

Programowanie 2009 Programming 2009

Lista zadań nr 4 Problem set no. 4

Na zajęcia 24–25 marca 2009 Due March 25, 2009

Grupy zasadnicze Basic groups

Na wykładzie rozważaliśmy język opisany następującą składnią abstrakcyjną:

```
 $\mathcal{N} ::= \text{integers}$   
 $\mathcal{X} ::= \text{identifiers}$   
 $\mathcal{A} ::= \mathcal{N} \mid \mathcal{X} \mid \mathcal{A} \mathcal{O} \mathcal{A}$   
 $\mathcal{O} ::= + \mid - \mid * \mid \text{div} \mid \text{mod}$   
 $\mathcal{B} ::= \text{true} \mid \text{false} \mid \mathcal{A} \mathcal{R} \mathcal{A} \mid \text{not } \mathcal{B} \mid \mathcal{B} \mathcal{P} \mathcal{B}$   
 $\mathcal{R} ::= < \mid \leq \mid > \mid \geq \mid = \mid \neq$   
 $\mathcal{P} ::= \text{or} \mid \text{and}$   
 $\mathcal{C} ::= \text{skip} \mid \mathcal{X} := \mathcal{A} \mid \mathcal{C} ; \mathcal{C} \mid \text{if } \mathcal{B} \text{ then } \mathcal{C} \mid \text{if } \mathcal{B} \text{ then } \mathcal{C} \text{ else } \mathcal{C} \mid \text{while } \mathcal{B} \text{ do } \mathcal{C}$ 
```

During the lecture we considered a programming language described by the following abstract syntax:

Język ten będziemy nazywać *Abstract While*.

We will call this language *Abstract While*.

Zadanie 1 (1 pkt). Język *While C*, to język *Abstract While* uzupełniony o C-podobną składnię konkretną. Przykładem programu w tym języku jest:

Problem 1 (1 p). The *While C* language is the *Abstract While* language augmented with a C-like concrete syntax. Here is an example of a program in this language:

```
n = 5;  
s = 1;  
while (n > 1) {  
    s = n * s;  
    n = n - 1;  
}
```

Podaj gramatykę opisującą ten język. Dostosuj parser DCG przedstawiony na wykładzie do Twojej gramatyki.

Give a grammar describing this language. Adjust the DCG parser presented during the lecture to your grammar.

Zadanie 2 (1 pkt). Język *While Pascal*, to język *Abstract While* uzupełniony o pascalopodobną składnię konkretną. Przykładem programu w tym języku jest:

Problem 2 (1 p). The *While Pascal* language is the *Abstract While* language augmented with a Pascal-like concrete syntax. Here is an example of a program in this language:

```
n := 5;  
s := 1;  
while n > 1 do  
begin  
    s = n * s;  
    n = n - 1  
end
```

Podaj gramatykę opisującą ten język. Dostosuj parser DCG przedstawiony na wykładzie do Twojej gramatyki. Szczegóły dotyczące języka Pascal znajdziesz m.in. na stronie:

<http://www.moorecad.com/standardpascal/>

Give a grammar describing this language. Adjust the DCG parser presented during the lecture to your grammar. Details of the Pascal language can be found at:

Zadanie 3 (1 pkt). Uzupełnij opisaną na wykładzie semantykę operacyjną wielkich kroków (semantykę naturalną) języka While, tj. podaj zbiór reguł wnioskowania definiujących relacje

$$\begin{aligned}\langle a, \cdot \rangle \rightarrow \cdot &\subseteq \Pi \times \mathbb{Z}, & a \in \mathcal{A} \\ \langle b, \cdot \rangle \rightarrow \cdot &\subseteq \Pi \times \{T, F\}, & b \in \mathcal{B} \\ \langle c, \cdot \rangle \rightarrow \cdot &\subseteq \Pi \times \Pi, & c \in \mathcal{C}\end{aligned}$$

Problem 3 (1 p). Complete the big step semantics (the natural semantics) of the While language given during the lecture, *i.e.*, give a set of inference rules that define the relations:

Zadanie 4 (1 pkt). Uzupełnij opisaną na wykładzie semantykę operacyjną małych kroków (strukturalną semantykę operacyjną) języka While, tj. podaj zbiór reguł wnioskowania definiujących relacje

$$\begin{aligned}\langle a, \pi \rangle \rightarrow_1 \langle a', \pi' \rangle, & \quad \langle a, \pi \rangle \rightarrow_1 n; & a, a' \in \mathcal{A}, \quad \pi, \pi' \in \Pi, \quad n \in \mathbb{Z} \\ \langle b, \pi \rangle \rightarrow_1 \langle b', \pi' \rangle, & \quad \langle b, \pi \rangle \rightarrow_1 t; & b, b' \in \mathcal{B}, \quad \pi, \pi' \in \Pi, \quad t \in \{T, F\} \\ \langle c, \pi \rangle \rightarrow_1 \langle c', \pi' \rangle, & \quad \langle c, \pi \rangle \rightarrow_1 \pi'; & c, c' \in \mathcal{C}, \quad \pi, \pi' \in \Pi\end{aligned}$$

Problem 4 (1 p). Complete the small step semantics (the structural operational semantics) of the While language given during the lecture, *i.e.*, give a set of inference rules that define the relations:

Zadanie 5 (1 pkt). Niech $\rightarrow$ będzie relacją tranzycji maszyny abstrakcyjnej języka While oraz niech p będzie następującym programem (do zapisu użyliśmy składni C-podobnej):

```
p : {
p1 :   n = 3;
p2 :   s = 1;
p3 :   while (n > 2) {
p31 :       s = n * s;
p32 :       n = n - 1;
      }
}
```

Problem 5 (1 p). Let $\rightarrow$ be the transition relation of the abstract machine of the While language and let p be the following program (we used the C-like syntax to write down this program):

Pokaż, jak maszyna abstrakcyjna języka While wykonuje ten program startując z pustą pamięcią, tj. wypisz ciąg kolejnych konfiguracji tej maszyny aż do konfiguracji końcowej:

Show how the abstract machine of the While language runs this program starting with the empty memory, *i.e.*, write down a list of subsequent configurations till the final one:

$$\langle [p], [], \emptyset \rangle \rightarrow \dots \rightarrow \langle [], [], \pi_f \rangle.$$

Zadanie 6 (1 pkt). Do zbioru instrukcji języka While dodajemy instrukcję **abort** przerywającą działanie programu:

Problem 6 (1 p). We add an **abort** instruction which stops the computation of a program to the set of instructions of the While language:

$\mathcal{C} ::= \dots \mid \text{abort}$

Pokaż jak zmodyfikować a) maszynę abstrakcyjną z wykładu, b) semantykę wielkich kroków z zadania 1, c) semantykę małych kroków z zadania 2 by uwzględnić to rozszerzenie. Uwaga: to jest trudne zadanie!

Show how to modify a) the abstract machine from the lecture, b) the big step semantics from Problem 1, c) the small step semantics from Problem 2 to take this extension into account. Warning: this problem is difficult!

Zadanie 7 (1 pkt). Do zbioru instrukcji języka While dodajemy instrukcję pętli z warunkiem sprawdzanym na końcu:

$$C ::= \dots \mid \text{repeat } C \text{ until } B$$

Wykonanie instrukcji pętli `repeat c until b` kończy się, gdy warunek b jest spełniony. Pokaż, jak zmodyfikować a) maszynę abstrakcyjną z wykładu, b) semantykę wielkich kroków z zadania 1, c) semantykę małych kroków z zadania 2 by uwzględnić to rozszerzenie.

Zadanie 8 (1 pkt). Wyrażenia arytmetyczne języka While rozszerzamy o operatory post- i pre-inkrementacji oraz dekrementacji:

$$\begin{aligned} \mathcal{A} &::= \mathcal{N} \mid \mathcal{X} \mid \mathcal{AOA} \mid \mathcal{XI} \mid \mathcal{IX} \\ \mathcal{I} &::= ++ \mid -- \end{aligned}$$

Teraz więc obliczenie wyrażeń może powodować skutki uboczne. Pokaż, jak zmodyfikować a) maszynę abstrakcyjną z wykładu, b) semantykę wielkich kroków z zadania 1, c) semantykę małych kroków z zadania 2 by uwzględnić to rozszerzenie.

Problem 7 (1 p). We add a loop instruction with a condition checked at the end to the set of instructions of the While language:

The computation of a loop instruction `repeat c until b` is terminated when the condition b is satisfied. Show how to modify a) the abstract machine from the lecture, b) the big step semantics from Problem 1, c) the small step semantics from Problem 2 to take this extension into account.

