

L^AT_EX på 10 minutter

Christian Bohr-Halling
<cbh@ruc.dk>

5. august 2004

Resumé

Nærværende tekst er en *meget* kort beskrivelse af det allermest nødvendige for at kunne starte skrivning under L^AT_EX med „skabelonen“ til projektrapporter fra akira.ruc.dk/~cbh/c/RUClatex.html. Der introduceres helt generelt til L^AT_EX. Dernæst beskrives, hvordan man f.eks. bruger lister, grafik, referencer og tekstformatering. Teksten blev i grundform skrevet meget hurtigt, meget kort og meget sent-om-natten til PIP-gruppen på NatBas, RUC efteråret 2003. Mindre opdateringer er senere kommet til.

Hvad er L^AT_EX

L^AT_EX generelt

L^AT_EX er et opmærkningssprog/system (markup language) til tekstformatering lige som f.eks. HTML. Det vil sige, at man under skrivning af en tekst angiver en strukturel/logisk beskrivelse (f.eks. angivelse af afsnit, fremhævet, fodnote osv.)¹.

L^AT_EX er oprindeligt skabt af Leslie Lamport i 1985 som en makropakke til TeX. LaTeX indeholder kommandoer på højere abstraktionsniveau i forhold til TeX og gør det dermed lettere at anvende.

Anvendelse af LaTeX har især vundet indpas til tekstproduktion inden for naturvidenskaben, specielt matematik og datalogi. Det skyldes blandt andet, at det er let at skrive symbolske formler så som matematiske formler, men der findes også et utal af pakker til L^AT_EX, der anvendelsesmæssigt udvider L^AT_EX' muligheder. Der findes f.eks. en pakke til node-skrivning.

L^AT_EX konkret

L^AT_EX tager kun rent tekstinput. Det vil sige, skrivning i Notepad er godt nok, og Word er overkill. Kildefilerne, det vil sige det rene tekstinput, til L^AT_EX skal behandles for at give et seværdigt formateret output. Der anvendes i aktuelle sammenhæng pdf_latex, der dirkete laver PDF-filer, men ellers er det „normale“ at få output i et specielt format kaldet .dvi (Device Independent), som så kan konverteres til PostScript og PDF. Til Windows findes L^AT_EX fx ved MikTeX, der frit kan hentes og anvendes².

Den L^AT_EX-fil, der samler det hele (i vores tilfælde er det mainPROJEKT.tex), skal køres med pdf_latex.exe: pdf_latex mainPROJEKT.tex. En rapport består af flere forskellige kildefiler, og disse inkluderes i mainPROJEKT.tex, som de skrives. Det gøres under delen

¹Oftes defineres den konkrete stilmæssige udformning særskilt fra den strukturelle/logiske

²Den lille distribution er nok, evt. ekstra pakker kan hentes senere/automatisk

„% *** SELVE INDHOLDET“ vha. `\input kapitel5PROJEKT.tex` („.tex“ kan udelades, hvis filen faktisk hedder dette til sidst).

Hvis man ikke gider at køre fra kommandolinjen, så er der fx følgende editore: WinShell (hvis jeg ikke husker galt, finder denne selv ud af opsætning af L^AT_EX (MikTeX)) og Crimson Editor. Disse kan sættes op til at køre pdflatex.

Diverse overordnede observationer

En kommando i L^AT_EX afgives som `\kommando[valgfrie par]{obligatorisk par}`. Man kan også komme ud for at skulle bruge såkaldte miljøer, hvor man inden i disse skriver en oftes større mængde tekst og kommandoer; et miljø er på formen `\begin{et miljø} bla bla\end{et miljø}`

Der er visse beskyttede tegn, man ikke kan skrive direkte. De skal skrives med en `\` foran. Disse tegn er bl.a. \$, &, %, {, }.

Når man oversætter til PDF-formatet, det vil sige kører pdflatex.exe på mainPROJEKT.tex, må der ikke være en åben udgave af en tidligere oversættelse i Acrobat, da den så brokker sig. Hvis man bruger GhostView i stedet, kan man til tider godt oversætte filerne mens en tidligere PDF-outputfil er åben. I denne sammenhæng bør det også nævnes, at man skal oversætte mindst 3 gange før indsholdsfortegnelse bliver opdateret, hvis altså man har ændret i tekst relevant herfor.

At skrive gåseøjne er en sag for sig selv. Den danske måde at gøre det på er „sådan“. I L^AT_EX gør man således for at skrive dette: ”sådan” — altså SHIFT-2+SHIFT-denVedBackSpace og til afslut SHIFT-2+denVedØ.

