

The `lplfitch` package*

John Etchemendy Dave Barker-Plummer Richard Zach

May 16, 2013

1 Introduction

The package `lplfitch` provides macros for typesetting natural deduction proofs in “Fitch” style, with subproofs indented and offset by scope lines. It produces proofs in the format used in the textbook *Language, Proof, and Logic* by Dave Barker-Plummer, Jon Barwise, and John Etchemendy. The package was originally written by John Etchemendy. The current version incorporates changes by Dave Barker-Plummer and Richard Zach.

2 Fitch Proofs by Example

In this section we will describe how to typeset the Fitch-style natural deduction proof shown in figure 1. This proof uses all of the macros that we have developed for typesetting proofs and therefore serves as a good illustration.

The commands for producing this proof are given in figure 2. This proof is a little hard to take in all at once, and so in the remainder of this section we will describe the development of this proof, which we hope is of use to readers.

We recommend developing the commands to typeset a proof from the inside out, starting with the most deeply nested subproof. We also recommend ignoring justifications until the proof is complete, primarily for simplicity but also because the line numbers are not known until the complete proof structure is present.

We begin by writing the lines of the innermost subproof (lines 3–6 of the main proof).

<code>\pline{Cube(a)}</code>	<code>Cube(a)</code>
<code>\pline{Cube(a)\lif Small(a)}</code>	<code>Cube(a) → Small(a)</code>
<code>\pline{Small(a)}</code>	<code>Small(a)</code>
<code>\pline{\exi{x}{Small(x)}}</code>	<code>∃x Small(x)</code>

The macro `\pline` produces a single line of the proof. The mandatory argument to `\pline` is the formula to appear within the step. You will notice that we have used the macro `\lif` to produce an conditional arrow, and `\exi` to produce

*This document corresponds to `lplproof` v0.9, dated 2013/05/16.

1.	$\forall x (Cube(x) \rightarrow Small(x))$	
2.	$\exists x Cube(x)$	
3.	a $Cube(a)$	
4.	$Cube(a) \rightarrow Small(a)$	\forall Elim: 1
5.	$Small(a)$	\rightarrow Elim: 4, 3
6.	$\exists x Small(x)$	\exists Intro: 5
7.	$\exists x Small(x)$	\exists Elim: 2, 3-6
8.	$\exists x Cube(x) \rightarrow \exists x Small(x)$	\rightarrow Intro: 2-7
9.	$(\forall x (Cube(x) \rightarrow Small(x)) \rightarrow (\exists x Cube(x) \rightarrow \exists x Small(x)))$	\rightarrow Intro: 1-8

Figure 1: An Example Proof

```

\fitchprf{
  \subproof{\pline[1.]{\uni{x}{(Cube(x)\lif Small(x))}}}{
    \subproof{\pline[2.]{\exi{x}{Cube(x)}}}{
      \boxedsubproof[3.]{a}{Cube(a)}{
        \pline[4.]{Cube(a)\lif Small(a)}[\lalle{1}]\
        \pline[5.]{Small(a)}[\life{4}{3}]\
        \pline[6.]{\exi{x}{Small(x)}}[\lexii{5}]
      }
      \pline[7.]{\exi{x}{Small(x)}}[\lexie{2}{3--6}]
    }
    \pline[8.]{\exi{x}{Cube(x)}\lif \exi{x}{Small(x)}}[\lifi{2--7}]
  }
  \pline[9.]{\brokenform{(\uni{x}{(Cube(x)\lif Small(x))}\lif {
    \formula{(\exi{x}{Cube(x)} \lif \exi{x}{Small(x))}}}\lifi{1--8}]
}

```

Figure 2: The commands for producing the proof of figure 1

an existentially quantified formula. These are members of a suite of macros that we find convenient for producing formulae. These are described in section 3.

