PROPOSITIONAL LOGIC AS A FORMAL LANGUAGE

We were deliberately informal about that, for our main focus was on trying to understand the precise mechanics of the natural deduction rules. However, it should have been clear that the rules we stated are valid for any formulas we can form, as long as they match the pattern required by the respective rule. For example

the application of the proof rule →e in

1 p → q premise

2 p premise

3 q →e 1, 2

is equally valid if we substitute p with p ∨ ¬r and q with r → p:

1 p ∨ ¬r → (r → p) premise

2 p ∨ ¬r premise

3 r → p →e 1, 2

This is why we expressed such rules as schemes with Greek symbols standing for generic formulas. Yet, it is time that we make precise the notion of 'any formula we may form.' Because this text concerns various logics, we will introduce in (1.3) an easy formalism for specifying well-formed formulas. In general, we need an unbounded supply of propositional atoms p, q, r,..., or p1, p2, p3,... You should not be too worried about the need for infinitely many such symbols. Although we may only need finitely many of these propositions to describe a property of a computer program successfully, we cannot specify how many such atomic propositions we will need in any concrete situation, so having infinitely many symbols at our disposal is a cheap way out. This can be compared with the potentially infinite nature of English: the number of grammatically correct English sentences is infinite, but finitely many such sentences will do in whatever situation you might be in (writing a book, attending a lecture, listening to the radio, having a dinner date, . . .).

Formulas in our propositional logic should certainly be strings over the alphabet {p, q, r,... }∪{p1, p2, p3,... } ∪ {¬,∧,∨, →,(,)}. This is a trivial observation and as such is not good enough for what we are trying to capture.

For example, the string (¬)() ∨ pq → is a word over that alphabet, yet, it does not seem to make a lot of sense as far as propositional logic is concerned. So what we have to define are those strings which we want to call formulas. We call such formulas well-formed.