Introduction to Computational Logic, SS 2006: Solution for assignment 7

Prof. Dr. Gert Smolka, Dipl.-Inform. Mathias Möhl

Exercise 7.1 (3-Element Specification)

$$\begin{array}{ll} \sigma_0: & \sigma 0 = 1 \\ \sigma_1: & \sigma 1 = 2 \\ \sigma_2: & \sigma 2 = 0 \end{array}$$

$$\begin{array}{ll} I_0: & 0 \to x = 1 \\ I_1: & 1 \to x = x \\ I_2: & 2 \to x = 1 \end{array}$$

$$\text{CA}: \quad f0 \to f1 \to f2 \to fx = 1$$

The last 4 axioms ensure that $|\mathcal{A}V| \leq 3$. Together with the first 3 axioms we can also show that each proper model has at least 3 elements:

$\underline{A, 0 = 1 \vdash x = 1:}$

$$\begin{array}{llr} x & = 1 \to x & I_1 \\ & = 0 \to x & 0 = 1 \\ & = 1 & I_0 \end{array}$$

$\underline{A, 2 = 0 \vdash x = 1:}$

$$\begin{array}{llr} 0 & = \sigma 2 & \sigma_2 \\ & = \sigma 0 & 0 = 2 \\ & = 1 & \sigma_0 \end{array}$$

$A, 2 = 0 \vdash x = 1$ now follows from $A, 0 = 1 \vdash x = 1$.

$\underline{A, 1 = 2 \vdash x = 1:}$

$$\begin{aligned} x &= 1 \rightarrow x && I_1 \\ &= 2 \rightarrow x && 1 = 2 \\ &= 1 && I_2 \end{aligned}$$

A specification of a set with 4 elements can be done analogously, but there is one further complication: you need two successor functions σ and σ'. One successor function σ alone is not sufficient, since you cannot deduce $x = 0$ from $x = \sigma(\sigma x)$. For example $0 = 2$ implies $1 = 3$, but not $1 = 0$ or $3 = 0$. This problem can be fixed with a second successor function σ' such that $\sigma' 0 = 2$ and $\sigma' 1 = 3$.

Exercise 7.2 (Stability)

a) $E = \{x = a\}$, $e = (\lambda x.x, \lambda x.a)$

b) the same as for a)

Exercise 7.3 (Deduction Rules) a) Let x be a variable such that $x \notin \mathcal{N}t$. Then we have the follwing derivation of $t = t$ from $\emptyset$.

$$\begin{aligned} &(\lambda x.t)x = t && \beta \\ &t = (\lambda x.t)x && \text{Sym} \\ &t = t && \text{Trans} \end{aligned}$$

b) Let $st = s't$ be an equation, $\hat{t} = \lambda f.ft$, and $f \in \mathcal{N}t$. Then we have the following derivation of $st = s't$ from $s = s'$.

$$\begin{aligned} &s = s' && \text{assumption} \\ &\hat{t}s = \hat{t}s' && \text{CR} \\ &\hat{t}s = st && \beta \\ &\hat{t}s' = s't && \beta \\ &\hat{t}s = s't && \text{Trans} \\ &st = \hat{t}s && \text{Sym} \\ &st = s't && \text{Trans} \end{aligned}$$

Exercise 7.4 (Conditionals) (a) $(x, y, y) = y$

$$
\begin{array}{rcll}
(x, y, y) & = & \overline{x}y + xy & \text{Def. Conditional} \\
& = & (x + \overline{x})y & \text{Dist} \\
& = & 1y & \text{Comp} \\
& = & y & \text{Iden}
\end{array}
$$

(b) $(x, y, z) = (\overline{x} + z)(x + y)$

$$
\begin{array}{rcll}
(x, y, z) & = & (\overline{x}y + xz) & \text{Def. Conditional} \\
& = & (x + \overline{x})(x + y)(z + \overline{x})(z + y) & \text{Dist (3x)} \\
& = & 1(x + y)(z + \overline{x})(z + y) & \text{Comp} \\
& = & (x + y)(z + \overline{x})(z + y) & \text{Iden} \\
& = & (x + y)(\overline{x} + z)(y + z) & \text{Comm} \\
& = & (x + y)(\overline{x} + z) & \text{Reso} \\
& = & (\overline{x} + z)(x + y) & \text{Comm}
\end{array}
$$

(c) $\overline{(x, y, z)} = (x, \overline{y}, \overline{z})$

$$
\begin{array}{rcll}
\overline{(x, y, z)} & = & \overline{\overline{x}y + xz} & \text{Def. Conditional} \\
& = & (\overline{\overline{x}} + \overline{y})(\overline{x} + \overline{z}) & \text{DeMo (3x)} \\
& = & (x + \overline{y})(\overline{x} + \overline{z}) & \text{Double Neg} \\
& = & (x, \overline{y}, \overline{z}) & \text{(b)}
\end{array}
$$

(d) $(\overline{x}, y, z) = (x, z, y)$

$$
\begin{array}{rcll}
(\overline{x}, y, z) & = & \overline{\overline{x}}y + \overline{x}z & \text{Def. Conditional} \\
& = & xy + \overline{x}z & \text{Dneg} \\
& = & (x, z, y) & \text{Def. Conditional}
\end{array}
$$

(e) $(x, y, z)u = (x, yu, zu)$

$$\begin{aligned}
(x, y, z)u &= (\overline{x}y + xz)u && \text{Def. Conditional} \\
&= \overline{x}yu + xzu && \text{Dist} \\
&= (x, yu, zu) && \text{Def. Conditional}
\end{aligned}$$