The `\pline` macro may only be used inside a proof context. We construct such a context by using the `\fitchprf` macro. This macro has two mandatory arguments, corresponding to the premises and the body of the proof. Each argument is a list of lines, separated by a newline (`\`) command. We therefore embed the lines of our example proof like this:

```
\fitchprf{\pline{Cube(a)}}{
  \pline{Cube(a) \lif Small(a)}\
  \pline{Small(a)}\
  \pline{\exi{x}{Small(x)}}
}
```

Typesetting a document containing this command results in the proof:

$$\left| \begin{array}{l} \text{Cube}(a) \\ \text{Cube}(a) \rightarrow \text{Small}(a) \\ \text{Small}(a) \\ \exists x \text{ Small}(x) \end{array} \right.$$

Our next step is to embed this proof as a subproof within the larger proof. This is simply achieved. We first change the use of `\fitchprf` to a `\subproof` command, and then treat this as a line of the body of another `\fitchprf` command. The result of doing this is:

```
\fitchprf{\pline{\exi{x}{Cube(x)}}}{
  \subproof{\pline{Cube(a)}}{
    \pline{Cube(a) \lif Small(a)}\
    \pline{Small(a)}\
    \pline{\exi{x}{Small(x)}}
  }
  \pline{\exi{x}{Small(x)}}
}
```

which yields the proof:

$$\left| \begin{array}{l} \exists x \text{ Cube}(x) \\ \left| \begin{array}{l} \text{Cube}(a) \\ \text{Cube}(a) \rightarrow \text{Small}(a) \\ \text{Small}(a) \\ \exists x \text{ Small}(x) \end{array} \right. \\ \exists x \text{ Small}(x) \end{array} \right.$$

The new outer `\fitchprf` has two elements in the second argument, one of which is the subproof that we just wrote and the second is a formula. As we embedded the subproof in the body of the main proof we changed the command from `\fitchprf`

to `\subproof`, the definitions of these two macros are almost identical but for the adjustment of vertical spacing after the use of a `\subproof` command. Note that no `\\` command is required after the use of a `\subproof` command.

Two further applications of this technique give us the command:

```
\fitchprf{}{
  \subproof{\pline{\uni{x}}{(Cube(x)\lif Small(x))}}{
    \subproof{\pline{\exi{x}}{Cube(x)}}{
      \subproof{\pline{Cube(a)}}{
        \pline{Cube(a)\lif Small(a)}\\
        \pline{Small(a)}\\
        \pline{\exi{x}}{Small(x)}
      }
      \pline{\exi{x}}{Small(x)}
    }
    \pline{\exi{x}}{Cube(x)\lif \exi{x}}{Small(x)}
  }
  \pline{\brokenform{(\uni{x}}{(Cube(x)\lif Small(x))}\lif}{
    \formula{(\exi{x}}{Cube(x)} \lif \exi{x}}{Small(x))}}{
  }
}
```

which produces the proof

$$\begin{array}{l}
 \vdash \quad \forall x (Cube(x) \rightarrow Small(x)) \\
 \vdash \quad \exists x Cube(x) \\
 \vdash \quad \quad Cube(a) \\
 \vdash \quad \quad \quad Cube(a) \rightarrow Small(a) \\
 \vdash \quad \quad \quad \quad Small(a) \\
 \vdash \quad \quad \quad \quad \exists x Small(x) \\
 \vdash \quad \quad \exists x Small(x) \\
 \vdash \quad \exists x Cube(x) \rightarrow \exists x Small(x) \\
 \vdash \quad (\forall x (Cube(x) \rightarrow Small(x)) \rightarrow \\
 \quad \quad (\exists x Cube(x) \rightarrow \exists x Small(x)))
 \end{array}$$

Notice that the last line contains a long formula, and we have used the `\brokenform` command to typeset this formula. The `\brokenform` command takes two arguments, the first is the first part of the formula, which will be typeset left justified in the space available, while the second argument contains the remaining pieces of the formula which will appear in subsequent lines. These lines should be separated by uses of the `\\` command, and each line should (usually) be a `\formula`. These lines will be typeset right justified in the available space.

