

Semantyka i weryfikacja programów 2021/22
Egzamin 4/02/2022, zadanie 2 (semantyka denotacyjna)

Napisz semantykę denotacyjną w stylu kontynuacyjnym dla deklaracji, instrukcji i wyrażeń języka o gramatyce:

$$\begin{aligned} \text{Num} \ni n &::= 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots \\ \text{Var} \ni x &::= x \mid y \mid \dots \\ \text{FName} \ni p &::= f \mid g \mid \dots \\ \text{EName} \ni \text{exn} &::= \text{exn}_0 \mid \text{exn}_1 \mid \dots \\ \text{Expr} \ni e &::= n \mid x \mid f() \mid e_1 + e_2 \mid e_1 * e_2 \mid e_1 - e_2 \\ \text{BExpr} \ni b &::= \text{true} \mid \text{false} \mid e_1 < e_2 \mid e_1 = e_2 \mid b_1 \wedge b_2 \mid \text{not } b \\ \text{Decl} \ni d &::= \text{var } x = e \mid \text{fun } f \text{ upon } \text{exn} \text{ return } e \text{ do } (I) \mid d_1; d_2 \\ \text{Instr} \ni I &::= x := e \mid I_1; I_2 \mid \text{if } b \text{ then } I_1 \text{ else } I_2 \mid \text{while } b \text{ do } I \mid \text{begin } d; I \text{ end} \mid \text{return } e \mid \text{throw } \text{exn} \end{aligned}$$

Jest to język z rekurencyjnymi funkcjami bezparametrowymi oraz z mechanizmem zgłaszania i wychwytywania wyjątków. Zmienne oraz funkcje deklarowane są w blokach `begin d; I end`; widoczność zmiennych oraz funkcji jest statyczna. Wyliczanie wyrażeń odbywa się od lewej do prawej (co ma znaczenie, gdyż funkcje mogą mieć efekty uboczne).

Wywołanie $f()$ funkcji wprowadzonej deklaracją `fun f upon exn return e do (I)` realizowane jest przez wykonanie instrukcji I (ciała funkcji).

- Instrukcja `return  $e'$` napotkana w ciele I funkcji f natychmiast przerywa to wywołanie. Zachowywany jest bieżący stan, a wartością tego wywołania $f()$ jest wartość wyrażenia e' wyliczona w tym stanie.
- Instrukcja `throw exn` przerywa wykonanie najgłębiej zagnieżdżonego wywołania funkcji f' , która zadeklarowała wyjątek o nazwie exn — a więc także bieżącego wykonania funkcji f — i, zachowując bieżący stan, przechodzi do obliczania wartości wyrażenia e' związanego w deklaracji funkcji f' z wyjątkiem exn . Wartość tego wyrażenia będzie wartością przerwanej wywołania $f'()$ (*vide* przykład poniżej).

Dopuszczamy możliwość, że obliczanie wartości wyrażenia e' z deklaracji funkcji f' może znów doprowadzić do wykonania kolejnej instrukcji `throw exn`: wtedy kończy ona wywołanie kolejnej najgłębiej zagnieżdżonej funkcji, która zadeklarowała wyjątek nazwie exn .

Wywołanie `throw exn` w sytuacji, gdy żadna z obecnie wywołanych funkcji nie zadeklarowała wyjątku o nazwie exn , może mieć dowolne działanie (treść zadania tego nie specyfikuje).

- Jeśli wykonanie ciała I zakończyło się bez wykonania instrukcji `return` ani `throw` to wynikiem wywołania $f()$ jest wartość 0.

Na przykład, w instrukcji:

```
begin
  var x = 10;
  fun t() upon exn0 return 10 do (throw exn3);
  fun f() upon exn1 return 20 do (throw exn2);
  begin
    fun g() upon exn2 return t() do (x := x + f());
    fun h() upon exn3 return 32 do (return x + g());
    fun i() upon exn3 return 30 do (x := x + h());

    x := x + i()
  end
end
```

przed wyjściem z zewnętrznego bloku zmienna x przyjmuje wartość 42, gdyż obliczenie `x := x + i()` spowoduje po kolei:

1. wywołanie funkcji i deklarującej wyjątek exn_3 ,
2. wywołanie funkcji h deklarującej wyjątek exn_3 ,

3. wywołanie funkcji g deklarującej wyjątek exn_2 ,
4. wywołanie funkcji f deklarującej wyjątek exn_1 ,
5. zgłoszenie wyjątku exn_2 ,
6. przerwanie obliczania funkcji g i przejście do obliczenia wyrażenia $t()$,
7. zgłoszenie wyjątku exn_3 ,
8. przerwanie obliczenia funkcji h i przejście do obliczenia wyrażenia 32 ,
9. którego wynik stanie się ostatecznym wynikiem wywołania $h()$,
10. i nastąpi powrót do obliczenia ciała funkcji i ,
11. co zwiększy x z 10 do 42,
12. a wynikiem wywołania $i()$ będzie domyślna wartość 0.

Semantyka pozostałych konstrukcji jest standardowa.

W rozwiązaniu należy podać:

- wszystkie potrzebne dziedziny pomocnicze i typy wszystkich funkcji semantycznych,
- równania semantyczne dla wyrażeń postaci x oraz $f()$,
- równania semantyczne dla instrukcji postaci **return** e oraz **throw** exn ,
- równania semantyczne dla deklaracji postaci **var** $x = e$ oraz **fun** f **upon** exn **return** e **do** (I) .

Pozostałe równania można pominąć, jeżeli są standardowe, zupełnie analogiczne do standardowych lub bardzo podobne do już napisanych (np. wyrażenia logiczne) – zaznaczając to w rozwiązaniu i opisując ewentualne potrzebne ich modyfikacje.

Można założyć, że programy nie korzystają z niezadeklarowanych funkcji. Można też założyć, że instrukcja **return** e może wystąpić tylko w ciele jakiejś funkcji. Można też przyjąć istnienie funkcji matematycznej *newloc* (zwracającej nową, nie używaną do tej pory lokację) o wybranym przez siebie typie (należy go explicite podać).