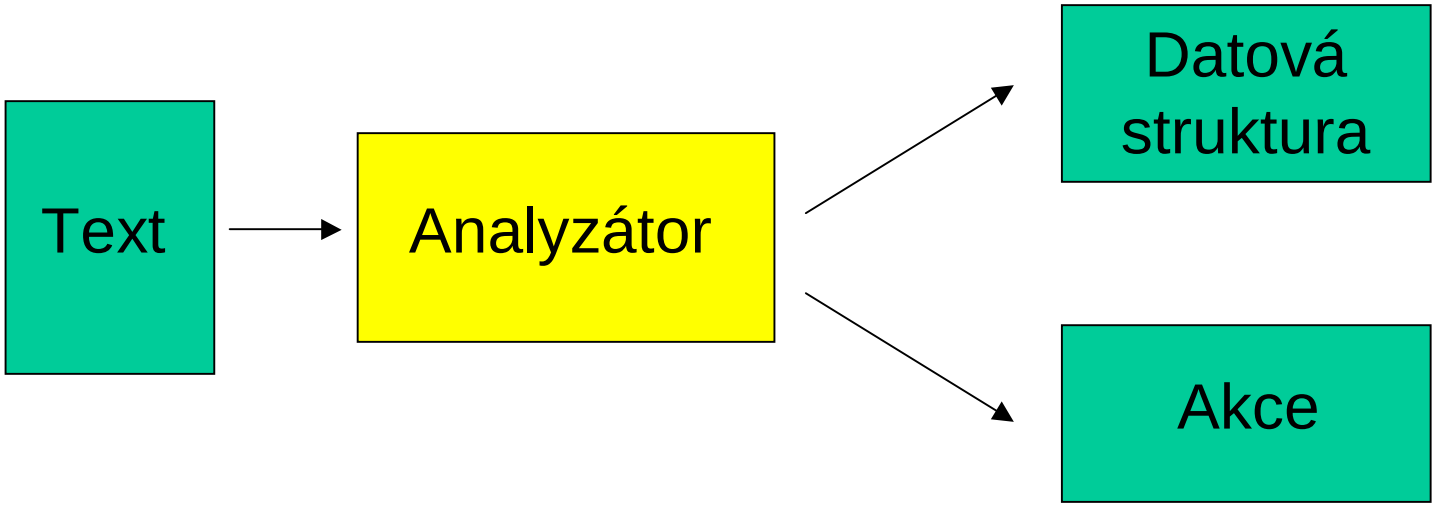
Analýza textu pomocí nástrojů lex/yacc

Miroslav Beneš
Katedra informatiky FEI VŠB-TU Ostrava

Miroslav.Benes@vsb.cz

Co chceme analyzovat?

- Konfigurační soubory
- Soubory příkazů (skripty)
- Strukturovaná data



Popis struktury textu

- Zpracovávaný text musí splňovat pravidla daná vlastnostmi *jazyka*
  - **lexikální struktura**
    (z jakých základních symbolů se text skládá – čísla, identifikátory, operátory, …)
  - **syntaktická struktura**
    (jak jsou tyto základní symboly organizovány do větších celků – výrazy, příkazy, …)