Problem 8 (1 p). We extend arithmetic expressions of the While language by post- and pre-incrementation and decrementation operators:

So now the computation of an arithmetic expressions can cause side effects. Show how to modify a) the abstract machine from the lecture, b) the big step semantics from Problem 1, c) the small step semantics from Problem 2 to take this extension into account.

Grupy rozszerzone

Extended groups

Zadanie 1 (1 pkt). Oto program w języku While (używamy składni C-podobnej):

```
p : {
p1 :   y = 3;
p2 :   while (x != 0) {
p21 :     y = y * n;
p22 :     x = x - 1;
      }
}
```

Udowodnij, że jeśli $x, n \in \text{Dom}(\pi)$ oraz $\pi(x) \geq 0$, to maszyna abstrakcyjna języka While uruchomiona w konfiguracji $\langle [p], [], \pi \rangle$ zatrzyma się w konfiguracji $\langle [], [], \pi' \rangle$, gdzie $\pi'(y) = \pi(n)^{\pi(x)}$. Ile kroków obliczeń wykona maszyna dla danej pamięci π ?

Zadanie 2 (1 pkt). Niech p będzie następującym programem w języku While (użyliśmy składni C-podobnej):

```
p : {
p1 :   s = 1;
p2 :   while (n > 2) {
p21 :     s = n * s;
p22 :     n = n - 1;
      }
}
```

Problem 1 (1 p). Here is a While program (we use the C-like syntax):

Prove that if $x, n \in \text{Dom}(\pi)$ and $\pi(x) \geq 0$, then the While abstract machine started with the configuration $\langle [p], [], \pi \rangle$ will halt in configuration $\langle [], [], \pi' \rangle$, where $\pi'(y) = \pi(n)^{\pi(x)}$. How many steps will the machine perform for a given memory π ?

Problem 2 (1 p). Let p be the following program (we used the C-like syntax):

Niech $\rightarrow_1$ będzie relacją obliczenia w jednym kroku strukturalnej semantyki operacyjnej języka While. Dla $k \in \mathbb{N}$ kładziemy

$$\begin{aligned}(\rightarrow_0) &= id \\ (\rightarrow_{k+1}) &= (\rightarrow_k) \cdot (\rightarrow_1)\end{aligned}$$

gdzie id jest relacją identycznościową, a $\cdot$ jest operacją składania relacji. Udowodnij, że jeśli $\mathbf{n} \in \text{Dom}(\pi)$ oraz $\pi(\mathbf{n}) \geq 0$, to istnieje takie $k \in \mathbb{N}$, że

$$\langle p, \pi \rangle \rightarrow_k \pi'$$

gdzie $\pi'(\mathbf{s}) = \pi(\mathbf{n})!$.

Zadanie 3 (1 pkt). Pokaż, że dla dowolnego wyrażenia logicznego b i dowolnej instrukcji c programy

$$p_1 = \text{if } b \text{ do } c; \text{ while } b \text{ do } c$$

oraz

Let $\rightarrow_1$ be the one-step computation relation of the structural operational semantics of the While language. Let for $k \in \mathbb{N}$

where id is the identity relation and $\cdot$ is the operation of composition of relations. Prove that if $\mathbf{n} \in \text{Dom}(\pi)$ and $\pi(\mathbf{n}) \geq 0$, then there exists such $k \in \mathbb{N}$, that

where $\pi'(\mathbf{s}) = \pi(\mathbf{n})!$.

Problem 3 (1 p). Prove that for any Boolean expression b and any instruction c programs

and

$$p_2 = \text{while } b \text{ do } c$$

są semantycznie równoważne względem semantyki operacyjnej wielkich kroków, tj. udowodnij, że $\langle p_1, \pi \rangle \rightarrow \pi'$ zachodzi wtedy i tylko wtedy, gdy $\langle p_2, \pi \rangle \rightarrow \pi'$, dla dowolnych stanów pamięci $\pi, \pi' \in \Pi$.

Zadanie 4 (1 pkt). W języku C aby wyznaczyć wartość wyrażenia logicznego $b_1 \parallel b_2$ (odp. $b_1 \&\& b_2$) najpierw oblicza się wartość wyrażenia b_1 i jeśli okaże się ono prawdziwe (odp. fałszywe), to wyrażenia b_2 nie oblicza się (zatem wyrażenia logiczne obliczane są w C leniwie). Definicja języka Pascal nie określa sposobu wyznaczania wartości takich wyrażeń, w szczególności dopuszcza się, by zawsze obliczane były oba wyrażenia (gorliwie). Pokaż, jak w definicji maszyny abstrakcyjnej, SOS i semantyki naturalnej odróżnić, czy wyrażenia logiczne obliczane są leniwie, czy gorliwie.