(g) $(x, y, z)(x, u, v) = (x, yu, zv)$

$$\begin{aligned}
(x, y, z)(x, u, v) &= (\overline{x}y + xz)(\overline{x}u + xv) && \text{Def. Conditional} \\
&= (\overline{x}y + xz)\overline{x}u + (\overline{x}y + xz)xv && \text{Dist} \\
&= \overline{x}u\overline{x}y + \overline{x}uxz + xv\overline{x}y + xvxz && \text{Dist} \\
&= 0vy + 0uz + \overline{x}\,\overline{x}yu + xxzv && \text{Comp} \\
&= 0 + 0 + \overline{x}\,\overline{x}yu + xxzv && \text{Domi} \\
&= \overline{x}\,\overline{x}yu + xxzv && \text{Iden} \\
&= \overline{x}yu + xzv && \text{Idem} \\
&= (x, yu, zu) && \text{Def. Conditional}
\end{aligned}$$

Exercise 7.5 (Dualisation)

$$\begin{aligned}
\widehat{(s, t_0, t_1)} &= \widehat{\bar{s}t_0 + st_1} && \text{Def. Conditional} \\
&= (\hat{\bar{s}} + \hat{t}_0)(\hat{s} + \hat{t}_1) && \text{Def. Dual} \\
&= (\hat{s}, \hat{t}_1, \hat{t}_0) && \text{7.4(b)}
\end{aligned}$$

Exercise 7.6 (All Prime Trees for $x < y$)

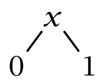
1:

1

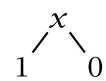
0:

0

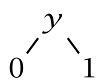
x:



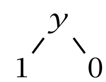
$\overline{x}$:



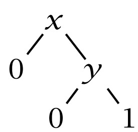
y:



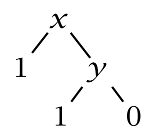
$\overline{y}$:



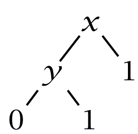
$x \wedge y$:



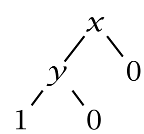
$\overline{x \wedge y}$:



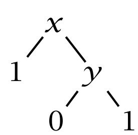
$x \vee y$:



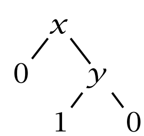
$\overline{x \vee y}$:



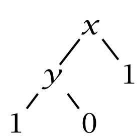
$x \rightarrow y$:



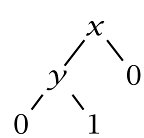
$\overline{x \rightarrow y}$:



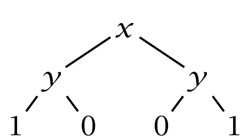
$y \rightarrow x$:



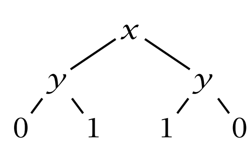
$\overline{y \rightarrow x}$:



$x \leftrightarrow y$:

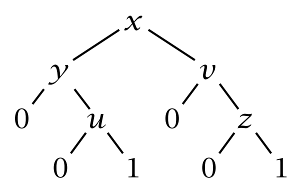


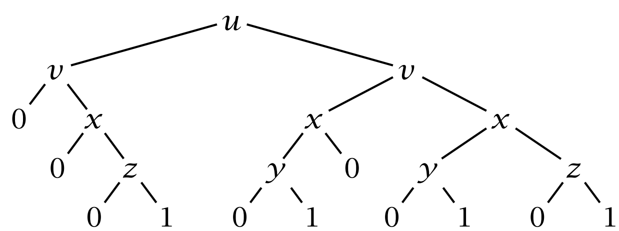
$\overline{x \leftrightarrow y}$:



Exercise 7.7 $x + \bar{y}y + \bar{z}z$

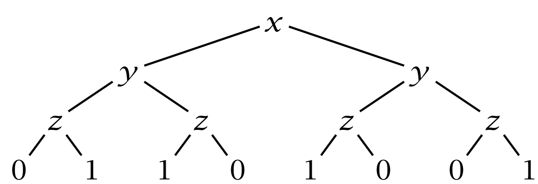
Exercise 7.8 (Order of Variables)



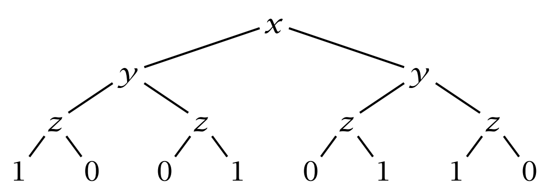


Exercise 7.9 (Computation of Prime Tree)

a) prime tree for s:



b) prime tree for $\bar{s}$:



c) the prime tree for $\hat{s}$ is the same as for s (exchange 0 and 1, and then mirror)

d) because of $s \wedge s = s$, the prime tree for $s \wedge s$ is the same as for s.

e) the prime tree for $s \vee \bar{s}$ is 1.

f) from (c) we know that $s = \hat{s}$. Therefore the prime tree for $s \leftrightarrow \hat{s}$ is 1.

Exercise 7.11 (Prime Tree Algorithms)

a)

$$
\begin{aligned}
(x, y, y) &= y \\
x \leftrightarrow 1 &= x \\
0 \leftrightarrow 0 &= 1 \\
(x, y, z) \leftrightarrow (x, y', z') &= (x, y \leftrightarrow y', z \leftrightarrow z') \\
(x, y, z) \leftrightarrow u &= (x, y \leftrightarrow u, z \leftrightarrow u) \\
x \leftrightarrow y &= y \leftrightarrow x
\end{aligned}
$$

b)

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

fun cond x s t = if s=t then t else D(x,s,t)

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