reversed non-italic epsilon
'34	g	\scriptg	lowercase variant g
'35	ḡ	\hookg	hooktop g
'36	Ḡ	\scg	small capital G
'37	γ	\nigamma	non-italic gamma

WSUIPA Characters			
Char Code	Char	IPAMACS Name	PULLUM & LADUSAW Name
'40	ɣ	\ipagamma	IPA Gamma
'41	ɤ	\babygamma	baby gamma
'42	hv	\hv	h-v ligature
'43	ħ	\crossh	crossed h
'44	fi	\hookg	hooktop g
'45	fj	\hookheng	hooktop heng
'46	ɥ	\invh	turned h
'47	ï	\bari	barred i
'50	ɨ	\dlbari	barred dotless i
'51	ι	\niota	non-italic greek iota
'52	ɪ	\sci	small capital I
'53	ɨ	\barsci	barred small capital I
'54	ȷ	\invf	barred dotless j
'55	ł	\tildel	l with tilde
'56	ł̄	\barl	barred l
'57	ł̂	\latfric	belted l
'60	ł̸	\taill	l with right tail
'61	ɭ	\lz	l-yogh ligature
'62	λ	\nilambda	non-italic greek lambda
'63	λ̸	\crossnilambda	crossed lambda
'64	ɱ	\labdentalnas	m with leftward tail at right
'65	ɯ	\invm	turned m
'66	ɱ̄	\legm	turned m with long right leg
'67	ɱ̂	\nj	n with leftward hook at left
'70	ɱ̄	\eng	eng
'71	ɱ̸	\tailn	n with right tail
'72	ɱ̄	\scn	small capital N
'73	⊙	\clickb	bull's eye
'74	ø	\baro	barred o
'75	ɔ̟	\openo	open o
'76	ω	\niomega	non-italic lowercase greek omega
'77	ω̄	\closedniomega	closed omega

WSUIPA Characters			
Char Code	Char	IPAMACS Name	PULLUM & LADUSAW Name
'100	∞	\oo	double o
'101	Ⓟ	\barp	barred p
'102	Ⓟ	\thorn	thorn
'103	ϕ	\nipher	non-italic lowercase greek phi
'104	ʀ	\flapr	fish hook r
'105	ɾ	\legr	r with long leg
'106	ɽ	\tailr	r with right tail
'107	ɿ	\invr	turned r
'110	ɿ̣	\tailinvr	turned r with right tail
'111	ɿ̣	\invlegr	turned long-legged r
'112	℞	\scr	small capital R
'113	℞̣	\invscr	inverted small capital R
'114	ʂ	\tails	s with right tail
'115	ʃ	\esh	esh
'116	ʃ̣	\curlyesh	curly-tail esh
'117	σ	\nisigma	non-italic lowercase greek sigma
'120	ɸ	\tailt	t with right tail
'121	ʈ	\tesh	t-esh ligature
'122	ɸ̣	\clickt	turned t
'123	θ	\nitheta	non-italic lowercase greek theta
'124	Ⓢ	\baru	barred u
'125	Ⓢ̣	\slashu	slashed u
'126	υ	\niupsilon	non-italic lowercase greek upsilon
'127	℄	\scu	small capital U
'130	Ⓢ̣	\barscu	barred small capital U
'131	Ⓢ	\scriptv	script v
'132	Ⓢ̣	\invw	inverted w
'133	χ	\nichi	non-italic lowercase greek chi
'134	ʎ	\invy	turned y
'135	℄	\scy	small capital Y
'136	ʐ	\curlyz	curly-tail z
'137	ʐ̣	\tailz	z with right tail

WSUIPA Characters			
Char Code	Char	IPAMACS Name	PULLUM & LADUSAW Name
'140	ɜ	\yogh	yogh
'141	ɝ	\curlyyogh	curly-tail yogh
'142	ʔ	\glotstop	glottal stop
'143	ɥ	\revglotstop	reversed glottal stop
'144	ɸ	\invglotstop	inverted glottal stop
'145	ɛ̥	\ejective	ejective
'146	ɛ̦	\reveject	reversed ejective
'147	ɱ	\dental#1	subscript bridge
'150	ˈ	\stress	vertical stroke (superior)
'151	ˌ	\secstress	vertical stroke (inferior)
'152	ˑ	\syllabic	syllabicity mark
'153	˒	\corner	corner
'154	˘	\upt	IPA pointer
'155	˙	\downt	IPA pointer
'156	˚	\leftt	IPA pointer
'157	˛	\rightt	IPA pointer
'160	˙	\halflength	half-length mark
'161	ː	\length	length mark
'162	˚	\underdots	subscript umlaut
'163	˘	\ain	reversed apostrophe
'164	ˆ	\upp	pointer
'165	ˆ	\downp	pointer
'166	ˆ	\leftp	pointer
'167	ˆ	\rightp	pointer
'170	˚	\overring	over-ring
'171	˚	\underring	under-ring
'172	˚	\open	subscript left half-ring
'173	˜	\midtilde	superimposed (mid-) tilde
'174	˜	\undertilde	subscript tilde
'175	˘	\underwedge	subscript wedge
'176	˘	\polishhook	polish hook
'177	˘	\underarch	subscript arch

제 3 절 음성문자 입력 예

- 사용법은 다른 꾸러미와 다를 바가 없다. 일반적인 표기 규약에 따라 [와] 사이에 국제음성문자를 기입하면 된다. 이 때 문자는 `wsuipa`에서 정의한 명령어를 사용하면 된다. 다음은 몇 가지 코드와 예이다.

A shibilant is a term occasionally found for a fricative corresponding to a “hushing” sound, e.g., IPA [ʃ] (more technically, a grooved laminal fricative with a sign \int).

A shibilant is a term occasionally found for a fricative corresponding to a “hushing” sound, e.g., IPA [ʃ] (more technically, a grooved laminal fricative with a sign \int).

- 몇 가지 예를 더 들면 다음과 같다. `wsuipa`는 글꼴 소스인 `*.mf`, `*.tfm` 파일이 함께 배포되기 때문에 없는 글자는 \TeX 의 해당 프로그램이 알아서 스스로 만든다.

\dots which is the case in the aforementioned instances, however, \int we get the other retroflex consonants: [ɲ], [ɭ], [ɻ], and [ɽ]. They are printed also as [ɲ], [ɭ], [ɻ], and [ɽ].

... which is the case in the aforementioned instances, however, **we get the other retroflex consonants: [ɲ], [ɭ], [ɻ], and [ɽ]**. They are printed *also as [ɲ], [ɭ], [ɻ], and [ɽ]*.

제 10 장

한글 \LaTeX 로 그림 삽입하기

\LaTeX 로 그림을 그리는 작업은 매우 정교하고도 전문적인 작업이지만, 일반 사용자가 그 방법을 다 알고 사용하는 것은 많은 시간을 투자해야 하는 일임에 틀림없다. 다행히도 언어학자들은 그림을 전문적으로 그릴 필요는 없다. 이런 전제 하에 이 장에서는 그림을 그리는 데 사용되는 응용 프로그램을 이용해서 작성된 그림 파일을 \LaTeX 문서에 삽입하는 방법만을 설명하고자 한다.

제 1 절 꾸러미 소개: `graphics.sty`와 `graphicx.sty`

\LaTeX 는 어떤 파일이 그림 파일인지 아닌지를 구분하지 못한다. 오직 구분하는 것은 삽입되는 파일의 외곽 크기가 얼마나 큰가 하는 점이다. 어떤 그래픽 파일이든 삽입되기 위해서는 자기가 얼마 넓은 장소를 차지하는가를 \TeX 에게 알려 주어야 한다. 이러한 크기에 관한 정보가 그림 파일내에 있는 `Bounding Box`라는 것이다.

\TeX 를 위한 PostScript 드라이버가 나오고 나서 \LaTeX 에 그림 ‘파일을 삽입’하고 ‘그림의 크기를 조절’하고 ‘그림을 회전’시키거나 색

깔을 넣는 일이 가능해졌다. 그러나 $\text{\LaTeX}2$ 가 나오기 전까지만 해도 꾸러미 파일들이 각기 다른 드라이버를 가지고 있어서 불편한 점이 많이 있었다. 그러나 \LaTeX 가 나오고 나서 `graphics.sty`과 `color.sty`의 도움으로 모든 드라이버를 이 꾸러미들의 선택사항으로 선택하기만 하면 되어 불편이 해소되었다.

외부 응용 프로그램으로 작성된 그림 파일을 \LaTeX 문서에 삽입하는 데 사용되는 꾸러미는 두 가지가 가장 대표적으로 꼽힌다. 첫째는 기본적인 `graphics.sty`이고, 다른 하나는 조금 더 확장이 된 `graphicx.sty`이다. 이 두 꾸러미는 기능상에서는 유사하지만 명령을 넣는 방법이 조금 다를 뿐이다.

이 꾸러미를 초기화영역에 기입하고 다음 설명을 따라서 해 보면 그림 파일이 자유롭게 문서에 삽입된다는 것을 확인할 수 있다.

제 2 절 그림 삽입하기: 기본 명령

\LaTeX 가 등장할 때까지만 해도 현재와 같은 다양한 포맷의 그림 파일은 존재하지 않았다. Bounding Box에 크기에 관한 정보가 있는 그림파일은 \TeX 에 삽입될 수 있다고 보아도 될 것이다. 이 장에서는 `*.png` 파일과 `*.eps` 파일에 국한해서 설명하기로 한다.

1. 그림 원판 준비하기:

Paintshop Pro, Photoshop 등 전문 그래픽 파일 생성 프로그램으로 `*.png` 파일과 `*.eps` 파일을 만든다. `*.png` 파일은 PDF 파일을 위한 것이고, `*.eps` 파일은 dvips를 이용하여 `*.ps` 파일을 만들기 위한 것이다.

2. 그림 원판 삽입하기:

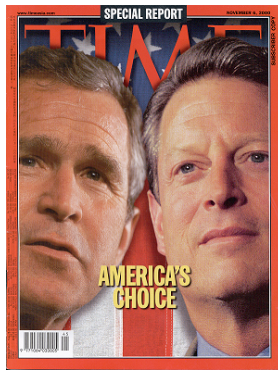
`*.tex` 파일에 그림이 들어 갈 자리에 다음과 같은 명령을 넣으면 그 자리에 그림이 삽입된다. 그림 파일은 \TeX 원본 파일과 동일한 디렉토리에 있는지 아니면 명령에 그림 파일이 있는 위치를 정확히 제시해야 한다.

```

\documentclass{article}
\usepackage{graphicx}
\begin{document}

\begin{center}
\includegraphics{pics/latex-graphics}
\end{center}
\end{document}

```



`\includegraphics` 명령어를 사용해서 여러 가지 형태로 그림을 원 관 파일에 삽입할 수 있다. 위에 제시한 코드는 `latex-graphics.eps` 파일이 현재 `*.tex` 파일의 아래 디렉토리인 `pics` 아래에 있는 경우의 코드이고 그 아래에 나오는 것은 조판 결과이다. 이 조판 결과는 `*.dvi` 파일에서는 볼 수 없고 `*.ps` 파일로 변환했을 경우에 볼 수 있다.

제 3 절 그림 삽입하기: 여러 가지 가능성

- 코드를 약간만 바꾸면 전체 그림의 크기가 반으로 축소된 크기로 그림을 삽입할 수 있다. `[scale=0.5]`가 그 명령어이다.

```

\begin{center}
\includegraphics[scale=0.5]{pics/latex-graphics}
\end{center}

```



- 같은 방법으로 너비를 1인치로 제한해서 그림을 삽입할 수 있다. 이 때 사용하는 명령어는 `[width=1in]`이다.

```
\begin{center}
\includegraphics[width=1in]{pics/latex-graphics}
\end{center}
```



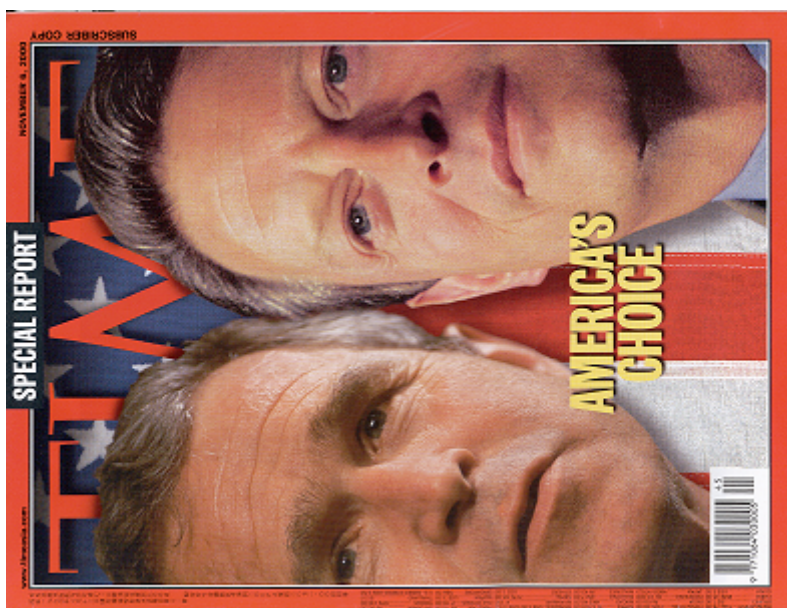
- 같은 방법으로 높이를 2.5 인치로 제한해서 그림을 삽입할 수도 있다. 이 때 사용하는 명령어는 `[height=2.5in]`이다.

```
\begin{center}
\includegraphics[height=2.5in]{pics/latex-graphics}
\end{center}
```



- `graphicx.sty` 꾸러미는 그림을 회전시켜 원문에 삽입할 수도 있다. 이 때 사용하는 명령어는 `[angle=90]`이다. `angle`=다음에 나오는 숫자는 임의의 각도를 가리킨다.

```
\begin{center}
\includegraphics [angle=90]{pics/latex-graphics}
\end{center}
```



- 축소해서 회전시켜 삽입할 수도 있다. `[angle=45,scale=0.40]`을 보면 무슨 의미인지 알 것으로 믿는다.

```
\begin{center}
\includegraphics[angle=45,scale=0.40]{pics/latex-graphics}
\end{center}
```



제 11 장

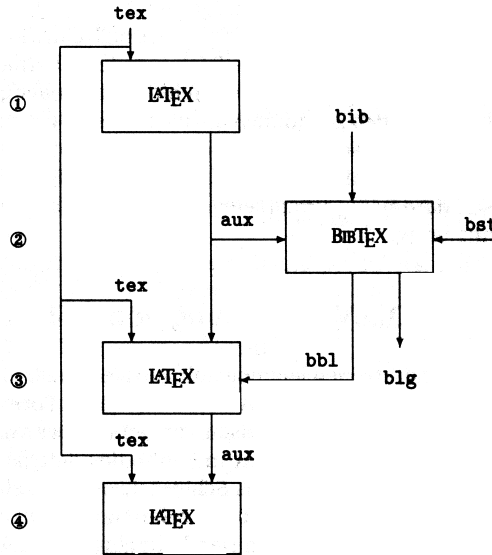
한글 \LaTeX 로 참고문헌을 자동으로 만들기

본 장에서는 `natbib.sty`과 관련 참고문헌 스타일 파일을 이용하여 참고문헌을 착오없이 정확하게 자동으로 만드는 방법을 간단하게 설명하고자 한다. 이 장에서 설명하는 방법은 최소한의 노력으로 $\text{BiB}\text{\TeX}$ 를 사용할 수 있도록 하는 목적으로 선택되었기 때문에 더 자세한 사항은 CTAN에서 내려 받을 수 있는 `natbib.sty`의 사용자 설명서나 다른 꾸러미에 대한 설명 혹은 $\text{BiB}\text{\TeX}$ 에 대한 일반적인 설명을 참고하기 바란다.

제 1 절 개념도

$\text{Bib}\text{\TeX}$ 와 \LaTeX 는 동시에 개발되지 않았지만, 아주 긴밀한 협동 작업을 통하여 참고문헌을 자동으로 검사하고 생성해 내는 임무를 완벽하게 수행해 낸다. 참고문헌이 만들어 지는 과정을 잘 이해하기 위해서는 $\text{Bib}\text{\TeX}$ 와 \LaTeX 가 어떻게 협동작업을 하는가를 살펴 볼 필요가 있다. $\text{Bib}\text{\TeX}$ 와 \LaTeX 의 협동 작업은 다음 개념도에 나타나는 바와 같이 크게 보아 네 단계로 나누어 살펴볼 수 있다. 우선 다음 개념도를 보고, 개념도 아래에 있는 설명을 읽어 보도록 한다. 개념도

에서는 각 단계가 ①, ②, ③, ④ 등으로 표시되어 있다.



1. 첫째 단계는 \LaTeX 를 돌려 $\backslash\text{cite}$ 로 표시가 되어 있는 참고문헌에 들어갈 인용 정보를 *.aux 파일에 기입해 놓는 단계;

2. 둘째 단계는 Bib \TeX 가 *.aux 파일에 있는 인용 정보를 보고 참고문헌 서지 사항들을 보아 놓은 파일인 *.bib에서 참고문헌 서지사항들을 추출해 *.bbl 파일에 기입해 놓고 다음 번 \LaTeX 를 이용하여 컴파일할 때 본문에 인용정보와 참고문헌 서지 정보를 기입할 준비를 하는 단계;

Bib \TeX 는 이 단계에서 어떤 스타일로 참고문헌 서지정보가 배열되어야 하는가를 *.bst파일에 정의되어 있는 명령을 보고 알고, 에러 메시지 등을 로그 파일인 *.blg에 남겨 놓는다.

