

Syntaktická analýza rekurzivním sestupem

(195.113.207.171:32)

Cíl: Seznámit se s rekurzivní programovou realizací syntaktického analyzátoru jazyků, generovaných LL(1) gramatikami.

Příklad 1: Gramatika 1

Mějme gramatiku, definovanou následujícími pravidly:

S -> **AB**

A -> **ah** | **abAc** | **aBc**

B -> **CDE**

C -> **d** | **e**

D -> **fD** | **f**

E -> **B** | **g**

- a) Jedná se o LL(1) gramatiku? Svůj závěr zdůvodněte!
- b) Navrhněte rekurzivní syntaktický analyzátor této nebo jí ekvivalentní gramatiky tak, že hlavní program bude součástí zdrojového kódu odpovídajícího lexikálního analyzátoru, realizovaného nástrojem *(f)lex*.

Příklady syntaktické analýzy:

Vstup_1 : ababababa

Výstup_1 : <<sekvenčně řazený seznam všech použitých a terminálů pravidel>
CHYBA, vstupní řetězec nepatří do zadane gramatiky!

Vstup_2 : adfffgcefg

Výstup_2 : <sekvenčně řazený seznam všech použitých a terminálů pravidel>
OK, aktuální vstup je generovan zadanou gramatikou.

Příklad 2: Gramatika 2

Napište syntaktický rekurzivní analyzátor jednoduché gramatiky výrazů (expressions), definované takto:

$$S \rightarrow id = E$$
$$E \rightarrow E + E \mid E * E \mid (E) \mid a \mid b$$

kde neterminální symboly jsou značeny velkými písmeny a terminály jsou výhradně jednoznakové.

- Jedná se o LL(1) gramatiku? Svůj závěr zdůvodněte!
- Navrhněte rekurzivní syntaktický analyzátor této nebo jí ekvivalentní gramatiky. Tentokrát ho ale vytvořte jako samostatný program v jazyce C, tj. bez použití specifických vývojových nástrojů.
- Kromě pravidel vytiskněte také sekvenční seznam aktuálně zpracovávaných terminálních symbolů.
- Pokuste se implementovat jednoduchý způsob obsluhy chyb, detekující problematický neterminál.

Příklady syntaktické analýzy:

Vstup_1 : aaaa

Výstup_1 : <sekvenčně řazený seznam všech použitých a terminálů pravidel>
ERROR v S

Vstup_2 : c=a+ ((b) +a)

Výstup_2 : <sekvenčně řazený seznam všech použitých a terminálů pravidel>
OK, aktuální vstup je generován zadanou gramatikou.

K urychlení práce můžete využít rozpracovaná řešení *h5_1_vzor.l* a *h5_2_vzor.c* z disku S. Sestavovací soubory (*makefile*) si vytvořte, případně editujte jejich verze z minulého týdne. Chování jednotlivých úloh můžete následně otestovat na datových souborech *dtg_1.txt* a *dtg_2.txt*, které přesměrujete na standardní vstup příslušného syntaktického analyzátoru.