2.1 Adding Line Numbers

The `\pline` command takes an optional argument which is the number of the line within the proof. We can simply add these now that we have all of the lines of the proof.

```
\fitchprf{}{
  \subproof{\pline[1.]{\uni{x}{(Cube(x)\lif Small(x))}}}{
    \subproof{\pline[2.]{\exi{x}{Cube(x)}}}{
      \subproof{\pline[3.]{Cube(a)}}{
        \pline[4.]{Cube(a)\lif Small(a)}\
        \pline[5.]{Small(a)}\
        \pline[6.]{\exi{x}{Small(x)}}
      }
      \pline[7.]{\exi{x}{Small(x)}}
    }
    \pline[8.]{\exi{x}{Cube(x)}\lif \exi{x}{Small(x)}}
  }
  \pline[9.]{\brokenform{(\uni{x}{(Cube(x)\lif Small(x))}\lif){
    \formula{(\exi{x}{Cube(x)} \lif \exi{x}{Small(x))}}}}
}
```

which results in the proof:

1.	$\forall x (Cube(x) \rightarrow Small(x))$
2.	$\exists x Cube(x)$
3.	$Cube(a)$
4.	$Cube(a) \rightarrow Small(a)$
5.	$Small(a)$
6.	$\exists x Small(x)$
7.	$\exists x Small(x)$
8.	$\exists x Cube(x) \rightarrow \exists x Small(x)$
9.	$(\forall x (Cube(x) \rightarrow Small(x)) \rightarrow (\exists x Cube(x) \rightarrow \exists x Small(x)))$

In the eventual proof, a boxed constant is introduced at line 3. We can convert the simple subproof into a boxed subproof. We use the `\boxedsubproof` macro to achieve this. The conversion is straightforward, we replace the first argument of the `\subproof` command which was `{\pline[3.]{Cube(a)}}` by the three arguments `[3.]{a}{Cube(a)}` obtaining

```
\boxedsubproof [3.]{a}{Cube(a)}{
  \pline[4.]{Cube(a)\lif Small(a)}\
  \pline[5.]{Small(a)}\
  \pline[6.]{\exi{x}{Small(x)}}
}
```

2.2 Adding Justifications

The final thing that we need to do to complete the desired proof is the addition of justifications. To produce a justified line we need to then add the optional justification argument.

```
\pline[5.]{Small(a)}[\ife{4}{3}]
```

The `\ife` macro produces a justification using the \rightarrow -elimination rule of the logic. This is one of a suite of macros that we developed to produce justifications in a uniform manner. These are described in section 4.

2.3 Command Reference

The `\pline` command can be used to typeset proof lines. It takes the formula as a mandatory argument, and line number and justification as optional arguments: `\pline[line number]{formula}[justification]`. The variant `\fppline` works the same, except it produces a focus slider. In case you want to typeset a line in an argument or proof that is not typeset as a formula (e.g., plain text), use the `\tpline` command. It takes a line number as optional argument.

2.4 Obsolete Commands

The original version of `lplfitch` set lines of proofs using two separate macros, the `\nline` and `\jline` commands. They are superseded by the `\pline` command and are included for backward compatibility only.

A numbered line is produced by `\nline`, with the optional argument the line number. A justified line is produced using the `\jline` macro, which has one mandatory and one optional argument. The optional argument contains the justification, while the mandatory argument must contain *all* of the arguments previously given to `\nline`. So the command

```
\nline[5.]{Small(a)}
```

produces exactly the same output as

```
\jline{[5.]{Small(a)}}
```

but notice that both the arguments to `\nline` are passed within the single argument to `\jline`. To produce a justified line using `\jline`, you would also include the justification as an optional argument, say,

```
\jline{[5.]{Small(a)}}[\life{2}{3}]
```

3 Logical Formulae

We set our formulae using sans-serif font and we also need to be in math mode because the use of logical connectives. The `\formula` command takes a formula to set as its argument, and handles both font and mode change. If the LPL convention of sans-serif formulas is not desired, you can simply redefine `\formula` as

```
\renewcommand{\formula}[1]{\ensuremath{#1}}
```

Many of the commands that follow require that they are used in a formula, or math, context.

3.1 Connectives

We introduce our own (usually) shorter names for logical connectives. All of these commands require a formula context.

Usage	Output
<code>\land</code>	\wedge
<code>\lor</code>	\vee
<code>\lif</code>	\rightarrow
<code>\liff</code>	\leftrightarrow
<code>\lnot</code>	\neg
<code>\lfalse</code>	\perp
<code>\lall</code>	\forall
<code>\lis</code>	\exists

3.2 Quantified Formulae

So that we don't have to think about spacing when setting quantified formulae, we defined the commands `\exi` and `\uni`. Use `\exi{x}{P(x)}` to obtain $\exists x P(x)$, and `\uni{x}{\exi{y}{(P(x)\lif Q(x,y))}}` for $\forall x \exists y (P(x) \rightarrow Q(x, y))$, for example. `\exi` and `\uni` require a formula context.