3. 셋째 단계는 다시 \LaTeX 가 수행하는데, 이 세번째 단계는 \LaTeX 가 *.tex 파일을 컴파일하면서 Bib \TeX 가 만들어 놓은 *.bbl 파일이 존재한다는 것을 알고 그 파일 안에 있는 정보를 읽어 들이는 단계;

4. 넷째 단계는 \LaTeX 가 세 번째로 돌면서 본문에 인용 표시가 되어 있는 부분에 *.bib 파일에서 추출해 온 서지 정보가 담긴

*.bb1에 적혀 있는 참고문헌 서지 정보와 인용정보를 본문에 완벽하게 삽입하는 단계;

이 때 L^AT_EX는 인용은 되어 있으나 참고문헌 서지 정보에는 없는 사항을 체크하고 경고 메시지를 남기며, 한 저자의 동일 년도 저작이 두 번 이상 등장하면 일관성을 유지 하면서 “Ryu (1999a)”, “Ryu (1999b)”와 같이 “a”, “b” 구분을 삽입한다. 이런 이유로 인용을 하고도 참고문헌의 서지에 나오지 않는 책이나 논문이 없는 초보자적 저작을 피할 수 있는 도구를 제공하고, (1999a), (1999b)가 혼동되어 출처를 잘 못 밝히는 우를 범하지 않도록 도와 준다.

가 그것이다. 종합해서 말하면, BibT_EX를 인용과 함께 완벽하게 사용하려면 다음과 같은 세 가지 꾸러미 혹은 스타일이 정의된 파일이 필요하다.

- *.sty 파일: 다양한 형식의 인용정보의 처리 등 참고문헌의 서지 정보와 연동하는 방법을 정의한 파일
이 장에서는 natbib.sty을 소개한다.
- *.bst 파일: 참고문헌의 포맷을 정의한 파일
이 장에서는 “언어와 정보”의 참고문헌 다는법과 동일한 포맷의 참고문헌을 만들어 주면서 natbib.sty과 완벽하게 연동하는 언어정보학회용 ksli-natbib.bst을 소개한다.
- *.bib 파일: 참고문헌의 서지 정보가 들어 있는 파일
이 화일은 이미 bibliography.bib이라는 이름으로
c:\localtexmf\bibtex\bib\ 아래에 저자가 모아 놓은 서지 정보가 담긴 파일을 사용한다.

제 2 절 참고문헌을 자동생성을 위한 설정

- natbib.sty은 Patrick W. Daly가 만든 꾸러미이다. 이 꾸러미를 사용하기 위해서는 c:\localtexmf\tex\latex\ 아래에 두고, 초기화 영역에 다음과 같이 기입하면 된다.

```
\usepackage[authoryear,colon]{natbib}
```

- `ksli-natbib.bst`는 참고문헌의 포맷을 규정해 놓은 파일로 `c:\localtexmf\bibtex\bst\` 아래에 두고, `*.tex` 원본 파일에는 `\begin{document}` 다음 `\end{document}` 앞에 기입해 놓아야 한다.
- `natbib.sty`과 함께 쓰이는 가장 일반적인 `*.bst` 파일은 `plainnat.bst`이지만, 우리는 여기에서 한국어언어정보학회 학회지인 “언어와 정보”의 참고문헌 다는 방법에 근거해서 만든 `ksli-natbib.bst`를 사용하는 것으로 가정한다.

```
\begin{document}

본문
\citet{Ryu:96a} 등을 기입한다.

\bibliographystyle{ksli-natbib}
\bibliography{bibliography}
\end{document}
```

- `*.bib` 파일은 `c:\localtexmf\bibtex\bib` 아래에 두고, `*.tex` 원본 파일에서는 `*.bst` 바로 다음에 두는 것이 일반적이다. 위에서는 `*.bib`의 이름이 `bibliography.bib`이다.

이제 구체적으로 참고문헌을 만드는 방법을 순서대로 적으면 다음과 같다.

1. 본문을 작성하면서 인용 등을 `natbib.sty`에서 정하는 citation 방법에 따라 기입한다 [기입방법은 다음 절 참고].
2. \LaTeX 를 돌린다 [WinEdt에서는 \LaTeX 아이콘을 누른다.].
3. $\text{Bi}\text{\TeX}$ 를 돌린다 [WinEdt에서는 Bib... 아이콘을 누른다.].
4. \LaTeX 를 다시 한번 돌린다 [WinEdt에서는 \LaTeX 아이콘을 누른다.].

5. L^AT_EX를 돌린 결과 *.log 파일에 citation에 관한 경고 메시지가 있으면 다시 한번 L^AT_EX를 돌린다 [WinEdt에서는 L^AT_EX 아이콘을 누른다.].

제 3 절 natbib.sty를 사용해서 본문에 인용구 달기

- natbib.sty은 다음과 같은 사항을 선택할 수 있게 하고 있다. 선택사항은 다음과 같이 usepackage와 natbib 사이에 기입한다.

<code>\usepackage[authoryear,colon]{natbib}</code>	
round	round parentheses are used (default)
square	square brackets are used [option]
curly	curly braces are used option
angle	angle brackets are used [option]
colon	multiple citations separated by colon (default)
comma	separated by comma
authoryear	selects author-year citations (default)
numbers	selects numerical citations
super	numerical citations as superscripts
sort	sorts multiple citations according to order in ref. list
sort&compress	like sort, but also compresses numerical citations
longnamesfirst	makes first citation full author list
sectionbib	puts bibliography in a <code>\section*</code> instead of <code>\chapter*</code>

- 옵션으로 [authoryear]를 선택했을 때, 다음과 같은 명령은 다음과 같은 조판 결과를 출력한다.

```
\citet{Ryu:96a} ==>> Ryu (1996a)
\citep{Ryu:96a} ==>> (Ryu, 1996a)
```

- 옵션으로 [numbers]를 선택했을 때, 다음과 같은 명령은 다음과 같은 조판 결과를 출력한다.

```
\citet{Ryu:96a} ==>> Ryu [21]
```

```
\citep{Ryu:96a} ==>> [21]
```

- 다중인용인 경우 [authoryear] 모드와 [numbers] 모드에서는 각각 오른쪽과 왼쪽과 같은 결과를 조판결과로 출력한다.

```
\citep{Jones:90,Smith:89}
```

```
==>> (Jones et al., 1990; Smith, 1989) || [21,24]
```

```
or (Jones et al., 1990, 1991) || [21,24]
```

```
or (Jones et al., 1990a,b) || [21,24]
```

- `\cite{key}`는 author-year mode에서는 `\citet{key}`와 같고, numerical mode에서는 `\citep{key}`와 같은 결과를 낸다.
- `\citet*`나 `\citep*`를 쓰면 모든 저자를 모두 기입한 결과를 출력한다.

```
\citep*{key} ==>> (Jones, Baker, and Williams, 1990)
```

- 다음과 같은 citation 방법을 natbib.sty이 제공한다.

```
\citep[chap. 2]{key} ==>> (Jones et al., 1990, chap. 2)
```

```
\citep[e.g., ]{key} ==>> (e.g., Jones et al., 1990)
```

```
\citep[see][page 34]{key}==>> (see Jones et al., 1990, page 34)
```

```
\citealt{key} ==>> Jones et al. 1990
```

```
\citealt*{key} ==>> Jones, Baker, and Williams 1990
```

```
\citealp{key} ==>> Jones et al., 1990
```

```
\citealp*{key} ==>> Jones, Baker, and Williams, 1990
```

```
\citeauthor{key} ==>> Jones et al.
```

```
\citeauthor*{key} ==>> Jones, Baker, and Williams
```

```
\citeyear{key} ==>> 1990
```

```
\citeyearpar{key} ==>> (1990)
```

```
\citetext{priv. comm.} ==>> (priv. comm.)
```

- 소문자가 들어 있는 이름을 인용할 때 (예를 들어 di Sciullo를), 대문자로 (예를 들어 Di Sciullo로) 본문에 넣고 싶으면 `\Citet{key}`, `\Citep{key}`, `\Citeauthor{dRob98}` 명령을 쓰면 된다.

```
\Citet{dSciullo:98}          ==>> Di Sciullo (1998)
\Citep{dSciullo:98}         ==>> (Di Sciullo, 1998)
\Citeauthor{dSciullo:98}    ==>> Di Sciullo
```

- `\citeindextrue`를 옵션으로 주면, 다음 장에서 소개하게 될 인덱스를 자동으로 만드는 자료파일에 인용된 저자들이 자동으로 저장된다. 이 옵션을 원하지 않으면 `\citeindexfalse`라고 입력하면 된다. 인덱스를 만들기 위해서는 초기화 영역에 `\makeindex`이라고 선언해야 한다.
- 이상의 명령을 다음과 같이 본문에서 사용하면 그 아래에 있는 조판결과를 얻을 수 있다.

```
\begin{citation}
\Citet{vonStechow:Wunderlich:91}에서는 의미론이 주제별로
잘 개관되어 있고, \citet{Pullum:Ladusaw:86}는 언어학자의
시각으로 {\sc ipa}를 정리했고, \citet{Pollard:Sag:94}에서는
{\sc avm}이 사용되었고 \citep[see][Chap.1]{Pollard:Sag:94},
\citeauthor*{Kamp:Reyle:93}는 새로운 {\sc drs}를 도입하고
있다 \citep{Kamp:Reyle:93}.
```

Von Stechow and Wunderlich (1991)에서는 의미론이 주제별로 잘 개관되어 있고, Pullum and Ladusaw (1986)는 언어학자의 시각으로 IPA를 정리했고, Pollard and Sag (1994)에서는 AVM이 사용되었고 (see Pollard and Sag, 1994, Chap.1), Kamp and Reyle는 새로운 DRS를 도입하고 있다 (Kamp and Reyle, 1993).

제 4 절 참고문헌 서지정보가 담긴 데이터베이스 만들기

- 참고문헌의 서지정보는 *.tex의 원본파일과는 별도로 작성되

어 있어야 한다. 이러한 서지 정보는 개인이 계속해서 어느 논문이나 책을 쓰든지 재사용할 수 있다. 말 그대로 참고문헌 데이터베이스이기 때문이다.

- 참고문헌 서지정보 데이터 베이스의 기재항은 대략 13 개 정도의 기준으로 분류되어 작성되어 있어야 한다. WinEdt에서는 Insert>BibTeX Items 항목에 기입환경이 정의되어 있어서 서지정보 데이터베이스를 작성하기가 쉽게 되어 있다.

@article	저널 논문
@book	출판사와 출판년도가 기입된 저서
@booklet	출판사와 출판년도가 기입되지 않은 저작물
@inbook	책의 일부분에 들어 있는 장이나 절
@incollection	편자가 있는 책의 일부분으로 독자적인 제목이 있는 저작물
@inproceedings	논문집에 나온 논문
@manual	기술문서
@masterthesis	석사학위 논문
@phdthesis	박사학위 논문
@proceedings	학술대회 논문집
@techreport	대학이나 연구소에서 발행하는 일련번호가 있는 부정기 간행물
@unpublished	미출판 저서나 논문
@misc	기타

- 다음은 개인적으로 사용해 온 참고문헌 서지정보 데이터베이스에서 추출한 샘플이다.

```
@phdthesis{Abney:87,
  author = {Steven Abney},
  title = {The English Noun Phrase in Its Sentential Aspect},
  school = {Massachusetts Institute of Technology (MIT)},
  year = 1987,
}
@unpublished{Alsina:90,
  author = {Alex Alsina},
```

```
        title = {Where's the Mirror Principle?},
        note = {ms., Stanford University},
        year = 1990,
    }
    @article{Chomsky:80,
        author = {Noam Chomsky},
        title = {On Binding},
        journal = {Linguistic Inquiry},
        volume = 11,
        pages = {1--46},
        year = 1980,
    }
    @incollection{Fillmore:68,
        author = {Fillmore, Charles},
        title = {The Case for Case},
        booktitle = {Universals in Linguistic Theory},
        publisher = {Holt, Reinhart and Winston, New York},
        year = {1968},
        editor = {Emmon Bach and Robert Harms},
        pages = {1--88},
    }
    @techreport{Grimshaw:88,
        author = {Grimshaw, Jane},
        year = 1988,
        title = {Adjuncts and Argument Structure},
        type = {MIT Lexicon Project Working Paper},
        number = 21,
        institution = {Center for Cognitive Science},
        address = {MIT},
    }
    @book{Pollard:Sag:94,
        author = {Carl Pollard and Ivan A. Sag},
        title = {Head-Driven Phrase Structure Grammar},
        publisher = {University of Chicago Press},
        address = {Chicago},
        year = 1994,
    }
```

```

@book{GMS:94,
  author = {Michel Goossens and Frank Mittelbach and Alexander
            Samarin},
  title = {The  $\text{\LaTeX}$  Companion},
  publisher = {Addison-Wesley},
  year = 1994
}
@inproceedings{Abeille:Godard:94,
  address = {Stanford University},
  author = {Anne Abeill $\text{\`e}$  and Dani $\text{\`e}$ le Godard},
  booktitle = {Proceedings of the Thirteenth West Coast
               Conference on Formal Linguistics},
  editor = {Raul Aranovich and William Byrne and Susanne
            Preuss and Martha Senturia},
  address = {Cambridge},
  publisher = {CSLI Publications/SLA},
  title = {The Complementation of French Auxiliaries},
  volume = 13,
  year = 1994,
}

```

제 5 절 언어와 정보 참고문헌 포맷: ksli-natbib.bst

언어정보학회의 저널인 “언어와 정보”의 참고문헌 스타일(fullname.bst)에서 정하고 있는 참고문헌 형식 중 몇 가지를 소개하면 다음과 같다. “언어학자를 위한 \LaTeX ”와 함께 배포되는 ksli-natbib.bst는 그런 사항을 반영하고 있고, 그 아래있는 log파일이 보여주는 바와 같은 정의를 담고 있다.

- 전체적으로 번호를 쓰지 않고 참고문헌을 배열하되, 인덴트를 주어 시각적으로 보기 좋게 한다.
- 저자명은 “성, 이름” 순으로 기입한다. 한국인 저자는 콤마 없이 바로 붙여 쓴다.
- 년도는 논문 (혹은 책)과 저자 사이에 오는데, 각각 마침표로

구분된다.

- 논문은 따옴표없이 보통 글씨로 쓰되, 첫 자만 대문자로 하고 나머지는 소문자로 쓴다.
- 저널명은 이탤릭체로 하고 호수와는 콤마로 구분하며 페이지 번호와는 콜론으로 구분한다.
- 저널논문의 페이지 번호는 pages나 pp. 혹은 여백이 없이 바로 콜론에 붙여 쓴다.
-

이미 소개한 바와 같이 `\end{document}` 앞에 `\bibliographystyle{ksli-natbib}`이라고 원관 파일에 적고 `\usepackage{natbib}`과 함께 쓸 것을 추천한다.

```
\documentclass[11pt]{article}
\usepackage{natbib}

\begin{document}

본문
\citet{Ryu:96a} 등을 기입한다.

\bibliographystyle{ksli-natbib}
\bibliography{bibliography}
\end{document}
```

```
*****
* This is Make Bibliography Style *
*****
It makes up a docstrip batch job to produce
a customized .bst file for running with BibTeX
Do you want a description of the usage? (NO)
```

`\yn=`

Enter the name of the MASTER file (default=merlin.mbs)

\mfile=

Name of the final OUTPUT .bst file? (default extension=bst)

\ofile=li-natbib.bst

Give a comment line to include in the style file.

Something like for which journals it is applicable.

\ans=Journal of Language and Information, Korean Society for Language and
Information (KSLI), BiBT ϵ X style file written by Prof. Dr. B.-R. Ryu

(Öò£Éíµ³Éᵇ¼À -íá-,³Éᵇ¼½

Do you want verbose comments? (NO)

\yn=

(merlin.mbs

<<< For more information about the meanings of

<<< the various options, see the section on

<<< Menu Information in the .mbs file documentation.

EXTERNAL FILES:

Name of language definition file (default=merlin.mbs)

\cfile=

Name of language file: \cfile=.

Include file(s) for extra journal names? (NO)

\yn=

No included files.

INTERNAL LANGUAGE SUPPORT (if no external language file)

(*) English words used explicitly

(b) Babel (words replaced by commands defined in babelbst.tex)

Select:

\ans=

You have selected: English

STYLE OF CITATIONS:

- (*) Numerical as in standard LaTeX
- (a) Author-year with some non-standard interface
- (b) Alpha style, Jon90 or JWB90 for single or multiple authors
- (c) Alpha style, Jon90 even for multiple authors
- (f) Alpha style, Jones90 (full name of first author)
- (c) Cite key (special for listing contents of bib file)

Select:

\ans=a

You have selected: Author-year

AUTHOR--YEAR SUPPORT SYSTEM (if author-year citations)

- (*) Natbib for use with natbib v5.3 or later
- (o) Older Natbib without full authors citations
- (l) Apalike for use with apalike.sty
- (h) Harvard system with harvard.sty
- (a) Astronomy system with astron.sty
- (c) Chicago system with chicago.sty
- (n) Named system with named.sty
- (d) Author-date system with authordate1-4.sty

Select:

\ans=

You have selected: Natbib

LANGUAGE FIELD

- (*) No language field
- (l) Add language field to switch hyphenation patterns temporarily

Select:

\ans=

You have selected: No language field

ANNOTATIONS:

- (*) No annotations will be recognized
- (a) Annotations in annote field or in .tex file of citekey name

Select:

\ans=

You have selected: No annotations

ORDERING OF REFERENCES (if author-year citations)

- (*) Alphabetical by all authors
- (l) By label (Jones before Jones and James before Jones et al)
- (k) By label and cite key instead of label and title, as above
- (d) Year ordered and then by authors (for publication lists)
- (c) Citation order (unsorted, only meaningful for numericals)

Select:

\ans=l

You have selected: By label

ORDER ON VON PART (if not citation order)

- (*) Sort on von part (de la Maire before Defoe)
- (x) Sort without von part (de la Maire after Mahone)

Select:

\ans=x

You have selected: Sort without von part

AUTHOR NAMES:

- (*) Full, surname last (John Frederick Smith)
- (f) Full, surname first (Smith, John Frederick)
- (i) Initials + surname (J. F. Smith)
- (r) Surname + initials (Smith, J. F.)
- (s) Surname + dotless initials (Smith J F)
- (x) Surname + pure initials (Smith JF)
- (y) Surname + spaceless initials (Smith J.F.)
- (a) Only first name reversed, initials (AGU style: Smith, J. F., H. K. Jones)
- (b) First name reversed, with full names (Smith, John Fred, Harry Kab Jones)

Select:

\ans=b

You have selected: First name reversed, with full names

EDITOR NAMES IN COLLECTIONS (if author names reversed)

- (*) Editor names NOT reversed as edited by John James Smith
- (r) Editor names reversed just like authors'

Select:

\ans=

You have selected: Editor names NOT reversed

POSITION OF JUNIOR (if author names reversed)

(*) Junior comes last as Smith, John, Jr.