Popis lexikální struktury

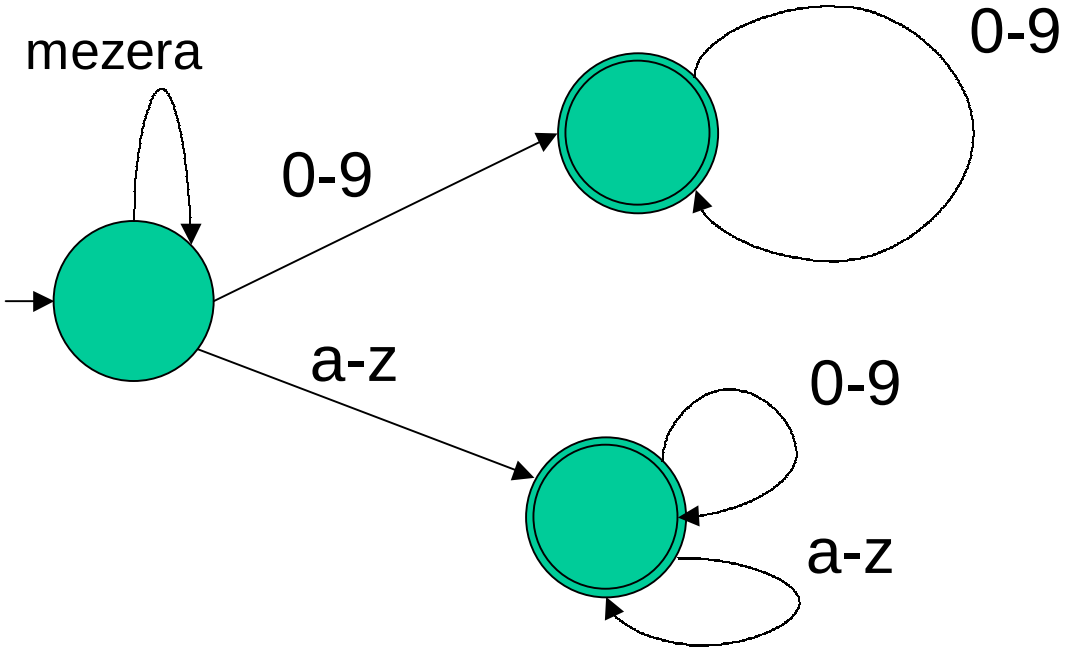
- gramatika

```
cislo      →    cis zbytek
zbytek     →    cis zbytek
           |    ε
cis        →    ‘0’
           |    …
           |    ‘9’
```

cislo ⇒ cis zbytek ⇒ cis ⇒’0’

Popis lexikální struktury

- konečný automat



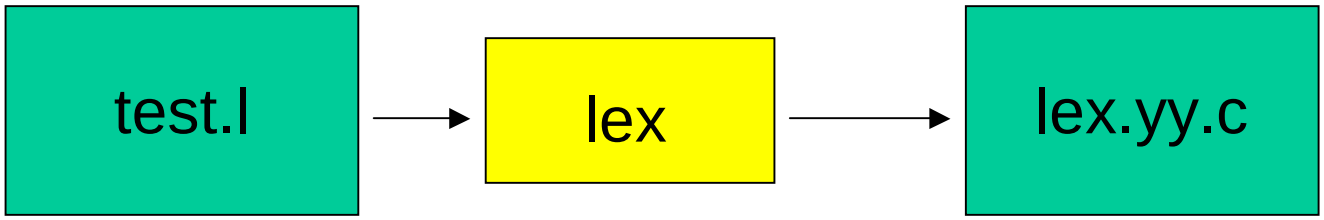
Popis lexikální struktury

- regulární výrazy
  - znaky: x “print” “/*” * \n
  - množiny znaků: [\n\t] [0-9] [0-9a-zA-Z] [^/]
  - volitelnost: x?
  - opakování: x* x+ [0-9]+
  - varianty: (x|y|z) [a-z]([0-9] | [a-z])*

[1-9][0-9]* | 0[0-7]* | 0x[0-9a-f]*+

Program lex (flex)

- Vstup: Regulární definice + akce
- Výstup: Program v C (C++)




```
int yylex();
```

