Universidade de Aveiro

Departamento de Electrónica, Telecomunicações e Informática

Linguagens Formais e Autómatos + Compiladores

Exame teórico-prático, parte 2 (Ano Letivo de 2018/2019) modelo

1. Sobre o alfabeto $T_1 = \{\texttt{t b z w a o v n}\}$ considere a gramática G_1 dada a seguir e seja L_1 a linguagem por ela descrita.

$$
\begin{array}{rcl}
\text{P} & \rightarrow & \varepsilon \mid \text{X I } \texttt{t} \text{ P} \mid \text{X } \texttt{b} \text{ P } \texttt{z} \text{ P} \\
\text{X} & \rightarrow & \varepsilon \mid \texttt{w C} \\
\text{I} & \rightarrow & \varepsilon \mid \texttt{a} \\
\text{C} & \rightarrow & \text{T} \mid \text{C } \texttt{o} \text{ T} \\
\text{T} & \rightarrow & \texttt{v} \mid \texttt{n T}
\end{array}
$$

[1,5] (a) Mostre que a palavra `a t w n v b z` pertence a L_1.

[1,5] (b) Avalie a veracidade da afirmação: $\{\texttt{w}, \texttt{t}\} \subset \texttt{first}(\text{X I t P})$.
Apresente os passos intermédios e/ou o raciocínio adequados para suportar a sua resposta.

[1,5] (c) Avalie a veracidade da afirmação: $\texttt{t} \in \texttt{follow}(\text{T})$.
Apresente os passos intermédios e/ou o raciocínio adequados para suportar a sua resposta.

[2,0] (d) Calcule o conjunto $\texttt{predict}(\text{P} \rightarrow \text{X I t P})$.
Apresente os passos intermédios e/ou o raciocínio adequados para suportar a sua resposta.

[2,0] (e) As produções começadas por P e C tornam a gramática G_1 inadequada à implementação de um reconhecedor descendente com *lookahead* de 1. Altere-a de forma a obter uma equivalente que o permita.

---

2. Considere o alfabeto $A = \{\texttt{a}, \texttt{b}, \texttt{c}\}$ e seja L_2 o conjunto de todas as expressões regulares definíveis sobre o alfabeto A. L_2 é uma linguagem independente do contexto definida sobre o alfabeto $T_2 = A \cup \{\texttt{(}, \texttt{)}, \texttt{*}, \texttt{|}\}$, em que `*` é o operador de fecho, `|` o operador de escolha e em que o operador de concatenação é implícito. Em termos de precedência, e da mais alta para a mais baixa, estão as operações de fecho, concatenação e escolha. Os parêntesis podem ser usados para alterar a precedência por defeito.

[3,0] (.) Construa uma gramática independente do contexto que represente a linguagem L_2.

Continua no verso

3. Sobre o alfabeto $T_3 = \{\texttt{NUM}, \texttt{BOX}, \texttt{CIRCLE}, \texttt{THICKNESS}, \texttt{COLOR}, \texttt{'\{'}, \texttt{'\}'}\}$, considere a gramática G_3 dada a seguir e seja L_3 a linguagem por ela descrita.

$$
\begin{array}{rcl}
\mathit{draw} & \rightarrow & \mathit{seq} \\
\mathit{seq} & \rightarrow & \varepsilon \\
 & | & \mathit{seq}\ \mathit{item} \\
\mathit{item} & \rightarrow & \texttt{COLOR NUM} \\
 & | & \texttt{THICKNESS NUM} \\
 & | & \texttt{CIRCLE}\ \mathit{point}\ \texttt{NUM} \\
 & | & \texttt{BOX}\ \mathit{point}\ \texttt{'\{'}\ \ \mathit{seq}\ \texttt{'\}'} \\
\mathit{point} & \rightarrow & \texttt{NUM NUM}
\end{array}
$$

Considere ainda a coleção de conjunto de itens usada na contrução de um reconhecedor ascendente parcialmente apresentada a seguir, onde $\delta(Z_i, a)$ representa a função de transição de estado.

