SEMANTYKA I WERYFIKACJA - ĆW. 3

1. Język TINY:

   - kategorie składniowe:
     - stałe liczbowe $N \in Num$,
     - zmienne $x \in Var$,
     - wyrażenia arytmetyczne $e \in Exp$,
     - wyrażenia logiczne $b \in BExp$,
     - instrukcje $S \in Stmt$
   - składnia:

     $$\begin{aligned} N &::= 0 \mid 1 \mid 2 \mid \cdots \\ x &::= \cdots \\ e &::= N \mid x \mid e_1 + e_2 \mid e_1 * e_2 \mid e_1 - e_2 \\ b &::= \texttt{true} \mid \texttt{false} \mid e_1 \leq e_2 \mid b_1 \wedge b_2 \mid \neg b' \\ S &::= \texttt{skip} \mid x := e \mid S_1; S_2 \mid \texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2 \mid \texttt{while } b \texttt{ do } S' \end{aligned}$$

     Na wykładzie podano semantykę operacyjną małych kroków dla instrukcji, korzystając przy tym z semantyki denotacyjnej wyrażeń. Zbiór konfiguracji to $(State \times Stmt) \cup State$, reguły np.:

     $$\frac{}{\texttt{skip},\ s \to s} \qquad \frac{[\![b]\!](s) = \texttt{ff}}{\texttt{while } b \texttt{ do } S,\ s \to \texttt{skip}, s}$$

2. **Zadanie:** Zapisać pozostałe reguły dla instrukcji.

3. **Zadanie:** Napisać semantykę operacyjną małych kroków dla wyrażeń boolowskich z leniwą ewaluacją lewostronną. Objaśnić parallel or. Przerobić semantykę instrukcji tak, by korzystała z semantyki operacyjnej wyrażeń.

   Uwaga: Podać dokładnie nowy zbiór konfiguracji.

4. **Zadanie:** Rozszerzyć język o inne konstrukcje iteracji:

   $$S ::= \cdots \mid \texttt{repeat } S' \texttt{ until } b \mid \texttt{for } x := e_1 \texttt{ to } e_2 \texttt{ do } S' \mid \texttt{do } e \texttt{ times } S' \mid \texttt{do } S' \texttt{ while } b$$

   i napisać dla nich semantykę operacyjną małych kroków.

5. **Zadanie:** Przerobić semantykę `for` powyżej tak, by e_2 było obliczane na nowo po każdym obrocie pętli.

6. **Zadanie:** Zrobić semantykę liczb binarnych z dodawaniem:

   $$e ::= \$ \mid e0 \mid e1 \mid e_1 + e_2$$

   (ten dolar to znak początku liczby, tzn. \$1001 reprezentuje liczbę binarną 1001).

Uwaga: to jest dosyć dziwna składnia, bardziej naturalne wydaje się

$$e ::= n \mid e_1 + e_2$$
$$n ::= \$ \mid e0 \mid e1$$

Rozwiązanie: Konfiguracje to wyrażenia. Dodawanie interpretujemy pisemnie:

$$\frac{}{e_1 0 + e_2 0 \to (e_1 + e_2)0} \qquad \frac{}{e_1 0 + e_2 1 \to (e_1 + e_2)1} \qquad \frac{}{e_1 1 + e_2 0 \to (e_1 + e_2)1}$$

kluczowa reguła dla przeniesienia:

$$\frac{}{e_1 1 + e_2 1 \to ((e_1 + e_2) + \$1)1}$$

kończenie obliczeń:

$$\frac{}{e + \$ = e} \qquad \frac{}{\$ + e = e}$$

(uwaga, nie można skorzystać z przemienności: $e_1 + e_2 \to e_2 + e_1$, bo to wprowadziłoby możliwość pętlenia), dodatkowo buchalteria:

$$\frac{e_1 \to e_1'}{e_1 + e_2 \to e_1' + e_2} \qquad \frac{e_2 \to e_2'}{e_1 + e_2 \to e_1 + e_2'}$$

$$\frac{e \to e'}{e0 \to e'0} \qquad \frac{e \to e'}{e1 \to e'1}$$

Pytanie: czy ta kluczowa reguła dla przeniesienia nie sprawia, że obliczenie może się nie zakończyć? Zauważmy, że wyrażenie się skomplikowało! *Odpowiedź:* Nie, dowód terminacji może iść przez indukcję po sumie dodawanych liczb.

Pytanie: A jak byśmy sobie poradzili w tej uproszczonej, naturalnej składni? *Odpowiedź:* Nijak. Konfiguracje muszą być zapisane w bogatszej składni. To częste zjawisko!

7. **Zadanie:** Dodajmy do tej semantyki operację odejmowania.