Universidade de Aveiro

Departamento de Electrónica, Telecomunicações e Informática

Linguagens Formais e Autómatos

(Ano Lectivo de 2013/14)

Exame de recurso 14 de Julho de 2015

NOTA: O exame tem 14 questões. As 2 mais mal classificadas serão cotadas a 1,0 valores cada; as restantes serão cotadas a 1,5 valores cada.

1. Considere, sobre o alfabeto $T = \{\, \texttt{a}, \texttt{b}, \texttt{c}, \texttt{d} \,\}$, as linguagens L_1, L_2, L_3, e L_4 definidas da seguinte forma:

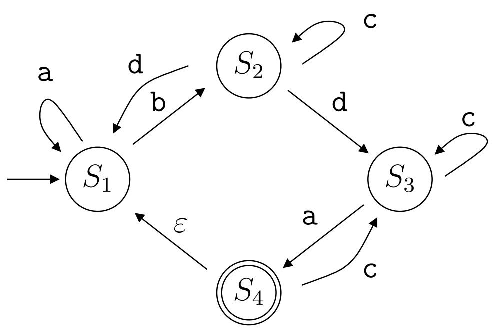
$$L_1 = \{\, \texttt{a}^n(\texttt{bcd})^k(\texttt{ca})^{n+1} \;:\; n \geq 0 \wedge k > 0 \,\}$$

$$L_2 = \{\, w \in T^* \;:\; w \texttt{ é gerada pela expressão regular } e_2 = \texttt{a}^*(\texttt{bcd})^+(\texttt{a}|\texttt{c})^* \,\}$$

$$L_3 = \{\, w \in T^* \;:\; w \texttt{ é reconhecida pelo autómato } M_3 \,\}$$

$$L_4 = \{\, w \in T^* \;:\; w \texttt{ é gerada pela gramática } G_4 \,\}$$

M_3



G_4

$$\begin{array}{lcl}
S & \rightarrow & \texttt{a}\ S\ \texttt{a} \mid X \\
X & \rightarrow & Z \mid Z\ \texttt{c}\ X \\
Z & \rightarrow & \varepsilon \mid \texttt{b}\ C\ \texttt{d} \\
C & \rightarrow & \varepsilon \mid \texttt{c}\ C
\end{array}$$

(a) Mostre que $\texttt{abcdca} \in (L_3 \cap L_4)$.

(b) Obtenha um **autómato finito, não generalizado**, que represente a linguagem L_2. Apresente o raciocínio e/ou os passos intermédios usados para chegar à sua resposta.

(c) Se se alterar o autómato M_3 passando S_1 a estado de aceitação, o autómato resultante representa a linguagem $L_3{}^*$ (fecho de L_3)? Justifique adequadamente a sua resposta.

(d) Construa um **autómato finito determinista** que reconheça a linguagem L_3. Apresente o raciocínio e/ou os passos intermédios usados para chegar à sua resposta.

(e) Determine uma **expressão regular** que represente a linguagem
$L = \{\omega \in T^* \;:\; \omega \in L_2 \vee \omega \in L_3\}$.
Apresente o raciocínio e/ou os passos intermédios usados para chegar à sua resposta.

(f) Mostre que $L_1 \subset L_3$.

(g) Projecte uma **gramática independente do contexto** que represente a linguagem L_1.

(h) Determine os conjuntos $\texttt{predict}(X \rightarrow Z)$ e $\texttt{first}(Z\ \texttt{c}\ X)$. Apresente o raciocínio e/ou os passos intermédios usados para chegar à sua resposta.

(i) Mostre que todos os símbolos não terminais da gramática G_4 são produtivos e acessíveis.

(j) Obtenha uma **gramática independente do contexto** sem transições ε (do tipo $A \rightarrow \varepsilon$) que represente a linguagem $(L_4 - \{\varepsilon\})$. Apresente o raciocínio e/ou os passos intermédios usados para chegar à sua resposta.

