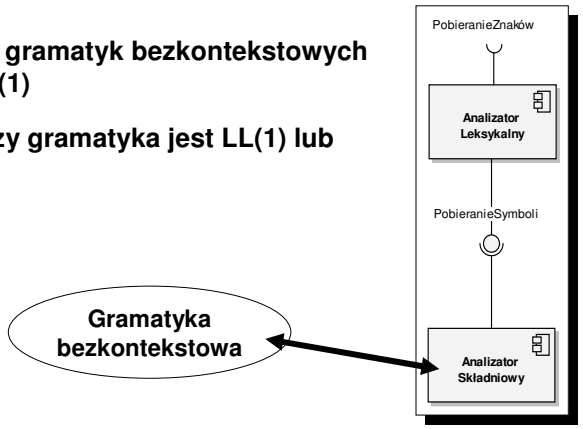
Przekształcanie gramatyk do LL(1) Gramatyki LL(k)

- **Przekształcanie gramatyk bezkontekstowych do gramatyk LL(1)**
- **Sprawdzanie, czy gramatyka jest LL(1) lub LL(k)**



Lewostronna faktoryzacja i zastępowanie

Gramatyka, gdzie:

- A → aB, A → aC, nie jest LL(1)

Lewostronna faktoryzacja polega na zastąpieniu kilku produkcji jedną oraz dodaniu nowego symbolu nieterminalnego:

- A →aX, X →B, X →C

Ta gramatyka nadal może nie być LL(1), gdy np.:

- A →aX, X →B, X → C, B →ab, C → ac

Zastępowanie polega na podstawieniu symboli terminalnych zamiast symboli nieterminalnych:

- A → aX, X →ab, X → ac, a następnie dokonaniu lewostronnej faktoryzacji
- A → aX, X →aY, Y → b, Y → c

Eliminacja lewostronnej rekurencji

Gramatyka jest lewostronnie rekurencyjna, gdy posiada produkcje postaci:

- $A \rightarrow A\alpha$, gdzie $A \in \Gamma$ i $\alpha \in (\Sigma \cup \Gamma)^*$

Eliminacja lewostronnej rekurencji polega na dodaniu nowego symbolu nieterminalego, który będzie zamieniał rekurencję na prawostronną.

Ogólnie, zestaw produkcji:

- $A \rightarrow A\alpha_1,\ A \rightarrow A\alpha_2,\ \ldots\ , A \rightarrow A\alpha_m$
- $A \rightarrow \beta_1,\ A \rightarrow \beta_2,\ \ldots\ , A \rightarrow \beta_n$

zamieniamy na następujący zestaw:

- $A \rightarrow \beta_1 A',\ A \rightarrow \beta_2 A',\ \ldots\ , A \rightarrow \beta_n A'$
- $A \rightarrow \alpha_1 A',\ A \rightarrow \alpha_2 A',\ \ldots\ , A \rightarrow \alpha_m A',\ A' \rightarrow \lambda$

Języki formalne i kompilatory © by Michał Śmiałek

Sprawdzanie gramatyk LL(1) i LL(k)

Gramatyka LL(k) wymaga znajomości „k" kolejnych symboli ze strumienia wejściowego, aby poprawnie zanalizować składnię zdania.

Dla gramatyk definiujemy zbiory „$first_k$", „$follow_k$" oraz „k-lookahead".

Definicje są podobne, jak dla zbiorów „first", „follow" i „lookahead". Zbiory t zawierają nie pojedyncze symbole, ale ciągi składające się z maksimum „k" (być może mniej) symboli.

Zbiory „follow" musimy obliczać wtedy, gdy elementy zbioru „first" mogą być krótsze niż „k".

Podobnie, jak dla gramatyk LL(1), gramatyka jest LL(k), gdy zbiory „k-lookahead" produkcji o tym samym symbolu nieterminanym z lewej strony, są rozłączne.

