



**KTH Datavetenskap  
och kommunikation**

# Introduktion till $\LaTeX$

Lars Engebretsen

April 2005

## Sammanfattning

$\LaTeX$  är ett mycket kraftfullt typsättningssystem. Det skapades av Leslie Lamport och bygger på  $\TeX$ , som i sin tur skapades av Donald Knuth för att typsätta bokserien *The Art of Computer Programming*. Man kan använda  $\LaTeX$  till i princip alla typer av dokument, men verktyget visar sin stora styrka i samband med vetenskapliga rapporter.  $\LaTeX$  är extremt stabilt och har inbyggt stöd för skapande av innehållsförteckningar, korsreferenser och referenslistor. Det är dessutom gratis och finns till så gott som alla plattformar. En mycket stor fördel är att filformatet är samma för alla plattformar – man kan alltså utan problem flytta dokument från en Windows-miljö till Unix och vice versa.

Denna skrift har flera syften. För det första ska den vara en introduktion till  $\LaTeX$  för den som aldrig använt verktyget tidigare. För det andra innehåller den en sammanfattning av svenska typografiska regler. För det tredje innehåller den tips som underlättar arbetet med  $\LaTeX$  och lösningar på vanliga  $\LaTeX$ -problem.

Mer information om  $\LaTeX$  finns på nätet. En bra startsida är Nadas  $\TeX$ -sida »<http://www.sgr.nada.kth.se/misc/tex/>«. Därifrån finns länkar till ytterligare dokumentation. En bra introduktion till  $\LaTeX$  är dokumentet *The Not So Short Introduction to  $\LaTeX 2\epsilon$*  som också finns på Nadas webbplats. På Nada används distributionen  $\text{teTeX}$  av Thomas Esser; all dokumentation som följer med  $\text{teTeX}$  finns i katalogen »[/pkg/teTeX/default/share/texmf-dist/doc](#)« på Nadas Unix-datorer. Det finns också böcker om  $\LaTeX$ ; en av de mest använda är  *$\LaTeX$ : A Document Preparation System* av Leslie Lamport.

## 1 Att komma igång

Att skriva ett dokument i  $\LaTeX$  är till viss del en annorlunda upplevelse mot program som Word. Dokumentet skrivs i ett textredigeringsprogram – exempelvis Emacs – och man fokuserar på att beskriva strukturen i dokumentet. För att se resultatet måste dokumentet först

kompileras till en så kallad DVI-fil (dvi står för *device independent*). Sedan kan man titta på resultatet på skärmen eller skriva ut det.

Man kan jämföra processen med hur det är att skriva en webbsida i HTML. Då skapar man också sidan i ett textredigeringsprogram och där beskriver man vilken funktion olika delar av texten ska ha. Sedan tittar man på resultatet med en webbläsare.

Fördelen med ovanstående arbetssätt är att man själv koncentrerar sig på vad som ska stå i dokumentet och låter ett verktyg sköta all layout. Nackdelen är att det blir lite mer omständligt att skapa ett dokument, speciellt i början innan man är van.

## 1.1 Att köra L<sup>A</sup>T<sub>E</sub>X på Nada

För att använda L<sup>A</sup>T<sub>E</sub>X på Nada måste du ladda modulen tex (det görs med kommandot »module add tex»). Du får då tillgång till kommandot »latex» som tar en L<sup>A</sup>T<sub>E</sub>X-fil och skapar en DVI-fil, kommandot »xdvi» som visar en DVI-fil på skärmen, kommandot »dvips» som skriver ut en DVI-fil och många andra L<sup>A</sup>T<sub>E</sub>X-relaterade kommandon. Låt oss titta på hur man typiskt arbetar med L<sup>A</sup>T<sub>E</sub>X.

### 1.1.1 Skapa en L<sup>A</sup>T<sub>E</sub>X-fil

Först skapar man alltså själva L<sup>A</sup>T<sub>E</sub>X-filen i Emacs. Filen kan heta i princip vad som helst, men den bör ha ändelsen »tex». Man bör också undvika att ha blanksteg i namnet.

```
\documentclass[a4paper,11pt]{article}

\title{A Tiny Example}
\author{Lars Engebretsen}

\begin{document}

\maketitle

\section{Introduction}

This is a tiny example.

\end{document}
```

Normalt sett är en L<sup>A</sup>T<sub>E</sub>X-fil indelad i två delar. Inledningen (*preamble* på engelska) anger vilken typ av dokument det är – ovanstående exempel är en artikel – och vad dokumentet har för titel och författare. Där kan man också ange dokumentets datum samt ladda till läggspaket om man behöver mer avancerad funktionalitet. Det bakvända snedstreckat, \, inleder L<sup>A</sup>T<sub>E</sub>X-kommandon. I det lilla exemplet ovan har vi till exempel \maketitle som skriver ut dokumentets titel. För de kommandon som tar en parameter – \section är exempel på ett sådant – skriver man parametern inom klamrar direkt efter kommandot. L<sup>A</sup>T<sub>E</sub>X har också ett omgivningsbegrepp. Omgivningar inleds med \begin{...} och avslutas med \end{...}. Det finns ett stort antal omgivningar inbyggda i L<sup>A</sup>T<sub>E</sub>X och ännu fler finns definierade i olika tilläggspaket. Själva dokumentet, utöver inledningen, ligger i en speciell omgivning som inleds med \begin{document} och avslutas med \end{document}.

### 1.1.2 Från L<sup>A</sup>T<sub>E</sub>X-fil till DVI-fil

För att skapa en DVI-fil från ovanstående L<sup>A</sup>T<sub>E</sub>X-fil använder man kommandot »latex». Om L<sup>A</sup>T<sub>E</sub>X-filen heter »tiny.tex» kan det till exempel se ut på följande sätt:

```
dator> latex tiny.tex
This is TeX, Version 3.14159 (Web2C 7.3.1)
(tiny.tex
LaTeX2e <1998/12/01> patch level 1
...
No file tiny.aux.
[1] (tiny.aux) )
Output written on tiny.dvi.
Transcript written on tiny.log.
```

Förutom DVI-filen skriver L<sup>A</sup>T<sub>E</sub>X också en logg som man kan titta i om man vill ha mer detaljerad information om vad som hände under körningen.

### 1.1.3 Titta på DVI-filen

Till slut använder man kommandot »xdvi» för att se resultatet på skärmen.

```
dator> xdvi tiny.dvi
```

Det räcker att starta »xdvi» en gång per dokument. Om man ändrar i dokumentet upptäcker »xdvi» detta och visar det ändrade dokumentet istället. Lämpligt är därför att köra »xdvi» i bakgrunden:

```
dator> xdvi tiny.dvi &
```

Man måste däremot köra latex för varje ändring i dokumentet. Vill man skriva ut resultatet på papper använder man »dvips».

```
dator> dvips tiny.dvi
```

För att få ut resultatet på en viss skrivare används väljaren -P, precis som i de flesta andra Unix-program.

```
dator> dvips -Pred tiny.dvi
```

### 1.1.4 Alternativa arbetssätt

Till MacOS X finns programmet T<sub>E</sub>XShop som tillhandahåller en textredigerare med inbyggd förhandsgranskare. T<sub>E</sub>XShop är installerat i Nadas Mac-system och är konfigurerat för att använda den L<sup>A</sup>T<sub>E</sub>X som finns installerad i Nadas Unix-system. Det går därför bra att flytta L<sup>A</sup>T<sub>E</sub>X-filer mellan Nadas MacOS- och Unix-system om man använder teckenkodningen *Latin-1* (se vidare avsnitt 5.3.1). T<sub>E</sub>XShop skapar PDF-filer istället för DVI-filer genom att använda programmet »pdflatex» (se vidare avsnitt 5.1.2).

Använder man Emacs kan man förenkla arbetet betydligt genom att använda en utökad L<sup>A</sup>T<sub>E</sub>X-mod som laddas genom att man lägger in kommandot

```
(require 'tex-site)
```

i sin ~/.emacs-fil.

