Programowanie rozszerzone

Lista zadań nr 7

Na ćwiczenia 20 i 21 kwietnia 2010

Zadanie 1 (2 pkt). Dany jest następujący program P w języku IMP:

$$y := 1; \mathsf{while}\ \neg(x = 0)\ \mathsf{do}\ (y := y * z; x := x - 1)$$

Pokaż, że jeżeli $\langle P, \sigma\rangle \to \sigma'$, to $\sigma'(y) = \sigma(z)^{\sigma(x)}$.

Zadanie 2 (2 pkt). Rozważmy instrukcję lokalnie definiującą zmienną:

$$\mathsf{let}\ x\ =\ a\ \mathsf{in}\ c$$

Intuicyjna semantyka tej instrukcji jest następująca. Najpierw liczona jest wartość wyrażenia a, która zostaje przypisana zmiennej x. W tak zmodyfikowanym stanie wykonywana jest instrukcja c, a po zakończeniu jej wykonywania zmiennej x przywracana jest jej pierwotna wartość.

1. Podaj semantykę naturalną instrukcji $\mathsf{let}\ x\ =\ a\ \mathsf{in}\ c$.
2. Podaj semantykę małych kroków instrukcji $\mathsf{let}\ x\ =\ a\ \mathsf{in}\ c$. Postaraj się nie wprowadzać żadnych konstrukcji pomocniczych.

Zadanie 3 (2 pkt). Rozważmy wyrażenia arytmetyczne języka IMP:

$$(\mathsf{Aexp}) \quad a \ ::= \ n \mid x \mid a_0 + a_1 \mid a_0 - a_1 \mid a_0 * a_1$$

Definiujemy język instrukcji:

$$\begin{array}{rrcl} (\mathsf{Instr}) & c & ::= & \mathsf{return}\ n \mid \mathsf{lookup}\ x \mid \mathsf{add} \mid \mathsf{sub} \mid \mathsf{mul} \\ (\mathsf{Code}) & cs & ::= & \epsilon \mid c : cs \end{array}$$

wraz z maszyną abstrakcyjną (wirtualną) interpretującą ciągi takich instrukcji, przy użyciu stosu argumentów/wyników:

$$(\mathsf{Stack}) \quad s \ ::= \ \epsilon \mid n : s$$

Przejścia maszyny definiuje relacja przejścia $\Rightarrow \subseteq \mathsf{Conf} \times \mathsf{Conf}$, gdzie $\mathsf{Conf} = \mathsf{Code} \times \Sigma \times \mathsf{Stack}$:

$$\begin{array}{rcl} \langle \mathsf{return}\ n : cs,\ \sigma,\ s\rangle & \Rightarrow & \langle cs,\ \sigma,\ n : s\rangle \\ \langle \mathsf{lookup}\ x : cs,\ \sigma,\ s\rangle & \Rightarrow & \langle cs,\ \sigma,\ \sigma(x) : s\rangle \\ \langle \mathsf{add} : cs,\ \sigma,\ n_0 : n_1 : s\rangle & \Rightarrow & \langle cs,\ \sigma,\ n_0 + n_1 : s\rangle \\ \langle \mathsf{sub} : cs,\ \sigma,\ n_0 : n_1 : s\rangle & \Rightarrow & \langle cs,\ \sigma,\ n_0 - n_1 : s\rangle \\ \langle \mathsf{mul} : cs,\ \sigma,\ n_0 : n_1 : s\rangle & \Rightarrow & \langle cs,\ \sigma,\ n_0 * n_1 : s\rangle \end{array}$$

1. Zdefiniuj funkcję $[\![\cdot]\!] : \mathsf{Aexp} \to \mathsf{Code}$ translującą wyrażenia arytmetyczne na instrukcje maszyny.
2. Udowodnij, że Twoja funkcja $[\![\cdot]\!]$ jest poprawna ze względu na semantykę naturalną wyrażeń arytmetycznych, tzn., że $\langle a, \sigma\rangle \to n$ wtedy i tylko wtedy, gdy $\langle [\![a]\!],\ \sigma,\ \epsilon\rangle \Rightarrow^* \langle \epsilon,\ \sigma,\ n : \epsilon\rangle$.