Języki formalne i metody kompilacji
Laboratorium nr 8

Stefan Sokołowski

Zadanie 1:

Napisać rekursywne parsery zstępujące **budujące drzewa** (por. zad. 2 z lab. 7[1]), dla gramatyk:

(a) $\langle X\rangle ::= \texttt{a} \mid \langle X\rangle\texttt{b}\langle X\rangle\texttt{c}$

(b)
$$\begin{aligned}\langle S\rangle &::= \texttt{a} \mid \langle T\rangle\texttt{b}\\ \langle T\rangle &::= \texttt{c} \mid \langle S\rangle\texttt{d}\end{aligned}$$

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**)**:**

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

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

Na przykład napis
`dekl dekl kom BEGIN dekl BEGIN dekl kom END END kom`
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. Liczba tych punktów zależy od oryginalności i elegancji rozwiązania.

---

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

Zadanie 3 (ew. dokończyć w domu)**:**

Przygotować tabelę pierwszeństw redukcji dla gramatyki

$$\begin{aligned}
\langle \text{program} \rangle &::= \langle \text{deklaracje} \rangle \ \langle \text{instrukcje} \rangle \\
\langle \text{deklaracje} \rangle &::= \texttt{d} \mid \langle \text{deklaracje} \rangle \ \texttt{d} \\
\langle \text{instrukcje} \rangle &::= \langle \text{komenda} \rangle \mid \langle \text{instrukcje} \rangle \ \langle \text{komenda} \rangle \\
\langle \text{komenda} \rangle &::= \texttt{k} \mid \texttt{(} \ \langle \text{program} \rangle \ \texttt{)}
\end{aligned}$$

Wskazówka:

Najpierw wyznaczyć dla każdego nieterminalu A zbiory

$$\mathit{fst}\, A \qquad \mathit{fst}^+ A \qquad \mathit{lst}\, A \qquad \mathit{lst}^+ A$$

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

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

oraz inna produkcja

$$B \to vxAu$$

Wtedy $A \in \mathit{fst}^+ A$, więc $x \doteq A$ oraz $x \lessdot A$.

W takim przypadku należy

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

  $$A' \to A'w$$

- wprowadzić nową produkcję prowadzącą z A do A':

  $$A \to A'$$

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.