## 1.2 Matematik

Matematiska formler är något som  $\LaTeX$  är exceptionellt bra på. De skrivs normalt mellan dollartecken. Dollartecknen talar om, både för  $\LaTeX$  och för den som skriver dokumentet, att det som står mellan tecknen är en matematisk formel. Exempelvis skrivs formeln  $E = mc^2$  som `$E = m c^2$`. Blankstegen har ingen betydelse i matematiska formler, de kan man sätta in som man vill för att det ska bli lättare att läsa formeln direkt i  $\LaTeX$ -filen. Det finns ett stort antal specialsymboler och matematiska konstruktioner tillgängliga som  $\LaTeX$ -kommandon, exempelvis skrivs formeln  $V = \frac{4}{3}\pi r^3$  för sfärens volym som `$V = \frac{4}{3}\pi r^3$`. I början kan det kännas lite otympligt att läsa formler i  $\LaTeX$ -notation, men man vänjer sig snabbt. Dessutom kan man alltid titta på hur det blir på skärmen med »xdvi».

Det är viktigt att alltid omgärda matematiska uttryck med dollartecken, de innebär nämligen att  $\LaTeX$  använder speciella regler för hur uttrycket ska typsättas. Exempelvis blir uttrycket `$5-2=3$` typsatt som  $5 - 2 = 3$ , utan dollartecken runt blir det istället  $5-2=3$ . Som synes är mellanrummen mellan tecknen i det senare fallet för små, och istället för minus-tecken får man ett bindestreck.

## 1.3 Felhantering

Felmeddelanden från  $\LaTeX$  ter sig en smula kryptiska i början. De två vanligaste felet man gör under normal  $\LaTeX$ -användning är att skriva något kommando fel eller att använda ett matematikkommando utan att omgärda det med dollartecken. Låt oss titta på hur ett typiskt felmeddelande ser ut. Antag att vi har stavat fel på `\maketitle`.  $\LaTeX$  stannar då med följande felmeddelande:

```
! Undefined control sequence.
1.28 \maketitle
?
```

På första raden står felmeddelandet, andra raden innehåller numret på den rad som var fel (28) och den text som var fel (`\maketitle`). Vi kan nu fortsätta på ett antal sätt. Enklast är att bara trycka RETURN, då ignoreras felet och körningen fortsätter. Vi kan sedan ändra felstavningen i  $\LaTeX$ -filen och köra den genom latex igen.

Glömmer man att omgärda matematikkommandon med dollartecken blir felet lite mer kryptiskt. Skriver man till exempel `A = 4 \pi r^2`, utan dollartecken runt, stannar  $\LaTeX$  med följande felmeddelande:

```
! Missing $ inserted.
<inserted text>
$
1.212 \pi
?
```

Felmeddelandet har samma struktur som tidigare, men vi får också veta att  $\LaTeX$  stoppade in ett dollartecken åt oss. Tyvärr vet inte  $\LaTeX$  när formeln tar slut, så för det mesta dyker det upp en hel del följdfel i detta läge. Genom att ge  $\LaTeX$  kommandot »R» följt av RETURN ignoreras alla efterföljande fel. Vi kan sedan ändra i filen och köra latex igen. I värsta fall kan man alltid trycka CTRL-C för att avbryta  $\LaTeX$  när man har fått ett felmeddelande. Nackdelen med det är att det då inte skapas någon DVI-fil.

## 1.4 Att skriva svenska texter

Det går alldeles utmärkt att skriva svenska texter med  $\LaTeX$  men man måste då ladda tre tilläggs paket. Detta gör man genom att lägga till följande rader i dokumenthuvudet:

```
\usepackage[swedish]{babel}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
```

Den första av dessa rader har två huvudsakliga effekter. För det första blir datumangivelser och vissa texter på svenska istället för på engelska, om man har en innehållsförteckning kommer den till exempel att få rubriken »Innehållsförteckning» istället för »Table of contents». För det andra laddas svenska avstavningsmönster. För att dessa mönster ska fungera måste man dessutom använda en utökad teckentabell – från början klarade  $\LaTeX$  bara engelska och använde en teckentabell som saknar bland annat å, ä och ö – detta åstadkoms med den andra raden ovan.

Den tredje raden talar om för  $\LaTeX$  vilken teckenkodning själva  $\LaTeX$ -filen har. På Unix-system har man normalt teckenkodningen *Latin-1*, även kallad ISO 8859-1. Kör man under Windows använder man normalt Microsofts något ändrade variant av *Latin-1*, då ska man istället ange

```
\usepackage[cp1252]{inputenc}
```

Under MacOS 9 eller tidigare används Apples alldeles egen teckenkodning, då ska man ange

```
\usepackage[applemac]{inputenc}
```

Under MacOS X bör man konfigurera sitt redigeringsprogram så att det sparar filen i *Latin-1* och sedan ange

```
\usepackage[latin1]{inputenc}
```

På så sätt kan man arbeta med filen även under andra operativsystem utan att behöva koda om den. Om flera personer arbetar på samma dokument är det viktigt att alla använder samma teckenkodning, se avsnitt 5.3.

Sammanfattningsvis kan därför en svensk motsvarighet till det lilla testdokumentet ovan se ut på följande sätt:

```
\documentclass[a4paper,11pt]{article}

\usepackage[swedish]{babel}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}

\title{Ett litet exempel}
\author{Lars Engebretsen}

\begin{document}

\maketitle

\section{Inledning}

Det här är ett litet exempel.

\end{document}
```

Prova att köra  $\LaTeX$  på detta dokument och titta på resultatet i »xdvi». Skriver man på svenska måste man också vara medveten om svenska skrivregler och svenska typografiska konventioner, mer om detta senare i häftet.

## 2 Ett längre dokument

Man behöver inte kunna många  $\LaTeX$ -kommandon för att skriva en typisk rapport. I det här avsnittet presenteras de vanligaste kommandona och konstruktionerna. Det finns en inbyggd  $\LaTeX$ -mod i Emacs och en utökad mod kan laddas genom att man lägger in instruktionen (`require 'tex-site`) i sin `.emacs`-fil.

### 2.1 Förstasidan

Den första sidan i en rapport innehåller normalt dokumentets titel, författare och datum. Vi har redan sett hur man anger titel och författare; datum anges med kommandot `\date`. Om man inte anger något datum kommer  $\LaTeX$  automatiskt sätta in dagens datum. Man måste till sist ange kommandot `\maketitle` för att titeln ska tas med i dokumentet. Dokumentets sammanfattning står normalt direkt efter titeln och anges med omgivningen »abstract». I det här dokumentet börjar sammanfattningen så här:

```
\begin{abstract}
\LaTeX\ är ett mycket kraftfullt
typsättningssystem. Det skapades av
Leslie Lamport och bygger på \TeX,
som i sin tur...
\end{abstract}
```

### 2.2 Stycken och avsnitt

Stycken markeras i  $\LaTeX$  med en blankrad i texten. Normalt beteende är att texten i det nya stycket typsätts på ny rad med ett litet indrag.

För att få denna text att inleda ett nytt stycke skrevs följande i  $\LaTeX$ -filen:

```
Stycken markeras i \LaTeX\ med en blankrad
i texten. Normalt beteende är att texten
i det nya stycket typsätts på ny rad med
ett litet indrag.
```

```
För att få denna text att inleda ett nytt
stycke skrevs följande i \LaTeX-filen:
```

Vi har redan sett att kommandot `\section` kan användas för att markera att ett nytt avsnitt inleds. På motsvarande sätt inleder kommandona `\subsection` och `\subsubsection` underavsnitt i olika nivåer. Dessa avsnitt numreras normalt; vill man inte ha något nummer lägger man till en asterisk efter kommandonamnet men före argumentet. Texten

```
\subsubsection*{Ett onummerat stycke}
```

typsätts exempelvis:

## Ett onummererat stycke

Numrerade stycken kommer automatiskt med i innehållsförteckningen, men för att få med innehållsförteckningen i dokumentet måste man lägga in kommandot `\tableofcontents` på den plats där man vill att innehållsförteckningen ska vara.

