

Typesetting Deductions in L^AT_EX*

Michael Kohlhase
Computer Science, Jacobs University
<http://kwarc.info/kohlhase>

September 11, 2009

Abstract

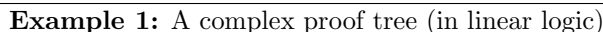
Proof trees are usually presented in a table-like fashion in logic, the `ded` package provides an infrastructure for typesetting them in L^AT_EX.

Contents

1	The User Interface	2
1.1	Inference Rules	2
1.2	Derivation Trees	2
1.3	Display Environments	3
1.4	Tableaux	3
1.5	Miscellanea	3
1.6	Acknowledgements	3
2	The Implementation	4
2.1	Inference Rules and Derivations	4
2.2	Display Environments	6
2.3	Tableaux	8
2.4	Miscellanea	9

*Version v1.1 (last revised 2009/09/11)

The `ded` package provides an infrastructure for marking up complex proof trees and formal derivations. See Figure 1 for an example.



The basic constructor of proof trees is the `\ian` macro for applications inference rules. These take a comma-separated list of assumptions as the first argument, the conclusion as the second, and the rule name as the third argument.

`\bianc` differs from `\bian` only in the bounding box. `\bianc` includes the rule name while `\bian` does not. This allows a precise layout of deduction trees. Basically, we use `\bianc`, if there is a subtree to the right of this one, and `\bian` if it is the rightmost one¹. So the simplest invocation of `\bian` is in `\bian{⟨ass1⟩,...,⟨assn⟩}{⟨concl⟩}{⟨name⟩}`, which will be typeset as

$$\frac{\text{ass}_1 \quad \dots \quad \text{ass}_n}{\text{concl}} \text{ name}$$

While the comma-separated list as the first argument to the `\ian` macro is a simple mechanism for passing arbitrary numbers of assumptions, it can sometimes be confusing to the author. Moreover, most inference lists take no more than four assumptions. Therefore — and for backwards compatibility with earlier versions of the `ded` package, we provide the macros `\ibn` (two assumptions), `\icn` (three) and `\idn` (four). Thus, `\ibn{⟨A⟩}{⟨B⟩}{⟨C⟩}{⟨N⟩}` is equivalent to `\ian{⟨A⟩,⟨B⟩}{⟨C⟩}{⟨N⟩}`.

The `\i*n` macro is useful in displaying single rules of inference, but it can do much more. I can be applied recursively to display derivation trees such as the one in Figure 2.

There are four variants of the basic `\ian` macro which differ by bounding box; we present this for `\ian*` here:

¹EDNOTE: sounds like we could really automate this.

<code>\begin{Displaynd}</code>	$\begin{array}{c} [h] \\ \vdots \\ a_1 \end{array} \quad \begin{array}{cc} a_2 & a_3 \end{array} \quad \frac{A_1 \quad A_2}{C} R \quad F$
<code>\icn{\ianr{\hypjuda{h}{a_1},a_2,a_3}{c}{r}}</code>	
<code>{\subtreec{A_1,A_2}{C}R}</code>	
<code>F</code>	
<code>D</code>	
<code>E</code>	
<code>\end{Displaynd}</code>	

Example 2: A Derivation Tree built from nested inference rules

Of course the bounding box variants `\ibn*`, `\icn*`, `\idn*`, and `\ien*` also exist for $* \in \{c, l, r\}$.

The main purpose of the bounding box is to place subtrees in a derivation. In Figure 2 we use

For representing subtrees — i.e. proof trees we do not want to show — we have a macro

`\subtree` `\subtree` that has 2 lines and a name for the subtree.

1.3 Display Environments

Even though inference rules can in principle be invoked any where in math mode, the `ded` package supplies a couple of specialized environments that initialize.

`displaynd` The `displaynd` and `Displaynd` environments allow for centered displayed derivation trees, much like the usual display style mathematics. The first leaves a bit more space between inference rules. The `cboxnd` is a variant that closes this in a box.

`textnd` The `textnd` and

`tboxnd` Finally, the environment is a variant of the L^AT_EX `figure` environment for derivation trees.

