Assignment 10
Introduction to Computational Logic, SS 2008

Prof. Dr. Gert Smolka, Dr. Chad Brown
www.ps.uni-sb.de/courses/cl-ss08/

Read in the lecture notes: Chapter 8

Exercise 10.1 We require that X is finite so that every Boolean function can be represented by a formula. Suppose X is infinite. How can we obtain a Boolean function that cannot be represented by a propositional formula?

Exercise 10.2 Find tableau proofs for the following tautologies:

a) $(\bot, x, y) = x$
b) $(\top, x, y) = y$
c) $(x, y, y) = y$

Exercise 10.3 Draw all prime trees containing no other variables but x and y. Assume $x < y$. For each tree give an equivalent propositional formula that is as simple as possible.

Exercise 10.4 Let s be the propositional formula $x = (y = z)$. Assume $x < y < z$. Draw the prime trees for the following formulas: s, $\neg s$, $s \wedge s$, $s \to s$.

Exercise 10.5 Four girls agree on some rules for a party:

i) Whoever dances which Richard must also dance with Peter and Michael.
ii) Whoever does not dance with Richard is not allowed to dance with Peter and must dance with Christophe.
iii) Whoever does not dance with Peter is not allowed to dance with Christophe.

Express these rules as simply as possible.

a) Describe each rule with a propositional formula. Do only use the variables c (Christophe), p (Peter), m (Michael), r (Richard).
b) Give the prime tree that is equivalent to the conjunction of the rules. Use the order $c < p < m < r$.

Exercise 10.6

a) Find a propositional formula s that contains the variables x, y, z and has x as its only significant variable.
b) Determine the significant variables of the formula $(x \to y) \wedge (x \vee y) \wedge (y \vee z)$.

Exercise 10.7 Develop an algorithm that for two prime trees s, t yields the prime tree for $s = t$. Implement the algorithm in Standard ML. Proceed as follows:

a) Complete the following equations so that they become tautologies on which the algorithm can be based.

$$\begin{aligned} (x = \top) &= \\ (\bot = \bot) &= \\ ((x, y, z) = (x, y', z')) &= \\ ((x, y, z) = u) &= \end{aligned}$$

b) Complete the declarations of the procedures *red* and *equiv* so that *equiv* computes for two prime trees s, t the prime tree for $s = t$. The variable order is the order of *int*. Do not use other procedures.

```
type var = int
datatype dt = F | T | D of var * dt * dt

fun red x s t =

fun equiv T t =
  | equiv s T =
  | equiv F F =
  | equiv F (D(y,t0,t1)) =
  | equiv (D(x,s0,s1)) F =
  | equiv (s as D(x, s0, s1)) (t as D(y, t0, t1)) =
      if x=y then
      else if x<y then
      else
```

Exercise 10.8 Let decision trees be represented as in Exercise 10.7, and let propositional formulas be represented as follows:

```
datatype pf = FF | TT | V of var | NEG of pf | AND of pf * pf
            | OR of pf * pf | IMP of pf * pf | EQ of pf * pf
```

Write a procedure $pi : pf \to dt$ that yields the prime tree for a propositional formula. Be smart and only use three auxiliary procedures *red*, *neg* and *conj*.

Exercise 10.9 Let s be the propositional formula $(x \wedge y \equiv x \wedge z) \wedge (y \wedge z \equiv x \wedge z)$. Assume the variable order $x < y < z$.

a) Draw the prime tree for s.

b) Draw a minimal BDD whose nodes represent the subtrees of the prime tree for s.

c) Give the table representation of the BDD. Label each non-terminal node of your BDD with the number representing it.