## 2.3 Framhäva delar av texten

Det är ofta vanligt att man vill framhäva enstaka ord i texten, exempelvis när man introducerar ett begrepp för första gången. I  $\LaTeX$  åstadkommer man enklast detta genom att använda kommandot `\emph{...}`. Exempelvis typsätts meningen »Ett tal är *rationellt* om det kan skrivas som en kvot av två heltal.» av texten

```
Ett tal är \emph{rationellt} om det kan
skrivas som en kvot av två heltal.
```

Om man citerar ett längre stycke text kan det vara praktiskt att innesluta den i en quote-omgivning. Texten

```
\begin{quote}
Emellertid kräver algoritmerna exponentiell
tid och matematiker är övertygade om att
det är omöjligt att konstruera enklare
algoritmer.
\end{quote}
```

typsätts som

```
Emellertid kräver algoritmerna exponentiell tid och matematiker är övertygade
om att det är omöjligt att konstruera enklare algoritmer.
```

Den citerade texten skiljer sig därmed klart och tydligt från omgivningen.

## 2.4 Tecken med speciell betydelse

Vissa tecken har speciell betydelse i  $\LaTeX$ . De vanligaste man stöter på är kombinationerna ‘ ’ , , << och >> som skapar olika varianter av citattecken: “ ” „ « ». De tre sista av dessa kombinationer fungerar bara om man har med raden

```
\usepackage[T1]{fontenc}
```

i dokumenthuvudet. Observera att man alltså aldrig ska använda tecknet " för att skapa ett citattecken. Bruket av de olika symbolerna för citattecken varierar; i engelsk text används oftast “ och ”, i tysk text ofta „ och ”. Mer information om svensk typografi finns i avsnitt 2.13.

Kombinationerna -- och --- skapar två olika typer av horisontella streck: *en-dash* (–) respektive *em-dash* (—). I engelsk text används det förra för att ange exempelvis sidhänvisningar, som i ss 17–21, medan det senare används som satsparentes. I svensk text används bara den korta varianten (se avsnitt 2.13).

Tecknen ^ och \_ kan bara användas i matematiska uttryck. De anger där exponenter respektive index;  $x^{a+b} = x^a x^b$  typsätts till exempel som  $x^{a+b} = x^a x^b$ . Tecknet & används för att separera kolumner i tabeller; för att få fram ett &-tecken måste man skriva `\&`. Tecknet \ inleder  $\LaTeX$ -kommandon. Ett kommando består normalt av bokstäver,

exempelvis finns kommandot `\LaTeX`, som skapar  $\LaTeX$ -logotypen. Efter ett kommando ignoreras blanksteg; vill man ha ett blanksteg måste man skriva `\` , dvs tecknet `\` följt av ett blanksteg:

```
\LaTeX är bra, \LaTeX\ är bättre
```

blir » $\LaTeX$  är bra,  $\LaTeX$  är bättre«. Vissa kommandon tar argument; de står normalt inom måsvingar. Om argumentet bara består av ett enda tecken kan man utelämna måsvingarna, men man får ha dem kvar om man vill.

Det enda kvarvarande specialtecknet av rang är `%`, som inleder kommentarer. Om det står ett procenttecken på en rad ignoreras all text från och med procenttecknet till och med radslut. Texten

```
Detta är % inte alls
sant
```

typsätts alltså som »Detta är sant«.

## 2.5 Icke-svenska bokstäver

Har man med raden

```
\usepackage[latin1]{inputenc}
```

i sitt dokument kan man skriva en lång rad mer eller mindre ovanliga bokstäver direkt i sitt dokument: `à`, `ï`, `å`, `æ`, `ø`, `ç` och `ñ` för att nämna några.  $\LaTeX$  har också ett antal kommandon för att skapa de bokstäver man inte får fram på tangentbordet. De vanligaste är `\'` , `\'` , `\^`  och `\^`  som skapar akut accent, grav accent, trema respektive circumflex. Exempelvis skapar `\'a` tecknet `à` och `\"i` tecknet `ï`. Lägga märke till att kommandot `\i` skapar ett `i` utan prick, `ı`, som man sedan sätter ett trema ovanför. På samma sätt skapar `\j` ett `j` utan prick: `j`. För att få fram bokstäverna `æ`, `ø` och `œ` finns kommandona `\ae` `\o` respektive `\oe` tillgängliga. Det finns också några mer exotiska accenter i  $\LaTeX$ , till exempel kan man skriva `Erd\H{o}s` för att få fram namnet på den ungerska matematikern Erdős och `Wa\l{k}{e}sa` genererar namnet på den polske Solidaritetsledaren och presidenten Wałęsa. Den sista accenten ovan fungerar bara om man har med raden

```
\usepackage[T1]{fontenc}
```

i sitt dokument.

## 2.6 Större matematiska uttryck

Vi har redan tittat på hur man kan skriva matematiska uttryck som står inne på en rad. Man kan också skriva mer komplicerade uttryck på helt separata rader:

$$\int \frac{dx}{\sqrt{1-x^2}} = \arcsin x + C$$

Ovanstående uttryck typsattes av följande text i  $\LaTeX$ -filen

```
\begin{displaymath}
\int \frac{dx}{\sqrt{1-x^2}}
=
\arcsin x + C
\end{displaymath}
```



Lägg märke till att radbrytningarna ovan inte har någon betydelse för hur formeln typsätts. Trots att  $\LaTeX$  lägger ner ganska mycket arbete på att typsätta matematiska formler så bra som möjligt måste man ibland göra manuella justeringar. Två användbara kommandon i detta läge är `\!` och `\,`, som drar ihop respektive skjuter isär; exempelvis kan man typsätta Stokes sats

$$\iint \vec{A} \cdot d\vec{S} = \oint \nabla \cdot \vec{A} \, dr$$

med följande text:

```
\begin{displaymath}
\int\!\!\int \vec{A} \cdot d\vec{S}
=
\int \nabla \cdot \vec{A} \, \!, \, dr.
\end{displaymath}
```

I vissa böcker typsätts integraler lite annorlunda mot vad som visats hittills – d:et i dx är då inte kursivt. Det enklaste sättet att åstadkomma det på är att definiera ett nytt kommando, låt oss kalla det `\ud` (för »upright d»), som typsätter d:et. Det görs på följande sätt:

```
\newcommand{\ud}{\mathrm{d}}
```

Kommandot `\mathrm` sätter sitt argument i det typsnitt som används för upprätt stil i matematiska formler. Vi kan sedan använda kommandot `\ud` för att typsätta Stokes sats:

$$\iint \vec{A} \cdot d\vec{S} = \oint \nabla \cdot \vec{A} \, dr.$$

```
\begin{displaymath}
\int\!\!\int \vec{A} \cdot \ud\vec{S}
=
\int \nabla \cdot \vec{A} \, \!, \, \ud r.
\end{displaymath}
```

Generellt är det ovanligt att använda uppräta d:n i integraler. Anhängare av systemet menar att eftersom d:et är en operator ska det typsättas i ett upprätt typsnitt precis som exempelvis operatoren sin; motståndarna hävdar att det är fullt att ha ett upprätt och ett lutande tecken ihop på det viset.

## 2.7 Listor och uppräknings

Ofta vill man dela upp en uppräknings i punkter.  $\LaTeX$  har inbyggt stöd för detta. Som exempel typsätts texten

```
\begin{enumerate}
\item En punkt.
\item En punkt till.
\begin{enumerate}
\item Underpunkter
\item skapas lätt.
\end{enumerate}
\item Sista punkten.
\end{enumerate}
```

på följande sätt:

1. En punkt.
2. En punkt till.
  - (a) Underpunkter
  - (b) skapas lätt.
3. Sista punkten.

Vill man inte ha numrerade listor byter man bara ut »enumerate» mot »itemize» ovan; då blir resultatet istället:

- En punkt.
- En punkt till.
  - Underpunkter
  - skapas lätt.
- Sista punkten.