(m) Junior between as Smith, Jr., John

Select:

\ans=

You have selected: Junior comes last

JUNIOR PART IN THE CITATION (if author-year citations)

(*) No 'junior' part in the citations but in the ref listing

(j) 'Junior' in citations as well as in ref listing

Select:

\ans=

You have selected: No 'junior' part in the citations

PUNCTUATION BETWEEN AUTHOR NAMES:

(*) Author names separated by commas

(s) Names separated by semi-colon

(h) Names separated by slash /

Select:

\ans=

You have selected: Author names separated by commas

ADJACENT REFERENCES WITH REPEATED NAMES:

(*) Author/editor names always present

(d) Repeated author/editor names replaced by dash

Select:

\ans=

You have selected: Author/editor names always present

NUMBER OF AUTHORS:

(*) All authors included in listing

(l) Limited authors (et al replaces missing names)

Select:

\ans=

You have selected: All authors

TYPEFACE FOR AUTHORS IN LIST OF REFERENCES:

- (*) Normal font for author names
 - (s) Small caps authors (`\sc`)
 - (i) Italic authors (`\it` or `\em`)
 - (b) Bold authors (`\bf`)
 - (u) User defined author font (`\bibnamefont`)
- Select:

`\ans=`

You have selected: Normal font for author names

FONT OF CITATION LABELS IN TEXT (if author-year citations)

- (*) Cited authors plain as result of `\cite` command
 - (i) Cited authors italic
 - (s) Cited authors small caps
 - (b) Cited authors bold
 - (u) User defined citation font (`\citenamefont`)
- Select:

`\ans=`

You have selected: Cited authors plain

FONT OF EXTRA LABEL (The extra letter on the year)

- (*) Extra label plain
 - (i) Extra label italic
- Select:

`\ans=`

You have selected: Extra label plain

LABEL WHEN AUTHORS MISSING (if author-year citations)

- (*) Year blank when KEY replaces missing author (for natbib 7.0)
 - (y) Year included when KEY replaces missing author
- Select:

`\ans=`

You have selected: Year blank when KEY replaces missing author

MISSING DATE (if author-year citations)

- (*) Missing date set to `????` in label and text
- (b) Missing date left blank

Select:

\ans=

You have selected: Missing date set to ????

DATE POSITION:

- (*) Date at end
- (b) Date after authors
- (j) Date part of journal spec. (as 1994;45:34-40) else at end
- (e) Date at very end after any notes

Select:

\ans=b

You have selected: Date after authors

DATE FORMAT (if author-year citations)

- (*) Year plain without any brackets
- (p) Year in parentheses as (1993)
- (b) Year in brackets as [1993]
- (c) Year preceded by colon as ': 1993'
- (d) Year preceded by period as '. 1993'
- (m) Date preceded by comma as ', 1993'
- (s) Year preceded by space only, as ' 1993'

Select:

\ans=d

You have selected: Year preceded by period

INCLUDE MONTHS:

- (*) Date is year only without the month
- (m) Include month in date

Select:

\ans=

You have selected: Date is year only

DATE PUNCTUATION (if date not at end)

- (*) Date with standard block punctuation (comma or period)
- (c) Colon after date as 1994:
- (s) Semi-colon after date as 1994;
- (p) Period after date even when blocks use commas
- (x) No punct. after date

Select:

\ans=p

You have selected: Period after date

BLANK AFTER DATE:

(*) Space after date and punctuation

(x) No space after date as 1994:45

Select:

\ans=

You have selected: Space after date

DATE FONT:

(*) Date in normal font

(b) Date in bold face

Select:

\ans=

You have selected: Date in normal font

TRUNCATE YEAR (if author-year citations)

(*) Year text full as 1990--1993 or 'in press'

(t) Year truncated to last 4 digits

Select:

\ans=

You have selected: Year text full

TITLE OF ARTICLE:

(*) Title plain with no special font

(i) Title italic (\em)

(q) Title and punctuation in single quotes ('Title,' ..)

(d) Title and punctuation in double quotes (''Title,') ..)

(g) Title and punctuation in guillemets (<<Title,>> ..)

(x) Title in single quotes ('Title', ..)

(y) Title in double quotes (''Title'', ..)

(z) Title in guillemets (<<Title>>, ..)

Select:

\ans=

You have selected: Title plain

CAPITALIZATION OF ARTICLE TITLE:

(*) Sentence style (capitalize first word and those in braces)

(t) Title style (just as in bib entry)

Select:

\ans=

You have selected: Sentence style

ARTICLE TITLE PRESENT:

(*) Article title present in journals and proceedings

(x) No article title

Select:

\ans=

You have selected: Article title present

JOURNAL NAMES:

(*) Periods in journal names are retained, as 'Phys. Rev.'

(x) Dotless journal names as 'Phys Rev'

Select:

\ans=

You have selected: Periods in journal names

JOURNAL NAME FONT:

(*) Journal name italics

(r) Journal name normal font

Select:

\ans=

You have selected: Journal name italics

THESIS TITLE:

(*) Thesis titles like books

(a) Thesis title like article

Select:

\ans=

You have selected: Thesis titles like books

TECHNICAL REPORT TITLE:

(*) Tech. report title like articles

(b) Tech. report title like books

Select:

\ans=

You have selected: Tech. report title like articles

TECHNICAL REPORT NUMBER:

(*) Tech. report and number plain as 'Tech. Rep. 123'

(i) Tech. report and number italic as '{\it Tech. Rep. 123}'

Select:

\ans=i

You have selected: Tech. report and number italic

JOURNAL VOLUME:

(*) Volume plain as vol(num)

(i) Volume italic as {\em vol}(num)

(b) Volume bold as {\bf vol}(num)

(d) Volume and number bold as {\bf vol(num)}

Select:

\ans=

You have selected: Volume plain

JOURNAL VOL AND NUMBER:

(*) Journal vol(num) as 34(2)

(s) Journal vol (num) as 34 (2)

(c) Journal vol, num as 34, 2

(n) Journal vol, no. num as 34, no. 2

(h) Journal vol, \# number as 34, \#2

(b) Journal vol number as 34 2

(x) Journal vol, without number as 34

Select:

\ans=x

You have selected: Journal vol, without number

VOLUME PUNCTUATION:

(*) Volume with colon as vol(num):ppp

(s) Volume with colon and space as vol(num): ppp

(h) Volume with semi-colon as vol(num); ppp

(c) Volume with comma as vol(num), ppp

(b) Volume with blank as vol(num) ppp

Select:

\ans=

You have selected: Volume with colon

YEAR IN JOURNAL SPECIFICATION:

(*) Journal year like others as given by date position

(v) Journal vol(year) as 34(1995)

(s) Journal vol (year) as 34 (1995)

(p) Year with pages as 34(2), (1995) 1345--1387

(c) Year, comma, pages as 34(2), (1995), 1345--1387

Select:

\ans=

You have selected: Journal year like others

PAGE NUMBERS:

(*) Start and stop page numbers given

(f) Only start page number

Select:

\ans=

You have selected: Start and stop page numbers

LARGE PAGE NUMBERS:

(*) No separators for large page numbers

(c) Comma inserted over 9999 as 11,234

(s) Thin space inserted over 9999 as 11 234

(p) Period inserted over 9999 as 11.234

Select:

\ans=

You have selected: No separators for large page numbers

WORD 'PAGE' IN ARTICLES:

(*) Article pages numbers only as 234-256

(p) Include 'page' in articles as pp. 234--256

Select:

\ans=

You have selected: Article pages numbers only

POSITION OF PAGES:

(*) Pages given mid text as is normal

(e) Pages at end but before any notes

Select:

\ans=

You have selected: Pages given mid text

WORD 'VOLUME' IN ARTICLES:

(*) Article volume as number only as 21

(p) Include 'volume' in articles as vol. 21

Select:

\ans=

You have selected: Article volume as number only

NUMBER AND SERIES FOR COLLECTIONS:

(*) Allows number without series and suppresses word "number"

(s) Standard BibTeX as: "number 123 in Total Works";

error if number and no series

Select:

\ans=

You have selected: Allows number without series

POSITION OF NUMBER AND SERIES:

(*) After chapter and pages as in standard BibTeX

(t) Just before publisher or organization

Select:

\ans=

You have selected: After chapter and pages

JOURNAL NAME PUNCTUATION:

(*) Comma after journal name

(x) Space after journal name

Select:

\ans=

You have selected: Comma after journal

BOOK TITLE:

- (*) Book title italic (`\em`)
 - (p) Book title plain (no font command)
- Select:

`\ans=`

You have selected: Book title italic

PAGES IN BOOKS:

- (*) Pages in book plain as pp. 50-55
 - (p) Pages in book in parentheses as (pp. 50-55)
 - (x) Pages in book bare as 50-55
- Select:

`\ans=`

You have selected: Pages in book plain

TOTAL PAGES OF A BOOK:

- (*) Total book pages not printed
 - (p) For book: 345 pages or pp.
- Select:

`\ans=`

You have selected: Total book pages not printed

PUBLISHER ADDRESS:

- (*) Publisher, address as Harcourt, New York
 - (a) Address: Publisher as New York: Harcourt
- Select:

`\ans=`

You have selected: Publisher, address

PUBLISHER IN PARENTHESES:

- (*) Publisher as normal block without parentheses
 - (p) Publisher in parentheses
 - (d) Publisher and date in parentheses (Oxford, 1994)
 - (c) Publisher and date in parentheses, no comma (Oxford 1994)
 - (f) Publisher and date without parentheses Oxford, 1994
 - (k) Publisher and date, no parentheses, no comma Oxford 1994
- Select:

\ans=

You have selected: Publisher as normal block

PUBLISHER POSITION:

- (*) Publisher after chapter, pages
- (p) Publisher before chapter, pages
- (e) Publisher after edition

Select:

\ans=

You have selected: Publisher after chapter, pages

ISBN NUMBER:

- (*) Include ISBN for books, booklets, etc.
- (x) No ISBN

Select:

\ans=x

You have selected: No ISBN

ISSN NUMBER:

- (*) Include ISSN for periodicals
- (x) No ISSN

Select:

\ans=x

You have selected: No ISSN

‘EDITOR’ AFTER NAMES (EDITED BOOKS WITHOUT AUTHORS):

- (*) Word ‘editor’ after name
- (a) ‘Name (editor),’ in parentheses, after name, comma after
- (b) ‘Name (Editor),’ as above, editor upper case
- (c) ‘Name, (editor)’ in parentheses, after name, comma between
- (d) ‘Name, (Editor)’ as above, editor upper case
- (e) ‘Name (editor)’ in parentheses, after name, no commas
- (f) ‘Name (Editor)’ as above, editor upper case

Select:

\ans=

You have selected: Word ‘editor’ after name

EDITOR IN COLLECTIONS:

- (*) Same as for edited book (names before booktitle)
- (b) In booktitle, edited by .. (where .. is names)
- (p) In booktitle (edited by ..)
- (c) In booktitle, (edited by ..)
- (e) In booktitle, editor ..
- (f) In booktitle, (editor) ..
- (k) In booktitle (editor..)
- (g) In booktitle, (editor..)
- (j) In booktitle, .., editor
- (m) In booktitle (.., editor)

Select:

\ans=

You have selected: Same as for edited book

PUNCTUATION BETWEEN SECTIONS (BLOCKS):

- (*) \newblock after blocks (periods or new lines with openbib option)
- (c) Comma between blocks
- (s) Semi-colon between blocks
- (b) Blanks between blocks
- (t) Period after titles of articles, books, etc else commas
- (a) Period after titles of articles else commas

Select:

\ans=t

You have selected: Period after titles of articles, books, etc

PUNCTUATION BEFORE NOTES (if not using \newblock)

- (*) Notes have regular punctuation like all other blocks
- (p) Notes preceded by period

Select:

\ans=

You have selected: Notes have regular punctuation

PUNCTUATION AFTER AUTHORS:

- (*) Author block normal with regular block punctuation
- (c) Author block with colon

Select:

\ans=

You have selected: Author block normal

PUNCTUATION AFTER 'IN':

- (*) Space after 'in' for incollection or inproceedings
- (c) Colon after 'in' (as 'In: ...')
- (i) Italic 'in' and space
- (d) Italic 'in' and colon
- (x) No word 'in' for edited works

Select:

\ans=

You have selected: Space after 'in'

'IN' WITH JOURNAL NAMES (if using 'in' with collections)

- (*) No 'in' before journal name
- (i) Add 'in' before journal name in style for incollection

Select:

\ans=

You have selected: No 'in' before journal name

FINAL PUNCTUATION:

- (*) Period at very end of the listed reference
- (x) No period at end

Select:

\ans=

You have selected: Period at very end

ABBREVIATE WORD 'PAGES' (if not using external language file)

- (*) 'Page(s)' (no abbreviation)
- (a) 'Page' abbreviated as p. or pp.
- (x) 'Page' omitted

Select:

\ans=

You have selected: 'Page(s)'

ABBREVIATE WORD 'EDITORS':

- (*) 'Editor(s)' (no abbreviation)
- (a) 'Editor' abbreviated as ed. or eds.

Select:

\ans=

You have selected: 'Editor(s)'

OTHER ABBREVIATIONS:

(*) No abbreviations of volume, edition, chapter, etc

(a) Abbreviations of such words

Select:

\ans=a

You have selected: Abbreviations

ABBREVIATION FOR 'EDITION' (if abbreviating words)

(*) 'Edition' abbreviated as 'edn'

(a) 'Edition' abbreviated as 'ed'

Select:

\ans=a

You have selected: 'Edition' abbreviated as 'ed'

MONTHS WITH DOTS:

(*) Months with dots as Jan.

(x) Months without dots as Feb Mar

Select:

\ans=

You have selected: Months with dots

EDITION NUMBERS:

(*) Editions as in database saving much processing memory

(w) Write out editions as first, second, third, etc

(n) Numerical editions as 1st, 2nd, 3rd, etc

Select:

\ans=n

You have selected: Numerical editions

Reading external language file \cfile=

STORED JOURNAL NAMES:

(*) Full journal names for prestored journals

(a) Abbreviated journal names

(s) Abbreviated with astronomy shorthands like ApJ and AJ

Select:

\ans=

You have selected: Full journal names

AMPERSAND:

(*) Use word 'and' in author lists

(a) Use ampersand in place of 'and'

(v) Use \BIBand in place of 'and'

Select:

\ans=

You have selected: Use word 'and'

COMMA BEFORE 'AND':

(*) Comma before 'and' as 'Tom, Dick, and Harry'

(n) No comma before 'and' as 'Tom, Dick and Harry'

(c) Comma even with 2 authors as 'Tom, and Harry'

Select:

\ans=

You have selected: Comma before 'and'

NO 'AND' IN REFERENCE LIST:

(*) With 'and' before last author in reference list

(x) No 'and' as 'Tom, Dick, Harry'

Select:

\ans=

You have selected: With 'and'

FONT OF 'ET AL':

(*) Plain et al

(i) Italic et al

(r) Roman et al even when authors something else

Select:

\ans=i

You have selected: Italic et al

ADDITIONAL REVTeX DATA FIELDS:

(*) No additional fields for REVTeX
(r) Include REVTeX data fields collaboration, eid, eprint, archive, numpages, url
Select:

\ans=
You have selected: No additional fields

E-PRINT DATA FIELD: (without REVTeX fields)

(*) Do not include eprint field
(e) Include eprint and archive fields for electronic publications
Select:

\ans=
You have selected: Do not include eprint field

URL ADDRESS: (without REVTeX fields)

(*) No URL for electronic (Internet) documents
(u) Include URL as regular item block
(n) URL as note
(l) URL on new line after rest of reference
Select:

\ans=
You have selected: No URL

REFERENCE COMPONENT TAGS:

(*) No reference component tags in the \bibitem entries
(b) Reference component tags like \bibinfo in the content of \bibitem
Select:

\ans=
You have selected: No reference component tags

EMPHASIS: (affects all so-called italics)

(*) Use emphasis ie, \em, allows font switching
(i) Use true italics ie, \it, absolute italics
(x) No italics at all
(u) Underlining in place of italics, best with ulem package
Select:

\ans=

```

You have selected: Use emphasis

NEW FONT SELECTION SCHEME:
(*) No NFSS use 2.09 font commands
(n) NFSS use \textbf, \emph, not \bf, \em
  Select:

\ans=
  You have selected: No NFSS
)

Finished!!
Batch job written to file 'li-natbib.dbj'
Shall I now run this batch job? (NO)

\yn=j

```

제 6 절 Bib \TeX 를 사용하지 않고 수동으로 참고문헌 목록 만들기

\LaTeX 의 장점을 제대로 사용할 수 없어서 추천하고 싶지 않은 방법이지만

covington.sty에서 제공하는 `\begin{reflist}`와 `\end{reflist}` 환경을 이용해서 참고문헌을 만드는 방법도 있다.