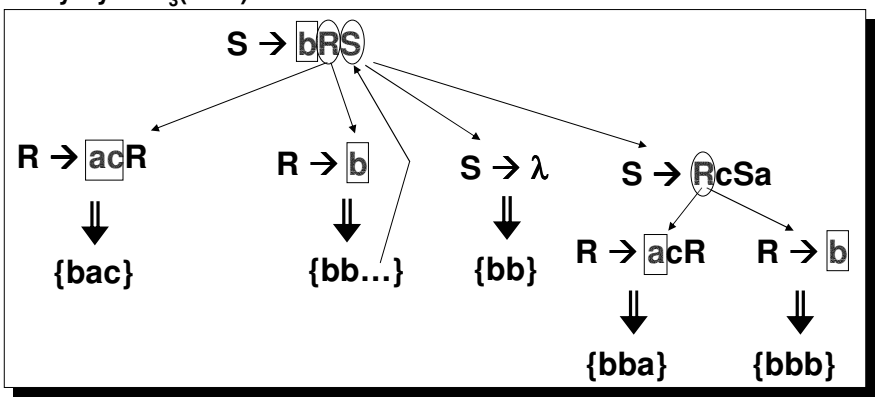
Języki formalne i kompilatory © by Michał Śmiałek

Sprawdzanie gramatyk LL(k) - ilustracja

Mamy gramatykę:

- S → bRS, S → RcSa, S → λ, R → acR, R → b

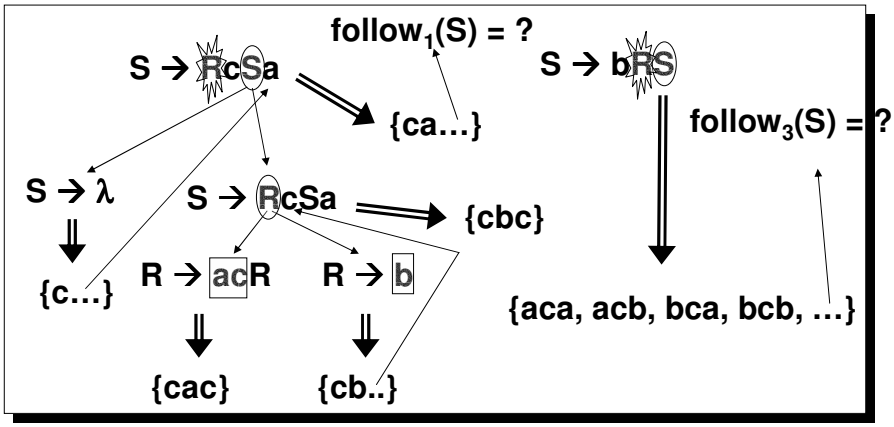
Liczymy $first_3$(bRS):



Języki formalne i kompilatory © by Michał Śmiałek

Sprawdzanie gramatyk LL(k) - ilustracja

Liczymy $follow_3$(R):



należy jeszcze policzyć $follow_1$(S) i $follow_3$(S)…

Języki formalne i kompilatory © by Michał Śmiałek

Sprawdzanie gramatyk LL(k) - ćwiczenia

Sprawdzamy, czy gramatyki o następujących zestawach produkcji są LL(1), LL(2), LL(3) (symbole pisane wielkimi literami są nieterminalne, pisane małymi literami są terminalne, S jest symbolem startowym):

- S → aA, A →S, A → λ
- S → C, S → D, C → aC, C → c, D → aD, D → d
- S → aAS, S → b, A → a, A →bS
- S → aAaB, S → bAbB, A → a, A → ab, B → aB, B → a
- S → AS, S → λ, A →aA, A →b

Proszę policzyć zbiory „lookahead" dla odpowiednich produkcji, porównać je i stwierdzić, czy gramatyka jest LL(1), LL(2), LL(3), czy też nie jest LL(k) gdzie k<4.