4 Justifications

Here are the macros that we have developed to typeset justifications within a proof. These do not require a formula context.

Usage	Output
<code>\landi{2, 3}</code>	\wedge Intro: 2, 3
<code>\lande{4}</code>	\wedge Elim: 4
<code>\lori{5}</code>	\vee Intro: 5
<code>\lore{6}{7--9}{11--13}</code>	\vee Elim: 6, 7-9, 11-13
<code>\noti{14--15}</code>	\neg Intro: 14-15
<code>\note{16}</code>	\neg Elim: 16
<code>\falsei{17}{19}</code>	\perp Intro: 17, 19
<code>\ffalsee{20}</code>	\perp Elim: 20
<code>\lifi{21--25}</code>	\rightarrow Intro: 21-25
<code>\life{27}{30}</code>	\rightarrow Elim: 27, 30
<code>\lifi{33--37}{40-50}</code>	\leftrightarrow Intro: 33-37, 40-50
<code>\liffe{55}{57}</code>	\leftrightarrow Elim: 55, 57
<code>\reit{4}</code>	Reit: 4
<code>\eqi</code>	$=$ Intro
<code>\eqe{7}{11}</code>	$=$ Elim: 7, 11
<code>\lalli{9--12}</code>	\forall Intro: 9-12
<code>\lalle{13}</code>	\forall Elim: 13
<code>\lexii{15}</code>	\exists Intro: 15
<code>\lexie{17}{18-30}</code>	\exists Elim: 17, 18-30

5 Other Macros

5.1 Arguments

When you want to typeset an argument, for example when setting an exercise requiring the proof of some conclusion from a collection of premises, you should use the `\fitcharg` command. This command takes two arguments, both mandatory, the premise lines which are set above the Fitch bar, and the conclusion lines, which go below. Here's an example. The argument

$$\left| \begin{array}{l} P(a) \\ Q(a) \\ \hline S(a) \end{array} \right.$$

is typeset by the command:

```
\fitcharg{
  \formula{P(a)}\
  \formula{Q(a)}
}{
  \formula{S(a)}
}
```

Notice that multiple lines are separated by a use of the \LaTeX command `\`.

5.2 Fitch Contexts

The `\fitchctx` command is used for setting proof fragments which do not include the horizontal Fitch bar. We use these, for example, when formally describing the inference rules of our logic. The command below, for example, typesets our description of the rule of conditional proof.

Notice here that we use the command `\ellipsesline` to produce a line of the “proof” containing a vertical ellipses (using the `\pline{\vdots}` results in the ellipses not being centered on the line).

Another new feature of this situation is the use of the `\fpline` command. This command works just like the `\pline` command and produces exactly what the `\pline` command would, except that a focus slider appears to the left of the Fitch bar in question. The `\fpline` command does not work in the argument lines of `\fitchprf` or `\fitcharg`.

```
\fitchctx{
  \subproof{\pline[$n$.]{P}}{
    \ellipsesline\
    \pline[$m$.]{Q}
  }
  \fpline{P \lif Q}[\lifi{$n$--$m$}]
}
```

$$\triangleright \left| \begin{array}{l} n. P \\ \hline \vdots \\ m. Q \end{array} \right. \rightarrow \text{Intro: } n\text{--}m$$

6 Other Packages

Two other \LaTeX packages provide functionality to typeset Fitch-style proofs. They differ from `lplfitch` in that they set the line numbers close to the left edge of the proof, while `lplfitch` sets them close to the formula. Another difference is that subproofs can easily be copied and work the same regardless of the depth of nesting.

Peter Selinger’s package is available at

<http://www.mathstat.dal.ca/~selinger/fitch/>,

and that by Johan Klüwer at

<http://folk.uio.no/johanw/FitchSty.html>.

7 Implementation

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{lpplfitch}[2013/05/07 -- Fitch Proofs a la LPL]
\formula Typesets a formula (in math mode, letters typeset in sans-serif as in LPL)
3 \newcommand*\formula[1]{\ensuremath{\sf{#1}}}
```

7.1 Connectives and Quantifiers

We provide convenient and short commands for logical symbols.