`fignd`

1.4 Tableaux

1.5 Miscellanea

`\rulename` The `\rulename` can be used to construct names of inference rules for a calculus $\langle Calc \rangle$ using `\rulename{\langle Calc \rangle}{\langle name \rangle}`.

`\inputlf` In some calculi that deal with computational linguistics we use the `\inputlf` macro for designing the input logical form.

1.6 Acknowledgements

An early version and the heart of the placement macro that is now `\inrulehelp` has been obtained from Frank Pfenning. The current `ded` package mainly adds periphery and documentation.

Florian Rabe has added functionality for hypothetical reasoning and Hella Hoffmann noticed an interface flaw that led to the introduction of `\i*nl` and `\i*nr`.

2 The Implementation

First we set up the lengths sizes.

```
1 (*package)
2 \def\@lineskipamount{4pt}
3 \def\@interlineskipamount{2pt}
```

\mud centers its argument in the current box.

```
4 \def\mud#1{\hfil $\displaystyle{#1}$\hfil}
```

\rig puts its argument to the right.

```
5 \def\rig#1{\hfil $\displaystyle{#1}$}
```

2.1 Inference Rules and Derivations

We first reserve some token registers and introduce a new switch, which we will use to create double lines if necessary.

```
6 \newbox\conc@box
7 \newbox\line@box
8 \newbox\name@box
9 \newbox\prem@box
10 \newbox\max@box
11 \newif\ifdouble\doublefalse
```

\inrulehelp The \inrulehelp macro is the main work horse for layouting the deduction trees. It constructs the layout for the upper part of the inference rule (including the horizontal bar and rule name) and saves it in the token register \conc@box. Then it constructs the conclusion and saves it in the box \line@box for later use. The

```
12 \newdimen\over@hang
13 \newdimen\tmp@dimen
14 \newdimen\max@wd
15 \newif\ifmax\maxfalse
16 \def\inrulehelp#1#2#3{%
17   \setbox\conc@box=\hbox{$\displaystyle{\mathstrut #2}$}%
18   \setbox\name@box=\hbox{$\; #3$}%
19   \setbox\line@box=\vbox{\vskip 2pt\halign{##\cr
20     \let\@tmpop=\relax
21     \mud{\@for\@I:={#1}\do{\@tmpop\@I\let\@tmpop=\quad}}\cr
22     \noalign{\vskip\the\lineskip}%
23     \noalign{\hrule height 0pt}%
24     \rig{\vbox to 0pt{\vss\hbox to 0pt{\copy\name@box \hss}\vss}}\cr
25     \noalign{\hrule}%
26     \ifdouble\noalign{\vskip\@interlineskipamount}\noalign{\hrule}\fi%
27     \noalign{\vskip\the\lineskip}%
28     \mud{\copy\conc@box}\cr}}%
29 \advance\max@wd by \wd\name@box
30 \tmp@dimen=\wd\line@box%
31 \advance\tmp@dimen by -\wd\conc@box%
32 \over@hang=.5\tmp@dimen}
```

\i*n

```
33 \def\ian#1#2#3{\lineskip\@lineskipamount%
34 \ifmax\inrulehelp{#1}{#2}{#3}\box\line@box\else%
35 \inrulehelp{#1}{#2}{#3}\hbox to \wd\conc@box{\hss\box\line@box\hss}\fi}
36 \def\ibn#1#2#3#4{\ian{#1}{#2}{#3}{#4}} %prem1, prem2, conc, name
37 \def\icn#1#2#3#4#5{\ian{#1}{#2}{#3}{#4}{#5}}
38 \def\idn#1#2#3#4#5#6{\ian{#1}{#2}{#3}{#4}{#5}{#6}}
```

```

39 \def\ien#1#2#3#4#5#6#7{\ian{#1},{#2},{#3},{#4},{#5}}{#6}{#7}}
40 \</package>
41 \*ltxml)
42 DefConstructor(' \ian{}{}{}', "<ltx:XApp>#3#1#2</ltx:XApp>");
43 DefConstructor(' \ibn{}{}{}{}', "<ltx:XApp>#4#1#2#3</ltx:XApp>");
44 DefConstructor(' \icn{}{}{}{}{}', "<ltx:XApp>#5#1#2#3#4</ltx:XApp>");
45 DefConstructor(' \idn{}{}{}{}{}{}', "<ltx:XApp>#6#1#2#3#4#5</ltx:XApp>");
46 DefConstructor(' \ien{}{}{}{}{}{}{}', "<ltx:XApp>#7#1#2#3#4#5#6</ltx:XApp>");
47 \</ltxml>

