

Semantyka języków programowania

II UW r 2013/14

Lista zadań nr 3

Na ćwiczenia 29 października 2013

Zadanie 1. Zastosuj zasadę indukcji względem definicji relacji $\langle c, \sigma \rangle \rightarrow \sigma'$ by udowodnić następujący fakt. Niech $Y \in \mathbf{Loc}$. Wówczas dla wszystkich instrukcji c oraz stanów σ i σ' , jeżeli $Y \notin \text{loc}_L(c)$ i $\langle c, \sigma \rangle \rightarrow \sigma'$, to $\sigma(Y) = \sigma'(Y)$.

Zadanie 2. Rozważmy język **IMP** rozszerzony o wyjątki. Niech **Exc** będzie (bliżej niezdefiniowaną) kategorią syntaktyczną nazw wyjątków, z metazmienną e . Rozszerzamy składnię abstrakcyjną języka o instrukcję rzucania wyjątków (**throw** e) oraz ich obsługi (**try** S **catch** $\{e_1 : c_1 \dots e_n : c_n\}$, $n \geq 0$):

$$c ::= \dots \mid \text{throw } e \mid \text{try } c \text{ catch } \{e_1 : c_1 \dots e_n : c_n\}$$

Wykonanie instrukcji **try** c **catch** $\{e_1 : c_1 \dots e_n : c_n\}$ polega na wykonaniu instrukcji c , a następnie na zareagowaniu na wynik tego wykonania: jeżeli S nie zgłasza nieobsłużonych wyjątków, to wynik jej wykonania jest wynikiem wykonania całej instrukcji **try**; jeżeli c zgłasza nieobsłużony dotąd wyjątek e_i , to zostaje on przechwycony i wykonana zostaje odpowiadająca mu instrukcja c_i obsługująca go; jeżeli zgłoszonego wyjątku nie ma na liście obsługiwanych wyjątków, to jest on propagowany. Rzucony w wyniku wykonania instrukcji **throw** e wyjątek e powoduje zignorowanie wszystkich pozostałych do wykonania instrukcji aż do najbliższej (w sensie zagnieżdżenia) instrukcji obsługującej e , o ile taka występuje.

Zadaj semantykę naturalną języka **IMP** z wyjątkami.

Zadanie 3. Rozważmy język **IMP**² będący rozszerzeniem języka **IMP** o wyrażenia warunkowe znane z języka **C**:

$$\begin{aligned} a & ::= n \mid X \mid a_0 + a_1 \mid b ? a_0 : a_1 \\ b & ::= \text{true} \mid \text{false} \mid a_0 = a_1 \mid a_0 \leq a_1 \mid \neg b \mid b_0 \wedge b_1 \mid b_0 \vee b_1 \mid b ? b_0 : b_1 \\ c & ::= \text{skip} \mid x := a \mid c_0 ; c_1 \mid \text{if } b \text{ then } c_0 \text{ else } c_1 \mid \text{while } b \text{ do } c \end{aligned}$$

1. Zdefiniuj semantykę naturalną języka **IMP**².

2. Pokaż, że każdą instrukcję języka **IMP**² można odcukrzyć do semantycznie równoważnej jej instrukcji języka **IMP**. *Wskazówka:* odcukrzanie zdefiniuj jako złożenie następujących transformacji: wyeliminowanie wyrażeń logicznych postaci $b ? b_0 : b_1$, przesunięcie wyrażeń arytmetycznych postaci $b ? a_0 : a_1$ do korzenia wyrażeń arytmetycznych, wyeliminowanie wyrażeń postaci $A_0 = A_1$ oraz $A_0 \leq A_1$, gdzie $A ::= b ? A_0 : A_1 \mid a$, wyeliminowanie instrukcji postaci $X := A$.

Zadanie 4. Pokaż, że relacja równoważności instrukcji jest kongruencją, tzn., że jest relacją równoważności oraz, że dla każdego kontekstu C , jeżeli $c_0 \sim c_1$, to $C[c_0] \sim C[c_1]$.

Które z poniższych par instrukcji są semantycznie równoważne?

1. $c_2; (\text{if } b \text{ then } c_0 \text{ else } c_1)$ oraz $\text{if } b \text{ then } (c_2; c_0) \text{ else } (c_2; c_1)$
2. $(\text{if } b \text{ then } c_0 \text{ else } c_1); c_2$ oraz $\text{if } b \text{ then } (c_0; c_2) \text{ else } (c_1; c_2)$

Zadanie 5. Udowodnij, że c_0 i c_1 są semantycznie równoważne wtedy i tylko wtedy, gdy są obserwacyjnie równoważne, przy założeniu, że obserwujemy wyłącznie terminację programów.