

Języki Formalne i Techniki Translacji

Ćwiczenia

Michał Budnik (234586)

Lista 8 - Zadanie 3

29 stycznia 2019

1 Zadanie

Opracuj szczegółowo algorytm analizy metodą pierwszeństwa operatorów, tak aby na wyjściu otrzymać odwrotność prawostronnego wyprowadzenia dla zadanej gramatyki operatorowej G i słowa wejściowego w . Pokaż krok po kroku jak działa algorytm dla gramatyki z zadania 1 i słowa `false or not (true or false) and true`.

2 Opracowanie algorytmu

Zadaniem relacji priorytetów jest oddzielenie uchwytów prawostronnych form zdaniowych, gdzie \leq oznacza lewy kraniec, \doteq oznacza środek uchwytu i \geq oznacza prawy kraniec. Przyjmijmy, że mamy prawostronną formę zdaniową gramatyki operatorowej. Z faktu, że po prawej stronie produkcji nie ma sąsiadujących nieterminali wynika, że nie ma ich też w prawostronnej formie zdaniowej. Możemy więc prawostronną formę zdaniową zapisać jako $\beta_0 a_1 \beta_1 \dots a_n \beta_n$, gdzie każde z β_i jest albo ϵ , albo pojedynczym nieterminalem, a każde z a_i jest pojedynczym terminalem.

Założmy, że między a_i i a_{i+1} zachodzi dokładnie jedna z relacji \leq , \doteq lub \geq . Ponadto, będziemy używali $\$$ do oznaczania krańca ciągu oraz ustalimy, że $\$ \leq b$ oraz $b \geq \$$ dla wszystkich symboli terminalnych b . Przypuśćmy teraz, że usuwamy nieterminale z ciągu i umieszczamy symbol właściwej relacji, \leq , \doteq lub \geq , między każdą parą terminali i między krańcowymi terminalami i znakami $\$$ oznaczającymi krańce ciągu. Uchwyt można znaleźć w następujący sposób:

1. Przeglądamy ciąg od lewej strony aż do napotkania pierwszego \geq .
2. Następnie przechodzimy do tyłu (w lewo) po kolejnych \doteq , aż do napotkania \leq .
3. Uchwyt składa się ze wszystkich symboli leżących między \geq z kroku pierwszego, oraz \leq z kroku drugiego. Włączają się do niego wszystkie zawarte lub otaczające nieterminale. (Dołączenie nieterminali jest konieczne, aby nie spowodować sąsiadowania dwóch nieterminali w prawostronnej formie zdaniowej).

W przypadku, gdy w uchwycie znajduje się id następuje jego redukcja do E . Po redukcji wszystkich id do E otrzymamy prawostronną formę zdaniową. Ponieważ nieterminale nie wpływają na analizę, nie musimy się przejmować ich odróżnianiem. Na stosie analizatora redukującego możemy przechowywać znacznik "nieterminal", oznaczający miejsce zajęte dla wartości atrybutów.

Jeśli między parą terminali nie zachodzi żadna z relacji, to wykryto błąd składniowy i trzeba wywołać procedurę obsługi błędu.

Algorytm formalnie przedstawiony poniżej, w rys. 2.1.

Wejście: Słowo wejściowe w i tablica relacji priorytetów (wyznaczona z gramatyki operatorowej G).

Wyjście: Jeżeli w poprawny - szkieletowe drzewo wyprowadzenia, w przeciwnym wypadku - informacje o błędzie.

Metoda: Początkowo stos zawiera $\$$, a bufor wejściowy ciąg $w\$$.

```

1: zainicjuj  $ip$  tak, aby wskazywało pierwszy symbol  $w\$$ ;
2: repeat
3:   if  $\$$  jest na wierzchołku stosu oraz  $ip$  wskazuje  $\$$  then
4:     return
5:   else begin
6:     niech  $a$  będzie symbolem z wierzchołka stosu a  $b$  - symbolem wskazywanym przez  $ip$ ;
7:     if  $a < b$  lub  $a \doteq b$  then begin
8:       odłóż  $b$  na stos;
9:       przesun  $ip$  do następnego symbolu wejściowego;
10:    else if  $a > b$  then
11:      repeat
12:        zdejmij element ze stosu;
13:      until szczytowy element ze stosu jest w relacji  $<$  z poprzednio zdjętym terminalem;
14:    else błąd()
15:    end if
16:  end if
17: until false

```

Rys 2.1: Algorytm analizy metodą pierwszeństwa operatorów

3 Działanie

Wywołanie algorytmu nastąpi dla gramatyki z zadania 1. Tablica relacji priorytetów utworzona z gramatyki operatorowej G :

	or	and	true	false	()	not	\$
or	>	<	<	<	<	>	<	>
and	>	>	<	<	<	>	<	>
true	>	>				>		>
false	>	>				>		>
(<	<	<	<	<	=	<	>
)	>	>				>		>
not	>	>	<	<	<	>	<	>
\$	<	<	<	<	<	<	<	

Rys 3.1: Tablica relacji priorytetów gramatyki G

Input: $w = \text{false}$ or not (true or false) and true
 Algorytm:

ip	Stos	Warunek	Zdjęty	Pozostały ciąg	Akcja po warunku
false	\$	\$ < false		or not (true or false) and true \$	Odłożenie
or	\$, false	false > or		not (true or false) and true \$	Redukcja
or	\$	\$ < false	false	not (true or false) and true \$	Koniec redukcji
or	\$	\$ < or	false	not (true or false) and true \$	Odłożenie
not	\$, or	or < not	false	(true or false) and true \$	Odłożenie
(\$, or, not	not < (false	true or false) and true \$	Odłożenie
true	\$, or, not, ((< true	false	or false) and true \$	Odłożenie
or	\$, or, not, (, true	true > or	false	false) and true \$	Redukcja
or	\$, or, not, ((< true	true	false) and true \$	Koniec redukcji
or	\$, or, not, ((< or	true	false) and true \$	Odłożenie
false	\$, or, not, (, or	or < false	true) and true \$	Odłożenie
)	\$, or, not, (, or, false	false >)	true	and true \$	Redukcja
)	\$, or, not, (, or	or < false	false	and true \$	Koniec redukcji
)	\$, or, not, (, or	or >)	false	and true \$	Redukcja
)	\$, or, not, ((> or	or	and true \$	Koniec redukcji
)	\$, or, not, ((\doteq)	or	and true \$	Odłożenie
and	\$, or, not, (,)) > and	or	true \$	Redukcja
and	\$, or, not, ((\doteq))	true \$	Redukcja
and	\$, or, not	not < ((true \$	Koniec redukcji
and	\$, or, not	not > and	(true \$	Redukcja
and	\$, or	or < not	not	true \$	Koniec redukcji
and	\$, or	or < and	not	true \$	Odłożenie
true	\$, or, and	and < true	not	\$	Odłożenie
\$	\$, or, and, true	true > \$	not	€	Redukcja
\$	\$, or, and	and < true	true	€	Koniec redukcji
\$	\$, or, and	and > \$	true	€	Redukcja
\$	\$, or	or < and	and	€	Koniec redukcji
\$	\$, or	or > \$	and	€	Redukcja
\$	\$	\$ < or	or	€	Koniec redukcji
\$	\$	\$ \$	or	€	HALT