Zadanie 5 (1 pkt). Do zbioru instrukcji języka While dodajemy skoki:

$$\mathcal{C} ::= \dots \mid \mathcal{L} : \mathcal{C} \mid \text{goto } \mathcal{L}$$

gdzie $\mathcal{L}$ jest kategorią *etykiet*. Semantykę języka można zadać *przez translację*, definiując odwzorowanie programów w tym języku na programy w języku, którego semantykę znamy. Opisz translację języka rozszerzonego o instrukcje skoku na oryginalny język While.

are semantically equivalent with respect to the big step operational semantics, *i.e.*, prove that $\langle p_1, \pi \rangle \rightarrow \pi'$ if and only if $\langle p_2, \pi \rangle \rightarrow \pi'$, for arbitrary memory states $\pi, \pi' \in \Pi$.

Problem 4 (1 p). To compute the value of a logical expression $b_1 \parallel b_2$ (resp. $b_1 \&\& b_2$) in the C programming language one has to compute the value of b_1 and if it turns out to be true (resp. false), then the expression b_2 is not evaluated (so logical expressions are evaluated lazily in C). The definition of Pascal does not define the way of evaluation of such expressions, in particular it is allowed to evaluate both expressions in every situation (eagerly). Show how to distinguish in the definitions of the abstract machine, SOS and the natural semantics the lazy and eager ways of evaluation of logical expressions.

Problem 5 (1 p). We add jumps to the set of instructions of the While language:

Zadanie 6 (1 pkt). Zbiór instrukcji języka While rozszerzamy o instrukcję wyboru:

$$\begin{aligned} \mathcal{C} &::= \dots \mid \text{case } \mathcal{A} \text{ of } \mathcal{D} \\ \mathcal{D} &::= \text{when } n \Rightarrow \mathcal{C} \mid \epsilon \\ \epsilon &::= \end{aligned}$$

Znaczenie instrukcji

`switch a of when  $n_1 \Rightarrow p_1 \dots \text{when } n_k \Rightarrow p_k$`

jest następujące: najpierw jest obliczane wyrażenie a , a następnie jego wartości poszukuje się wśród liczb $n_1, \dots, n_k$. Jeśli pewna liczba występuje wśród liczb $n_1, \dots, n_k$ więcej niż raz, wybiera się pierwsze jej wystąpienie. Gdy już wybrano liczbę n_i , wykonuje się instrukcję p_i . Jeżeli wśród liczb $n_1, \dots, n_k$ nie ma wartości wyrażenia a , nie wykonuje się żadnej akcji. Zadać formalnie semantykę języka While rozszerzonego o instrukcję wyboru a) przez translację do oryginalnego języka While, b) definiując maszynę abstrakcyjną, c) definiując SOS, d) definiując semantykę naturalną.

Zadanie 7 (1 pkt). Zestaw instrukcji języka While rozszerzamy o instrukcje wejścia/wyjścia:

$$\mathcal{C} ::= \dots \mid \text{read } \mathcal{X} \mid \text{write } \mathcal{A}$$

Jeżeli program może się komunikować z otoczeniem, to stan obliczeń nie jest opisany jedynie przez zawartość pamięci, ale wymaga również uwzględnienia interakcji z otoczeniem. Niech $\mathbb{Z}^0 = \epsilon$, $\mathbb{Z}^{k+1} = \mathbb{Z} \times \mathbb{Z}^k$, $\mathbb{Z}^* = \bigcup_{k=0}^{\infty} \mathbb{Z}^k$ (skończone, być może puste ciągi liczb całkowitych) oraz $\mathbb{Z}^\omega = \mathbb{Z} \times \mathbb{Z}^\omega = \prod_{k=1}^{\infty} \mathbb{Z}$ (nieskończone ciągi liczb całkowitych). Przyjmujemy, że $[]$ oznacza ciąg pusty oraz że $[x|s]$ oznacza ciąg złożony z liczby x i wszystkich elementów (skończonego lub nie) ciągu s . Stan obliczeń z uwzględnieniem operacji wejścia/wyjścia można opisać za pomocą krotki $\langle \sigma, i, o \rangle \in \mathbb{S} \times \mathbb{Z}^\omega \times \mathbb{Z}^*$, gdzie $\sigma : \mathbb{A} \rightarrow \mathbb{Z}$ jest opisem pamięci, $i \in \mathbb{Z}^\omega$ reprezentuje nieskończony ciąg danych wejściowych, zaś $o : \mathbb{Z}^*$ reprezentuje skończony ciąg danych wyjściowych wypisanych przez program. Zdefiniuj maszynę abstrakcyjną, SOS i semantykę naturalną tak rozszerzonego języka.

