Course on Formal Methods 2010-2011

Logic & Theorem Proving

3 February 2011
Some exercises

1. Given the formula $((x \wedge y) \vee (x \Rightarrow y)) \Rightarrow x$, give an assignment to variables x, y that makes the formula true and another assignment that makes it false. Transform the given formula into CNF.

2. Prove $A \vee (B \Rightarrow C) \Rightarrow (B \Rightarrow (A \vee C))$ using natural deduction by indicating the rule applied at each step.

3. Prove $(\forall x.(P\,x) \vee (Q\,x)) \Rightarrow (\forall x.(Q\,x) \vee (P\,x))$ using natural deduction by indicating the rule applied at each step.

4. Given the formula

$$f = (\exists x.\forall y.p(x, z, y) \vee q(g(x), y)) \vee (\forall x.\exists z.p(y, x, x) \wedge q(a, z))$$

and the λ-expression $e = \lambda xy.xx(\lambda z.zy)z$, mark each variable occurrence in f and e as binding, bound or free. For each bound occurrence, indicate the corresponding binding occurrence.

5. Apply β-reduction to the term $(\lambda xy.x(\lambda z.zy)x)(\lambda z.zy)$. Underline the redex at each step.

6. Let $\Sigma = \{f : \sigma \to \rho \to \tau,\ g : \sigma \to \rho\}$ and $\Gamma = \{x : \sigma, y : \sigma\}$. Derive a type judgement for the term $\lambda x^{\sigma}.f\,x\,(g\,y)$.

7. A *stack* over any type T of elements is a data structure that can be either empty or made by an element of T followed by a stack over T. Give a definition of the type `stack` using the derived rule for data type definition, discuss its characteristics and finally give an example of a term of the defined data type.