- 우선 covington.sty을 초기화 영역에 기입하고, 참고문헌이 들어 갈 자리에 `\begin{reflist}`로 시작해서 `\end{reflist}`로 끝나는 환경을 설정한다.

```
\documentclass[11pt]{article}
\usepackage{hangul}
\usepackage{covinton}

\begin{document}

....(본문)...

\begin{reflist}
...참고문헌...
\end{reflist}
\end{document}
```

- `\begin{reflist}`로 시작해서 `\end{reflist}`로 끝나는 환경에 일반적인 문단이 있는 글을 쓰는 것처럼 쓰면 자동으로 LSA 스타일의 참고문헌이 생성된다. 다음은 코드와 조판결과이다.

```
\begin{reflist}
Barton, G. Edward; Berwick, Robert C.; and Ristad, Eric Sven.
1987. Computational complexity and natural language.
Cambridge, Massachusetts: MIT Press.

Chomsky, Noam. 1965. Aspects of the theory of syntax.
Cambridge, Massachusetts: MIT Press.

Covington, Michael. 1993. Natural language processing
for Prolog programmers. Englewood Cliffs, New Jersey:
Prentice--Hall.
\end{reflist}
```

Barton, G. Edward; Berwick, Robert C.; and Ristad, Eric Sven.
1987. Computational complexity and natural language.
Cambridge, Massachusetts: MIT Press.

Chomsky, Noam. 1965. Aspects of the theory of syntax. Cam-
bridge, Massachusetts: MIT Press.

Covington, Michael A. 1993. Natural-language processing for Prolog programmers. Englewood Cliffs, New Jersey: Prentice-Hall.

제 12 장

한글 \LaTeX 로 자동으로 색인 만들기

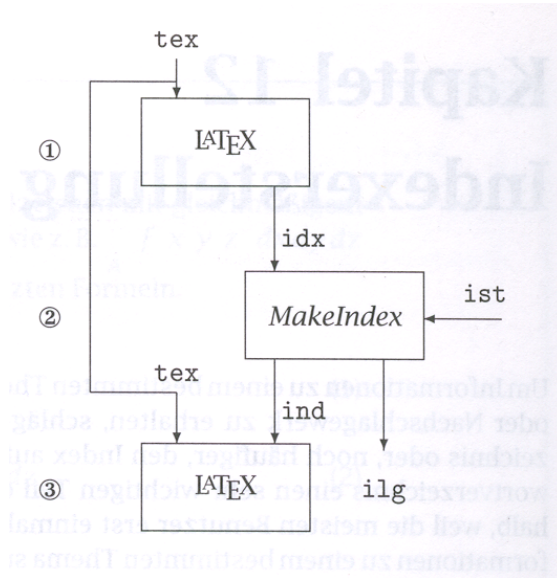
본 장에서는 `index.sty`를 이용하여 색인을 자동으로 만드는 방법을 간단하게 설명하고자 한다. 자세한 사항은 CTAN에서 내려 받을 수 있는 `index.sty`의 사용자 설명서를 보기 바란다.

제 1 절 개념도

`index.sty`를 사용하여 색인을 자동으로 생성하는 작업은 다음 그림과 같이 세 단계로 나누어 설명하는 것이 이해가 쉽다고 생각된다.

1. 첫째, `index.sty`가 초기화 영역 (Preamble)에 선언된 명령에 따라 색인 작업에 필요한 자료를 파일에 담아 놓는 단계;
2. 둘째, `index.sty`가 본문에서 추출한 색인 자료를 `MakeIndex`라는 프로그램을 이용하여 \LaTeX 문서에 삽입될 수 있는 형태로 가공하는 단계;
3. 셋째, `MakeIndex`라는 프로그램이 순서대로 가공해 놓은 자료를 본문의 지정된 위치에 삽입하는 단계

가 그것이다. 위와 같은 단계는 다음 개념도에서 확인해 보면 훨씬 이해가 쉬울 것으로 생각한다. 개념도에서는 위의 각 단계가 ①, ②, ③으로 표시되어 있다.



1. 첫째 단계가 끝나면 `index.sty`의 명령에 맞게 초기화 영역 (Preamble)에 선언된 명령에 따라 `*.adx`, `*.ldx`, `*.ndx`, `*.idx` 등 색인 작업에 필요한 자료를 파일이 생성된다.
2. 위에서 생성된 `*.adx`, `*.ldx`, `*.ndx`, `*.idx` 등의 파일을 `MakeIndex`를 이용하여 `LATEX` 문서에 삽입될 수 있는 형태로 가공하면 `*.and`, `*.lnd`, `*.nnd`, `*.ind` 등의 파일이 새로이 생성된다. 이때 `MakeIndex`는 `gind.ist`, `gglo.ist` 등 인덱스 생성용 스타일 파일을 참조하고 로그 파일로 `*.ilg`를 남겨 놓는다.
3. 셋째 단계로 `MakeIndex`라는 프로그램이 가공해 놓은 `*.and`, `*.lnd`, `*.nnd`, `*.ind` 등의 파일을 `index.sty`가 `\printindex[aut]`, `\printindex[not]`, `\printindex[list]`, `\printindex` 등이 기입된 자리로 읽어 와 본문에 삽입하게 된다.

제 2 절 색인을 생성하기 위한 문서의 구조

색인을 자동으로 생성하기 위해서는 다음과 같은 절차를 따라야 한다.

- `index.sty`를 `c:\localtexmf\tex\latex` 아래에 두어야 하고 이 파일이 그곳에 있다는 것을 \TeX 가 미리 알고 있어야 하기 때문에 \TeX 의 Refresh Filename Database를 먼저 실행시켜야 한다.
- 그런 다음 `\usepackage{index}`를 초기화 영역(Preamble)에 선언해야 하고, 이어서 역시 `\begin{document}`가 시작되기 전 초기화 영역(Preamble)에 다음과 같이 적어야 한다.

```
\makeindex
\newindex{aut}{adx}{and}{Author Index}
\newindex{not}{ndx}{nnd}{Index of Notations}
\newindex{list}{ldx}{lnd}{Subject Index}

\proofmodefalse
\def\adx{\index[aut]}
\def\idx{\index[not]}
\def\tdx{\index[list]}
```

- 위의 명령에서 `\newindex`의 마지막 논항으로 나온 `Author Index` 등이 각 인덱스가 출력될 때 제목으로 붙는 말이다.
- 그런 다음 `\begin{document}` 다음에 오는 본문에서 색인에 넣고 싶은 단어를 `\adx{}`, `\ndx{}`, `\ldx{}` 등의 `{ }` 안에 기입한다.
`\adx{}`는 이름 색인용으로, `\ndx{}`는 표기법 색인용으로, `\ldx{}`는 주제어 색인용 등으로 역할 분담을 시켜 놓고 색인을 동시에 세 개 생성해 내는 것이 일석삼조로 일을 처리하는 방법이다.
- 그런 다음 `\end{document}`가 나오기 전 색인을 넣고 싶은 곳에 다음과 같은 명령을 기입하고 다시 \LaTeX 로 컴파일하면 된다. 페이지 수가 정확하게 맞으려면 \LaTeX 를 두 번 이상 컴파일시키는 것이 좋다.

```

\printindex[aut]
\printindex[not]
\printindex[list]
\printindex

```

- 위의 그림에서 ①과 ③ 단계는 L^AT_EX가 자동으로 처리하나 MakeIndex가 처리해야 하는 단계인 ② 단계는 각각 다음과 같은 명령을 도스창에서 처리해 주어야 한다. 이 절차가 약간 불편하게 생각될 수 있는 절차이나, 사람이 몇 백 페이지에 달하는 책의 색인을 만드는 것과 비교하면 불편하다고 생각하는 것 자체가 사치임에 틀림없고, 또 L^AT_EX가 최종적으로 만들어 낸 색인의 질을 보면 이 절차는 즐거운 일로 간주되어도 된다.¹
- *.adx 파일로부터 *.and 파일을 만들기 위해서는, 파일이름이 sample.tex를 컴파일해서 나온 sample.adx 파일이라면
c:\texmf\miktex\bin\makeindex.exe -o sample.and sample.adx라고 도스창에서 처리해야 한다. 이것을 배치 파일로 만들어 두면 편리하다.
- 마찬가지로 sample.ldx를 sample.lnd로 만들기 위해서는
c:\texmf\miktex\bin\makeindex.exe -o sample.lnd sample.ldx라고 도스창에서 처리해야 한다. 이것도 역시 배치 파일로 만들어 두면 편리하다.
- 마찬가지로 sample.ndx를 sample.nnd로 만들기 위해서는
c:\texmf\miktex\bin\makeindex.exe -o sample.nnd sample.ndx라고 도스창에서 처리해야 한다. 이것도 역시 배치 파일로 만들어 두면 편리하다.
- MakeIndex가 남긴 로그파일은 대강 다음과 같은 내용을 담고 있다.

This is MAKEIN~1, portable version 2.12 [26-May-1993]

¹WinEdt에 있는 MakeIndex 아이콘을 이용하여 다양한 확장자로 된 색인 자료 파일을 처리하려면 매번 옵션을 바꿔 주어야 하거나 Popup Menu 자체를 에디팅해야 하기 때문에 초보자로서는 부담되는 일이다. 여기에서는 단순한 방법인 DOS창에서 작업하는 방법 혹은 sample-adx.bat 등을 이용하는 방법을 권한다.

```
(MiKTeX 1.20e).
Scanning input file latex4ling-leaflet-091.ldx....done
(5 entries accepted, 0 rejected).
Sorting entries....done (10 comparisons).
Generating output file latex4ling-leaflet-091.lnd...
done (19 lines written, 0 warnings).
Output written in latex4ling-leaflet-091.lnd.
Transcript written in latex4ling-leaflet-091.ilg.
```

지금까지 설명한 파일의 구조를 소스로 나타내면 다음과 같다.

```
\documentclass{book}
\usepackage{hangul,index}
\makeindex
\newindex{aut}{adx}{and}{Author Index}
\newindex{not}{idx}{nnd}{Index of Notations}
\newindex{list}{ldx}{lnd}{Subject Index}

\proofmodefalse
\def\adx{\index[aut]}
\def\idx{\index[not]}
\def\tdx{\index[list]}

\begin{document}
... 본문 ...
\printindex[aut]
\printindex[not]
\printindex[list]
\printindex
\end{document}
```

제 3 절 본문에 색인 표제어 표시하기

본문에 색인 항목을 표시해 두는 일은 색인 작업의 기본이다. 초기화 영역에서 정의한 바대로 명령을 써서 색인 항목을 표시해 주어야 한다. 우리는 다음과 같은 색인 정의를 하고 있다고 가정한다.

```
\def\adx{\index[aut]}
\def\idx{\index[not]}
\def\tdx{\index[list]}
```

따라서 우리는 `\adx{색인어1}`, `\idx{색인어2}`, `\tdx{색인어3}` 등으로 본문을 쓰면서 기입해 두면 된다. 다음은 Sag and Wasow (1999)에 나오는 한 구절이다.

```
\adx{Sag}\adx{Wasow} \Citet{Sag:Wasow:99} write as follows:\\
Although the boundaries are not sharp,
'syntax' contrasts with \tdx{semantics} 'semantics'
(the study of linguistic meaning),
\tdx{morphology} 'morphology' (the study of word structure),
and \tdx{phonology} 'phonology' (the study of the sound
patterns of language) in ways that 'grammar' does not.
\tdx{grammar!descriptive|)}
\tdx{grammar!prescriptive|)}
```

Sag and Wasow (1999) write as follows:

Although the boundaries are not sharp, 'syntax' contrasts with 'semantics' (the study of linguistic meaning), 'morphology' (the study of word structure), and 'phonology' (the study of the sound patterns of language) in ways that 'grammar' does not.

제 4 절 색인 위치 지정

색인은 일반적으로 참고문헌 다음에 나오지만, 이 장의 경우처럼 예시 목적으로는 본문의 어디에도 나올 수 있다. 색인이 어느 곳에 나오게 하느냐는 색인이 출력되게 하는 명령을 어디에 기입했느냐는 문제와 동일하다. 다음 예는 색인이 나오는 위치를 일반적인 경우처럼 참고문헌 다음에 문서가 끝나기 전에 나오도록 명령인

```
\printindex[aut]%Author Index
\printindex[not] %Index of Notations
```

```
\printindex[list] %Subject Index
\printindex %The Index
```

을 넣은 경우의 예이다.

```
\bibliographystyle{ksli-natbib}
\bibliography{bibliography} %Literatur
%
\printindex[aut]%Author Index
\printindex[not] %Index of Notations
\printindex[list] %Subject Index
\printindex %The Index
\end{document}
```

지금 이 장치럼 바로 다음 쪽에 인덱스가 나오게 하려면 바로 이 문단 다음에 다음과 같은 인덱스 출력 명령어를 쓰면 된다.

```
\printindex[aut]%Author Index
\printindex[not] %Index of Notations
\printindex[list] %Subject Index
\printindex %The Index
```


































제 13 장







한글 \LaTeX 로 유인물 만들기

이 장에서는 논문발표시 많이 쓰이는 유인물 (handout)을 만드는 방법을 간단히 소개하고자 한다. Handout을 만들기 편하도록 명령을 모아 놓은 꾸러미는 없지만, 일반적으로 저자가 사용하는 형식을 여기에 소개하고자 한다. Handout에 가끔 쓰이는 특수 기호를 구현하기 위해 `pifont.sty`을 설치해서 만들어 보고자 한다.

제 1 절 특수문자 꾸러미: `pifont.sty`

Windows 98/2000 등에서 볼 수 있는 Dingbat 글꼴을 \LaTeX 에서 사용할 수 있게 해주는 꾸러미가 `pifont.sty`이다. 이 꾸러미를 이용해 우리는 다음 표에 나와 있는 것과 같은 특수문자 등을 문서에 쓸 수 있다.

<code>\ding{33}</code>		<code>\ding{34}</code>		<code>\ding{35}</code>	
<code>\ding{36}</code>		<code>\ding{37}</code>		<code>\ding{38}</code>	
<code>\ding{39}</code>		<code>\ding{40}</code>		<code>\ding{41}</code>	
<code>\ding{42}</code>		<code>\ding{43}</code>		<code>\ding{44}</code>	
...
<code>\ding{100}</code>		<code>\ding{101}</code>		<code>\ding{102}</code>	
<code>\ding{103}</code>		<code>\ding{104}</code>		<code>\ding{105}</code>	
<code>\ding{106}</code>		<code>\ding{107}</code>		<code>\ding{108}</code>	
<code>\ding{109}</code>		<code>\ding{110}</code>		<code>\ding{111}</code>	
<code>\ding{112}</code>		<code>\ding{113}</code>		<code>\ding{114}</code>	
<code>\ding{115}</code>		<code>\ding{116}</code>		<code>\ding{117}</code>	
<code>\ding{118}</code>		<code>\ding{119}</code>		<code>\ding{120}</code>	
...

<code>\ding{172}</code>	①	<code>\ding{173}</code>	②	<code>\ding{174}</code>	③
<code>\ding{175}</code>	④	<code>\ding{176}</code>	⑤	<code>\ding{177}</code>	⑥
<code>\ding{178}</code>	⑦	<code>\ding{179}</code>	⑧	<code>\ding{180}</code>	⑨
<code>\ding{181}</code>	⑩	<code>\ding{182}</code>	⑪	<code>\ding{183}</code>	⑫
<code>\ding{184}</code>	⑬	<code>\ding{184}</code>	⑭	<code>\ding{185}</code>	⑮
...
<code>\ding{246}</code>		<code>\ding{247}</code>		<code>\ding{248}</code>	
<code>\ding{249}</code>		<code>\ding{250}</code>		<code>\ding{251}</code>	

제 2 절 구성 코드

`pifont.sty`를 이용해서 큰 단위의 아이콘을 만들고 나머지는 `itemize` 환경을 이용하여 소단위의 아이콘을 만든다. `itemize` 환경은 계속해서 내포시킬 수 있다.

```
\begin{itemize}
\item 작은 단위
\item 작은 단위
  \begin{itemize}
```



```

\item 더 작은 단위
\item 더 작은 단위
\end{itemize}

```

```

\item 작은 단위
\end{itemize}

```

다음에 적은 예는 참고용이다.

```

\NeedsTeXFormat{LaTeX2e}[1996/12/01]
\documentclass[a4paper, 12pt]{article}
\usepackage{hangul}
\usepackage{pifont}
\begin{document}
\usefont{H}{mj}{m}{n} % 명조
\newpage
\null
\noindent
한국언어정보학회 논문 발표\
2000년 6월 17일 (토), 태우재단

```

```

\begin{center}%
{\LARGE\bf
{\rmfamily\bfseries 한국어 공간 지식의 어휘확에 관한 \
개념적/어휘 의미론적 분석}
\par}%
\vskip 1.5em%
\lineskip .5em%
%
\begin{tabular}[t]{c}%
{\bf 류병택}\}[0.5em]
{\small 충남대학교 언어학과}\}[0.5em]
{\small\tt ryu@cnu.ac.kr}\}
{\small\tt http://www.cnu.ac.kr/\sim$ryu/}
\end{tabular}
\vskip 1em%
\end{center}%
\par
\vskip 1.5em
\pagestyle{plain}

\section{Introduction}
\noindent \ding{110} { } {\bf Motivations}:
\begin{itemize}
\item
“‘Semantic encoding (of dimensional adjectives) in the
lexicon could be biologically fixed, hence universal.’”
[\citet[p.3]{Bierwisch:67b}]

\item
“‘The dimensions languages pick out are far from arbitrary:
They appear to be just those dimensions the human perceptual
apparatus is tuned to pick out.’”
[\citet[p.534]{Clark:Clark:77}]

\item
“‘All languages seem to have a certain sample of lexical
items to make reference to spatial dimensions such as height,
length, width, distance, etc.’” [\citet[p.1]{Lang:94b}]

```

위의 예는 다음과 같이 조판되어 나올 것이다.