Zadanie 8 (1 pkt). Rozważmy program

`while b do ( $x := a$ ;  $c$ )`

Jakie warunki musi spełniać wyrażenie a , by powyższy program był semantycznie równoważny programowi

`$x := a$ ; while b do c`

Problem 6 (1 p). We extend the set of instructions of the While language with the choice instruction:

is the following: first the expression a is evaluated, then its value is searched through the numbers $n_1, \dots, n_k$. If there is more than one occurrence of the same number in the sequence $n_1, \dots, n_k$, we choose the first of them. If the number n_i has been selected, we perform the instruction p_i . If there is no value of the expression a among the numbers $n_1, \dots, n_k$, then we perform no action. Define formally the semantics of the While language augmented with the choice instruction a) by translation to the original language, b) defining its abstract machine, c) defining its SOS, d) defining its natural semantics.

Problem 7 (1 p). We extend the set of instructions of the While language with the input/output operations:

If a program can communicate with its environment, then the state of computation is not described entirely by the contents of the memory, but the interaction with the environment has to be taken into account. Let $\mathbb{Z}^0 = \epsilon$, $\mathbb{Z}^{k+1} = \mathbb{Z} \times \mathbb{Z}^k$, $\mathbb{Z}^* = \bigcup_{k=0}^{\infty} \mathbb{Z}^k$ (finite, maybe empty sequences of integers), and $\mathbb{Z}^\omega = \mathbb{Z} \times \mathbb{Z}^\omega = \prod_{k=1}^{\infty} \mathbb{Z}$ (infinite sequences of integers). Let $[]$ stands for the empty sequence and $[x|s]$ — for a sequence consisting of an element x and all the elements of a (finite or infinite) sequence s . The state of computation including input/output operations can be described by a tuple $\langle \sigma, i, o \rangle \in \mathbb{S} \times \mathbb{Z}^\omega \times \mathbb{Z}^*$, gdzie $\sigma : \mathbb{A} \rightarrow \mathbb{Z}$, where $\sigma : \mathbb{A} \rightarrow \mathbb{Z}$ is the description of the memory, $i \in \mathbb{Z}^\omega$ represents the infinite sequence of input data, and $o : \mathbb{Z}^*$ represents a finite sequence of output data written by the program. Define the abstract machine, SOS, and natural semantics of the extended language.

Problem 8 (1 p). Consider a program

What conditions does the expression a have to satisfy for the program above being equivalent to

Podaj dowód używając jednej z definicji semantyki operacyjnej języka While.

Give a proof using one of the definitions of the operational semantics of While.

Grupa zaawansowana

Advanced group

Zadanie 1 (2 pkt). Udowodnij, że maszyna abstrakcyjna języka While jest deterministyczna.

Problem 1 (2 p). Prove that the While abstract machine is deterministic.

Zadanie 2 (2 pkt). Niech $\rightarrow_1$ będzie relacją obliczenia w jednym kroku SOS języka While i niech dla $k \in \mathbb{N}_+$

Problem 2 (2 p). Let $\rightarrow_1$ be the one-step computation relation of the While SOS and let for $k \in \mathbb{N}_+$

$$\frac{\langle a, \pi \rangle \rightarrow_1 \langle a', \pi' \rangle \quad \langle a', \pi' \rangle \rightarrow_k n}{\langle a, \pi \rangle \rightarrow_{k+1} n} \quad \frac{\langle b, \pi \rangle \rightarrow_1 \langle b', \pi' \rangle \quad \langle b', \pi' \rangle \rightarrow_k v}{\langle b, \pi \rangle \rightarrow_{k+1} v} \quad \frac{\langle c, \pi \rangle \rightarrow_1 \langle c', \pi' \rangle \quad \langle c', \pi' \rangle \rightarrow_k \pi''}{\langle c, \pi \rangle \rightarrow_{k+1} \pi''}$$