```

\i*nc

```

48 \<package>
49 \def\ianc#1#2#3{\lineskip\@lineskipamount%
50 \ifmax\inrulehelp{#1}{#2}{#3}\box\lin@box\else%
51 \inrulehelp{#1}{#2}{#3}\box\line@box\hskip\wd\name@box\fi}
52 \def\ibnc#1#2#3#4{\ianc{#1},{#2}}{#3}{#4}} %prem1, prem2, conc, name
53 \def\icnc#1#2#3#4#5{\ianc{#1},{#2},{#3}}{#4}{#5}}
54 \def\idnc#1#2#3#4#5#6{\ianc{#1},{#2},{#3},{#4}}{#5}{#6}}
55 \def\ienc#1#2#3#4#5#6#7{\ianc{#1},{#2},{#3},{#4},{#5}}{#6}{#7}}
56 \</package>
57 \*ltxml)
58 Let(' \ianc', '\ian');
59 Let(' \ibnc', '\ibn');
60 Let(' \icnc', '\icn');
61 Let(' \idnc', '\idn');
62 Let(' \ienc', '\ien');
63 \</ltxml>

```

\i*nl

```

64 \<package>
65 \def\ianl#1#2#3{\lineskip\@lineskipamount%
66 \ifmax\inrulehelp{#1}{#2}{#3}\box\line@box\else%
67 \hskip\overhang\hbox to \wd\conc@box{\hss\box\line@box\hss}\fi}
68 \def\ibnl#1#2#3#4{\ianl{#1},{#2}}{#3}{#4}} %prem1, prem2, conc, name
69 \def\icnl#1#2#3#4#5{\ianl{#1},{#2},{#3}}{#4}{#5}}
70 \def\idnl#1#2#3#4#5#6{\ianl{#1},{#2},{#3},{#4}}{#5}{#6}}
71 \def\ienl#1#2#3#4#5#6#7{\ianl{#1},{#2},{#3},{#4},{#5}}{#6}{#7}}
72 \</package>
73 \*ltxml)
74 Let(' \ianl', '\ian');
75 Let(' \ibnl', '\ibn');
76 Let(' \icnl', '\icn');
77 Let(' \idnl', '\idn');
78 Let(' \ienl', '\ien');
79 \</ltxml>

```

\i*nr

```

80 \<package>
81 \def\ianr#1#2#3{\lineskip\@lineskipamount%
82 \ifmax\inrulehelp{#1}{#2}{#3}\box\line@box\else%
83 \inrulehelp{#1}{#2}{#3}\hbox to \wd\conc@box{\hss\box\line@box\hss}\hskip\overhang\hskip\wd\name@box\fi}
84 \def\ibnr#1#2#3#4{\ianr{#1},{#2}}{#3}{#4}} %prem1, prem2, conc, name
85 \def\icnr#1#2#3#4#5{\ianr{#1},{#2},{#3}}{#4}{#5}}
86 \def\idnr#1#2#3#4#5#6{\ianr{#1},{#2},{#3},{#4}}{#5}{#6}}
87 \def\ienr#1#2#3#4#5#6#7{\ianr{#1},{#2},{#3},{#4},{#5}}{#6}{#7}}
88 \</package>
89 \*ltxml)
90 Let(' \ianr', '\ian');

```

```

91 Let('ibnr','ibn');
92 Let('icnr','icn');
93 Let('idnr','idn');
94 Let('ienr','ien');
95 </ltxml>

```

`\i*nm`

```

96 (*package)
97 \def\ianm#1#2#3{\lineskip\@lineskipamount\maxtrue\inrulehelp{#1}{#2}{#3}\box\line@box}
98 </package>

```

`\subtree`

```

99 (*package)
100 \def\subtree#1#2#3{\doubletrue\ian{#1}{#2}{#3}}
101 \def\subtreec#1#2#3{\doubletrue\ianc{#1}{#2}{#3}}
102 </package>
103 <ltxml>
104 Let('subtree','ian');
105 Let('subtreec','ian');
106 </ltxml>

