

AI3 -- Knowledge Representation and Inference I

---

Assessed Practical Exercise 1

Your task is to implement first-order predicate logic in LISP. This involves:

- 1) Choosing and implementing a representation for well-formed formulae and theorems as LISP data objects.
- 2) Implementing the inference rules as functions taking one or more premises (and possibly some other arguments) and producing a conclusion.

A summary of the syntax and inference rules of first-order logic appears below.

Your implementation of the inference rules should fail (by calling the nullary function "abort") when applied to inappropriate arguments. For example, an attempt to apply Double Negation 1 (see the summary of inference rules below) to a theorem which is not of the form  $L \vdash \sim\sim A$  should fail. Thus it should be possible to use your program to mechanically check the validity of proofs.

You might find it easier to implement propositional logic first and then extend your implementation to handle full first-order logic rather than tackling first-order logic at the start. There are a few subtleties to first-order logic which you will have to take into account, for example the fact that wffs which differ only in the names of bound variables, for example

$\text{all } x.(P(x) \ \& \ Q(a))$     and     $\text{all } y.(P(y) \ \& \ Q(a))$

are regarded as identical by a rule such as MPP which requires two theorems which match in some way. A successful implementation of propositional logic alone will be worth 70% of the available marks.

Issued: Monday 8th November 1987

Due: Monday 30th November 1987

Solutions, including appropriate documentation and examples, should be submitted to Don Sannella (dts@itspna) by electronic mail. To send the file "yoursolution", issue the command: `mail dts < yoursolution`