

Ferramenta de Análise Preditiva Tabular

Êndril Castilho da Silveira, Leonardo Pellegrini Silva

Disciplina de Linguagens Formais

Departamento de Informática – UNISC

Campus de Santa Cruz do Sul

96.810-206 - Santa Cruz do Sul - RS - Brasil

endrilcs@mx2.unisc.br, leonardopellegrini@mx2.unisc.br

1. Sobre o projeto

Desenvolvido em Python 3, o aplicativo é capaz de reconhecer uma gramática, fazer a fatoração e remoção de recursividade a esquerda, tanto a direta com a indireta, Além disso, é também capaz de obter *first* e *follow* das produções, gerar a tabela de predição e tabela de reconhecimento de uma entrada fornecida pelo usuário.

2. Modo de usar

Primeiramente, deve-se inserir quantos símbolos não-terminais e terminais serão utilizados e qual dos símbolos não-terminais será o símbolo inicial. A sentença vazia não precisa ser informada, ela poderá ser inserida nas produções.

```
Construindo a gramática:
Quantos simbolos não-terminais você deseja? (De 1 a 10)
3
Seus simbolos não-terminais são:
'A', 'B', 'C'
Quantos simbolos terminais você deseja?
2
Seus simbolos terminais são:
'a', 'b',
Qual será o símbolo inicial? Escolha entre: 'A', 'B', 'C'.
A
```

Imagem 1. Construção da gramática.

Logo após, deverão ser informadas as produções da gramática, informando o lado esquerdo, e em seguida, o lado direito de cada produção. Quando informar todas as produções, pressione 0 para terminar e exibir os dados sobre a gramática inserida.

```

Agora as produções: (MÁXIMO 10)
Produção 1 - Lado esquerdo
A
Produção 1 - Lado direito
CB
Produção 2 - Lado esquerdo (0 para parar)
B
Produção 2 - Lado direito
bCB
Produção 3 - Lado esquerdo (0 para parar)
B
Produção 3 - Lado direito
&
Produção 4 - Lado esquerdo (0 para parar)
C
Produção 4 - Lado direito
a
Produção 5 - Lado esquerdo (0 para parar)
0

```

Imagem 2. Inserção das Produções.

Fornecidas estas informações, o console informará os passos que a aplicação percorreu até o momento. Serão exibidas informações como entrada e saída das funções de remoção de ambiguidade, recursão e a tabela de predição.

```

Entrando na Fatoração

Produções Iniciais: [[A] -> [C, B], [B] -> [b, C, B], [B] -> [&], [C] -> [a]]

Produções pronta para a fatoração: [[A] -> [C, B], [B] -> [b, C, B], [B] -> [&], [C] -> [a]]

Produções Resultantes: [[A] -> [C, B], [B] -> [b, C, B], [B] -> [&], [C] -> [a]]

```

Imagem 3. Eliminação de ambiguidade por meio de fatoração.

```

Entrando na Remoção de Recursão à Esquerda

Produções Iniciais: [[A] -> [C, B], [B] -> [b, C, B], [B] -> [&], [C] -> [a]]

Produções Resultantes: [[A] -> [C, B], [B] -> [b, C, B], [B] -> [&], [C] -> [a]]

Firsts:
[C -> [a], A -> [a], B -> [b, &]]

Follows:
[A -> [$], B -> [$], C -> [b, $]]

```

Imagem 4. Remoção de recursão à esquerda e definição de Firsts e Follows.

Em seguida, o programa gerará a tabela de predição baseada nos *Firsts* e *Follows* definidos anteriormente. Algo importante de ser notado é que a tabela somente exibirá os itens que possuem relação, não exibindo entradas e elementos na pilha que não são relacionados.

Tabela:

| NT na Pilha | T na Entrada | Saída |
|-------------|--------------|------------------|
| C | a | [C] -> [a] |
| A | a | [A] -> [C, B] |
| B | b | [B] -> [b, C, B] |
| B | & | [B] -> [&] |
| B | \$ | [B] -> [&] |

Imagem 5. Tabela de Predição.

Após isso, o programa espera que o usuário forneça uma entrada que será, ou não, reconhecida pela ferramenta.

```
Insira a entrada a ser reconhecida (sem espaços): abaa
+-----+-----+-----+
| Pilha | Entrada | Saída |
+=====+=====+=====+
| $A    | abaa$   |       |
+-----+-----+-----+
| $BC   | abaa$   | [A] -> [C, B] |
+-----+-----+-----+
| $Ba   | abaa$   | [C] -> [a]    |
+-----+-----+-----+
| $B    | baa$    |       |
+-----+-----+-----+
| $BCb  | baa$    | [B] -> [b, C, B] |
+-----+-----+-----+
| $BC   | aa$     |       |
+-----+-----+-----+
| $Ba   | aa$     | [C] -> [a]    |
+-----+-----+-----+
| $B    | a$      |       |
+-----+-----+-----+
Erro: Entrada não reconhecida.
```

Imagem 6. Inserção de uma entrada não reconhecida.

```

Insira a entrada a ser reconhecida (sem espaços): ababa
+-----+-----+-----+
| Pilha | Entrada | Saída |
+=====+=====+=====+
| $A    | ababa$  |       |
+-----+-----+-----+
| $BC   | ababa$  | [A] -> [C, B] |
+-----+-----+-----+
| $Ba   | ababa$  | [C] -> [a]   |
+-----+-----+-----+
| $B    | baba$   |       |
+-----+-----+-----+
| $BCb  | baba$   | [B] -> [b, C, B] |
+-----+-----+-----+
| $BC   | aba$    |       |
+-----+-----+-----+
| $Ba   | aba$    | [C] -> [a]   |
+-----+-----+-----+
| $B    | ba$     |       |
+-----+-----+-----+
| $BCb  | ba$     | [B] -> [b, C, B] |
+-----+-----+-----+
| $BC   | a$      |       |
+-----+-----+-----+
| $Ba   | a$      | [C] -> [a]   |
+-----+-----+-----+
| $B    | $       |       |
+-----+-----+-----+
| $     | $       | [B] -> [&]   |
+-----+-----+-----+

Entrada reconhecida com sucesso!

```

Imagem 7. Reconhecimento Preditivo Tabular.