```

`\small/normalnd`

```

107 (*package)
108 \def\smallnd{\def\Rulespacing{\renewcommand{\arraystretch}{3}\arraycolsep 0em}}
109 \def\normalnd{\def\Rulespacing{\renewcommand{\arraystretch}{4}\arraycolsep 0em}}
110 \normalnd
111 \def\normalspacing{\renewcommand{\arraystretch}{1}}
112 </package>
113 <ltxml>
114 DefConstructor('smallnd','');
115 DefConstructor('normalnd','');
116 DefConstructor('normalspacing','');
117 </ltxml>

```

`\dedover`

```

118 (*package)
119 \def\dedover#1#2{\hbox{\vbox{\$displaystyle\mathstrut
120      #1}\$}\vbox{\$displaystyle\mathstrut #2}\$}}
121 </package>

```

`\hypjuda` `\hypjuda{h}{c}` for a hypothetical judgment with hypothesis `h` and conclusion `c`, `\hypjudb` for two hypotheses etc.

```

122 (*package)
123 \newcommand{\ded@atop}[3][\genfrac{}{}{0pt}{#1}{#2}{#3}]
124 \newcommand{\hypjuda}[2]{\ded@atop[0]{\ded@atop[0]{\left[#1\right]}\{\vdots\}}{#2}}
125 \newcommand{\hypjudb}[3]{\ded@atop[0]{\left[#1\right]}\{\hypjuda{#2}{#3}\}}
126 \newcommand{\hypjudc}[4]{\ded@atop[0]{\left[#1\right]}\{\hypjudb{#2}{#3}{#4}\}}
127 </package>
128 <ltxml>
129 </ltxml>

```

2.2 Display Environments

We first set some lengths

```

130 (*package)
131 \def\Displaynd@pre@space{0em}
132 \def\Displaynd@post@space{-1em}

```

```

133 \def\cbox@pre@space{-.5em}
134 \def\cbox@post@space{-2.5em}
135 \def\cbox@left@space{.3em}
136 \def\cbox@right@space{.3em}
137 \end{package}

```

displaynd

```

138 \newenvironment{displaynd}%
139 {\begin{displaymath}\Rulespacing\begin{array}{c}}%
140 {\end{array}\end{displaymath}\aftergroup\ignorespaces}
141 \end{package}
142 \end{package}
143 \end{package}
144 DefEnvironment('displaynd', '<ltx:Math><ltx:XMt>#body</ltx:XMt></ltx:Math>', mode=>'display_math');
145 \end{package}

```

Displaynd

```

146 \newenvironment{Displaynd}[1]%
147 {\vspace*{\Displaynd@pre@space}\begin{displaymath}\Rulespacing\begin{array}{#1}}%
148 {\end{array}\end{displaymath}\aftergroup\ignorespaces\vspace*{\Displaynd@post@space}}
149 \end{package}
150 \end{package}
151 \end{package}
152 DefEnvironment('Displaynd', '<ltx:Math><ltx:XMt>#body</ltx:XMt></ltx:Math>', mode=>'display_math');
153 \end{package}

```

textnd

```

154 \newenvironment{textnd}%
155 {\displaystyle\Rulespacing\begin{array}{c}}%
156 {\end{array}}%
157 \end{package}
158 \end{package}
159 \end{package}
160 DefEnvironment('textnd', '<ltx:Math><ltx:XMt>#body</ltx:XMt></ltx:Math>', mode=>'inline_math');
161 \end{package}

```

\ndsepline

```

162 \newenvironment{ndsepline}%
163 {\hline\[-7ex]}
164 \end{package}
165 \end{package}
166 \end{package}

```

cboxnd

```

167 \newenvironment{cboxnd}%
168 {\vspace*{\cbox@pre@space}
169 \begin{displaymath}\Rulespacing
170 \begin{array}{|@{\hspace{\cbox@left@space}}c@{\hspace{\cbox@right@space}}|\}\hline}%
171 {\hline\end{array}\end{displaymath}
172 \aftergroup\ignorespaces
173 \vspace*{\cbox@post@space}}
174 \end{package}
175 \end{package}
176 \end{package}
177 \end{package}