## 2.8 Hänvisningar

Ibland behöver man hänvisa till andra delar av texten. I  $\LaTeX$  löser man detta med två kommandon: `\label` och `\ref`. Det förra sätter en etikett på en viss plats i dokumentet; det senare lägger in en hänvisning. Exempelvis inleds detta stycke med  $\LaTeX$ -koden

```
\subsection{Hänvisningar}
\label{hänvisning}
```

Om vi sedan lägger in `\ref{hänvisning}` någonstans i texten kommer numret på detta avsnitt (2.8) att sättas ut. För att hänvisningarna ska bli korrekta måste man ibland köra sitt dokument genom latex flera gånger. Ifall det finns risk för att någon hänvisning är fel skrivs varningen

```
LaTeX Warning: Label(s) may have changed.
Rerun to get cross-references right.
```

ut när man kör latex. Förutom stycken kan man hänvisa till punkter i numrerade listor, matematiska uttryck som skrivs inuti en equation-omgivning samt tabeller och figurer.

## 2.9 Tabeller

Att skriva en tabell i  $\LaTeX$  är enkelt men kräver lite eftertanke. Man använder en speciell omgivning, »tabular», för att göra tabellen. Man anger hur tabellens kolumner ska justeras genom att ge ett argument till omgivningen; i exemplet nedan är första kolumnen vänsterjusterad (l), andra kolumnen centrerad (c), tredje kolumnen högerjusterad (r). Tecknet | anger att det ska vara ett streck mellan andra och tredje kolumnen. Inuti omgivningen skrivs själva tabellen; kolumner avdelas med & och rader med `\\`. För att få horisontella streck använder man kommandot `\hline` inuti tabellen.

```

\begin{tabular}{lc|r}
Ett: & Denna kolumn centreras & 17 \\
\hline
Två: & Titta & 4711 \\
Tre: & och & 4 \\
\end{tabular}

```

blir typsatt som

Ett:	Denna kolumn centreras	17
Två:	Titta	4711
Tre:	och	4

### 2.9.1 Tabeller först på en rad

Normalt är det en liten marginal till vänster om den första kolumnen i tabeller. Skriver man tabellen först på en rad bör man ta bort denna, det ordnar formateringsargumentet @. Det betyder ”byt ut det normala mellanrummet mellan kolumner mot argumentet till @” – för att få bort mellanrummet helt använder man således @{}:

```

blir typsatt som\
\begin{tabular}{@{}lc|r}
Ett: & Denna kolumn centreras & 17 \\
\hline
Två: & Titta & 4711 \\
Tre: & och & 4 \\
\end{tabular}

```

blir typsatt som

Ett:	Denna kolumn centreras	17
Två:	Titta	4711
Tre:	och	4

### 2.9.2 Flyttbara objekt

Ofta vill man inte att tabellen ska stå mitt i texten.  $\LaTeX$  har då en mekanism för att definiera flyttbara objekt (*floats*) som kan sättas ut där det passar. Denna mekanism används normalt enbart för tabeller och figurer. För att göra en flyttbar version av ovanstående tabell kan man skriva

```

\begin{table}
\begin{center}
\begin{tabular}{lc|r}
Ett: & Denna kolumn centreras & 17 \\
\hline
Två: & Titta & 4711 \\
Tre: & och & 4 \\
\end{tabular}
\end{center}
\caption{En tabell som placeras ut där
den passar. Det kan vara bra med en
förklarande text till den. Om texten
ska placeras ovanför eller nedanför
tabellen beror på vilka typografiska
konventioner man använder.}

```

Ett:	Denna kolumn centreras	17
Två:	Titta	4711
Tre:	och	4

**Tabell 1.** En tabell som placeras ut där den passar. Det kan vara bra med en förklarande text till den. Om texten ska placeras ovanför eller nedanför tabellen beror på vilka typografiska konventioner man använder.

```
\label{tabell}
\end{table}
```

Omgivningen »center» centrerar sitt innehåll på raden. Med kommandot `\caption` kan man lägga in en förklarande bildtext. Om man vill kunna referera till sin tabell i texten måste man lägga in en etikett inuti table-omgivningen. Etiketten *måste* stå efter `\caption`-kommandot. De flyttbara tabellerna numreras; tabell 1 visar resultatet av ovanstående  $\LaTeX$ -kod.

## 2.10 Figurer och bilder

För att lägga in bilder i ett  $\LaTeX$ -dokument används kommandot `\includegraphics`. Det tar ett argument, namnet på den fil som innehåller bilden. Detta kommando är inte inbyggt i  $\LaTeX$  utan finns i tilläggs paketet »graphicx». För att få tillgång till det måste man därför ha med raden

```
\usepackage{graphicx}
```

i början av dokumentet. Det är i allmänhet enklast att alltid ha figurer i EPS-format; mer om detta i avsnitt 5.2. Figurer läggs normalt i en flyttbar omgivning, precis som tabeller:

```
\begin{figure}
  \begin{center}
    \includegraphics{bild.eps}
  \end{center}
  \caption{En bild som föreställer
    något fint.}
  \label{bild}
\end{figure}
```

## 2.11 Rad- och sidbrytningar

För det mesta väljer  $\LaTeX$  lämpliga rad- och sidbrytningar automatiskt, men ibland behöver man sätta in dem själv. Radbrytningar sätts in med kommandot `\newline`, som också kan skrivas `\\`. För att framtvunga en sidbrytning använder man kommandot `\newpage`. Det finns också ett kommando `\clearpage` som sätter in en sidbrytning och skriver ut alla flyttbara objekt som inte hunnit sättas ut någonstans ännu.

## 2.12 Inkludera oformaterad text

Ibland vill man att text ska typsättas precis som den ser ut. Vanligast är detta för källkod till dataprogram, men ibland får man av sina medarbetare tabeller som bara ser snygga ut om alla tecken är lika breda. För att lösa detta använder man en speciell omgivning. Till exempel kan vi typsätta det klassiska första C-programmet:

```
#include <stdio.h>

main()
{
    printf("hello, world\n");
}
```

genom att skriva in

```
\begin{verbatim}
#include <stdio.h>

main()
{
    printf("hello, world\n");
}
\end{verbatim}
```

I ett tilläggspaket som heter just »verbatim» definieras ett kommando `\verbatiminput` som kan användas för att läsa in textfiler och typsätta dem. Kommandot tar ett argument, namnet på filen som ska läsas in.

## 2.13 Svenska skrivregler

Eftersom många läser stora mängder engelsk text är det lätt att omedvetet använda engelska skrivregler även till svensk text. I detta avsnitt sammanfattas några av de viktigaste skillnaderna mellan svenska och engelska skrivregler.

I svenska används bara en längd på tankstreck, den längd man får genom att skriva `--` i  $\LaTeX$ -filen. När tankstreckets används som satsparentes – vilket det gör här – ska det alltid omgärdas av blanksteg.

Vad gäller förkortningar skiftar bruket i svenskan. Språknämnden rekommenderar att förkortningar som bildats genom avkortning av ord, exempelvis *civ.ing.*, avdelas med punkter utan mellanrum mellan de olika delarna i förkortningen. I förkortningar som bildats genom sammandragning av ord, exempelvis *hr* eller *ngn*, ska man inte använda punkt. Efter måttenheter ska det aldrig vara punkt.

Citattecken skrivs i svenska texter alltid åt »samma» håll och man behöver därför bara använda `'`. Som alternativ till citationstecknet kan man använda så kallade »gåsögon», dessa ska i svensk text enbart peka åt höger. Tecknet för gåsögon kan vara lite svårt att få fram – i Emacs kan man trycka `C-x 8 >` men andra redigeringsverktyg har normalt inte den funktionaliteten – som alternativ kan man skriva `>>` i  $\LaTeX$ -filen.

Observera att det tecken man får genom att trycka `Shift-2` på ett svenskt tangentbord inte ska användas, det genererar för det första inte ett korrekt citattecken (`"`) utan en helt rak variant (`"`) och för det andra används det för att ge specialinstruktioner till  $\LaTeX$ .

I svenskan ska vissa sammansatta ord avstavas på speciella sätt; till exempel ska ordet tillåta avstavas som till-låta. För att instruera  $\LaTeX$  om detta skriver man `ti"llåta`. Här betyder `"ll` alltså »ll som ska avstavas ll-l». Denna instruktion fungerar för `"bb`, `"dd`, `"ff`, `"gg`, `"ll`, `"mm "nn`, `"pp`, `"rr`, `"ss` och `"tt`.