한국언어정보학회 논문발표
2000년 6월 17일 (토), 대우재단

한국어 공간 지식의 어휘화에 관한 개념적/어휘 의미론적 분석

류병래

충남대학교 언어학과

ryu@cnu.ac.kr

<http://www.cnu.ac.kr/~ryu/>

제 1 절 Introduction

■ Motivations:

- “Semantic encoding (of dimensional adjectives) in the lexicon could be biologically fixed, hence universal.” [Bierwisch (1967, p.3)]
- “The dimensions languages pick out are far from arbitrary: They appear to be just those dimensions the human perceptual apparatus is tuned to pick out.” [Clark and Clark (1981, p.534)]
- “All languages seem to have a certain sample of lexical items to make reference to spatial dimensions such as height, length, width, distance, etc.” [Lang (1994, p.1)]

제 2 절 2쪽을 1쪽으로 축소하기: pstops

Handout은 일반적으로 두 쪽을 한 면에 인쇄를 하는 것이 일반적이다. 이런 포맷을 우리는 2up이라고 말하는데, *.ps 파일을 우선 만들고 이 *.ps를 다시 pstops.exe를 이용해서 두 쪽을 한 면에 인쇄되는 새로운 *.ps를 만드는 것이 주요 골자이다.

pstops.exe는 명령이 복잡한 것처럼 보이는데, 이 명령도 배치 파일로 만들어 두고 *.ps 파일의 이름만 바꾸면 편리하다.

다음이 두 쪽을 한 면에 나타내는 *.ps 파일을 만드는 pstops.exe의 옵션이다.

```
pstops.exe 2:0L@.7(21cm,0)+1L@.7(21cm,14.85cm) in.ps out.ps
```

Happy TeXing!

부록 A

한글 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 시스템 및 관련 지원 프로그램 설치하기

제 1 절 $\text{MikT}_{\text{E}}\text{X}$ 설치하기

1.1 $\text{MikT}_{\text{E}}\text{X}$ 란 무엇인가

- $\text{MikT}_{\text{E}}\text{X}$ 는 Windows용 무료 $\text{T}_{\text{E}}\text{X}$ 배포 프로그램입니다.
- URL은 <http://www.miktex.org/>입니다.
- 저자는 독일의 Christian Schenk (cs@miktex.org)입니다.
- 2000년 10월 1일 현재 가장 최근의 공식적인 $\text{MikT}_{\text{E}}\text{X}$ 배포판의 버전은 1.20e (one-twenty-e)이고, Comprehensive TeX Archive Network (CTAN)의 [systems/win32/miktex/1.20](http://www.ctan.org/systems/win32/miktex/1.20)에서 내려 받을 수 있습니다. 2000년 10월 20일 현재 $\text{MikT}_{\text{E}}\text{X}2.0$ 버전의 Beta 6가 (<http://www.miktex.org/beta.html>) 배포되고 있습니다. 일반적으로 beta 버전은 버그가 많이 포함되어 있을 수 있기 때문에 여기서는 공식버전인 $\text{MikT}_{\text{E}}\text{X}1.20\text{e}$ 버전을 설치하는 방법을 설명합니다.
- 포함되어 있는 프로그램들은 다음과 같습니다.

- TeX 3.14159: The traditional TeX compiler
- Yap (Yet Another Previewer) 0.97: A tool to view TeX output
- METAFONT 2.718: Font compiler
- LaTeX2e <99/06/01> Macro package: Standard LaTeX packages
- Babel, psnfss, ...
- Dvips 5.86: Converts TeX output into PostScript
- Dvipdfm 0.12.6e: Converts TeX output into PDF
- pdfTeX 0.14d: A variant of the TeX compiler which produces PDF
- e-TeX 2.1: A feature-enhanced version of TeX
- Omega 1.8: A 16-bit-enhanced version of TeX
- METAPOST 0.641: A tool to produce PostScript figures
- Computer Modern PostScript Font: Standard TeX fonts in Type1 format
- AMSFonts PostScript Fonts: Math fonts in Type1 format
- MakeIndex 2.12: A tool to produce indices
- BibTeX 0.99c: A tool to produce bibliographies
- DVICopy: A tool to manipulate TeX output files
- psutils (psselect, pstops, ...): Tools to manipulate PostScript files

1.2 MikTeX 설치 방법

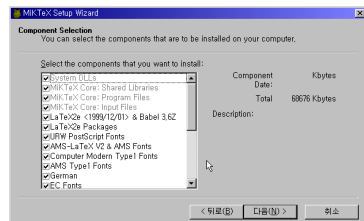
1. 언어학자를 위한 L^AT_EX CD를 CD-ROM Drive에 넣는다. 여기에는 1.20e를 설치하는 방법을 설명합니다. 2.0-beta 버전을 설치하는 방법은 `setupwiz.exe`를 실행해서 설치하는 방식은 동일하지만, `setupwiz.exe`가 실행되고 나서 해야 하는 조치가 약간 다를 수 있습니다.

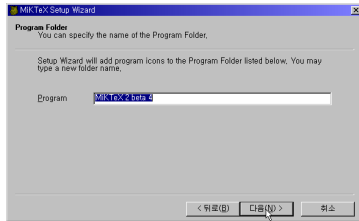
2. miktex\1.20e를 연다.
3. 아래의 왼쪽 그림과 같이 setupwiz.exe를 실행시킨다.
4. 오른쪽 그림과 같은 화면이 나타나면 환영 메시지가므로 바로 다음을 클릭한다.

이름	크기	종류	변환 날짜
latex.lst	19KB	LST 파일	99-12-17 오후 1:54
latex.zip	1,201KB	WinZip File	99-12-17 오후 1:54
latexgpl.lst	1KB	LST 파일	99-09-12 오전 12:00
latexgpl.zip	13KB	WinZip File	99-09-12 오전 12:00
latexplg.lst	68KB	LST 파일	99-12-17 오후 1:54
latexplg.zip	3,289KB	WinZip File	99-12-17 오후 1:54
lcore.lst	2KB	LST 파일	99-12-17 오후 1:54
lcore.zip	990KB	WinZip File	99-12-17 오후 1:54
lfonts.lst	7KB	LST 파일	99-04-01 오전 12:00
lfonts.zip	466KB	WinZip File	99-04-01 오전 12:00
metapost.lst	4KB	LST 파일	99-04-01 오전 12:00
metapost.zip	768KB	WinZip File	99-04-01 오전 12:00
msdll.lst	1KB	LST 파일	99-07-09 오전 12:00
msdll.zip	699KB	WinZip File	99-07-09 오전 12:00
news.html	29KB	HTML Do...	99-12-17 오후 1:54
omega.lst	16KB	LST 파일	99-04-01 오전 12:00
omega.zip	1,209KB	WinZip File	99-04-01 오전 12:00
pdfdtx.lst	3KB	LST 파일	99-12-17 오후 1:54
pdfdtx.zip	672KB	WinZip File	99-12-17 오후 1:54
psutils.lst	3KB	LST 파일	99-04-01 오전 12:00
psutils.zip	948KB	WinZip File	99-04-01 오전 12:00
setupwiz.exe	304KB	응용 프로그...	99-12-17 오후 1:54
tesinfo.lst	2KB	LST 파일	99-12-17 오후 1:54
tesinfo.zip	769KB	WinZip File	99-12-17 오후 1:54
news.txt	69B	LST 파일	99-04-13 오전 12:00
news.zip	1,380KB	WinZip File	99-04-13 오전 12:00
wcore.lst	7KB	LST 파일	99-12-17 오후 1:54
wcore.zip	1,709KB	WinZip File	99-12-17 오후 1:54
womega.lst	1KB	LST 파일	99-04-01 오전 12:00
womega.zip	325KB	WinZip File	99-04-01 오전 12:00
wandy.lst	43KB	LST 파일	99-04-01 오전 12:00
wandy.zip	1,040KB	WinZip File	99-04-01 오전 12:00



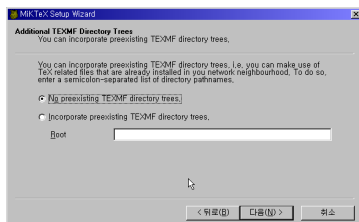
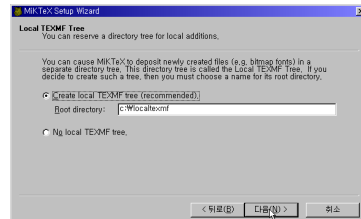
5. 아래 왼쪽의 그림처럼 Component Selection 창이 뜨면, 모두 설치한다고 체크하고 다음을 클릭한다.
6. 아래 오른쪽 화면처럼 Program Folder를 묻는 화면이 뜨면 기본적으로 제공하는 이름을 그대로 두고 (이 경우 1.20e 버전에서는 “MikTeX” 이고 2.0 beta버전에서는 “MikTeX 2 beta 6” 이다) 다음을 클릭한다.





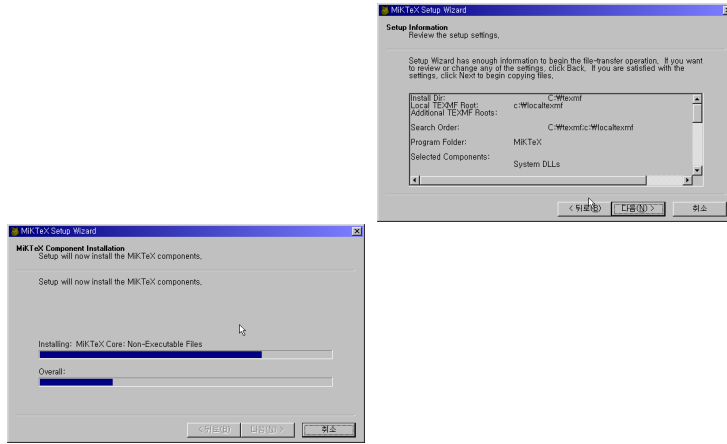
7. 그 다음에는 아래 왼쪽 그림처럼 Local TEXMF를 어떻게 설정하는지 묻는 화면이 뜨는데, “Creat local TEXMF tree [recommended]”를 선택하고 다음을 클릭한다.

8. 그 다음에는 아래 오른쪽 그림처럼 Additional TEXMF Trees를 설정할지 여부를 묻는 화면이 뜨는데, “No Preexisting TEXMF trees”를 선택하고 다음을 클릭한다.

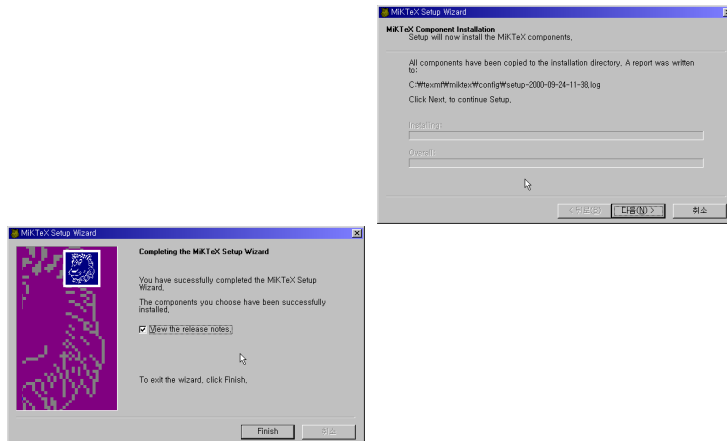


9. 다음에는 아래 왼쪽의 그림처럼 지금까지 선택한 Setup Information을 보여 주는데, 이의가 없으면 바로 다음을 클릭한다.

10. 다음을 클릭하자마자 아래 오른쪽의 그림처럼 MikTeX Component Installation이라는 화면과 함께 두 개의 상태창이 뜨면서 파일들을 하드에 복사하고 있는 상태를 보여준다.



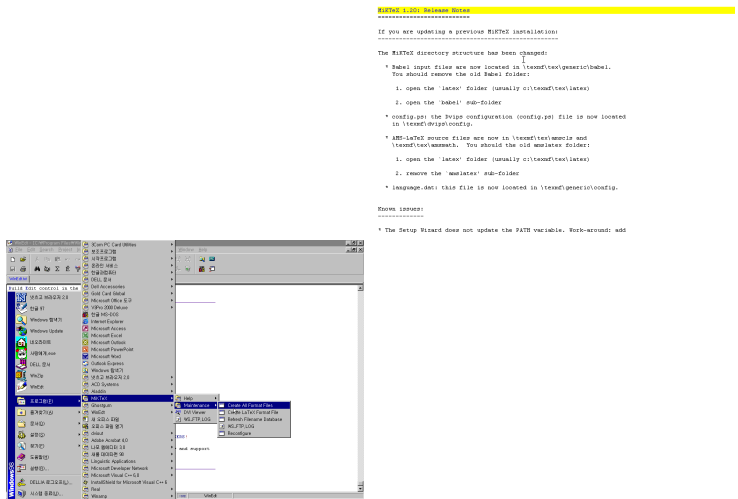
11. MikTeX Component Installation이라는 화면에 파일들이 모두 복사되고 난 상태에서, “다음” 단추를 누를 수 있도록 단추가 보인다. 이 때 “다음”을 클릭한다.
12. 아래 오른쪽의 그림처럼 Completing MikTeX Setup Wizard 화면이 뜨고 “View the release notes”라는 체크박스가 나타나는 데 이 옵션을 선택하고 “마침”이라는 단추를 누른다.



13. 마지막으로 다음 왼쪽 그림과 같은 텍스트 파일이 열리는데, 이 파일의 내용을 (대개는 읽지 않지만) 처음부터 읽어 본 후 창을 닫는다.
14. 이렇게 하면 일단은 MikTeX가 c:\texmf라는 새로운 디렉토

리에 설치되었다. 그러나 MikTeX를 사용하기까지는 다음 오른쪽 그림처럼

[시작] > [프로그램] > [MikTeX] > [Maintenance]를 가서 “Creat All Format Files”를 선택하고 Format File이 만들어지고 나면, “Refresh Filename Database”를 실행시킨다. 특히 “Refresh Filename Database”는 “Ghostscript”나 “GSView” 그리고 “한글 \LaTeX ”를 설치하고 난 다음에도 항상 반복해서 실행시켜야 한다.



제 2 절 Ghostscript 설치하기

2.1 Ghostscript란 무엇인가

- Ghostscript는 Laser Printers에 사용되는 PostScript Page Description Language에 대한 Interpreter입니다. Ghostscript를 이용하면 PostScript를 보여줄 수 없는 컴퓨터에서도 Postscript를 읽고 보여 줄 수 있게 만들어 주고, 프린터가 PostScript를 지원하지 않아도 프린팅할 수 있게 해 줍니다.
- URL은 <http://www.cs.wisc.edu/~ghost/gsview/>입니다.
- Ghostscript는 Aladdin Enterprises팀에 의해 작성되었으며, Al-

addin Enterprises가 저작권을 가지고 있습니다. 따라서 shareware도 아니고 public domain에 있지도 않습니다.

- Ghostscript는 3 가지 종류의 라이선스로 배포되는데, “Aladdin Ghostscript”라고 되어 있는 Aladdin Ghostscript 버전은 Aladdin Free Public License에 준해 배포됩니다. Aladdin Free Public License에 의하면 자유롭게 사용하고, 복사하고 배포할 수 있지만, 상업적으로 배포할 수는 없습니다.
- 2000년 10월 1일 현재 가장 최근의 공식적인 Ghostscript 배포판의 버전은 6.01 (2000년 3월 17일)이고, <http://www.cs.wisc.edu/~ghost/aladdin/ghostscript/>에서 내려 받을 수 있습니다.
- 포함되어 있는 프로그램들과 그 기능은 다음과 같습니다.
 - An interpreter for the PostScriptTM language and the Adobe Portable Document Format (PDF – sometimes confused with Acrobat, Adobe’s PDF browser and editor product);
 - Input modules (utilities) for reading a variety of raster formats, including CMYK, GIF, JFIF (JPEG), MIFF, PBM/PGM/PPM, and PCX;
 - Output modules (drivers) for a wide variety of window systems (including X Windows and Microsoft Windows), raster file formats, and printers, as well as a PDF output module that implements functions similar to Adobe’s AcrobatTM DistillerTM product; and
 - The Ghostscript library, a set of procedures to implement the graphics and filtering capabilities that are primitive operations in the PostScript language and in PDF.

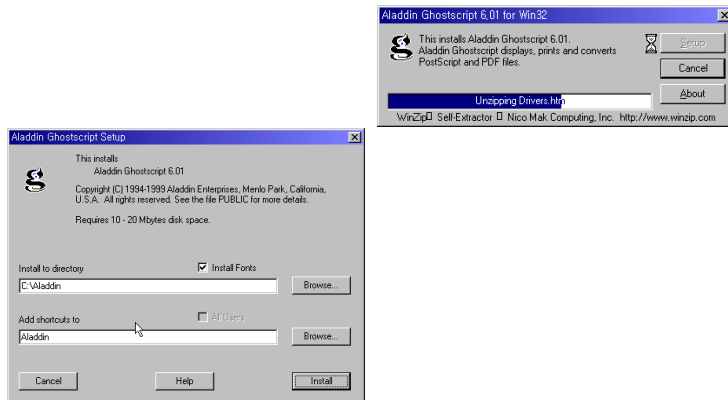
2.2 Ghostscript 설치방법

1. 언어학자를 위한 L^AT_EX CD를 CD-ROM Drive에 넣는다.
2. `support\gs601w32`를 연다.

