

Cvičení 7: Úvod do syntaktické analýzy nástrojem *yacc/bison*

(195.113.207.171:32)

Cíl: Seznámit se se syntaxí a základními možnostmi programu *yacc/bison*, včetně způsobů jeho komunikace s generátorem lexikálních analyzátorů *lex/flex*

Příklad 1

Syntakticky analyzujte gramatiku, známou z předchozího cvičení:

S -> AB**A -> ah | abAc | aBc****B -> CDE****C -> d | e****D -> fD | f****E -> B | g**

Vzhledem k nutnosti vzniku sdíleného souboru „*.tab.h“ spusťte generátor s přepínačem *-d*. Protože zadaná gramatika je typu LALR(1), nemusíte ji nijak upravovat. K ošetření chyb použijte vestavěnou funkci *yyerror()*.

Nezapomeňte, že analyzátor implicitně očekává znak konce souboru (*Ctrl-D*). Pracujete-li interaktivně, je třeba ho za každý vstupní řetězec doplnit ručně.

Dále se seznámte s trasovacím módem programu *yacc/bison*. Ten se zapíná nastavením hodnoty proměnné *yydebug* na 1. Ve vzorovém příkladu, kde je navíc ošetřena vícenásobná definice, zapínáme trasování hodnotou 1 makra *YYDEBUG*.

Příklad 2: Jednoduchý kalkulátor - kontrola syntaxe zadaných aritmetických výrazů

Mějme gramatiku, definovanou následujícími předpisy:

- Příkazem je každý aritmetický výraz implicitního typu integer.
- Aritmetický výraz může obsahovat libovolný počet následujících operátorů:
+, -, *, /, unární mínus, unární plus a spárované závorky ()
aplikovaných buď na aritmetický výraz, nebo na číslo.

Zapište tuto gramatiku pomocí produkčních pravidel v syntaxi generátoru *yacc/bison*. Záměrně nerespektujte asociativitu a prioritu operátorů ani pravidel. Problematická místa, sumárně reportovaná v průběhu překladu:

```
h6_2.y: warning: 24 shift/reduce conflicts
```

pak najděte v souboru *h6_2.output*, vygenerovaném pomocí přepínače *-v*.

Závěrem všechny konflikty odstraňte v souladu se zavedenými zvyklostmi pro zpracování aritmetických výrazů.