

Teoria da Computação
Licenciatura em Engenharia Informática
Terceiro TPC

7 de Dezembro de 2012

Considere o alfabeto $\Sigma = \{\text{lpar}, \text{rpar}, \text{comma}, \text{underscore}, \text{digit}, \text{letter}\}$ onde:

- **lpar** pode ser realizado pelo caracter “(”;
- **rpar** pode ser realizado pelo caracter “)”;
- **comma** pode ser realizado pelo caracter “,”;
- **underscore** pode ser realizado pelo caracter “-”;
- **digit** pode ser realizado por qualquer um dos caracteres seguintes 0, 1, 2, 3, 4, 5, 6, 7, 8, 9; and
- **letter** pode ser realizado por qualquer um dos caracteres seguintes a, b, c, ..., z.

Com base no alfabeto Σ , foi definida uma linguagem livre de contexto gerada pela gramática $G = \langle V, \Sigma, E, R \rangle$ onde

$$V = \{I, Lexp, Atom, List, Lseq, Lc, Num, Nc, ID, F, IDc\}$$

e R contém as seguintes regras:

$$\begin{array}{ll} I & \rightarrow Lexp\$ \\ Lexp & \rightarrow Atom \\ Lexp & \rightarrow List \\ Atom & \rightarrow Num \\ Atom & \rightarrow ID \\ List & \rightarrow \text{lpar } Lseq \text{ rpar} \\ Lseq & \rightarrow Lexp Lc \\ Lc & \rightarrow \text{comma } List \\ Lc & \rightarrow \epsilon \\ Num & \rightarrow \text{digit } Nc \\ Nc & \rightarrow \text{digit } Nc \\ Nc & \rightarrow \epsilon \\ ID & \rightarrow F \text{ letter } IDc \\ F & \rightarrow \text{underscore} \\ F & \rightarrow \epsilon \\ IDc & \rightarrow \text{letter } IDc \\ IDc & \rightarrow \text{digit } IDc \\ IDc & \rightarrow \epsilon \end{array}$$

Exemplos de palavras aceites e rejeitadas:

<i>Aceites</i>	<i>Rejeitadas</i>
abc	_136
(24)	()
((24))	(x(24))
(_xy, (xy))	(_xy, xy)
(_xy, (xy, (34)))	(_xy, (x_y, (34)))

- Tem que implementar (em Java) um parser determinístico LL(1) para a gramática G . Para tal, tem que definir a função de transição

$$\delta : V \times (\Sigma \cup \{\$\}) \rightarrow (R \cup \{\text{SYNTAXERR}\})$$

- Calcular o conjunto $First(w)$ para cada regra $A \rightarrow w$ da gramática.
- Calcular o conjunto $Follow(H)$ para cada não terminal $H \in V$.
- Construir a tabela de transição δ .
- Implementar em Java o parser para a gramática G , representado os símbolos como uma classe Java e o autómato determinístico LL(1) usando uma pilha, tal como foi definido na disciplina. O seu programa deve ler do `stdin` uma string terminada por uma mudança de linha e apresentar a resposta **Syntax Error** ou **Accepted** numa única linha. O código pode conter mais do que um ficheiro, mas a classe principal (que inclui a função `main`) deve ter o nome **Main**.

As instruções de submissão do código ao Mooshak estão na página da disciplina no Moodle.