2. Sobre o alfabeto $T_3 = \{\texttt{NUM}, \texttt{SUBDRAW}, \texttt{LINE}, \texttt{COLOR}, \texttt{END}\}$, considere a gramática G dada a seguir e seja L a linguagem por ela descrita.

$$
\begin{array}{rcl}
\textit{draw} & \to & \textit{seq}\ \texttt{END} \\
\textit{seq} & \to & \varepsilon \\
 & | & \textit{item}\ \textit{seq} \\
\textit{item} & \to & \texttt{COLOR}\ \texttt{NUM} \\
 & | & \texttt{LINE}\ \textit{point}\ \textit{point} \\
 & | & \texttt{SUBDRAW}\ \textit{point}\ \textit{draw}\ \texttt{END} \\
\textit{point} & \to & \texttt{NUM}\ \texttt{NUM}
\end{array}
$$

(a) Trace a árvore de derivação da palavra

`LINE NUM NUM NUM NUM SUBDRAW NUM NUM COLOR NUM LINE NUM NUM NUM NUM END END`

Se quiser, ao traçar a árvore, pode abreviar a designação dos símbolos, desde que isso não afete a interpretação da sua resposta.

(b) Preencha a tabela de reconhecimento (*parsing*) para um reconhecedor descendente, com *lookahead* de 1, para a gramática G.

(c) A construção de um reconhecedor (*parser*) ascendente para uma gramática baseia-se na coleção de conjuntos de itens. O elemento inicial dessa coleção para a gramática G está parcialmente descrito a seguir.

$$Z_0 = \{\textit{draw} \to \bullet\ \textit{seq}\ \texttt{END}\} \cup \cdots$$

Complete-o e determine também os elementos diretamente alcançáveis a partir dele.

(d) Uma palavra na linguagem dada por G descreve um desenho definido por um conjunto de linhas de cores diversas, usando 3 prinitivas gráficas:

- `COLOR NUM`, que permite mudar a cor da caneta de desenho para a dada por `NUM`.
- `LINE` *point point*, que desenha uma linha entre os pontos dados, usando a cor de desenho ativa.
- `SUBDRAW` *point draw* `END`, que cria um sub-desenho com um *offset* dado por *point* em relação ao (sub-)desenho dentro do qual fica. O ponto (0,0) do sub-desenho é o ponto *point* do (sub-)desenho onde está incluído.

Apenas o símbolo terminal `NUM` tem um atributo associado, designado v, que representa um número. O símbolo não terminal *point* representa as coordenadas X e Y de um ponto. A configuração inicial do sistema é caraterizada por cor 0 e *offset* (0,0). Finalmente, considere que dispõe da função $\texttt{drawLine}(x_1, y_1, x_2, y_2, c)$ que desenha uma linha entre o ponto (x_1, y_1) e o ponto (x_2, y_2), usando uma caneta de cor c. Pretende-se construir uma gramática de atributos, parcialmente definida abaixo, que permita invocar a função `drawLine` de forma adequada no contexto da produção *item* → `LINE` *point point*. Complete a gramática de atributos.

produção	**regra semântica**
...	...
... *item* → `LINE` *point point* ...	... `drawLine`(...) ...
...	...

Nos algoritmos seguintes considere uma gramática genérica $G = (T, N, S, P)$.

ALGORITMO do `first`(α), com $\alpha \in (T \cup N)^*$:

```
{
    if (α == ε) then
        return {ε}
    else if (α == a and a ∈ T) then
        return {a}
    else if (α == B and B ∈ N) then
        M = {}
        foreach (B → γ) ∈ P
            M = M ∪ first(γ)
        return M
    else        /* |α| > 1 */
        x = head(α)      /* the first symbol */
        β = tail(α)      /* all but the first symbol */
        M = first(x)
        if ε ∉ M then
            return M
        else
            return (M − {ε}) ∪ first(β)
}
```

ALGORITMO do `follow`:

(a) $\$ \in \texttt{follow}(S)$.

(b) se $(A \to \alpha\ B) \in P$, então $\texttt{follow}(B) \supseteq \texttt{follow}(A)$.

(c) se $(A \to \alpha\ B\ \beta) \in P$ e $\varepsilon \notin \texttt{first}(\beta)$, então $\texttt{follow}(B) \supseteq \texttt{first}(\beta)$.

(d) se $(A \to \alpha B \beta) \in P$ e $\varepsilon \in \texttt{first}(\beta)$, então $\texttt{follow}(B) \supseteq ((\texttt{first}(\beta) - \{\varepsilon\}) \cup \texttt{follow}(A))$.

ALGORITMO do `predict`:

$$\texttt{predict}(A \to \alpha) = \begin{cases} \texttt{first}(\alpha) & \varepsilon \notin \texttt{first}(\alpha) \\ (\texttt{first}(\alpha) - \{\varepsilon\}) \cup \texttt{follow}(A) & \varepsilon \in \texttt{first}(\alpha) \end{cases}$$