Struktur på dokumentet

Når man i L^AT_EX vil lave kapitel og afsnit gør man det, som meget andet, ved hjælp af kommandoer: `\chapter{Kapitel om neuronerne i et neuralt netværk}`. Der findes selvfølgelig flere forskellige niveauer, og man starter i teksten med det „største“. De er: `\chapter{}`, `\section{}`, `\subsection{}`, `\subsubsection{}`, `\paragraph{}`, `\subparagraph{}`. Det mindste afsnit dannes uden en kommando (wow!); i kildekoden laver man blot en tom linje mellem to tekststykker – resultatet er en ny linje med et indhak i teksten præcist som afsnittet heruder.

I særlige tilfælde kan man lave en eller flere nye linjeskift med `\\`, men denne metode skal ikke bruges som et pseudoafsnit, når man vil have lidt luft mellem de mindste afsnit (altså dem, der i koden afskildes af en tom linje og som resultat i PDF giver en ny linje plus et indryk). Dertil bruger man `\paragraph{}` evt. som her uden et indhold i{}.

Det er smart at have en seperat fil til hvert `\chapter{ }`, altså til hvert kapitel af rapporten. På den måde er det endnu lettere at forskellige personer arbejder på forskellige dele. I tilfældet af store afsnit, kan disse også deles ud på flere filer, men for mange filer kan nu også være forvirrende. Under alle omstændigheder bør man ikke inkludere underafsnitsfiler i afsnits-filer, da det let kan blive et mindre mareridt at finde ud af, hvor hvad står i kildefilerne. Inkluderer i stedet alle filer i mainPROJEKT.tex-filen.

At lave lister/punktopstilling

Et eksempel siger det hele...

```
1 \begin{itemize}
2   \item noget
3   \item noget
```

```
4 \end{itemize}
```

Dette giver følgende ...

- noget
- noget

Det er også muligt at „neste“ lister. Dette gøres ved at begynde et nyt „miljø“ (`\begin{itemize}...`) dér, hvor der normalt skulle have stået `\item` bla.

Tabeller

Beskrivelse ved et eksempel, man kan arbejde ud fra. Hvis der skal noget „fancy“ på, må det komme senere.

Type	Navn	Kendetegn
a	Q	Spiser
a	S	Tygger
d	D	Lever

Koden til eksempelet...

```
1 \begin{tabular}{rlc} %r: right, l: left, c: center
2 Type & Navn & Kendetegn\\
3 a & Q & Spiser\\
4 a & S & Tygger\\
5 d & D & Lever
6 \end{tabular}
```

Tekstformatering

At skrive med kursiv eller fed gøres hhv. med `\textit{aaaaaa}` og `\textbf{aaaaaa}`

At skrive en kommentar i kildekoden, dvs. noget \LaTeX ikke ser på, gør man ved at sætte `%` foran linjerne med kommentarer. Hvis man vil skrive en lille kommentar til ens læsere af PDF-filen, så gøres det i stedet med `\marginpar{Skal dette bulumpaafsnit med?}`

Orddeling er noget \LaTeX selv tager sig af³. Hvis den mod forventning deler et ord forkert, kan man gå ind og vise \LaTeX , hvordan det pågældende ord må deles. Man skriver `\- de steder i ordet, hvor en deling må forekomme, fx bu\~lum\~pa`

Matematik

At skrive matematik i \LaTeX er en sag for sig selv, og den gider jeg ikke beskrive andet end meget grundlæggende, da det meste alligevel er noget, man skal slå op fx mht. tegn. (se fx www.ams.org/tex/short-math-guide.html).

At skrive et enkelt-linje matematisk stykke inden i teksten, altså som en del af teksten, gøres vha. `\(2^x \)`. I tilfældet, man ønsker det på ny linje og ikke blandet sammen med teksten på en linje, gøres det vha. `\[2^x \]`. Et eksempel på af skrive på en linje (første metode) $\frac{3^{2x}}{4^i}$ skrives som: en masse tekst og midt i denne noget uha `\(\frac{3^{2x}}{4^i}\)` og så noget mere tekst.

³husk iøvrigt at sætte MikTeX op til dansk under MikTeX-mappen, MikTeX-options, Languages, Danish.