I ord som innehåller bindestreck tillåter  $\LaTeX$  normalt ingen avstavning efter bindestrecket. Har man ett långt ord efter bindestrecket, exempelvis A-kasseersättning, vill man ofta kunna avstava även andra delen av ordet. Genom att skriva `A-"kasseersättning` i  $\LaTeX$ -filen tillåts avstavning även efter bindestrecket.

I svensk text använder man decimalkomma medan man i engelsk text ofta använder decimalpunkt. Detta ställer till ett mindre problem eftersom  $\LaTeX$  normalt lägger till ett litet mellanrum efter varje kommatecken i matematisk text –  $f(x, y) = 3,14$  typsätts därför  $f(x, y) = 3,14$ . I svensk text vill man ha kvar detta mellanrum efter det första kommatecknet ovan men inte efter det andra. Tilläggs paketet »icomma» löser detta genom att ändra funktionen hos kommatecknet i matematisk text: Om kommatecknet följs av ett blanksteg läggs ett mellanrum till, annars läggs inget mellanrum till. Om man använder paketet »icomma» typsätts alltså  $f(x, y) = 3,14$  som  $f(x, y) = 3,14$ .

För att göra tabeller med decimaljusterade tal kan man använda paketet »dcolumn». Det definierar en ny kolumn typ D som tar tre argument. Det första anger decimalseparatorn i  $\LaTeX$ -filen, det andra hur decimalseparatorn ska typsättas och det tredje hur många decimaler tabellen ska ha plats för:

```
\begin{tabular}{|D{,}{,}{2,5}|D{,}{,}{2,5}|}
  \multicolumn{1}{|c|}{En jättelång titel}
    & \multicolumn{1}{c|}{B} \\
  3,14159 & 3,14159 \\
  17      & 17 \\
  0,5     & 0,5
\end{tabular}
```

Tabellen har alltså två kolumner som båda decimaljusteras, kolumnerna åtskiljs av lodräta streck och har plats för två siffror före decimalseparatorn och fem efter. Kolumnrubrikerna kräver specialbehandling – de ska nämligen inte decimaljusteras. För att ange speciell formatering i enstaka rutor används kommandot `\multicolumn` som tar tre argument. Det första är antalet kolumner rutan ska gå över (en etta i exemplet ovan), det andra anger hur texten i rutan ska justeras (centreras med vertikala streck omkring i exemplet ovan), det tredje anger texten i rutan. Man behöver inte justera sin  $\LaTeX$ -kod så att `&`-tecknen kommer rakt under varandra, men det är lite lättare att läsa koden då.

En jättelång titel	B
3,14159	3,14159
17	17
0,5	0,5

Paketet »icomma» definierar om kommatecknet – det blir ett så kallat *aktivt tecken* – vilket gör att syntaxen `D{,}{,}{...}` slutar att fungera. Lyckligtvis tillhandahåller paketet »icomma» ett kommando `\mathcomma` som genererar det vanliga kommatecknet. Det går därför utmärkt att använda paketen »dcolumn» och »icomma» tillsammans om man byter ut `D{,}{,}{...}` i tabellhuvudet ovan mot `D{,}{\mathord\mathcomma}{...}`. För att slippa skriva så mycket är det nog enklast att i detta fall definiera en ny kolumn typ. Man kan kalla den vad man vill, jag har valt `k` nedan:

```
\newcolumn type{k}[1]{%
  {D{,}{\mathord\mathcomma}{#1}}}
```

Tabeller skrivs sedan på samma sätt som tidigare i  $\LaTeX$ -filen, fast man använder kolumn typen `k` istället. Eftersom kommandot `\mathcomma` definieras av paketet »icomma» fungerar ovanstående enbart om man använder paketen »dcolumn» och »icomma» tillsammans. Använder man bara paketet »dcolumn» kan man istället definiera kolumn typen med två kommatecken:

```
\newcolumn type{k}[1]{D{,}{,}{#1}}
```

Första kolumnen	Andra	Tredje
4,234	2,948	276,6
17,87	0,06	15,8
4711,0	17,6324	3,236

**Tabell 2.** Ett exempel på en trelinjetabell med decimaljusterade tal.

Man behöver naturligtvis inte ha vertikala streck i sina tabeller; det är i själva verket vanligt att istället ha tre horisontella streck, som i tabell 2. Den tabellen typsattes av följande  $\LaTeX$ -kod som använder den nya kolumntypen `k`, definierad enligt ovan:

```

\begin{table*}
\begin{center}
\begin{tabular}{k{4,3}k{2,4}k{3,3}}
\hline
\multicolumn{1}{c}{Första kolumnen}
& \multicolumn{1}{c}{Andra}
& \multicolumn{1}{c}{Tredje} \\
\hline
4,234 & 2,948 & 276,6 \\
17,87 & 0,06 & 15,8 \\
4711,0 & 17,6324 & 3,236 \\
\hline
\end{tabular}
\end{center}
\caption{Ett exempel på en trelinjetabell
med decimaljusterade tal.}
\label{trelinje}
\end{table*}

```

## 2.14 Svensk stavningskontroll

På Nadas Unix-datorer finns programmet »stava», ett program för stavningskontroll av svensk text, installerat i standardmiljön. Det tar ett filnamn som argument och skriver ut alla felstavade ord i filen. För att använda »stava» på  $\LaTeX$ -filer bör man ange väljaren »t» för att inte alla  $\LaTeX$ -kommandon ska skrivas ut som felstavade ord. Det är tillåtet att använda »stava» i ickekomersiella sammanhang; källkoden finns på Nadas FTP-area.

## 2.15 Svenska avstavningsmönster

Svenska avstavningsmönster finns med i de flesta nyare  $\TeX$ -distributioner. Ibland måste de aktiveras med hjälp av något konfigurationsprogram, se den aktuella  $\TeX$ -distributions dokumentation för detaljer.

## 2.16 Svensk referenslista

För att få svenska referenslistor med  $\BibTeX$  kan man använda stilarna `sweabbrv`, `swealpha` och `sweplain`, som ersätter standardstilarna `abbrv`, `alpha` respektive `plain`. De finns installerade på Nada och går att ladda ner från Nadas FTP-area »ftp://ftp.nada.kth.se/pub/tex/

local/sebst/». Tyvärr innebär inte de svenska filerna att man får referenslistan sorterad enligt svenska konventioner; sorteringsordningen är inkompilerad i BibTeX-programmet och kan inte ändras på något enkelt sätt. Istället får man helt enkelt gå igenom referenslistan manuellt och flytta de referenser som hamnar fel; de är vanligtvis inte så många.

### 3 Rapportskrivning med L<sup>A</sup>T<sub>E</sub>X

Att skriva en rapport med L<sup>A</sup>T<sub>E</sub>X blir mycket lättare om man tänker på ett par saker under arbetets gång. Först och främst ska man utforma sin rapport med tanke på publiken. Skriver man en laborationsrapport är huvudsyftet ofta att övertyga läraren om att man har förstått laborationen; skriver man en vetenskaplig artikel är målet oftast att få läsaren att förstå vad man har gjort. Naturligtvis får detta konsekvenser för den färdiga rapportens inriktning.

En svårighet vid rapportskrivning i största allmänhet är att man ofta samarbetar med andra. Man måste då utarbeta rutiner som gör att arbetet går smidigt. Man måste också tänka på att använda sina verktyg på rätt sätt – trots att L<sup>A</sup>T<sub>E</sub>X är bra på mycket ger det inte automatiskt ett perfekt resultat. Några tips på vägen ges nedan.

#### 3.1 Arbeta tillsammans med andra

Arbetar man tillsammans med andra måste man tänka på att alla deltagare i projektet ska kunna arbeta på dokumentet. Det innebär att *alla* filer som behövs för att köra dokumentet genom L<sup>A</sup>T<sub>E</sub>X måste finnas tillgängliga för alla projektdeltagare. Se därför till att inte använda några exotiska paket som kräver speciella systemresurser. Var också försiktig med att skicka L<sup>A</sup>T<sub>E</sub>X-filer med e-post. På grund av brister i tidig programvara lägger många posthanteringsprogram till ett »större än» först på varje rad som börjar med »From». Detta yttrar sig sedan som att det står antingen »>From» eller »¿From» istället för »From» när dokumentet körs genom L<sup>A</sup>T<sub>E</sub>X.

