Problem Set 1

Due: April 13

Problem 1. Prove Theorem 4.5.22 (term-model completeness for simply typed lambda calculus) in the Mitchell text.

Problem 2. Consider simple types defined starting from type constants $b_1, b_2, \ldots$. A *monotone model* assigns to each type constant, b, a meaning $[\![b]\!]_0$ which is a *pointed cpo* (*cf.*, Mitchell §5.2.2). The meaning of function types is given by

$$[\![\sigma \to \tau]\!]_0 ::= [\![\sigma]\!]_0 \to_m [\![\tau]\!]$$

where $P_1 \to_m P_2$ denotes the *monotonic total functions* from a partial order, P_1, to a partial order, P_2 (*cf.*, Mitchell, §5.2.3).

(a) Prove that the monotone model is a model of the simply typed lambda-calculus (what Mitchell calls a *Henkin Model* in §4.5.3).

Let σ be a simple type and f be a function in $[\![\sigma \to \sigma]\!]_0$ in a monotone model. Define the set $F(f) \subseteq [\![\sigma]\!]_0$ inductively as follows

- $\bot_\sigma \in F(f)$,
- if $s \in F(f)$, then $f(s) \in F(f)$, and
- if S is a totally ordered subset of $F(f)$, then the least upper bound, $\bigvee S$, is in $F(f)$.

(b) Prove that $F(f)$ is totally ordered. *Hint:* Structural induction on the definition of $F(f)$.

(c) Let $a_f ::= \bigvee(F(f))$. Show that a_f is a *least fixed point* of f (*cf.*, Mitchell §5.2.4).

(d) Define $\mu_\sigma : [\![\sigma \to \sigma]\!]_0 \to [\![\sigma]\!]_0$ by the rule $\mu(f) ::= a_f$. Prove that $\mu_\sigma \in [\![(\sigma \to \sigma) \to \sigma]\!]_0$.

Problem 3. (Semigroup Word Problem). We reduced the question whether a length n 2-Counter Machine terminates to a semigroup word problem involving the $n+3$ symbol alphabet $\{\$, !, 0, \ldots, n\}$. Explain how to do it using an alphabet of only two symbols.

Problem 4. Consider the following the distributivity axioms as directed rewrite rules:

$$\begin{aligned}
(e \texttt{ * } (f \texttt{ + } g)) &\longrightarrow ((e \texttt{ * } f) \texttt{ + } (e \texttt{ * } g)), \\
((f \texttt{ + } g) \texttt{ * } e) &\longrightarrow ((f \texttt{ * } e) \texttt{ + } (g \texttt{ * } e)).
\end{aligned}$$

An expression is *flattened* when neither of these rules is applicable to it.

The directed distributivity rules are actually terminating: starting with h, *no matter where the rules are successively applied*, a flattened expression will be reached. This fact is not obvious because if e is a "large" subexpression, then the righthand side of the rule with two occurrences of e may be larger, have more redexes, *etc*. than the lefthand side with only one e.

There is an ingenious, simple way to verify this termination claim. Define the measure, m(h), of an arithmetic expression h, by induction:

- $m(h) = 2$ if h is `0`, `1`, or a variable.
- $m((e \texttt{ + } f)) = m(e) + m(f) + 1$.
- $m((e \texttt{ * } f)) = m(e) \times m(f)$.

(a) Let h' be the result of an applying one of the directed distributivity rules to some subexpression of h. Prove that $m(h') < m(h)$. Explain why termination follows immediately from this observation. *Hint:* If h is $e \texttt{ * } (f \texttt{ + } g)$ and h' is $(e \texttt{ * } f) \texttt{ + } (e \texttt{ * } g)$, then you should verify that $m(h') < m(h)$. But the general claim does not follow *solely* from this fact, since the expression that gets rewritten may be a proper subexpression of h, not the whole of h.

(b) We extend the directed distributivity rules to handle arithmetic expressions with the unary negation operator, `-`, as well:

$$\begin{aligned}
-(-e) &\longrightarrow e \\
-(f+g) &\longrightarrow (-f)+(-g) \\
-(f*g) &\longrightarrow (-f)*g
\end{aligned}$$

Verify that the rewrite system consisting of the directed distributivity rules and the three rules above is terminating on all arithmetic expressions.