



Linguagens Formais e Autómatos

(Ano lectivo de 2013/14)

Guiões das aulas práticas

Guião #07

Implementação de Reconhedores Gramaticais em **bison**

Sumário

Implementação de reconhedores gramaticais em **bison**.

~~Exercício 1~~ Sobre o alfabeto $T = \{a, b, c\}$, considere a gramática

$$P \rightarrow a P a \mid b P b \mid c$$

que descreve a linguagem $L = \{wcw^R \mid w \in \{a, b\}^*\}$, em que w^R representa o reverso da palavra w . O programa em **bison** seguinte aceita um conjunto de palavras separadas/terminadas por `'\n'` e para cada uma delas indica se pertencem ou não à linguagem L .

```
%{
int yylex(void);
int yyerror(char *s);
#include <stdio.h>
%}

%%

S : L
  ;

L : /* \lambda */
  | P '\n' { printf(" --> a valid word\n"); } L
  | '\n' { printf(" --> an empty word\n"); } L
  | error '\n' { printf(" --> an invalid word\n"); } L
  ;

P : 'a' P 'a'
  | 'b' P 'b'
  | 'c'
  ;

%%

#include <stdio.h>

int main(void)
{
    yyparse();
    return 0;
}

int yylex(void)
```

```

{
    return getchar();
}

int yyerror(char *s)
{
    return 0;
}

```

Teste-o.

~~Exercício 2~~ Considere a gramática

$$P \rightarrow \lambda \mid a P b P \mid c P$$

e seja L a linguagem por ela gerada.

- ~~1.~~ Que representa a linguagem L ?
- ~~2.~~ Com base nesta gramática, construa um programa em **bison** que seja um analisador sintático (parser) das palavras da linguagem L . O programa deve aceitar um conjunto de palavras separadas/terminadas por '**\n**' e para cada uma delas deve indicar se pertencem ou não a L .

~~Exercício 3~~ Considere a gramática

$$\begin{aligned}
 E &\rightarrow T \mid E + T \\
 T &\rightarrow F \mid T \cdot F \\
 F &\rightarrow K \mid \sim F \\
 K &\rightarrow 0 \mid 1 \mid (E)
 \end{aligned}$$

que representa o conjunto L das expressões booleanas.

- ~~1.~~ Com base nesta gramática, construa um programa em **bison** que seja um analisador sintático (parser) das palavras da linguagem L . O programa deve aceitar um conjunto de palavras separadas/terminadas por '**\n**' e para cada uma delas deve indicar se pertencem ou não a L .
- ~~2.~~ Altere o programa e/ou a gramática feitos na alínea anterior de modo a que se possam usar espaços nas expressões. Por exemplo, a expressão " 1 + 0 " deve ser válida e equivalente a "1+0" .

continua

Exercício 4 Sobre o alfabeto $T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, +, (,)\}$, considere a gramática

$$E \rightarrow T \mid E + T$$

$$T \rightarrow F \mid T * F$$

$$F \rightarrow N \mid (E)$$

$$N \rightarrow D \mid D N$$

$$D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$

que representa o conjunto L das expressões aritméticas usando os operadores $+$ e $*$.

- ~~1.~~ Com base nesta gramática, construa um programa em **bison** que seja um analisador sintático (parser) das palavras da linguagem L . O programa deve aceitar um conjunto de palavras separadas/terminadas por ' $\backslash n$ ' e para cada uma delas deve indicar se pertencem ou não a L .
- ~~2.~~ Altere a gramática e o programa de modo a suportar as operações de subtração e divisão.
- ~~3.~~ Altere a gramática e o programa de modo a suportar números reais.
- ~~4.~~ Altere a gramática e o programa de modo a que se possam usar espaços nas expressões. Por exemplo, as expressões $18+20$ e $18 + 20$ devem ser equivalentes.
5. Altere a gramática e o programa de modo a suportar a operação de potência.