

Języki formalne i techniki translacji

Lista nr 5

Zadanie 1

Niech atrybut syntetyzowany *wart* zawiera wartość liczby binarnej wygenerowanej za pomocą następującej gramatyki

$$\begin{aligned} S &\rightarrow L.L \mid L \\ L &\rightarrow LB \mid B \\ B &\rightarrow 0 \mid 1 \end{aligned}$$

Użyj atrybutów syntetyzowanych do wyznaczenia $S.wart$.

Zadanie 2

Rozważmy gramatykę

$$\begin{aligned} W &\rightarrow W + S \mid S \\ S &\rightarrow L.L \mid L \\ L &\rightarrow LC \mid C \\ C &\rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \end{aligned}$$

generującą wyrażenia utworzone przez wyrażenia złożone z liczb rzeczywistych i całkowitych oraz operacji dodawania. Gdy dodawane są dwie liczby całkowite to wynik jest całkowity, w przeciwnym przypadku wynik jest rzeczywisty.

Podaj definicję sterowaną składnią wyznaczającą typ i wartość każdego podwyrażenia.

Zadanie 3

Rozszerz definicję sterowaną składnią z poprzedniego zadania o translację wyrażeń do postaci postfiksowej. Użyj jednoargumentowego operatora `int2real` do przekształcania wartości całkowitych tak aby oba argumenty dodawania były zawsze tego samego typu.

Zadanie 4

Rozważmy gramatykę

$$\begin{aligned} E &\rightarrow E \text{ or } T \mid T \\ T &\rightarrow T \text{ and } F \mid F \\ F &\rightarrow \text{not } F \mid (E) \mid \text{true} \mid \text{false} \end{aligned}$$

Podaj definicję sterowaną składnią dla tej gramatyki.

Zadanie 5

Podaj gramatykę atrybutową generującą wyrażenia boolowskie i arytmetyczne zbudowane ze stałych boolowskich, liczb całkowitych i rzeczywistych oraz standardowych operacji i relacji na tych wyrażeniach. Gramatyka powinna sprawdzać, czy wyrażenie da się otypować oraz wyliczać wartość generowanych wyrażeń. Zakładamy, że dla każdej stałej znamy typ i wartość oraz, że mamy niejawną konwersję z typu całkowitego na rzeczywisty.

Zadanie 6

Co wydrukuje poniższy program przy założeniu, że parametry przekazywane są do procedury p przez (a) wartość, (b) referencję, (c) kopiowanie i odtwarzanie, (d) nazwę.

```
program main;
  procedure p(x,y,z);
    begin
      y:=y+1;
      z:=z+x;
    end;
begin
  a:=2;
  b:=3;
  p(b,a,a);
  write(a);
  write(b);
end.
```

Zadanie 7

Przełóż poniższy program na kod trójadresowy, wyodrębnij bloki podstawowe i narysuj graf przepływu.

```
main()
{
  int i, a[10];
  i=0;
  while( i<10 )
  {
    a[i]=0;
    i=i+1;
  }
}
```

Zadanie 8

Napisz schemat translacji instrukcji if-else i for języka C na kod trójadresowy.