Formát souboru pro lex

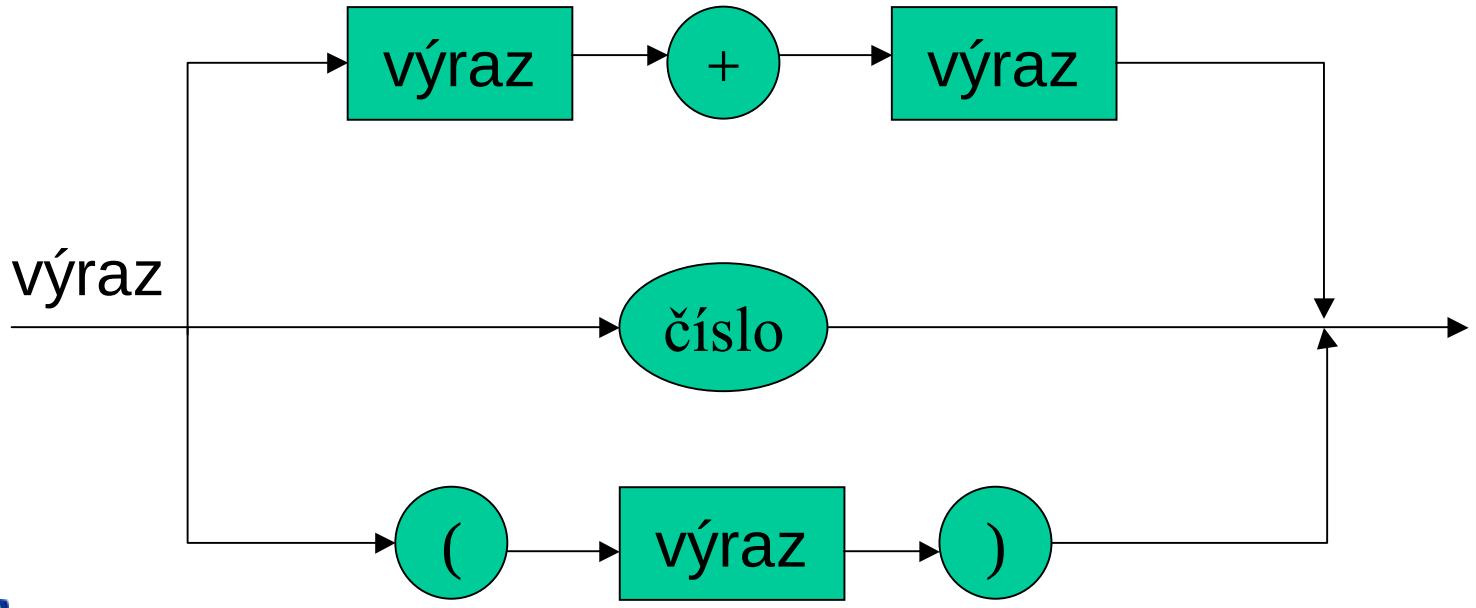
```
/* Použití jako filtr */
%{
# include “defs.h”
%}
cislice   [0-9]
pismeno   [a-z0-9]
%%
[ \n\t]+   ;
{cislice}+ printf("%s\n", yytext);
%%
int yywrap(void) { return 1; }
int main(void) { yylex(); return 0; }
```

Formát souboru pro lex

```
/* Použití jako lexikální analyzátor */
%{
# include “y_tab.h”
# include “defs.h”
%}
%%
[ \n\t]+ ;
[0-9]+   return CISLO;
div      return DIV;
mod      return MOD;
.        return yytext[0];
```

