Assignment 6, Semantics of Programming Languages

Prof. Gert Smolka, Thorsten Brunklaus
www.ps.uni-sb.de/courses/sem-ws01/

Hand in until December, 12th, 2001

Notice: Assignment 6 is the last graded assignment before the first exam. To be admitted to the exam, your total score must be at least 300 points.

Exercise 6.1: (10) See exercise 4.3.8 on page 247 in Mitchell's book.

Exercise 6.2: (5) See exercise 4.3.9(b) on page 247 in Mitchell's book.

Exercise 6.3: (10) See exercise 4.3.15 on page 254 in Mitchell's book.

Exercise 6.4: (40) Consider the lambda calculus with $\rightarrow, \times, +, \mathrm{null}$ and the following term constants

$$\begin{aligned} \mathrm{zero}_\sigma &: \mathrm{null} \rightarrow \sigma \\ \mathrm{dneg}_\sigma &: ((\sigma \rightarrow \mathrm{null}) \rightarrow \mathrm{null}) \rightarrow \sigma \end{aligned}$$

Prove the following formulas of propositional logic valid by constructing a closed term of the corresponding type.

(a) $\neg a \vee \neg b \Rightarrow \neg(a \wedge b)$

(b) $\neg a \wedge \neg b \Rightarrow \neg(a \vee b)$

(c) $(a \Rightarrow c) \vee (b \Rightarrow c) \Rightarrow (a \wedge b \Rightarrow c)$

(d) $(a \wedge b) \vee (a \wedge c) \Rightarrow a \wedge (b \vee c)$

(e) $(a \Rightarrow b) \wedge (a \Rightarrow \neg b) \Rightarrow \neg a$

(f) $\neg a \vee b \Rightarrow (a \Rightarrow b)$

(g) $(a \wedge \neg b \Rightarrow c) \wedge \neg c \wedge a \Rightarrow b$

(h) $a \vee \neg a$

(i) $\neg(a \vee b) \Rightarrow \neg a \wedge \neg b$

Formulas are translated into types as follows:

$$\begin{aligned} \neg F &\quad\longmapsto\quad F \rightarrow \mathrm{null} \\ F_1 \wedge F_2 &\quad\longmapsto\quad F_1 \times F_2 \\ F_1 \vee F_2 &\quad\longmapsto\quad F_1 + F_2 \\ F_1 \Rightarrow F_2 &\quad\longmapsto\quad F_1 \rightarrow F_2 \end{aligned}$$

Hints: The first difficult formula is (f). From now on you have to use the constant dneg, which introduces the equivalence $\neg\neg F \Leftrightarrow F$. The underlying proof system is known as natural deduction. Natural deduction was discovered independently of the λ-calculus. For a very readable introduction to natural deduction (not involving λ-calculus) with many examples consult Section 1.2 of the book "Logic in Computer Science" by Huth and Ryan.

Exercise 6.5: (15) Which $\lambda^{\rightarrow}$ signature accommodates first-order predicate logic with

a, b	*constants*
f	*unary function*
g	*binary function*
p	*unary predicate*
q	*binary predicate*
$\mathrm{false}, \neg, \wedge, \exists$	*logic symbols*

in the lambda calculus?

Use the type constants i for individuals and t for truth values. Which term describes the formula

$$\neg\mathrm{false} \wedge \exists x \exists y (p(f\ (x)) \wedge q(a, g(y, x)))$$

Exercise 6.6: (20) Consider the λ-calculus with $\rightarrow, \times, +, \mathrm{unit}, \mathrm{zero}$.

(a) State the axioms and inference rules for type judgements $\Gamma \triangleright M : \sigma$.

(b) State the axioms and inference rules for equality judgements $\Gamma \triangleright M = N : \sigma$.

Hint: This exercise is straightforward since the axioms and inference rules asked for are given in Mitchell's book. You should know the rules by heart. This is easy if you understand them.