```
4 \newcommand*\lif{\rightarrow}
5 \newcommand*\liff{\leqtrightharpoonrightarrow}
6 \newcommand*\lfalse{\bot}
7 \newcommand*\lall{\forall}
8 \newcommand*\lis{\exists}
```

The following provide for commands for setting quantified formulae, inserting correct space between variable and formula. It can be redefined using `\renewcommand` if you want to use them but want quantifiers set differently (e.g., with parentheses)

```
9 \newcommand*\quant[3]{#1 #2\;#3}
10 \newcommand*\exi[2]{\quant{\lis}{#1}{#2}}
11 \newcommand*\uni[2]{\quant{\lall}{#1}{#2}}
```

7.2 Justifications

The following commands are used to generate justifications, e.g., `\landi{1, 2}` produces “**^ Intro:** 1, 2”.

```
12 \newcommand*\intro[1]{\formula{#1\,}\{\bf Intro:}}
13 \newcommand*\elim[1]{\formula{#1\,}\{\bf Elim:}}
14
15 \newcommand*\landi[1]{\intro{\land} #1}
16 \newcommand*\lande[1]{\elim{\land} #1}
17 \newcommand*\lori[1]{\intro{\lor} #1}
18 \newcommand*\lore[3]{\elim{\lor} #1, #2, #3}
19 \newcommand*\lnoti[1]{\intro{\lnot} #1}
20 \newcommand*\lnote[1]{\elim{\lnot} #1}
21
22 \newcommand*\lfalsei[2]{\intro{\lfalse} #1, #2}
23 \newcommand*\lfalsee[1]{\elim{\lfalse} #1}
24
25 \newcommand*\lifi[1]{\intro{\lif} #1}
26 \newcommand*\life[2]{\elim{\lif} #1, #2}
27
28 \newcommand*\liffi[2]{\intro{\liff} #1, #2}
29 \newcommand*\liffe[2]{\elim{\liff} #1, #2}
30
31 \newcommand*\reit[1]{\{\bf Reit:} #1}
32
```

```

33 \newcommand*\eqi[0]{\$=\,$\{\bf Intro}}
34 \newcommand*\eqe[2]{\elim{=} #1, #2}
35
36 \newcommand*\lalli[1]{\intro{\lall} #1}
37 \newcommand*\lalle[1]{\elim{\lall} #1}
38
39 \newcommand*\lexii[1]{\intro{\lis} #1}
40 \newcommand*\lexie[2]{\elim{\lis} #1, #2}

```

7.3 Dimensions

The following dimensions can be redefined with `\setlength`

```

\fitchargwidth Width of formulas in arguments
41 \newlength{\fitchargwidth}
42 \setlength{\fitchargwidth}{3.5in}

\fitchprfwidth Width of formulas in proofs
43 \newlength{\fitchprfwidth}
44 \setlength{\fitchprfwidth}{3.0in}

\fitchctxwidth Width of formulas in Fitch contexts
45 \newlength{\fitchctxwidth}
46 \setlength{\fitchctxwidth}{1in}

\fitchsep Distance between scope line and formula
47 \newlength{\fitchsep}
48 \setlength{\fitchsep}{10pt}

```

7.4 Contexts, Arguments, Proofs

```

\fitchctx Typesets a Fitch context, e.g., specification of a rule, in which a focus slider is
typeset to the left of the conclusion
49 \newcommand{\fitchctx}[1]{%
50 \advance \fitchctxwidth by -\fitchsep%
51 \advance \fitchctxwidth by .5pt%
52 \begin{tabular}[t]{r@{}|p{\fitchctxwidth}@{}l}
53 \phantom{\slider} \ll[-1.75ex]
54 #1 \ll[-1.75ex] & &
55 \end{tabular}
56 \advance \fitchctxwidth by \fitchsep%
57 \advance \fitchctxwidth by -.5pt%
58 }

\fitchprf Typesets a fitch proof
59 \newcommand{\fitchprf}[2]{%
60 \advance \fitchprfwidth by -\fitchsep%
61 \advance \fitchprfwidth by .5pt%

```

