

Programowanie rozszerzone

Lista zadań nr 5

Na ćwiczenia 30 i 31 marca 2010

Zadanie 1 (2 pkt). Rozważmy język IMP wprowadzony na wykładzie. Niech $\text{vars}(a)$ będzie zbiorem wszystkich zmiennych występujących w wyrażeniu arytmetycznym a , $\text{vars}(b)$ – zbiorem wszystkich zmiennych występujących w wyrażeniu logicznym b , a $\text{vars}(c)$ – zbiorem wszystkich zmiennych występujących w instrukcji c .

1. Używając podanej na wykładzie zasady indukcji strukturalnej dla wyrażeń arytmetycznych udowodnij, że jeżeli $x \notin \text{vars}(a)$ i $\langle a, \sigma \rangle \rightarrow m$, to $\langle a, \sigma[n/x] \rangle \rightarrow m$, dla każdego n .
2. Sformułuj zasadę indukcji strukturalnej dla zbioru poprawnie zbudowanych wyrażeń logicznych języka IMP, a następnie udowodnij, że jeżeli $x \notin \text{vars}(b)$ i $\langle b, \sigma \rangle \rightarrow t$, to $\langle b, \sigma[n/x] \rangle \rightarrow t$, dla każdego n .
3. Sformułuj zasadę indukcji strukturalnej dla zbioru poprawnie zbudowanych instrukcji języka IMP. Czy pozwala ona na udowodnienie twierdzenia mówiącego, że jeżeli $x \notin \text{vars}(c)$ i $\langle c, \sigma \rangle \rightarrow \sigma'$, to $\langle c, \sigma[n/x] \rangle \rightarrow \sigma'$, dla każdego n ?

Zadanie 2 (2 pkt). Zadań semantykę naturalną każdej z poniższych instrukcji, albo uzasadnij, że semantyka naturalna nie nadaje się do opisu danej instrukcji:

1. if b then c ,
2. repeat c until b ,
3. for x from a_1 to a_2 do c ,
4. abort (awaryjne przerwanie wykonania całego programu),
5. $c_1 \parallel c_2$ (równoległe wykonanie instrukcji c_1 i c_2).

Zadanie 3 (2 pkt). Rozważmy zbiór Nlist wszystkich skończonych list liczb całkowitych, zdefiniowany indukcyjnie w następujący sposób:

$$ns ::= \text{nil} \mid \text{cons}(n, ns)$$

1. Sformułuj zasadę indukcji strukturalnej dla zbioru NList .
2. Zdefiniuj indukcyjnie (podaj odpowiednie aksjomaty i reguły wnioskowania) relacje $\text{append} \subseteq \text{Nlist} \times \text{Nlist} \times \text{Nlist}$ (odpowiadającą predykatowi append w Prologu) oraz $\text{reverse} \subseteq \text{Nlist} \times \text{Nlist}$ (odpowiadającą predykatowi reverse w Prologu i zdefiniowaną za pomocą relacji append).
3. Udowodnij, że dla dowolnych $ns_1, ns_2, ns_3 \in \text{Nlist}$, jeżeli $\text{reverse}(ns_1, ns_2)$ i $\text{reverse}(ns_2, ns_3)$, to $ns_1 = ns_3$.