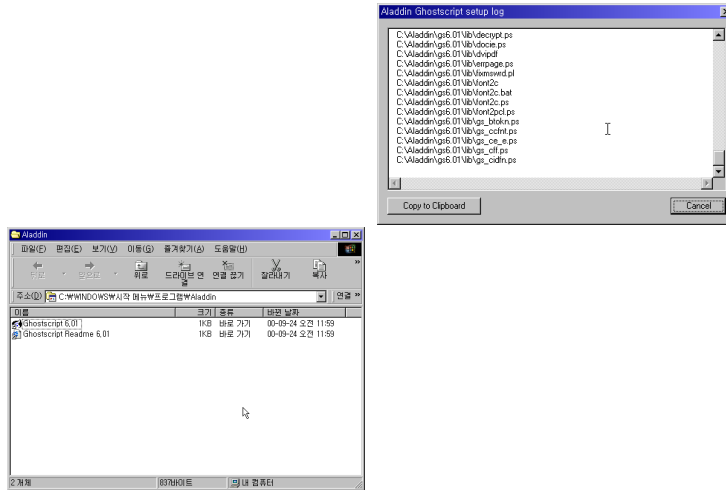
3. 아래 왼쪽 그림과 같이 gs601w32.exe를 실행시킨다.
4. 아래 오른쪽 그림과 같은 화면이 뜨면 “Setup” 단추를 누른다.



5. 아래 왼쪽 그림처럼 Setup 프로그램이 압축 파일을 자동으로 푼다.
6. 그 다음에는 아래 오른쪽 그림처럼 Setup 환경 변수를 묻는데, 그림처럼 “Install fonts”를 체크하고 (다른 폴더를 지정해도 되지만) c:\Aladdin이라는 폴더를 지정하고 프로그램 명칭을 “Aladdin”이라고 주고 “Install” 단추를 누른다.



7. “Install” 단추를 누르면 아래 왼쪽 그림처럼 Setup 로그 파일을 보여주는 창이 뜨고 이를 통해 자기의 하드 디스크에 Ghostscript가 설치되고 있음을 알 수 있다.
8. 그 다음에는 “시작” > “프로그램”에 아래 오른쪽 그림과 같은 단축 아이콘이 생성되었음을 보여주고 설치작업을 종료된다.



9. 그 다음에는 GSView를 다음 절의 안내에 따라 설치한다.

제 3 절 GSView 설치하기

3.1 GSView란 무엇인가

- GSView는 A Graphical Interface for Displaying, Printing, and Converting Postscript and PDF Files
- URL은 <http://www.cs.wisc.edu/~ghost/gsview/>입니다.
- GSView는 Ghostgum Software Pty Ltd의 Russell Lang에 의해 작성되었으며, Ghostgum Software Pty Ltd.가 저작권을 가지고 있습니다. 따라서 shareware도 아니고 public domain에 있지도 않습니다.
- GSView는 Aladdin Free Public License에 준해 배포됩니다. Aladdin Free Public License에 의하면 자유롭게 사용하고, 복사하고 배포할 수 있지만, 상업적으로 배포할 수는 없습니다. Aladdin Free Public Licence에 의하면 사용자는 프로그램을 사용한 댓가로 지불할 필요는 없지만, GSView 개발, 유지, 보수

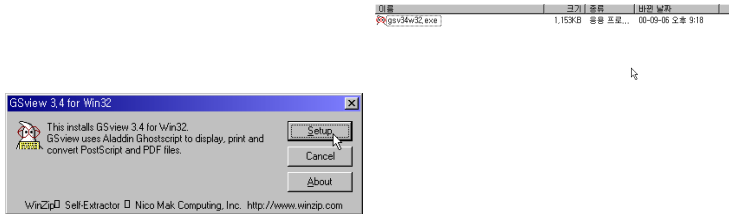
지원을 위해 등록할 수 있습니다. 현재 등록비는 AUD \$40이고, <http://www.ghostgum.com.au/>에서 온라인 등록이 가능하다.

- 2000년 10월 1일 현재 가장 최근의 공식적인 GSView 배포판의 버전은 3.4 (2000년 7월 29일)이고, <http://www.cs.wisc.edu/~ghost/gsview/get34.html>에서 내려 받을 수 있습니다.
- 포함되어 있는 프로그램들과 그 기능은 다음과 같습니다.

- Display and print PostScript and PDF files.
- View pages in arbitrary order (Next, Previous, Goto).
- Page size and Orientation are automatically selected from DSC comments or can be selected using the menu.
- Print selected pages using Ghostscript.
- Convert pages to bitmap, PDF or PostScript.
- Selectable display resolution, depth, alpha.
- Single button zoom.
- Extract selected pages to another file.
- Copy display bitmap to clipboard.
- Save clipboard bitmap as BMP file.
- Add bitmap or user preview to EPS file (Interchange, TIFF or Windows Metafile)
- Graphically select and show bounding box for EPS file.
- Extract bitmap preview or PostScript from DOS EPS file.
- Extract text or search for text.
- Can read gzip and bzip2 compressed PostScript and PDF files.
- On-line help.
- Win32 and OS/2 executables. Older versions support Win16.

3.2 GSView 설치 방법

1. 언어학자를 위한 L^AT_EX CD를 CD-ROM Drive에 넣는다.
2. support\gsv34w32를 연다.
3. 아래 왼쪽의 그림과 같이 gsv34w32.exe를 실행시킨다.
4. 아래 오른쪽의 그림과 같이 초기 화면이 나타나면 “Setup”을 클릭한다.

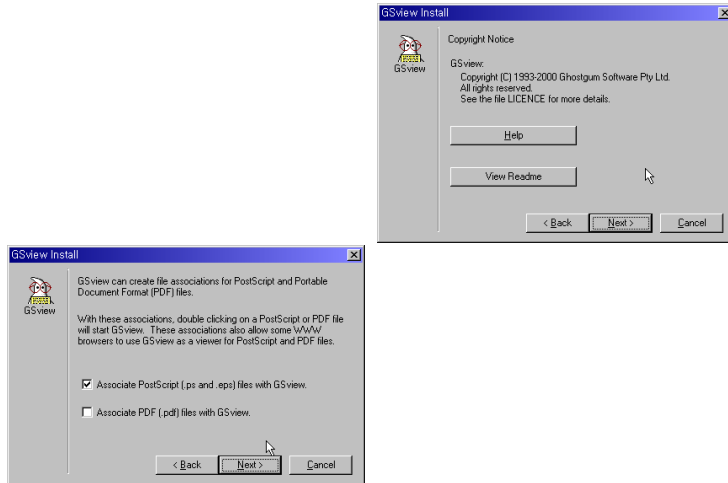


5. 그 다음에는 아래 왼쪽 그림처럼 “Select Languages” 화면이 뜨는데, 원하는 언어를 선택하면 아래 오른쪽 화면으로 넘어간다.
6. 그 다음에는 아래 오른쪽 그림과 같이, 3M의 하드 디스크를 차지할 것이라는 메시지와 함께 설치 안내가 나오는데 이 때 “Next”를 클릭한다.

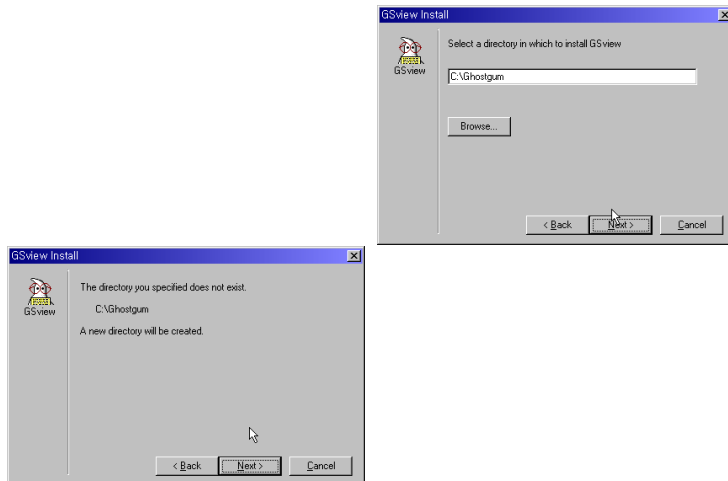


7. 그 다음에는 아래 왼쪽 그림과 같이 Copyright Notice가 나오는데, 한번쯤 감사의 마음을 갖고 “Next”를 클릭한다.
8. 그 다음에는 어느 어느 파일 유형을 GSView로 볼 것인지를 묻는데, Acrobat Reader를 설치했거나 설치할 예정이면 .ps와 .eps파일을 GSView로 보겠다는 첫번째 체크박스만을 선택하고 “Next”를 누른다. 만약 Acrobat Reader를 설치하지 않을 계획이면 둘 다 체크하고 “Next”를 클릭해야 하나, Acrobat

Reader를 설치하고 그 프로그램으로 .pdf 파일을 보는 것이 좋다고 추천하고 싶다.

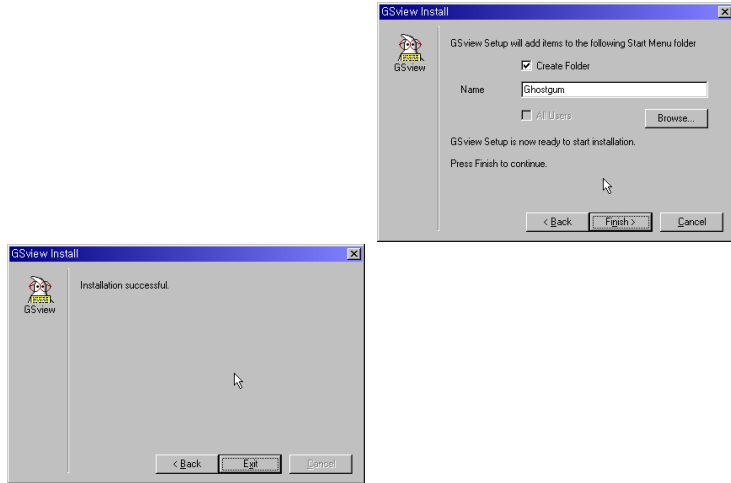


9. 그 다음에는 어느 디렉토리에 GSView를 설치할 것인가를 묻는데, 아래 왼쪽 그림처럼 (사실은 다른 디렉토리도 상관없지만 추천한대로) c:\ghostgum이라고 쓰고 “Next”를 클릭한다.
10. 그러면 c:\ghostgum 디렉토리가 새로 생성됨을 알리는 화면이 뜨는데, 이 때 “Next”를 클릭한다.

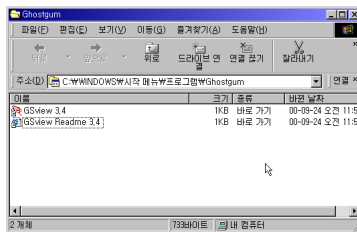


11. 그 다음에는 메뉴폴더의 명칭을 묻는데, 아래 왼쪽 그림처럼 자동으로 추천한 대로 두고 “Finish”를 클릭한다.

12. 그러면 아래 오른쪽 그림처럼 설치가 끝났음을 알리는 화면이 뜬다.



13. 그리고 아래 그림처럼, 메뉴폴더가 생성되었음을 보여주는 화면이 뜨면 설치는 끝난 것이다.



제 4 절 Winedt 설치하기

4.1 Winedt란 무엇인가

- Winedt는 A Powerful, Extremely Flexible and Versatile Native Win32 Text Editor and Shell for $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ Documents
- URL은 <http://www.winedt.com/>입니다.
- Winedt는 Alexander Simonic 박사에 의해 작성되었으며, shareware (try-before-you-buy software)입니다. 평가를 위해 31일 동

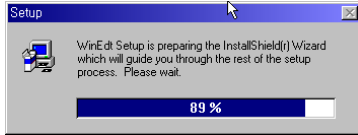
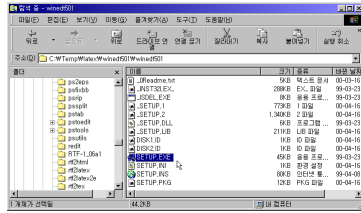
안 지불하지 않고 사용할 수 있으나, 그 이후는 The Registration Network (<http://www.reg.net/>)에서 혹은 orders@winedt.com으로 이메일을 보내 등록해야 합니다. 등록비는 학생 라이선스는 US \$30이고, 교육용 라이선스는 US \$40 입니다.

- 2000년 10월 1일 현재 가장 최근의 공식적인 Winedt 배포판의 버전은 5.1 (2000년 4월 15일)인데, 지난 8월 14일 이후 Patch 4 [Build: 20000814 (v. 5.1)] (patch4.zip 820 KB)를 ,
<ftp://nt-newton.fho-emen.de/pub/software/pc/win/winedt32/patch4.zip>에서 내려 받을 수 있습니다.
- WinEdt는 일반적으로 에디터가 가지고 있는 기능 외에 다음과 같은 기능들이 더 포함되어 있다.
 - Spell Checking with automatic highlighting of misspelled words and word completion...
 - Active Strings that can be used to define an automatic response macro which is activated upon typing certain strings. This functionality allows the automatic completion of environments or correction of frequently misspelled words, etc...
 - Invisible Popup Menus that can be used to implement double shortcut keystrokes (eg. equivalents of emacs keyboard interface)
 - Modes and submodes that can be used to set multilingual or multipurpose configurations (WinEdt32 only)
 - A Gather interface that can be used to collect data in your projects for easy referencing and navigation through your documents
 - ... and much more ...

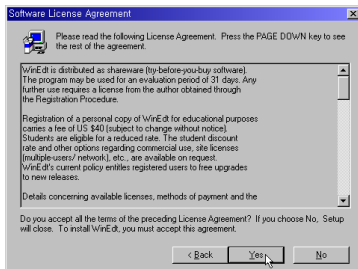
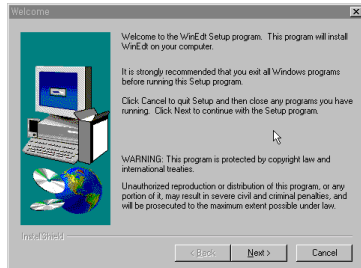
4.2 Winedt 설치 방법

1. 언어학자를 위한 L^AT_EX CD를 CD-ROM Drive에 넣는다.

2. editors\winedt\winedt501을 연다.
3. 아래 왼쪽 그림과 같이 setup.exe를 실행시킨다.
4. 그러면 아래 오른쪽 그림처럼 Setup 프로그램이 설치를 시작한다.

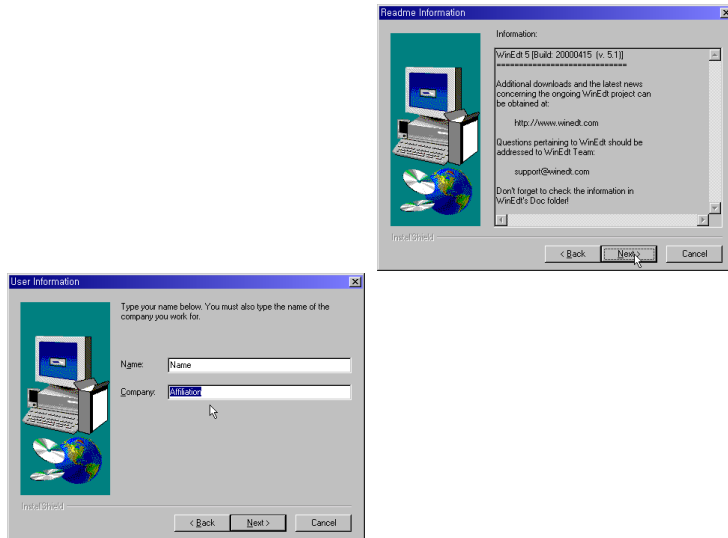


5. 아래 왼쪽과 같은 화면이 나오면 “Next”를 클릭한다.
6. 아래 오른쪽과 같은 화면이 나오면 “Yes”를 클릭한다.



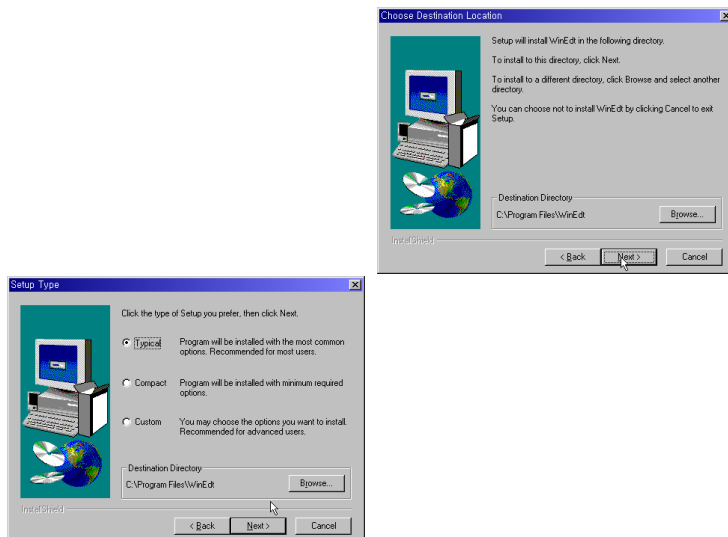
7. 그 다음에는 아래 왼쪽 화면처럼 초기화 정보가 요약되어 나오는데 여기서 “Next”를 클릭한다.

