

Języki formalne i metody kompilacji

Laboratorium nr 8

Stefan Sokołowski

Zadanie 1:

Proszę napisać rekursywne parsery zstępujące **budujące drzewa** (por. zad. 2 z lab. 7¹), dla gramatyk:

$$(a) \quad \langle X \rangle ::= a \mid \langle X \rangle b \langle X \rangle c \qquad (b) \quad \begin{array}{l} \langle S \rangle ::= a \mid \langle T \rangle b \\ \langle T \rangle ::= c \mid \langle S \rangle d \end{array}$$

Wskazówka:

Punktem wyjścia powinny być programy rozwiązujące zad. 2 z lab. 7, należy je tylko wzbogacić o budowanie drzew. Można wzorować się na programie demonstrowanym na wykładzie

`iis.ans-elblag.pl/~stefan/Dydaktyka/JezForm/Wyklady/07-pars-zstep.c`

— proszę zajrzeć do funkcji `czynnik`, która jest czytelnie skomentowana.

Zadanie 2 (domowe specjalne):

Proszę napisać rekursywny parser zstępujący, **budujący drzewa**, zintegrowany z analizatorem leksykalnym, dla gramatyki:

```

$$\begin{array}{l} \langle \text{program} \rangle ::= \text{BEGIN } \langle \text{deklaracje} \rangle \langle \text{instrukcje} \rangle \text{END} \\ \langle \text{deklaracje} \rangle ::= \text{dekl} \mid \langle \text{deklaracje} \rangle \text{dekl} \\ \langle \text{instrukcje} \rangle ::= \langle \text{komenda} \rangle \mid \langle \text{instrukcje} \rangle \langle \text{komenda} \rangle \\ \langle \text{komenda} \rangle ::= \text{kom} \mid \langle \text{program} \rangle \end{array}$$

```

Na przykład napis

BEGIN dekl dekl kom BEGIN dekl BEGIN dekl kom END END kom END

jest programem w sensie tej gramatyki.

Analizator leksykalny powinien być osobną funkcją, czytającą tekst z wejścia i podającą parserowi kolejne leksemy, odpowiadające napisom

dekl, kom, BEGIN, END

Uwaga:

Pierwszej osobie, która przyśle mi (na `s.sokolowski@ans-elblag.pl`) poprawne rozwiązanie tego zadania, przysługują punkty do zaliczenia laboratorium/ćwiczeń. Liczba tych punktów zależy od oryginalności i elegancji rozwiązania. W mailu proszę koniecznie podać swoje **nazwisko, numer zajęć i numer zadania**.

¹`iis.ans-elblag.pl/~stefan/Dydaktyka/JezForm/Slajdy/07c.pdf`.

Zadanie 3 (ew. dokończyć w domu):

Proszę przygotować tabelę pierwszeństw redukcji dla gramatyki

```
⟨program⟩ ::= ⟨deklaracje⟩ ⟨instrukcje⟩
⟨deklaracje⟩ ::= d | ⟨deklaracje⟩ d
⟨instrukcje⟩ ::= ⟨komenda⟩ | ⟨instrukcje⟩ ⟨komenda⟩
⟨komenda⟩ ::= k | ( ⟨program⟩ )
```

Wskazówka:

Najpierw wyznaczyć dla każdego nieterminalu A zbiory

$$fst A \quad fst^+ A \quad lst A \quad lst^+ A$$

Następnie użyć tych zbiorów do wyznaczenia tabeli pierwszeństw.

Może się okazać, że to nie jest gramatyka z pierwszeństwem, to znaczy w którejś klatce tabelki znajdzie się więcej niż jedna relacja. Tak się stanie na przykład wtedy, gdy w gramatyce istnieje produkcja leworekursywna

$$A \rightarrow Aw \tag{1}$$

oraz inna produkcja

$$B \rightarrow vxAu$$

Wtedy $A \in fst^+ A$, więc $x \doteq A$ oraz $x < A$.

W takim przypadku należy

- wprowadzić nowy nieterminal A' i wszystkie produkcje z nieterminala A zastąpić analogicznymi produkcjami z A' ; np. produkcję (1) zastąpić przez

$$A' \rightarrow A'w$$

- wprowadzić nową produkcję prowadzącą z A do A' :

$$A \rightarrow A'$$

Dla tak poprawionej gramatyki sporządzić nową tabelę pierwszeństw. Ta tabela podana jest w iis.ans-elblag.pl/~stefan/Dydaktyka/JezForm/Slajdy/08c-rozw3.pdf — ale proszę tam nie zaglądać przed solidną próbą wykonania zadania samemu.

Zadanie 4:

Użyć tabeli pierwszeństw wygenerowanej w zadaniu 3 do analizy składniowej następujących słów:

- (a) dk
- (b) ddk(dkk)k
- (c) dk(d(dkk)(dk)k)
- (d) ddkdkk

Uwaga: w przypadku (d) powinien zostać zasygnalizowany błąd.