At skrive multi-linjer (justeret) kan gøres vha. miljøet align som følgende...

```
1 \end{align}
2 (x + y) \cdot (x + \overline{y}) &= x + (y \cdot \overline{y}) \\
3 &= x + 0 \\
4 &= x
5 \end{align}
```

... der ser sådan ud...

$$(x + y) \cdot (x + \bar{y}) = x + (y \cdot \bar{y}) \quad (1)$$

$$= x + 0 \quad (2)$$

$$= x \quad (3)$$

(og nej, + og \cdot er i dette eksempel ikke hhv. addering og multiplikation, men står for andre operationer)

Grafik

At indsætte billeder er relativt nemt, når altså først det er sat op. Man skal bare huske, at kun følgende typer kan bruges: TIFF, PNG og JPEG.

```
1 \begin{figure}[h]
2   \centering
3   \scalebox{2.2}{\includegraphics{Bulumpa.jpg}} %scalebox: ændr størrelse
4   \caption{Billede af en Bulumpa} %Billedetekst
5   \label{BillBup} %Så man kan lave en reference til billede
6 \end{figure}
```

Referencer rundt i teksten

Hvordan laver man så en krydsreferenece til et billede — og for den sags skyld andre ting — man har sat en `\label{referencen}` på? Jo, man skriver da bare `\ref{referencen}`

Nu er det jo sådan, at RUC-rapporter meget gerne skal have kildereferencer — i hvert fald, hvis man ønsker at bestå. Sådan en tilføjes ganske let. I filen `referencesPROJEKT.tex` tilføjer man følgende for hver ny kilde-reference, der bruges i rapporten: `\bibitem[læserensRef]{latexsRef}`Selve referencen. Når man så et sted i rapporten skal lave en reference til kilden gør man følgende `\cite[evt side]{latexsRef}`

Fodnoter sættes lige efter teksten, den skal stå til. Den sættes med `\footnote{noget noget noget}`

Skrivning af programkode

Inkludering af programkode på en flot måde kan ske på følgende måder. 1) Kildekode inline, det vil sige korte kodelinjer midt i et afsnit: `\lstinline [language=prolog]!prolo-kildekoden!`. 2) Kildekoden for sig selv, dvs. større mængder kildekode, der ikke skal stå og rode rundt midt et afsnits tekst:

```
\begin{lstlisting}[language=prolog, captionpos={b},
                  caption={Beskrivelse af koden},
                  label={enReferenceHertil}]
  større mængde prolog-kode
\end{lstlisting}
```

Det er også muligt at lave en inkludering af kildekode direkte fra kildekodefilerne vha. `\lstinputlisting [language=prolog]{koden.pl}`

Standardopsætningen er sat til java-kode, så hvis det er dette, man vil have til at stå, behøver man ikke eksplicit at angive typen af koden vha `[language=prolog]`.

Egne kommandoer

Kapitel-intro

Det giver et godt overblik over hvert kapitel (chapter), hvis disse indledes med en lille kapitel-intro. En kapintro indeholder kort følgende: hvorfor er dette kapitel vigtigt, dets indhold, hvad man får ud af det. Til det formål er lavet en kommando, `\kapintro{Dette bulumpa–kap er vigtigt...}`, som angives lige efter et nyt `\chapter{Bulumpa–kapitel 3}`.

Større citater

Når man vil have større citater med, det vil sige over en linjes længde, så er det lettest at læse, hvis disse står ud fra den omgivende tekst. Dertil anvendes miljøet `quote2`. Det gør teksten lidt mindre og kursiv samt rykker den ind i begge sider.

```
\begin{quote2}
bla bla bla \upshape\cite{d}
\end{quote2}
```

Bemærk, at før citat-referencen kommer der en „upshape“. Det gør, at selve referencen ikke står kursiv (ser pænere ud). Dette er ikke blevet automatiseret endnu...

Problemer?

- Hvordan skrives en \sim , fx i en web-adresse? Sådan: `\(\sim\)`.
- Stavekontrol? Ja, Aspell.

Tilretning af skabelonen til de enkelte projekter

I filen „preamblePROJEKT.tex“ skal der tilrettes to linjer til det enkelte projekt. Dette har CBH sikkert allerede gjort, hvis CBH er med i gruppen. Der skal tilrettes hhv. projekt-titel og gruppenummer i de følgende to linjer fra nævnte fil.

```
\fancyhead[R0,LE]{Projekt X Y Z}
\fancyfoot[CO,CE]{\scriptsize{Gr.~X~|~RUC~|~NatBas}}
```

Forsiden Forsiden ligger i filen `title.tex`. Denne skal naturligvis også tilrettes. Det siger vel sig selv, hvad der skal tilrettes. Hvis man ønsker et andet uddannelseslogo i bunden til højre kan dette opnås ved at ændre ved `\scalebox{0.25}{\includegraphics{grafik/forside–rucGenerelt.png}}`. Datalogis er direkte en mulighed (ukommenteret).

Mere info

Hvis man vil have en længere, generel introduktion kan man fx se „The Not So Short Introduction to LaTeX2e“, www.ctan.org/tex-archive/info/lshort/english/lshort.pdf.