8. 그 다음에는 아래 오른쪽 화면처럼 “Name”과 “Company”를 묻는데, 이름과 소속을 입력하고 “Next”를 클릭한다.

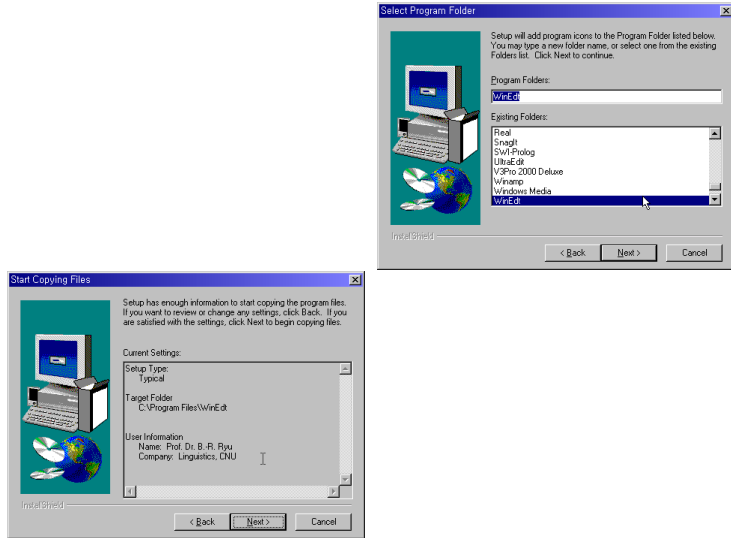


9. 그 다음에는 아래 왼쪽 화면처럼, 어디에 설치할 것이지를 묻는데, 이때 추천하는 대로 두고 “Next”를 클릭한다.

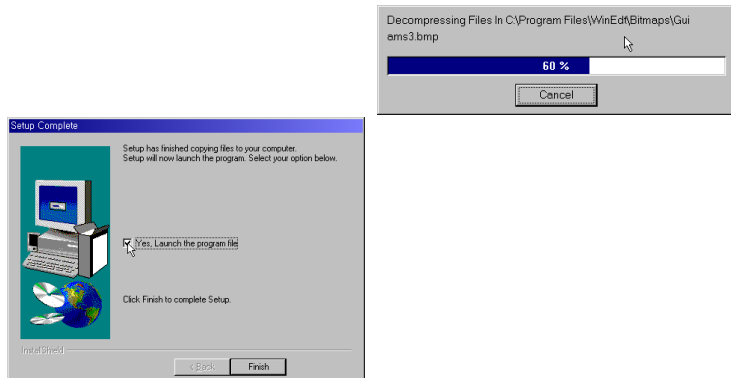
10. 그 다음에는 아래 오른쪽 화면처럼 어느 버전을 설치할 것이지 묻는데, “typical”을 체크하고 “Next”를 클릭한다.



11. 그 다음에는 아래 왼쪽 화면처럼, 메뉴폴더의 구성에 대해서 묻는데, 이때 추천하는 대로 두고 “Next”를 클릭한다.
12. 그 다음에는 아래 오른쪽 화면처럼 설치정보를 요약해서 보여주는데, 확인 후 이상이 없으면 “Next”를 클릭한다.

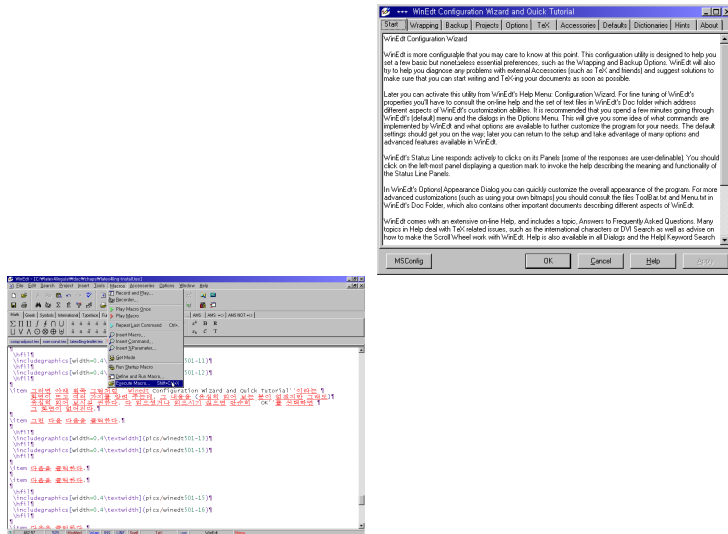


13. 그러면 아래 왼쪽 그림처럼 설치 프로그램이 하드 디스크에 프로그램을 설치한다.
14. 프로그램의 복사가 다 된 다음에는 아래 오른쪽 화면처럼 “프로그램을 바로 실행할 것인가”를 묻는 체크박스가 나타나는데 이 옵션을 선택하고 “Finish”를 클릭한다.



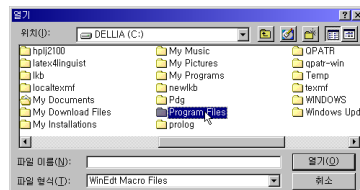
15. 그러면 아래 왼쪽 그림처럼 “WinEdt Configuration Wizard and Quick Tutorial”이라는 화면이 뜨고 여러 가지를 알려 주는데, 그 내용을 (유심히 읽어 보는 분이 없겠지만 그래도) 유심히 읽어 보시길 권한다. 다 읽으셨거나 읽으시기 싫으면 단순히 “OK”를 선택하면 그 화면이 없어진다.

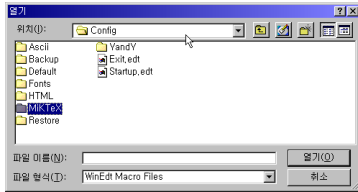
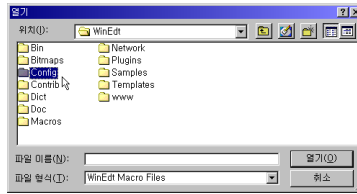
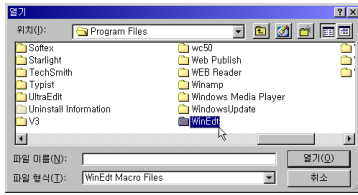
16. 그런 다음 아래 오른쪽 그림처럼 [Macros] > [Execute Macro ...]라는 메뉴를 선택한다.



17. 그러면 아래 왼쪽 그림처럼 “WinEdt Macro files”를 찾는 “열기” 화면이 나타나는데 이 때 WinEdt가 설치되어 있는 폴더를 찾아 아래 오른쪽 그림처럼 “Config” 폴더를 찾아 다시 그 아래에 있는 “MikTeX”라는 폴더를 찾아 그 폴더를 연다. 일반적인 경우라면 다음과 같은 경로 아래에 있다:

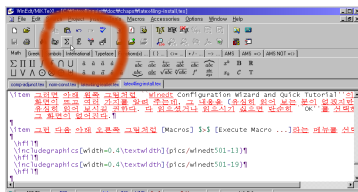
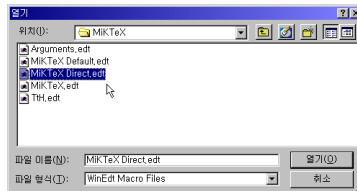
c:\Program Files\WinEdt\Config\MikTeX





18. 그 폴더를 열면 아래 오른쪽 그림과 같이 5개의 .edt 파일이 들어 있는데, 그 중에서 그림처럼 “MikTeX Direct.edt”를 선택한다.

19. 그런 다음 아래 오른쪽 그림처럼 Pull-down 메뉴 아래에 있는 아이콘들 중에서 Σ 를 선택한다. 그러면 여러 가지를 특수기호를 선택만 하면 바로 입력해 주는 모드로 설정이 된다. 이렇게 하면 WinEdt의 설치가 끝난다.



제 5 절 Acrobat Reader 설치하기

5.1 Acrobat Reader란 무엇인가

- Acrobat Reader는 Portable Data Format (PDF) 파일을 화면에 보여주고, 프린팅할 수 있게 해 주는 프로그램이다.
- URL은 <http://www.adobe.com/acrobat>입니다.
- Acrobat Reader는 현재까지는 Freeware입니다.
- 2000년 10월 1일 현재 가장 최근의 공식적인 Acrobat Reader 한국어 배포판의 버전은 4.05 (2000년 4월 15일)입니다.
- 사실 PDF파일은 GSView에서도 아무런 문제없이 화면에서 볼 수 있고 프린트할 수 있기 때문에 Acrobat Reader를 따로 설치할 필요는 없으나, 경험상으로 보면 GSView 보다는 더 선명한 화면을 제공하기 때문에 이 점을 선호하는 사람은 설치해도 좋다고 생각한다.

5.2 Acrobat Reader 설치 방법

1. 언어학자를 위한 L^AT_EX CD를 CD-ROM Drive에 넣는다.
2. `support\acrobat_reader\ar405kor.exe`를 실행시키고 안내에 따라 설치를 하면 된다. 더 자세한 설치 방법은 생략하기로 한다.

제 6 절 Graphic Utilities 설치하기

6.1 L^AT_EX와 Graphic Applications

다음 프로그램들은 모두 구입하셔야 합니다. 우선 며칠은 shaeware로 사용하실 수 있습니다.

- .eps파일을 만드는데 유용한 프로그램: Mayura Draw, Paint Shop Pro, Adobe Photoshop
- .png파일을 만드는데 유용한 프로그램: Paint Shop Pro
- .eps, .png, .gif, .jpg파일 등을 변환하는데 유용한 프로그램: Paint Shop Pro

6.2 Graphic Utilities 설치 방법

1. 언어학자를 위한 L^AT_EX CD를 CD-ROM Drive에 넣고 support 폴더를 연다.
2. 설치하고 싶은 프로그램이 들어 있는 폴더를 열어 그 안에 있는 setup.exe를 실행시키고 안내에 따라 설치를 하면 된다. 더 자세한 설치 방법은 생략하기로 한다.

부록 B

L^AT_EX에 한글을 설치하기 - H^LA_TE_X

제 1 절 L^AT_EX에서 한글 사용

이미 앞 장에서 설명했듯이 지금까지 나온 한글L^AT_EX를 정리하면 다음과 같습니다.

- **한T_EX Ver 1.5:** (주) 한글과 컴퓨터에서 Windows에서 사용할 수 있게 만들어 상업용으로 출시한 제품; 현재 더 이상 지원하지 않음.
- **원레이텍 Ver 1.0:** (주) LG 소프트에서 Windows에서 사용할 수 있게 만들어 출시한 제품; 현재 더 이상 지원하지 않으나 공개 ftp 자료실에 가면 구할 수 있음.
- **YahT_EX Ver 0.6:** DOS 환경에서 돌아 가는 emT_EX3.1415에 한글을 사용할 수 있도록 만든 ‘Yet Another Hangul T_EX’; 천리안 Math 동우회나 하이텔 (13. 공개자료실) 등에 가면 구할 수 있음.
- **hLaTeXp:** KAIST 수학과 차제춘 님이 만든 한글 L^AT_EX; 96/02/12

배포판 이후 업데이트가 안되고 있는 것으로 보임.

- **Werner Lemberg의 한글 T_EX**: 국내에는 거의 알려지지 않은 한글 T_EX로 1994년 독일의 Werner Lemberg에 의해 만들어진 한글 T_EX; 이 한글 T_EX역시 k-시리즈 한글 글꼴에 바탕을 두고 있음.
- **HLaTeX**: 1995년 독일 튀빙엔 대학교의 은광희 님에 의해 만들어진 한글 L^AT_EX; 글꼴 소스와 함께 배포된 현재 사용자가 많은 한글 L^AT_EX; 2000년 2월 20일 이후 현재까지 0.991 버전이 배포되고 있음.

현재 한글L^AT_EX는 은광희님이 만드신 H^AL^AT_EX (버전 0.991)와 차재춘님이 만드신 hLaTeXp의 두 가지가 가장 많이 퍼져 있습니다. 둘은 각기 장단점이 있으나 어느 쪽도 잘 쓸 수 있습니다. 그러나 두 가지를 한 PC에서 사용하는 것은 추천하고 싶지 않은 방법입니다.

1.1 H^AL^AT_EX-0.991

은광희 님이 만드신 H^AL^AT_EX는 버전 0.991이 배포되고 있습니다 (2000.2.20).

H^AL^AT_EX-0.991에서는

- 한글 글꼴이 0.98에서 제공하던 완성형 한글 위주에서 UHC (Unified Hangul Code)폰트 중심으로 바뀌었습니다. 완성형 글꼴은 더 이상 지원하지 않는다고 합니다.
- Lambda(Omega)에서도 동일한 UHC 한글 폰트를 씁니다.
- 완성형 한글을 문화부 TTF 폰트를 이용하여 쓸 수 있습니다.

1.2 hLaTeXp

차재춘 님이 만드신 hLaTeXp는 X-series 글꼴이라는 것을 쓰는데, 글꼴의 소스를 공개하지 않아 프린터의 해상도에 맞는 300dpi 혹은 600dpi .pk 파일을 내려받아 써야 한다. X-series PK font는 양이 많고, TeX의 한 특징인 고급스러운 윤곽선을 얻기가 힘들다는 것이 결

정적인 약점으로 작용합니다. 그래서 현재로서는 hLaTeXp로 만든 한글 문서는 PDF로 변환했을 때 품위가 좀 떨어지는 것으로 알려져 있습니다.

그러나 윈도우용 T_EXsolution인 TeXplus와 hLaTeXp가 아주 잘 호환된다는 점(HLaTeX은 좀 어렵습니다)과, hTeXp라는 한글 T_EX을 제공한다는 점을 생각하면 hLaTeXp를 선호하는 분도 있을 것이라고 생각합니다. TeXplus나 hLaTeXp에 관심이 있으신 분은 한글 T_EX홈페이지 (<http://knot.kaist.ac.kr/htex/>)를 방문하셔서 정보를 얻으시기 바랍니다. 이 글에서는 hLaTeXp에 대해서는 더 이상 설명하지 않겠습니다.

1.3 CJK 패키지

한글 구현의 다른 방법으로 CJK 패키지가 있습니다. 이 패키지의 장점은 한글 이외에 일본어 및 중국어 등도 입력만 가능하다면 처리할 수 있다는 장점이 있습니다만, 공개된 글꼴의 수가 적고 무료 글꼴의 품위가 조금 낮은 듯한 것이 흠입니다. H_AT_EX-0.98에서 쓰던 완성형 폰트를 이용할 수도 있게 되어 있습니다. 이 패키지는 CTAN에서 구할 수 있습니다.

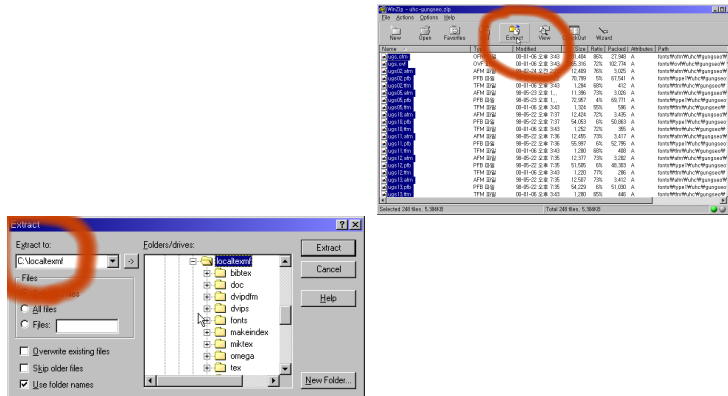
제 2 절 H_AT_EX 설치하기

이 장은 조진환(chofchof@kias.re.kr)님의 “MikT_EX 1.2e에 H_AT_EX 0.991 설치하기”와 은광희님의 “한글H_AT_EX길잡이 0.99 (99년 1월 18일)”를 중심으로 H_AT_EX를 설치하는 방법을 기술한 것입니다. H_AT_EX의 설치하는 첫째 글꼴 설치, 둘째 꾸러미 및 한글 처리 매크로 그리고 설정변경으로 나누어 설명하겠습니다.

2.1 글꼴 설치

1. 언어학자를 위한 L_AT_EX CD의 hlatex\fonts\uhc 폴더에 가서 해당 압축 파일만 풀면 자동으로 해당 폴더에 설치되도록 설정해 놓았습니다.

2. 글꼴을 설치할 때에는 Winzip과 같은 .exe 압축 파일을 풀 수 있는 도구를 이용해서 다음 그림의 왼쪽과 같이 Extract를 하되 저장될 root 폴더를 오른쪽 그림과 같이 c:\localtexmf로 지정하십시오.
3. 아래의 세 개의 파일들은 필수적인 글꼴을 포함하고 있습니다. 반드시 설치하셔야 합니다.



- HLaTeX-0.99 UHC 명조 글꼴
hlatex/uhc/uhc-myoungjo.exe (7.75MB -> 10.4MB)
- HLaTeX-0.99 UHC 고딕 글꼴
hlatex/uhc/uhc-gothic.exe (5.66MB -> 8.38MB)
- HLaTeX-0.99 UHC 타자 글꼴
hlatex/uhc/uhc-taza.exe (408KB -> 1.60MB)

4. 아래의 글꼴들은 필요에 따라 설치하여 사용할 수 있습니다. 설치하지 않아도 사용에는 지장이 없습니다. HLaTeX-0.99의 매뉴얼 중 DVI 포맷으로 된 hlguide.dvi 파일을 YAP으로 보려면 꼭 필요합니다만 hlguide.ps.gz 파일은 이 글꼴들을 설치하지 않아도 GSview를 이용해서 보거나 출력할 수 있습니다.