```

62 \hspace*{.35em}%
63 \begin{tabular}[t]{|p{0pt}@{}p{\fitchprfwidth}@{\hspace*{\fitchsep}}|}
64 \multicolumn{3}{@{}l@{}}{\ }\\[-2.35ex]
65 #1 \\
66 \ \\[-2.5ex] \cline{1-1}\\[-2ex]
67 #2 \\ \multicolumn{3}{@{}l@{}} \ \\[-2.35ex]
68 \end{tabular}
69 \advance \fitchprfwidth by \fitchsep%
70 }

```

`\fitcharg` Typesets an argument (without line numbers). Takes two arguments, the premises and conclusion lines of the argument.

```

71 \newcommand{\fitcharg}[2]{
72 \advance \fitchargwidth by -\fitchsep%
73 \hspace*{.35em}
74 \begin{tabular}[t]{|p{\fitchsep}@{}p{\fitchargwidth}@{}|}
75 \multicolumn{2}{@{}l@{}}{\ }\\[-2.35ex]
76 #1 \\
77 \ \\[-2.5ex] \cline{1-1}\\[-2ex]
78 #2 \\ \multicolumn{2}{@{}l@{}} \ \\[-2.35ex]
79 \end{tabular}
80 }

```

`\slider` Typesets the focus arrow.

```

81 \newcommand*\slider{\mbox{\$ \triangleright \;$}}

```

`\pline` Typesets a proof line with an optional line number (before the formula argument) and an optional justification (after the formula argument)

```

82 \def\pline{\@ifnextchar[\@plinenum{\@plinenum[\@empty]}}
83 \def\@plinenum[#1]#2{\@ifnextchar[{\@plines{#1}{#2}}{\%
84 \@plines{#1}{#2}[\@empty]}}
85 \def\@plines#1#2[#3]{ & #1\formula{\; #2} & #3}

```

`\fpline` Like `pline` except produces a focus slider on the left

```

86 \newcommand*\fpline{\slider\pline}

```

`\tline` Typesets proof line containing text with an optional line number.

```

87 \newcommand*\tline}[2][{}]{ & #1\hspace{.6em}{#2}}

```

The following macros `\nline`, `\jline`, and `\fline` are the commands provided in the original version of `lplfitch` and are included for backwards compatibility.

```

88 \newcommand*\nline}[2][{}]{ & #1\formula{\;#2}}

```

justified line (justification optional) `\jline[justification]{args for line}`

```

89 \newcommand*\jline}[2][{}]{\nline#2 & #1}

```

focused line

```

90 \newcommand*\fline}[1]{\slider \jline#1}

```

`\ellipsesline` Producing vertical ellipses in a line on their own. This takes care of centering the ellipses in their line.

```
91 \newcommand*\ellipsesline[0]{\nline{\;\raise.65ex\hbox{\vdots}}}
```

`\subproof` This command sets a subproof within a proof. The arguments are the line before, and the lines after the horizontal fitch bar.

```
92 \newcommand\subproof[2]{&\fitchprf{#1}{#2}\}
```

`\boxedsubproof` Produces a new boxed subproof. `\boxedsubproof` [*n*] [*c*] [*premise*] [*body*], where *n* is the number of the premise line, *c* is the constant(s) to be boxed, *premise* is the premise of the subproof, and *body* contains the lines of the subproof.

```
93 \newcommand\boxedsubproof[4][{}]{
94   \subproof{\nline[#{1}]{\fbox{\formula{#2}}\;#3}}{#4}
95 }
```

`\brokenform` A crude attempt at dealing with a formula which need to be broken to fit in the proof. We take two arguments, the first line of the formula which is left justified, and the remainder of the lines, which are right justified. The first argument is set as a `\formula` but the second is not (as it may contain line breaks itself).

```
96 \newcommand\brokenform[2]{
97   \advance\fitchprfwidth by-\fitchsep
98   \begin{tabular}[c]{rp{\fitchprfwidth}}
99     \multicolumn{1}{p{\fitchprfwidth@{}}}{\formula{#1}}\
100    \formula{#2}
101   \end{tabular}
102   \advance\fitchprfwidth by\fitchsep
103 }
```

Change History

v0.1		v0.9
General: Initial version 1	General: First public beta
	 1