



Homework

1. (main - this exercise is a part of the first of the two bigger assignments of this course)

Implement a small kernel of a proof system for minimal propositional logic:

- (a) Choose a programming language. The task has been defined in such a way that the choice of the language should not influence the complexity of the task.
 - In functional languages defining the datatypes is easy thanks to algebraic datatypes (ML, Scala, Haskell, variants of Lisp, ...)
 - In object-oriented languages defining the types and functions is easy due to classes and inheritance (C++, Python, ...)
 - For pure C, datatypes become structs of unions, but implementing reading and evaluation should be easy
- (b) Choose the datatypes used for the objects of the logic
 - Do you need types? terms? formulas? theorems?
 - What are the constructors?
- (c) Choose the inference rules
 - At least minimal proposition logic
 - Do you need substitution?
- (d) Implement a mechanism that lets you read proofs in a simple syntax, verify them and print them, that only accepts proofs in the given calculus.
 - Example possible input format (invalid proof):
`introduce?(eliminate?(assume?(x), substitute?(y,z)))`

2. (optional - small) Find a HOL-Light proof using rules (without the automated rules) that shows:

$$\vdash (a \implies b \implies c) \implies (a \implies b) \implies a \implies c$$

Without using automated tactics like `TAUT`. The list of rules can be found for example at:

<http://www.cl.cam.ac.uk/~jrh13/hol-light/holchart.txt>