```

tboxnd

```

178 \newenvironment{tboxnd}%

```

```

179 \newenvironment{tboxnd}%
180 {$\displaystyle\Rulespacing
181 \begin{array}{|@{\hspace{\cbox@left@space}}c@{\hspace{\cbox@right@space}}|\hline}%
182 {\hline\end{array}$}
183 \end{package}
184 \end{ltxml}
185 \end{ltxml}

```

fignd

```

186 \package
187 \newenvironment{fignd}[2]%
188 {\begin{figure}[htb]\def\fignd@label{fig:#1}\def\fignd@capt{#2}\begin{cboxnd}}%
189 {\end{cboxnd}\caption{\fignd@capt}\label{\fignd@label}\end{figure}}
190 {\def\endfignd{\end{cboxnd}\caption{\fignd@capt}\label{\fignd@label}\end{figure}}
191 \end{package}
192 \end{ltxml}
193 \end{ltxml}

```

2.3 Tableaux

tableau Tableaux are modeled as arrays in \LaTeX , and as nested proof terms in \LaTeXML . For the latter we need to disable the newline macro.

```

194 \package
195 \newenvironment{tableau}%
196 {\arraycolsep .2em\def\arraystretch{.9}\begin{array}{c}}%
197 {\end{array}}
198 \end{package}
199 \end{ltxml}
200 DefEnvironment('tableau',
201               '<ltx:XMAp><ltx:XTok meaning="tableau" omcd="tableaux"/>#body</ltx:XMAp>',
202               beforeDigest=>sub { DefMacro("\\\\",')'); } );
203 \end{ltxml}

```

displaytableau*

```

204 \ltxmlRawTeX(
205 \package | ltxml)
206 \newenvironment{displaytableau*}%
207 {\begin{displaymath}\begin{tableau}}%
208 {\end{tableau}\end{displaymath}\aftergroup\ignorespaces}
209 \end{package | ltxml)
210 \ltxml);

```

displaytableau

```

211 \package
212 \newenvironment{displaytableau}[1]% label
213 {\begin{equation}\label{tab:#1}\begin{tableau}}%
214 {\end{tableau}\end{equation}\aftergroup\ignorespaces}
215 \end{package}
216 \end{ltxml}
217 DefEnvironment('displaytableau',{', '<ltx:Math><ltx:XMATH>#body</ltx:XMATH></ltx:Math>',
218               mode=>'display_math');
219 \end{ltxml}

```

branches

```

220 \package
221 \newenvironment{branches}[1]% format
222 {\begin{array}[t]{#1}\begin{array}[t]{c}}%

```



```

223 {\end{array}\end{array}}
224 \end{package}
225 \end{*ltxml}
226 DefEnvironment('{branches}{}',
227               '<ltx:XMAp><ltx:XTok meaning="branches" omcd="tableaux"/>'
228               . '<ltx:XMAp><ltx:XTok meaning="tableau" omcd="tableaux"/>'
229               . '#body'
230               . '</ltx:XMAp>'
231               . '</ltx:XMAp>');
232 \end{*ltxml}

```

\nextbranch

```

233 (package)\def\nextbranch{\end{array}&\begin{array}[t]{c}}
234 \end{*ltxml}
235 DefConstructor('\nextbranch','</ltx:XMAp><ltx:XMAp><ltx:XTok meaning="tableau" omcd="tableaux"/>');
236 \end{*ltxml}

```

2.4 Miscellanea

EdNote(2) 2

```

237 (*package)
238 \def\rulename#1#2{\mbox{\sf{#1:#2}}}
239 \end{package}
240 \end{*ltxml}
241 DefConstructor('\rulename-{}-{}','<ltx:XTok omcd="FIXME" meaning="#1-#2"/>');
242 \end{*ltxml}

```

EdNote(3) 3

```

243 (*package)
244 \def\inputlf#1{\fbox{\ensuremath{#1}}}
245 \end{package}
246 \end{*ltxml}
247 DefConstructor('\inputlf{}','\#1');
248 \end{*ltxml}
249 \end{package}

```

²EdNOTE: fix the current theory

³EdNOTE: fix me!