- HLaTeX-0.99 UHC 봄 글꼴
hlatex/uhc/uhc-bom.exe (427KB -> 1.43MB)
- HLaTeX-0.99 UHC 그래픽 글꼴
hlatex/uhc/uhc-graphic.exe (1.43MB -> 3.52MB)

- HLaTeX-0.99 UHC 궁서 글꼴
hlatex/uhc/uhc-gungseo.exe (3.97MB -> 5.25MB)
- HLaTeX-0.99 UHC 자모 글꼴
hlatex/uhc/uhc-jamo.exe (3.90MB -> 4.89MB)
- HLaTeX-0.99 UHC 펜 글꼴
hlatex/uhc/uhc-pen.exe (1.00MB -> 3.03MB)
- HLaTeX-0.99 UHC 필기 글꼴
hlatex/uhc/uhc-pilgi.exe (805KB -> 2.49MB)
- HLaTeX-0.99 UHC 신문 글꼴
hlatex/uhc/uhc-shinmun.exe (3.39MB -> 4.70MB)
- HLaTeX-0.99 UHC 옛글 글꼴
hlatex/uhc/uhc-yetgul.exe (4.04MB -> 5.35MB)
- HLaTeX-0.99 문화부 트루타입 글꼴
hlatex/moonttf/moonttf.exe (7.51MB -> 12.8MB)

5. 아래의 글꼴들은 글자의 가로폭을 변화시킨 장평 글꼴들입니다. 장평 글꼴이 필요한 특별한 경우가 아니면 설치하지 않는 것이 좋습니다.

- HLaTeX-0.99 UHC 명조 장평 글꼴
hlatex/uhc/uhc-myungjo-extra.exe (3.04MB -> 9.01MB)
- HLaTeX-0.99 UHC 고딕 장평 글꼴
hlatex/uhc/uhc-gothic-extra.exe (3.01MB -> 9.08MB)
- HLaTeX-0.99 UHC 타자 장평 글꼴
hlatex/uhc/uhc-taza-extra.exe (1.06MB -> 4.65MB)
- HLaTeX-0.99 UHC 봄 장평 글꼴
hlatex/uhc/uhc-bom-extra.exe (891KB -> 3.81MB)
- HLaTeX-0.99 UHC 그래픽 장평 글꼴
hlatex/uhc/uhc-graphic-extra.exe (2.09MB -> 7.63MB)
- HLaTeX-0.99 UHC 궁서 장평 글꼴
hlatex/uhc/uhc-gungseo-extra.exe (1.46MB -> 4.43MB)

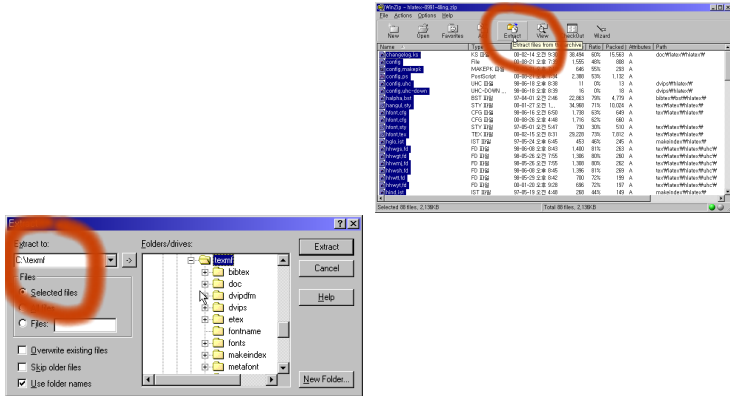
- HLaTeX-0.99 UHC 자모 장평 글꼴
hlatex/uhc/uhc-jamo-extra.exe (3.17MB -> 13.3MB)
- HLaTeX-0.99 UHC 펜 장평 글꼴
hlatex/uhc/uhc-pen-extra.exe (1.78MB -> 7.63MB)
- HLaTeX-0.99 UHC 필기 장평 글꼴
hlatex/uhc/uhc-pilgi-extra.exe (1.62MB -> 6.58MB)
- HLaTeX-0.99 UHC 신문 장평 글꼴
hlatex/uhc/uhc-shinmun-extra.exe (1.46MB -> 4.50MB)
- HLaTeX-0.99 UHC 옛글 장평 글꼴
hlatex/uhc/uhc-yetgul-extra.exe (1.29MB -> 4.53MB)
- HLaTeX-0.99 문화부 트루타입 장평 글꼴
hlatex/moonttf/moonttf-extra.exe (648KB -> 1.04MB)

6. 위의 1번과 2번 항목의 글꼴들을 모두 설치한 경우에만 `c:\localtexmf\doc\latex\hlatex\hlguide.dvi` 파일을 YAP으로 볼 수 있습니다. pk글꼴은 시스템이 자동으로 감지해서 글꼴이 없으면 스스로 만들어 씁니다. 논리적으로 처음에는 아무런 한글 글꼴도 없는 상황이므로 맨 처음 pk글꼴을 생성하게 되는데, 이 글꼴 생성에 매우 오랜 시간이 (성능이 낮은 컴퓨터에서는 1 시간 이상) 걸린다는 점을 알아 두시는 것이 당황하지 않는 방법입니다.

2.2 HIAT_EX-0.991을 위한 MiK_TE_X 1.20e 세팅 변경

- 다음에는 언어학자를 위한 HIAT_EX-0.991 꾸러미 및 구성 파일 설치 및 세팅 변경 차례입니다. 이글에서는 루트 TEXMF 트리가 `c:\texmf`에 있고 로컬 TEXMF 트리가 `c:\localtexmf`에 있는 것으로 가정하고 설명하겠습니다.
- 이미 기본적인 세팅을 다 끝내서 저장한 상태이므로 여러분은 압축파일을 풀기만 하면 됩니다. 그 파일은 다음과 같습니다.
hlatex\configured\hlatex-0991-4ling.exe (1.11MB -> 2.22MB)

- 꾸러미 및 구성 파일을 설치할 때에도 Winzip과 같은 .zip 압축 파일을 풀 수 있는 도구를 이용해서 다음 그림의 왼쪽과 같이 Extract를 하되, 이번에는 저장될 root 폴더를 c:로 지정하십시오.¹



- 다음에 기술할 세팅변경은 hlatex-0991-4ling.exe. 파일을 풀면 1항-7항까지는 자동으로 변경해 주도록 이미 세팅을 해 놓았습니다. 따라서 여러분은 hlatex-0991-4ling.exe. 파일을 풀고 나서 바로 8번 항목으로 가셔서 MikTeX의 Refresh Filename Database 를 수행하면 됩니다. 그 명령은 [시작] > [프로그램] > [MiKTeX] > [Maintenance] 에 있습니다.

1. UHC (Unified Hangul Code) 글꼴은 PostScript Type I 글꼴 (.pfb)로 만들어져 있으므로 ps2pk라는 프로그램을 통해 PK 글꼴을 얻게 됩니다. 현재 Device Independent Files (.dvi)을 보여주는 YAP은 .pk 글꼴만을 사용하기 때문에, 먼저 YAP에게 PostScript Type I 글꼴을 .pk 로 바꾸는 방법을 알려주어야 합니다. 이 역할을 하는 것이 MiKTeX 1.20e 에서는 config.makepk 라는 파일입니다.

↳ c:\texmf\miktex\config\config.makepk 의 마지막에 p uhc-down.map+ 줄이 있는지 살펴보고 없으면 추가한다.

¹오른쪽 그림에는 c:\texmf아래에 설치하는 것처럼 나와 있으나 c:아래 설치하도록 하십시오. 더 정확하게 말하면 texmf 폴더와 localtexmf를 모두 포괄하는 바로 위 상위 폴더를 지정해서 푸셔야 제대로 설치됩니다.

- 이제 DVIPS 를 위해서 `config.ps`를 고칠 차례입니다. 만약 DVIPS 로 PS 파일을 만들 때 PK 글꼴을 넣길 원하시면 이 부분은 생략해도 됩니다.

↳ `c:\texmf\dvips\config\config.ps` 의 `p+myfonts.map` 다음 줄에 `p+uhc-down.map` 줄이 있는지 살피고 없으면 추가한다.

- 혹시 PDF 파일로 출력을 원하시는 분들은 MiKTeX 1.20e 에는 `dvipdfm` 이라는 강력한 프로그램이 들어 있기 때문에 이를 이용하면 PDF파일을 쉽게 만들 수 있습니다. 사용방법은 DVIPS 와 거의 유사하며 간단히 `dvipdfm test.dvi` 하면 `test.dvi` 가 `test.pdf` 로 바뀝니다. 세팅방법은 아래와 같습니다.

↳ `c:\texmf\dvipdfm\config\config` 의 마지막에 `f uhc-down.map` 줄이 있는지 살피고 없으면 추가한다.

- PDFTeX 을 선호하시는 분들은 `c:\texmf\pdfTEX\config` 아래에 있는 `pdfTEX.cfg` 의 세팅을 변경해야 합니다.

↳ `c:\texmf\pdfTEX\config\pdfTEX.cfg` 의 마지막에 `map uhc-down.map+` 줄이 있는지 살피고 없으면 추가한다.

- 위에서 문화부 트루타입 글꼴을 설치한 경우에는 `c:\texmf\ttf2tfm\base\ttfonts.map` 파일을 변경해 주어야 합니다.

↳ 텍스트 에디터를 사용해서 이 파일의 마지막에 `c:\localtexmf\ttf2tfm\base\moonttf.map` 파일을 복사합니다.
혹은 도스창에서 아래의 명령을 실행하면 됩니다.

```
copy c:\texmf\ttf2tfm\base\ttfonts.map  
+c:\localtexmf\ttf2tfm\base\moonttf.map
```

```
c:\texmf\ttf2tfm\base\ttfonts.map
```

단, H_AT_EX-0.98 에서 문화부 트루타입 글꼴을 사용했던 분들의 경우 이미 `ttfonts.map` 이 정보를 가지고 있으므로 다시 이 작업을 할 필요는 없습니다.

- 이제 H_AT_EX가 어떤 글꼴을 기본 글꼴로 사용해야 하는지 세팅해야 할 차례입니다. 이 부분은 `hfont.cfg`가 담당하고 있기 때문에 `hfont.cfg`의 해당부분을 변경해 주어야 합니다. `hfont.cfg`는 `c:\texmf\tex\latex\hlatex` 아래에 있으므로 이 파일을 찾아 텍스트 에디터로 열어 세팅을 바꿔 준다.

→ 열어 본 결과 다음 다음과 같이 되어 있으면,

```
% UHC 글 자체를 한글 LaTeX 운영 체계의 기본 글 자체로 미리 설정하기
%% 위해서는 다음과 같이 한다.
% \MapHangulFamily{mj}{wmj,wmj,wmj}
% \MapHangulFamily{gt}{wgt,wgt,wgt}
% \MapHangulFamily{tt}{wtt,wtt,wtt}
% \MapHangulFamily{sm}{wmj,wmj,wmj}
% \MapHangulFamily{gs}{wgs,wgs,wgs}
% \MapHangulFamily{gr}{wgr,wgr,wgt}
% \DeclareTextFontCommand{\textsh}{\shfamily}
% \DeclareRobustCommand\shfamily
%      {\not@math@alphabet\shfamily\mathsh}
%      \hfontfamily{sh}}
% \MapHangulFamily{sh}{wsh,wsh,wsh}
```

다음과 같이 각 행이 주석문으로 % 표시가 되있는 것을 % 표시를 없애고 저장한다.

```
% UHC 글 자체를 한글 LaTeX 운영 체계의 기본 글 자체로 미리 설정하기
%% 위해서는 다음과 같이 한다.
\MapHangulFamily{mj}{wmj,wmj,wmj}
\MapHangulFamily{gt}{wgt,wgt,wgt}
```

```

\MapHangulFamily{tt}{wtt,wtt,wtt}
\MapHangulFamily{sm}{wmj,wmj,wmj}
\MapHangulFamily{gs}{wgs,wgs,wgs}
\MapHangulFamily{gr}{wgr,wgr,wgt}
\DeclareTextFontCommand{\textsh}{\shfamily}
\DeclareRobustCommand\shfamily
    {\not@math@alphabet\shfamily\mathsh
    \hfontfamily{sh}}
\MapHangulFamily{sh}{wsh,wsh,wsh}

```

7. Winedt는 $\text{T}_\text{E}\text{X}$ 가 컴파일 과정에서 에러를 만났을 때, 그곳이 어느 곳인지 알려주는 기능이 있습니다. 이 기능은 아주 유용한 기능이므로 다음과 같이 MikTeX의 초기화 파일 (`c:\texmf\miktex\config\miktex.ini`) 수정해서 사용하시기를 추천합니다.

↳ 텍스트 에디터를 사용해서 `miktex.ini` 파일을 열고 이 파일의 마지막 부분에

```

;; Editor=notepad "%f"
;; Editor=winedt "[Open('%f');SelLine(%l,7)]"
;; Editor=emacs +%l "%f"
Editor=notepad "%f"

```

라고 되어 있으면, 다음과 같이 고치고 저장한다.

```

;; Editor=notepad "%f"
Editor=winedt "[Open('%f');SelLine(%l,7)]"
;; Editor=emacs +%l "%f"
;; Editor=notepad "%f"

```

이 명령은 MikTeX와 연동하는 에디터로 Notepad가 아닌 Winedt를 사용하겠다는 뜻을 MikTeX에 전하는 역할을 한다.

8. 이제 마지막으로 MikTeX의 Refresh Filename Database 를 수행해야 합니다. 이것은 [시작]> [프로그램]> [MiKTeX] > [Maintenance] 에 있습니다.

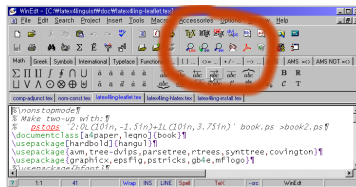
9. 아래와 같이 간단한 test.tex 파일을 만든 후 latex test 라고 쳐서 test.dvi 를 만든 후 dvips test 하면 test.ps가 나오는지, 그리고 dvi2pdf test라고 치면 test.pdf가 제대로 생성되는지 확인하십시오.

```

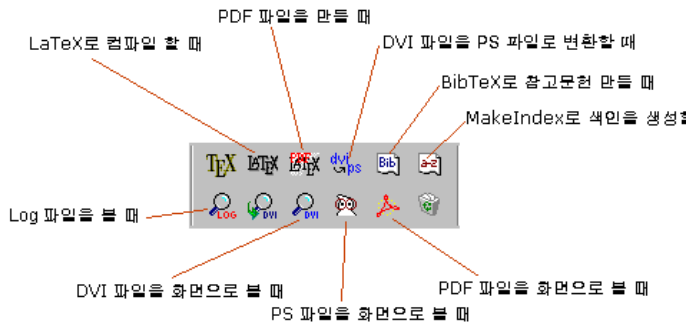
\documentclass{article}
\usepackage{hangul}
\begin{document}
이것은 HLaTeX-0.991 연습입니다.
\end{document}

```

10. Winedt에서는 아래에 있는 그림처럼 아이콘만 누르면 명령이 자동으로 수행됩니다.



11. 관련이 있는 아이콘을 조금 확대하면 다음과 같습니다. 그리고 각 아이콘이 무슨 명령을 수행하도록 프로그래밍되었는지 아래 그림을 보면 알 수 있습니다.



H^AT_EX-0.991 에 관한 자세한 사항은 `c:\localtexmf\doc\latex\hlatex` 에 있는 `hlguide.ps.gz` 파일을 참조하십시오. GSview 를 이용하여 보거나 출력할 수 있습니다.

참고 문헌

- Bierwisch, Manfred. 1967. Some semantic universals of german adjectives. *Foundations of Language*, 3:1–36.
- Cha, Jae-Choon. 1996. 한글 T_EX 홈페이지 인터넷 사이트. Ms., 한글 T_EX의 역사.
- Clark, Herbert H. and Eve V. Clark. 1981. *Psychology and Language: An Introduction to Psycholinguistics*. Harcourt Brace Jovanovich, New York.
- Eun, Koang-Hi. 2000. 한글H_AT_EX 길잡이 ver. 0.99. Ms., H_AT_EX 0.99 배포판 사용자 설명서.
- Kamp, Hans and Uwe Reyle, editors. 1993. *Form Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer Academic Publishers, Dordrecht.
- Knuth, Donald E. 1986a. *METAFONT: The Program*. Addison-Wesley, Reading.
- Knuth, Donald E. 1986b. *The METAFONTbook*. Addison-Wesley, Reading.
- Knuth, Donald E. 1986c. *T_EX: The Program*. Addison-Wesley, Reading.
- Knuth, Donald E. 1986d. *The T_EXbook*. Addison-Wesley, Reading.

- Lamport, Leslie. 1987. *LaTeX – A Document Preparation System – User’s Guide and Reference*. Addison-Wesley, Reading.
- Lang, Ewald. 1994. Basic dimension terms: A first look at universal features and typological variation, paper presented at FAS Conference “Linguistics, Universals and Typological Variation”, Berlin; March 17-19, 1994.
- Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.
- Pullum, Geoffrey K. and William A. Ladusaw. 1986. *Phonetic Symbol Guide*. University of Chicago Press, Chicago.
- Sag, Ivan A. and Tom Wasow. 1999. *Syntactic Theory – A Formal Introduction*. CSLI Publication, Stanford.
- von Stechow, Arnim and Dieter Wunderlich, editors. 1991. *Semantik: Ein Internationales Handbuch der zeitgenössischen Forschung*. de Gruyter, Berlin.