Latex: Writing Logical Expressions*[WiP]

Artur Wegrzyn

2021-04-20

Contents

1 Introduction **1**

2 Logical symbols: basics **2**
2.1 Writing any expressions 2
2.2 Logical connectives 3
2.2.1 Connectives: *object language* 3
2.2.2 Connectives: *observer's language* 3
2.3 *Observer's* versus *object language*: one more remark 4

3 Gentzen-styl sequent calculus proofs, package 'bussproofs' **4**

4 Fitch-style proofs with package 'lplfitch' **5**

1 Introduction

The purpose of this document is to investigate writing of logical expressions in Latex. I start with the simple cases (most importantly: logical connectives and quantifiers) and then proceed to more difficult ones (writing proofs in

*What I mean hear by 'expression' is any sequence of symbols in an arbitrary alphabet of choice. The word 'expression' should be considered as element of the dictionary of the spoken (verncular?) English. By the way, this explanation constitutes a situation where the distinction *suppositio formalis* vs *suppositio materialis* evinces itself. For more on this, cf. [1], p. 59

sequent calculus and Fitch-style proofs).

Importantly, it is acknowledged that author is writing down these words to serve his own needs: there is not intention on his side to make them user-friendly and readily comprehensible to others as their primary purpose in his mind is that they act as a store of relevant, concise information in the area considered.

Last but not least, the conventions I assume are - at least to some extent - subjective - and are by no means universal. This is especially pronounced with respect to the distinction between *observer's* and *object* languages whose importance cannot be stressed enough.

2 Logical symbols: basics

2.1 Writing any expressions

The most obvious way to write atoms is writing them as ordinary single-dollar-sign-inline equations like the following one: P, Q, which is attained using the following Latex command:

```
$P$
```

For an equation shown in a separate line, use dobule-dollar equations. That is, in order to get:

$$P \rightarrow Q$$

use:

```
$$ P \rightarrow Q$$
```

Last but not least there are the Latex equations. Example of numbered equation:

$$P \rightarrow (Q \rightarrow P) \tag{1}$$

This can be obtained using the following code:

```
\begin{equation}
P \rightarrow (Q \rightarrow P)
\end{equation}
```

In order to suppress the automatic numbering of equations, append asterisk to the end of the equation invocation like this:

```
\begin{equation*}
P \rightarrow (Q \rightarrow P)
\end{equation*}
```

which will be compiled into:

$$P \rightarrow (Q \rightarrow P)$$

2.2 Logical connectives

It is of utmost importance to distinguish between the logician's two languages here, before moving on, i.e. between the *observer's language* and *object language*. Hence, this section splits in two parts, corresponding to the two languages.

2.2.1 Connectives: *object language*

Let's consider a PL language with two atoms P, Q. The symbols that can be used to connect them are:

1. Conjunction $P \wedge Q$, : `P \wedge Q`
2. Disjunction (OR, i.e. inclusive) $P \vee Q$: there are two possibilities - either `P \vee Q` or `P \lor Q` [1]
3. Material conditional $P \rightarrow Q$: `P \rightarrow Q`
4. Equivalence: $P \leftrightarrow Q$: `P \leftrightarrow Q`
5. Negation: $\neg P$: `\lnot P`

2.2.2 Connectives: *observer's language*

Here, contrary to the previous section, the PL language atoms are not available. Given that the observer's language here is English, let P, Q be *propositions* in English[2].

1. Conjunction $P \bigwedge Q$, : `P \wedge Q`

[1] Pun not intended.

[2] Point well worth elaborating upon: what is meant here by a proposition in English is, very roughly, a statement of which it is possible to say whether it is true or false.

2. Disjunction (OR) $P \bigvee Q$: `P \bigvee Q`
3. Material conditional $P \Rightarrow Q$: `P \Rightarrow Q`
4. Equivalence: $P \Leftrightarrow Q$: `P \Leftrightarrow Q`
5. Negation: $\neg P$: `\lnot P`

2.3 *Observer's* versus *object language*: one more remark

Important symbols that are *related* to the problem of consequence are:

$$\rightarrow, \therefore$$

They can be obtained using commands: `\rightarrow, \therefore`
As regards the *meaning* of these symbols: I follow the notation of Smith[3] [2], so $\rightarrow$ stands for logical connective of the *object language* representing *material conditional*, as mentioned above. The symbol $\therefore$ is also a member of the *object language* and is used to represent an argument, which is whole given in the *object language*.

3 Gentzen-styl sequent calculus proofs, package 'bussproofs'

Before presenting the relevant symbols, I make a detour to clarify what is meant here by 'sequent calculus'. I follow the classical position Kleene [3], in which he defines the Gentzen-style calculus in paragraph 9. First symbols that come to mind are the semantic entailment $\models$ and synactic entailment $\vdash$ that can be obtained via Latex commands `\models` and `\vdash` .

Elegant way of writing *modus ponens* is:

$$\frac{\text{A} \qquad \text{B}}{\text{D}}$$

Next topic of interest here is a device for coherent writing of proofs/deductions. Let's present the above-mentioned Gentzen-style calculus as given in [3] (page 387):

[3]Given clarity with which [2] is written, cf. section 18.8 therein for further accessible exposition of the use of these symbols.

1a. $P \to (Q \to R)$

1b. $(P \to Q) \to ((A \to (B \to C)) \to (A \to C))$

2. Modus ponens:

4 Fitch-style proofs with package 'lplfitch'

References

[1] A. Tarski, *Introduction to Logic and to the Methodology of Deductive Sciences*. Dover, 1995.

[2] P. Smith, *An Introduction to Formal Logic*. Logic Matters, 2020.

[3] S. C. Kleene, *Mathematical Logic*. Dover, 1967.