

# Programowanie (M)

## Lista zadań nr 3

Na ćwiczenia 20 i 22 marca 2012

**Zadanie 1 (1 pkt).** W oparciu o semantykę naturalną języka **While**, zaimplementuj jego interpreter w Prologu.

**Zadanie 2 (1 pkt).** Zadać semantykę naturalną instrukcji `repeat S until b` (o intuicyjnej semantyce: powtarzaj instrukcję  $S$  aż warunek  $b$  będzie spełniony):

1. bezpośrednio podając odpowiednie reguły (które nie mogą odwoływać się do instrukcji `while`).
2. jako cukier syntaktyczny.

Pokaż następnie równoważność obu definicji.

**Zadanie 3 (1 pkt).** Zadać semantykę naturalną instrukcji `for x := a1 to a2 do S` (o intuicyjnej, celowo niesprecyzowanej, semantyce: powtarzaj  $S$  dla  $x$  od wartości  $a_1$  do wartości  $a_2$ ):

1. bezpośrednio podając odpowiednie reguły (które nie mogą odwoływać się do instrukcji `while`).
2. jako cukier syntaktyczny.

Pokaż następnie równoważność obu definicji.

**Zadanie 4 (1 pkt).** Udowodnij, że relacja  $\Rightarrow$  definiująca strukturalną semantykę operacyjną języka **While** jest deterministyczna. Następnie zdefiniuj indukcyjnie jej przechodnio-zwrotne domknięcie  $\Rightarrow^*$  i pokaż, że dla dowolnych  $S, s, s'$  i  $s''$ , jeżeli  $\langle S, s \rangle \Rightarrow^* s'$  oraz  $\langle S, s \rangle \Rightarrow^* s''$ , to  $s' = s''$ .

**Zadanie 5 (1 pkt).** Rozważmy język **While** rozszerzony o wyjątki. Niech **Exc** będzie (bliżej niezdefiniowana) kategorią syntaktyczną nazw wyjątków, z metazmienną  $e$ . Rozszerzamy składnię abstrakcyjną o instrukcję rzucania wyjątków (`throw e`) oraz ich przechwytywania i obsługi (`try S catch {e1 : S1 ... en : Sn}`,  $n \geq 0$ ):

$$S ::= \dots \mid \text{throw } e \mid \text{try } S \text{ catch } \{e_1 : S_1 \dots e_n : S_n\}$$

Wykonanie instrukcji `try S catch {e1 : S1 ... en : Sn}` polega na wykonaniu instrukcji  $S$ , 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  $S$  zgłasza nieobsłużony dotąd wyjątek  $e_i$ , to zostaje on przechwycony i wykonana zostaje odpowiadająca mu instrukcja  $S_i$  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 zagnieźdzenia) instrukcji obsługującej  $e$ , o ile taka występuje.

Zadać semantykę naturalną oraz zgodną z nią strukturalną semantykę operacyjną języka **While** z wyjątkami.

**Zadanie 6 (1 pkt).** Rozszerzamy język **While** o instrukcję  $\text{random}(x)$ , której wykonanie powoduje przypisanie zmiennej  $x$  dowolnej, niedeterministycznie wygenerowanej, liczby naturalnej. Zadać semantykę naturalną oraz zgodną z nią strukturalną semantykę operacyjną tej instrukcji.

Założmy teraz, że w języku **While** występuje instrukcja niedeterministycznego wyboru  $S_1$  or  $S_2$  z wykładu. Zaproponuj taką instrukcję  $S$ , że  $\langle \text{random}(x), s \rangle \rightarrow s'$  wtedy i tylko wtedy, gdy  $\langle S, s \rangle \rightarrow s'$ , dla wszystkich  $s$  i  $s'$ . Czy powinniśmy w takim przypadku uznać, że instrukcje te są semantycznie równoważne?