oraz

and

$$\frac{\langle a, \pi \rangle \rightarrow_1 n}{\langle a, \pi \rangle \rightarrow_* n} \quad \frac{\langle a, \pi \rangle \rightarrow_1 \langle a', \pi' \rangle \quad \langle a', \pi' \rangle \rightarrow_* n}{\langle a, \pi \rangle \rightarrow_* n} \\ \frac{\langle b, \pi \rangle \rightarrow_1 v}{\langle b, \pi \rangle \rightarrow_* v} \quad \frac{\langle b, \pi \rangle \rightarrow_1 \langle b', \pi' \rangle \quad \langle b', \pi' \rangle \rightarrow_* v}{\langle b, \pi \rangle \rightarrow_* v} \\ \frac{\langle c, \pi \rangle \rightarrow_1 \pi'}{\langle c, \pi \rangle \rightarrow_* \pi'} \quad \frac{\langle c, \pi \rangle \rightarrow_1 \langle c', \pi' \rangle \quad \langle c', \pi' \rangle \rightarrow_* \pi''}{\langle c, \pi \rangle \rightarrow_* \pi''}$$

Udowodnij, że dla dowolnych $a \in \mathcal{A}$, $b \in \mathcal{B}$ oraz $c \in \mathcal{C}$ jest

Prove that for all $a \in \mathcal{A}$, $b \in \mathcal{B}$ and $c \in \mathcal{C}$ one has

$$\begin{aligned} \{(\pi, n) \mid \langle a, \pi \rangle \rightarrow_* n\} &= \bigcup_{k=1}^{\infty} \{(\pi, n) \mid \langle a, \pi \rangle \rightarrow_k n\} \\ \{(\pi, v) \mid \langle b, \pi \rangle \rightarrow_* v\} &= \bigcup_{k=1}^{\infty} \{(\pi, v) \mid \langle b, \pi \rangle \rightarrow_k v\} \\ \{(\pi, \pi') \mid \langle c, \pi \rangle \rightarrow_* \pi'\} &= \bigcup_{k=1}^{\infty} \{(\pi, \pi') \mid \langle c, \pi \rangle \rightarrow_k \pi'\} \end{aligned}$$

Zadanie 3 (2 pkt). Udowodnij równoważność semantyk małych i wielkich kroków języka While, tj. pokaż, że relacje $\rightarrow$ semantyki naturalnej i relacje $\rightarrow_*$ z poprzedniego zdania są równe.

Problem 3 (2 p). Prove the equivalence of small and big step semantics of the While language, *i.e.*, prove that the relations $\rightarrow$ of SOS and the relations $\rightarrow_*$ defined in the previous problem are equal.

Zadanie 4 (2 pkt). Udowodnij równoważność maszyny abstrakcyjnej i semantyki małych kroków języka While, tj. pokaż, że $\langle [p], [], \pi \rangle \rightarrow \langle [], [], \pi' \rangle$ wtedy i tylko wtedy, gdy $\langle p, \pi \rangle \rightarrow_* \pi'$. Czy można jakoś porównać relację $\rightarrow$ tranzycji maszyny abstrakcyjnej i relację $\rightarrow_1$ obliczenia w jednym kroku w SOS?

Problem 4 (2 p). Prove the equivalence of the small and big step semantics of the While language, *i.e.*, show that $\langle [p], [], \pi \rangle \rightarrow \langle [], [], \pi' \rangle$ if and only if $\langle p, \pi \rangle \rightarrow_* \pi'$. Is it possible to relate somehow the transition relation $\rightarrow$ of the abstract machine and the relation $\rightarrow_1$ of one-step computation of the SOS?

Zadanie 5 (2 pkt). Do zbioru instrukcji języka While dodajemy skoki:

Problem 5 (2 p). We add jumps to the set of instructions of the While language:

$$\mathcal{C} ::= \dots \mid \mathcal{L} : \mathcal{C} \mid \text{goto } \mathcal{L}$$

gdzie $\mathcal{L}$ jest klasą *etykiet*. Zdefiniuj maszynę abstrakcyjną, SOS i semantykę naturalną tak rozszerzonego języka.

where $\mathcal{L}$ is the category of *labels*. Define the abstract machine, SOS and natural semantics of the extended language.