Det är bra att använda någon form av versionshantering vid rapportskrivning, både för att undvika att flera personer skriver på samma avsnitt samtidigt och för att kunna gå tillbaka och hämta en gammal version av rapporten ifall man hamnar i en återvändsgränd. Under Unix är systemen RCS och CVS vanliga; det finns stöd för dem även under andra operativsystem.

#### 3.2 Vanliga misstag

Trots att L<sup>A</sup>T<sub>E</sub>X i allmänhet gör en väldigt bra layout automatiskt måste man tänka på hur man instruerar L<sup>A</sup>T<sub>E</sub>X. I synnerhet är detta viktigt när man skriver matematisk text. Det gäller att hitta en lämplig balans mellan ord och matematiska symboler och en lämplig balans mellan matematiska uttryck som står inne på en rad och sådana som står på separata rader. Vissa symboler passar inte så bra inne på en rad – kanske ska man skriva  $x^{1/2}$  istället för  $\sqrt{x}$  och  $(a + b)/2$  istället för  $\frac{a+b}{2}$  i löpande text. Att skriva ett matematiskt uttryck på en egen rad ger det en extra tyngd och indikerar för läsaren att det är mer centralt än de uttryck som står i texten. Samtidigt vill man inte stycka sönder texten med en massa korta matematiska uttryck som står på egna rader.

Det finns många matematiska symboler definierade i L<sup>A</sup>T<sub>E</sub>X och ännu fler finns definierade i olika tilläggs paket. Själv tycker jag att man ska införa så få symboler som möjligt – det finns till exempel ingen anledning att använda  $\mathbb{R}$  för de reella talen, ett **R** duger lika bra



och passar bättre ihop med den omgivande texten. Försök att hela tiden fundera på vad som är det centrala budskapet i texten och hur läsaren bäst tar till sig det.

Fotnoter är något annat som kräver eftertanke. Det finns naturligtvis inte några universella regler för vad man ska och inte ska göra, men jag tycker att fotnoter i princip aldrig ska användas. Om det som står i fotnoten är viktigt ska det istället stå med i själva texten; om det bara är en sidokommentar som inte tillför något ska det inte stå med alls. När man läser direkt i L<sup>A</sup>T<sub>E</sub>X-filen står fotnotstexten alltid tillsammans med texten den refererar till, och då kan det kanske se naturligt ut, men när texten typsätts hamnar fotnoten i allmänhet långt ifrån texten den refererar till och då blir resultatet ofta svårläst.

Ett annat misstag som många gör i början är att använda listor alldeles för mycket. Eftersom det är enkelt att skapa listor med L<sup>A</sup>T<sub>E</sub>X är det lätt att använda dem för mycket. Om man har en lista med väldigt många och/eller långa punkter i sitt dokument bör man fundera på om man kan uttrycka det man vill få sagt på något annat sätt.

## 4 Lokala Nada-paket

Det finns några lokalt utvecklade paket på Nada. Som exempel finns dokumentklasser för examensarbeten, avhandlingar och brev. Aktuell dokumentation och beskrivningar av de olika paketen finns på Nadas webbplats »<http://www.sgr.nada.kth.se/misc/tex/>» samt i katalogen »/pkg/teTeX/share/texmf-local/doc/latex» på Nadas Unix-datorer.

Det går även att använda de lokalt utvecklade paketen utanför Nada; de kan laddas ner färdigpacketerade från Nadas FTP-area »<ftp://ftp.nada.kth.se/pub/tex/local/>». De flesta moderna TeX-installationer har en eller flera kataloger för lokala tillägg – arkiven som kan laddas ner från »<ftp://ftp.nada.kth.se/pub/tex/local/>» ska packas upp i en sådan katalog *med katalogstrukturen intakt*. Använder man teTeX även hemma är det förmodligen enklast att installera de lokala Nada-paketen i katalogen »~/texmf» men det går också att installera dem i »/usr/local/teTeX/share/texmf-local». Använder man MiKTeX ska paketen normalt sett installeras i »C:\localtexmf». När arkiven packats upp måste den så kallade *filnamnsdatabasen* uppdateras – i teTeX görs det genom att man kör kommandot »texhash», i MiKTeX använder man det grafiska konfigurationsgränssnittet.

Om du planerar att använda något av de lokala Nada-paketen för att skriva dokument tillsammans med andra, tänk på att tidigt kontrollera att det går bra för alla dina medarbetare att använda paketen. Samma sak gäller naturligtvis om du planerar att arbeta med ett dokument på många olika platser.

### 4.1 Modifiering av Nadas TeX-installation

Om man behöver lägga till egna paket till Nadas TeX-installation eller modifiera beteendet hos TeX-programmen är det förmodligen enklast att skapa en katalog »texmf» i sin hemkatalog. Den katalogen söks automatiskt igenom av Nadas TeX-installation. Filstrukturen ska vara enligt *TeX Directory Standard*, TDS. En definition av TDS finns på nätet, för det mesta räcker det att veta att L<sup>A</sup>T<sub>E</sub>X-relaterade filer ska vara i katalogen »~/texmf/tex/latex», avstavningsmönster i »~/texmf/tex/generic» och referensdatabaser i »~/texmf/bibtex/bib».

Vill man ändra något som ligger mer djupt i installationen, till exempel lägga till något avstavningsmönster och/eller installera något eget PostScript-typsnitt måste man skapa katalogerna ».texmf-config» och ».texmf-var» i sin hemkatalog. Därefter kan man använda de konfigurationsprogram som kommer med teTeX för att ändra inställningar. Ändrade konfigurationsfiler kommer då att skrivas till »~/texmf-config» och ändrade formatfiler skrivs till

»~/texmf-var». Sökvägen till konfigurationsfilen »texmf.cnf» är dock inkompilerad i  $\TeX$ -binärerna; har man behov av att ändra den måste man sätta miljövariabeln `TEXMFCNF` till att peka på katalogen där den ändrade varianten ligger. Det huvudsakliga konfigurationsprogrammet i  $\TeX$  heter »texconfig» – det är menystyrt och tämligen självinstruerande.

## 5 Vanliga problem

Det finns några få saker som många  $\LaTeX$ -användare får problem med. I detta avsnitt finns förklaringar till dessa problem och förslag på lösningar. Det finns många så kallade FAQ:er – samlingar med vanliga frågor och svar till dessa – på nätet, en av de bästa underhålls av *The UK  $\TeX$  Users Group* och finns tillgänglig på »<http://www.tex.ac.uk/cgi-bin/texfaq2html?introduction=yes>».

### 5.1 Konvertera $\LaTeX$ till PDF

Det format som idag verkar vara bäst för publicering på webben är PDF (*Portable Document Format*). PostScript är visserligen väldigt vanligt i Unix-världen, men helt okänt för PC-användare. Det går i allmänhet utmärkt att skapa PDF-filer från  $\LaTeX$ -dokument, men man måste tänka på några saker.

#### 5.1.1 $\LaTeX$ -fil till DVI till PostScript till PDF

Den enklaste vägen att få en PDF-fil från sin DVI-fil är att generera en PostScript-fil med »dvips» och sedan göra om PostScript-filen till PDF med kommandot »distill» (senaste versionen nås med »module add distiller» på Nadas Solaris-datorer). För att få bra kvalitet på PDF-dokumentet är det mycket viktigt att använda enbart så kallade *Type1-typsnitt* i PostScript-filen. Normalt använder »dvips» enbart speciella bitkartetypsnitt som genereras med programmet MetaFont, men genom att lägga till väljaren »-Ppdf» kan man (på Nadas Solaris-datorer) få Type1-versioner av många av de vanliga  $\TeX$ -teckensnitten. Vi har site-licens på dessa typsnitt och det är tillåtet för oss att publicera dokument som innehåller dessa typsnitt på webben. Använder man en annan  $\LaTeX$ -installation måste man själv se till att rätt sorts typsnitt läggs in i PDF-filen. De flesta moderna färdigpaketerade distributioner har redan stöd för detta med väljaren »-Ppdf» till »dvips». Symptom på att bitkartetypsnitt används är att tecknen ser kantiga ut när PDF-filen visas på skärmen. I Acrobat Reader kan man kontrollera vilka typsnitt som finns med i PDF-filen med meny `File`→`Document Info`→`Fonts`. Det ska stå *Type 1* vid varje typsnitt.