Popis syntaktické struktury

- syntaktický graf



Popis syntaktické struktury

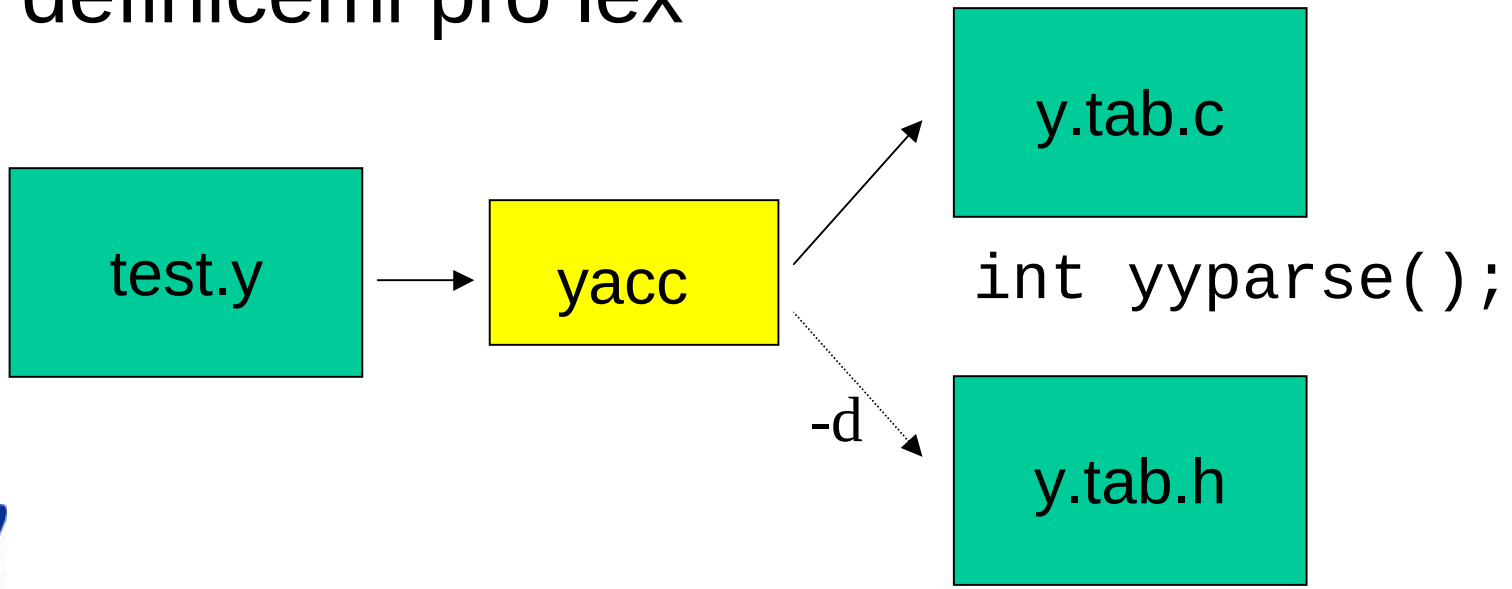
- gramatika

```
vyraz  →   vyraz + vyraz
       |   ( vyraz )
       |   CISLO
```

vyraz ⇒ vyraz+vyraz ⇒ CISLO+vyraz ⇒
CISLO+(vyraz) ⇒ CISLO+(CISLO)

Program yacc (bison)

- Vstup: Gramatika + akce v C/C++
- Výstup: Program v C/C++, záhlaví s definicemi pro lex



Formát souboru pro yacc

```
%{
# include “defs.h”
%}
%term CISLO
%%
vyraz:      vyraz '+' vyraz
      |     '(' vyraz ')'
      |     CISLO
%%
void yyerror(char* txt) {…}
int main(void) { yyparse(); return 0; }
```

Atributy

```
%{
# define YYSTYPE double
%}
%%
prog:       prog vyraz ';'
                  { printf("%g\n", $2); }
      |     /* e */
vyraz:      vyraz '+' vyraz { $$=$1+$3;}
      |     '(' vyraz ')' { $$=$2; }
      |     CISLO
```

`YYSTYPE yylval;` - nastavuje lex. analyzátor

Atributy

```
%union {
  int i;
  double d;
}
%term <i> INUM
%term <d> DNUM
%type <d> vyraz
%%
vyraz:   vyraz '+' vyraz { $$=$1+$3;}
   |     '(' vyraz ')' { $$=$2; }
   |     INUM { $$ = (double)$1; }
   |     DNUM
```

Další zdroje informací

- **`http://www.cs.vsb.cz/benes/vyuka/pre`**
  - stránky předmětu Překladače (PRE)
- **`http://epaperpress.com/lexandyacc/index.html`**
  - A Compact Guide to Lex&Yacc
- **`http://www.geocities.com/SiliconValley/Campus/3754/index.htm`**
  - Lex&Yacc Tutorial