Man bör också explicit ange sidformat för att få rätt sidformat i PDF-filen. För att få A4-format använder man väljaren »-ta4» till »dvips», för att få letter-format väljaren »-tletter». Tag för vana att alltid kontrollera sidformatet i Acrobat Reader när PDF-filen har skapats.

Bilder bör vara i vektorformat (se avsnitt 5.2). Om man använder bitkartebilder måste man se till att de får en lämplig upplösning. För det första måste bildfilen i sig ha tillräckligt hög upplösning. För det andra måste man se till att Distiller-programmet inte ändrar upplösningen. Inställningen *PrintOptimized* är normalt lämplig om man inte har mycket stora bilder och behöver få ner filstorleken på bekostnad av den grafiska kvalitén. På Mac eller Windows väljs detta i Distiller-programmets grafiska gränssnitt. I Solaris används väljaren »-joboptions +PrintOptimized.joboptions» (kör »distill -help» för mer information om väljare).

### 5.1.2 L<sup>A</sup>T<sub>E</sub>X-fil direkt till PDF

Det finns en utökad version av L<sup>A</sup>T<sub>E</sub>X som skapar PDF-filer direkt istället för DVI-filer. Den heter pdfL<sup>A</sup>T<sub>E</sub>X och körs normalt med kommandot »pdflatex». De senaste versionerna av pdfL<sup>A</sup>T<sub>E</sub>X fungerar i allmänhet mycket bra – när detta skrivs är den senaste versionen 1.10b – tidigare versioner hade vissa problem.

Även i detta fall är det viktigt att använda vektortypsnitt och pdfL<sup>A</sup>T<sub>E</sub>X stöder både Type1- och TrueType-typsnitt. Om man typsätter det allra första exemplet i häftet med kommandot »pdflatex» får man ungefär följande utmatning:

```
This is pdfTeX, Version 3.14159-1.10b
(Web2C 7.4.5) ... LaTeX2e <2001/06/01>
...
(./tiny.aux) )
<.../type1/bluesky/cm/cmr10.pfb>
<.../type1/bluesky/cm/cmbx12.pfb>
<.../type1/bluesky/cm/cmr12.pfb>
<.../type1/bluesky/cm/cmr17.pfb>
Output written on tiny.pdf
(1 page, 19791 bytes).
Transcript written on tiny.log.
```

De sista raderna ovan anger vilka typsnitt som läggs in i PDF-filen. Filer som har ändelsen pfa eller pfb är i allmänhet Type1-typsnitt, ändelsen ttf anger TrueType-typsnitt. Ser man filer med ändelsen av typen 600pk eller liknande tyder det på att bitkartetypsnitt har använts.

Om man använder pdfL<sup>A</sup>T<sub>E</sub>X kan man inte använda EPS-bilder, alla bilder måste konverteras till PDF-format. Detta kan göras med Distiller; kör man Linux kan man istället använda programmet »epstopdf» eller »convert». Man kan använda samma L<sup>A</sup>T<sub>E</sub>X-fil till både vanliga L<sup>A</sup>T<sub>E</sub>X och pdfL<sup>A</sup>T<sub>E</sub>X om man har följande konvention för bilder: Skapa både en EPS-version och en PDF-version av varje bild och lägg dem i filer med samma namn och ändelserna eps respektive pdf; om bildfilerna heter »bild.eps» respektive »bild.pdf» ska de sedan läsas in med kommandot

```
\includegraphics{bild}
```

i L<sup>A</sup>T<sub>E</sub>X-filen. Beroende på om L<sup>A</sup>T<sub>E</sub>X eller pdfL<sup>A</sup>T<sub>E</sub>X används kommer då »bild.eps» respektive »bild.pdf» att läsas in. Om man skapar sina bilder i MetaPost (se avsnitt 5.2) kan de läsas in direkt av pdfL<sup>A</sup>T<sub>E</sub>X ifall man lägger de färdiga EPS-bilderna i filer med ändelsen mps.

Numera kommer de flesta färdigpaketerade L<sup>A</sup>T<sub>E</sub>X-system med fullt fungerande stöd för pdfL<sup>A</sup>T<sub>E</sub>X och ett stort utbud av typsnitt i vektorformat. Situationen kompliceras dock av att det länge inte fanns några gratistypsnitt som kunde användas tillsammans med svenska dokument. Raden

```
\usepackage[T1]{fontenc}
```

som måste finnas med för att svensk avstavning ska fungera gör nämligen att en typsnittsfamilj som länge inte fanns i Type1-format används. Numera finns ett paket vid namn »CM-super» som löser detta, men det finns inte med i så många färdigpaketerade lösningar. Det går dock att ladda ner från nätet och det är enkelt att installera. En annan lösning på problemet är att använda paketet *Latin Modern* som laddas med raden

```
\usepackage{lmodern}
```

För närvarande (januari 2005) ser det ut som att den senare lösningen, med paketet *Latin Modern*, är den som kommer att stödjas i de stora T<sub>E</sub>X-distributionerna.

## 5.2 Konstruera snygga bilder

De bilder som enklast blir snygga i  $\LaTeX$ -dokument är skalbara linjebaserade illustrationer. Det går naturligtvis att inkludera bitkartebilder, både i gråskalor och i färg, men då måste man anpassa bildens kvalitet efter den skrivare dokumentet ska skrivas ut på.

### 5.2.1 Vektorbaserade ritprogram

För att skapa en skalbar linjebaserad bild måste man naturligtvis använda ett program som stöder detta. På Unix-system finns i allmänhet programmet »xfig» installerat. Det finns en version av »xfig» även till Windows-baserade system; man kan också använda Adobe Illustrator. Se till att din bild exporteras som en EPS-fil; då klarar  $\LaTeX$  garanterat av att läsa den. Tyvärr är det vanligt att program som hävdar att de genererar EPS-filer i själva verket genererar något annat. Under Windows brukar det ofta hamna skräpcken både i början och i slutet av filen. Dessa måste man ta bort för hand. En EPS-fil ska börja med %! och slutar normalt med %%EOF.

Nästa komplexitetsnivå kommer när man vill lägga in text i sina figurer. Ofta vill man att texten ska ha samma typsnitt som brödtexten i dokumentet, ibland vill man använda matematiska formler i texten. Det finns stöd för detta i »xfig» och i princip bygger lösningen på att bilden separeras i dels en linjebild utan text och dels en samling  $\LaTeX$ -kommandon som ritar ut den text man vill ha.

Ett annat verktyg som genererar linjebaserade bilder och har stöd för att lägga in text genererad med  $\LaTeX$  i bilden är MetaPost. Det fungerar ungefär som MetaFont – programmet som genererar de typsnitt som ursprungligen användes till  $\LaTeX$  – men det genererar bilder istället för tecken. MetaPost är mycket kraftfullt men har en högre inlärningströskel än »xfig» och Adobe Illustrator. Bilderna skapas genom att man skriver ett litet program som ritar ut kurvor och texter, detta program konverteras sedan till en EPS-fil av MetaPost. En stor fördel med MetaPost är att bilder genererade med programmet kan läsas in direkt i pdf $\LaTeX$  utan att först konverteras till PDF. Det följer med både en introduktion och en referensmanual till MetaPost med de flesta moderna  $\LaTeX$ -distributioner; på Nadas Unix-datorer ligger de i katalogen »/pkg/teTeX/default/share/texmf-dist/doc/metapost/base».

### 5.2.2 Bitkartebilder

Använder man  $\LaTeX$  är det förmodligen enklast att konvertera sin bitkartebild till EPS-format; då kan den läggas in precis som alla andra EPS-bilder. Använder man däremot pdf $\LaTeX$  bör man istället lagra sina bilder i PNG-format, dessa bilder kan läsas in direkt av pdf $\LaTeX$ . Tänk på att upplösningen i bilden måste anpassas efter den skrivare som ska skriva ut dokumentet. Skapar man PDF-filer av sina dokument kan det vara lämpligt att ha versioner anpassade för utskrift respektive läsning på skärmen. Ett mycket användbart kommando för att konvertera bilder mellan olika format är »convert». På Nadas Unix-datorer ligger kommandot i gnome-modulen; på Linux-system är kommandot i allmänhet tillgängligt i standardmiljön.

## 5.3 Flytta filer mellan system

Trots att  $\LaTeX$  är väldigt portabelt kan man stöta på problem när man flyttar dokument mellan olika datorsystem. Störst risk för problem är det naturligtvis om man flyttar dokument

mellan olika operativsystem, men det är inte helt riskfritt att flytta ett dokument mellan två datorer som kör samma operativsystem.

### 5.3.1 Teckenkodning

Man måste bestämma sig för vilken teckenkodning  $\LaTeX$ -filen ska använda om man vill kunna flytta den mellan olika operativsystem. Givet att man har angivit teckenkodningen med `inputenc`-paketet kan man alltid köra filen genom  $\LaTeX$ , oberoende av vilket operativsystem man är på. Däremot kan det vara svårt att redigera filen om den har fel teckenkodning.

Generellt är det för närvarande bäst att använda teckenkodningen *Latin-1* om man ska flytta dokument mellan olika plattformar. Det är då viktigt att inte stoppa in några Windows-specifika tecken i filen när man redigerar den under Windows. Den teckenkodning som normalt används under Windows, *Microsoft Codepage 1252*, omfattar *Latin-1* men innehåller tecken på positioner som ska vara tomma enligt *Latin-1*. Redigerar man filen under MacOS X är det viktigt att man konfigurerar sina redigeringsprogram så att filen sparas i *Latin-1*.

Om man skriver på engelska och endast behöver mata in ett fåtal icke-engelska tecken kan det vara enklast att använda vanlig ASCII – dvs inte använda `inputenc`-paketet alls – i synnerhet om man arbetar på dokumentet tillsammans med människor i andra länder. Man måste då mata in svenska tecken med den speciella  $\TeX$ -syntaxen `\r{a}` för å, `\"a` för ä, osv.

I nyare  $\TeX$ -distributioner finns visst stöd för teckenkodningen UTF-8. Vanligtvis understöds då text med latinskt och kyrilliskt alfabet.

### 5.3.2 Olika typsnitt och paket

Det är viktigt att de typsnitt och tilläggspaket du använder finns på alla system där dokumentet ska behandlas. Använder man en färdigpaketerad  $\LaTeX$ -installation utan att själv installera något extra är man i allmänhet på säkra sidan. På Nada finns ett antal lokala paket som beskrivs i avsnitt 4. Dessutom finns den kommersiella typsnittsfamiljen *TMMath* som inte säkert finns på andra system.

### 5.3.3 Olika nyradtecken

Olika operativsystem använder olika konventioner för att markera att en ny rad börjar. Använder man Emacs kan man konvertera mellan olika nyradskonventioner. Ofta fungerar  $\LaTeX$  oberoende av nyradskonvention, men hanteringen av EPS-bilder är mer känslig. Den vanligaste orsaken till felmeddelandet

```
! Unable to read an entire line
Please increase buf_size in texmf.cnf.
```

är att man använder en EPS-bild genererad på en Mac under Unix. Eftersom Mac och Unix har olika nyradtecken tolkas hela bilden som en enda lång rad, och eftersom LaTeX läser in bilden radvis för att ta reda på hur stor den är blir inbufferten full. Om man då byter nyradtecken i filen från ASCII 13 (CTRL-M) till ASCII 10 (CTRL-J) fungerar allt utan att man behöver ändra `buf_size`. Om man trots allt måste öka `buf_size` är det enklast att lägga en kopia av filen »`texmf.cnf`» i någon lämplig katalog och sedan sätta miljövariabeln »`TEXMFCNF`» till att peka på den katalogen. Sedan kan man ändra parametrar i sin lokala kopia av konfigurationsfilen.

## 5.4 Att köra L<sup>A</sup>T<sub>E</sub>X hemma

Att själv administrera ett L<sup>A</sup>T<sub>E</sub>X-system är en smula komplext. Absolut enklast är att använda någon färdigpaketerad installation. För Unix finns te<sub>E</sub>X av Thomas Esser och för Windows finns MiK<sub>T</sub>E<sub>X</sub> av Mikael Schenk. Dessutom finns projektet *T<sub>E</sub>X Live* som tillhandahåller CD-skivor med färdiga L<sup>A</sup>T<sub>E</sub>X-system för många olika plattformar.

De färdigpaketerade systemen är i allmänhet utformade så att man kan installera egna tilläggspaket och det finns beskrivet i dokumentationen hur man går tillväga. Eftersom L<sup>A</sup>T<sub>E</sub>X är mycket filintensivt finns det ofta en databas över alla filer i systemet. Denna databas måste uppdateras när man har lagt till ett eget paket, annars hittar inte L<sup>A</sup>T<sub>E</sub>X paketet. Exakt hur det går till varierar mellan olika system och finns beskrivet i den dokumentation som kommer med systemet.

## 6 Framtiden

Trots att själva T<sub>E</sub>X-programmet inte utvecklas längre finns det några vidareutvecklingar på gång. Den första, som har funnits ganska länge, är e<sub>T</sub>E<sub>X</sub>. Den innehåller ett fåtal primitiver som åtgärdar brister i T<sub>E</sub>X. En mer radikal vidareutveckling är Omega, som har som mål att kunna typsätta alla världens språk. Det klarar därför av att skriva text i olika riktningar och kan hantera många fler typografiska konventioner än T<sub>E</sub>X. Tyvärr finns det mycket sparsamt med dokumentation om Omega-projektet och programmet ändras ofta ganska mycket. Detta gör att man kan behöva ändra i sina dokument när en ny version av Omega släpps. Mer information om Omega finns på »<http://omega.cse.unsw.edu.au:8080/>».

Som alternativ till L<sup>A</sup>T<sub>E</sub>X finns ConT<sub>E</sub>Xt. Det är rimligt färdigt och verkar mer välde-signat än L<sup>A</sup>T<sub>E</sub>X, i synnerhet om man vill skriva icke-teknisk text. En brist i L<sup>A</sup>T<sub>E</sub>X ser man ganska tydligt i dokument som har texten i två kolumner. Man skulle då gärna vilja att baslinjerna alltid låg på samma höjd i de två kolumnerna. L<sup>A</sup>T<sub>E</sub>X är tyvärr inte konstruerat med detta i åtanke och det är mycket komplicerat att konfigurera om L<sup>A</sup>T<sub>E</sub>X på ett sådant sätt. ConT<sub>E</sub>Xt, å andra sidan, har stöd för detta inbyggt som standard. ConT<sub>E</sub>Xt används normalt tillsammans med pdf<sub>T</sub>E<sub>X</sub> och har bra integrerat stöd för bildhantering. Nackdelen är naturligtvis att det är svårt att samarbeta med andra, som inte använder ConT<sub>E</sub>Xt. Mer information om ConT<sub>E</sub>Xt finns på »<http://www.ntg.nl/context/>».

Det finns också projekt som försöker skapa ett grafiskt användargränssnitt till L<sup>A</sup>T<sub>E</sub>X. Tanken är att man ska kunna utnyttja den textformatering som L<sup>A</sup>T<sub>E</sub>X tillhandahåller och samtidigt få ett mer intuitivt sätt att mata in text på. På Nadas Unix-datorer finns en sådan miljö installerad. Den heter Ly<sub>X</sub> och nås med »`module add lyx`»; kommandot »`lyx`» startar sedan programmet. Mer information om Ly<sub>X</sub> finns på »<http://www.lyx.org/>».