$$
\begin{aligned}
Z_0 &= \{\mathit{draw} \rightarrow \bullet\ \mathit{seq}\ ,\ \mathit{seq} \rightarrow \bullet\ ,\ \mathit{seq} \rightarrow \bullet\ \mathit{seq}\ \mathit{item}\} \\
Z_1 &= \delta(Z_0, \mathit{seq}) = \{\mathit{draw} \rightarrow \mathit{seq}\ \bullet\ ,\ \mathit{seq} \rightarrow \mathit{seq} \bullet \mathit{item}\ ,\ \mathit{item} \rightarrow \bullet\ \texttt{COLOR NUM}\ ,\ \mathit{item} \rightarrow \bullet\ \texttt{THICKNESS NUM}\ , \\
&\qquad \mathit{item} \rightarrow \bullet\ \texttt{CIRCLE}\ \mathit{point}\ \texttt{NUM}\ ,\ \mathit{item} \rightarrow \bullet\ \texttt{BOX}\ \mathit{point}\ \texttt{'\{'}\ \mathit{seq}\ \texttt{'\}'}\} \\
Z_2 &= \delta(Z_1, \mathit{item}) = \{\mathit{seq} \rightarrow \mathit{seq}\ \mathit{item}\ \bullet\} \\
Z_3 &= \delta(Z_1, \texttt{COLOR}) = \{\mathit{item} \rightarrow \texttt{COLOR} \bullet \texttt{NUM}\} \\
Z_4 &= \delta(Z_1, \texttt{THICKNESS}) = \{\mathit{item} \rightarrow \texttt{THICKNESS} \bullet \texttt{NUM}\} \\
Z_5 &= \delta(Z_1, \texttt{CIRCLE}) = \{\cdots\} \\
Z_6 &= \delta(Z_1, \texttt{BOX}) = \{\cdots\} \\
Z_7 &= \delta(Z_3, \texttt{NUM}) = \{\mathit{item} \rightarrow \texttt{COLOR NUM} \bullet\} \\
Z_8 &= \delta(Z_4, \texttt{NUM}) = \{\mathit{item} \rightarrow \texttt{THICKNESS NUM} \bullet\}
\end{aligned}
$$

[2,0] (a) Preencha as linhas da tabela de reconhecimento (*parsing*) para um reconhecedor ascendente relativamente aos estados Z_0 a Z_4.

[2,0] (b) Determine os conjuntos de itens definidores dos estados Z_5, Z_6 e de mais três, além dos apresentados.

---

4. Considere novamente a gramática G_3 dada no exercício anterior. Uma palavra na linguagem dada por G_3 descreve um desenho definido por uma sequência das seguintes operações gráficas (*item*):

   - `COLOR NUM`, que permite mudar a cor da caneta de desenho para a dada por `NUM`.
   - `THICKNESS NUM`, que permite mudar a espessura da caneta de desenho para a dada por `NUM`.
   - `CIRCLE` *point* `NUM`, que desenha um circunferência centrada no ponto dado por *point* e com raio dado por `NUM`, usando a caneta de desenho ativa.
   - `BOX` *point* `'{'` *seq* `'}'`, que cria um sub-desenho com um *offset* dado por *point* em relação ao desenho dentro do qual fica. O ponto (0,0) do sub-desenho é o ponto *point* do desenho onde está incluído.

   Apenas o símbolo terminal `NUM` tem um atributo associado, designado v e 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, espessura 1 e *offset* (0,0). Finalmente, considere que dispõe da função `drawCircle(x, y, r, c, t)` que desenha uma circunferência centrada no ponto `(x,y)`, com raio `r`, usando uma caneta de desenho com cor `c` e espessura `t`.

[1,5] (a) Trace a árvore de derivação da palavra

   `COLOR NUM CIRCLE NUM NUM NUM BOX NUM NUM '{' THICKNESS NUM CIRCLE NUM NUM NUM '}'`

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

[3,0] (b) Construa uma gramática de atributos que permita invocar a função `drawCircle` de forma adequada para cada circunferência incluída num desenho.

CÁBULA OFICIAL

ALGORITMO do `first`:

```
first(α) {
    if (α == λ) then
        return {λ}
    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 λ ∉ M then
            return M
        else
            return (M − {λ}) ∪ first(β)
}
```

ALGORITMO do `follow`:

1. $\$ \in \texttt{follow}(S)$.
2. se $(A \to \alpha\ B) \in P$, então $\texttt{follow}(B) \supseteq \texttt{follow}(A)$.
3. se $(A \to \alpha\ B\ \beta) \in P$ e $\lambda \notin \texttt{first}(\beta)$, então $\texttt{follow}(B) \supseteq \texttt{first}(\beta)$.
4. se $(A \to \alpha\ B\ \beta) \in P$ e $\lambda \in \texttt{first}(\beta)$, então $\texttt{follow}(B) \supseteq ((\texttt{first}(\beta) - \{\lambda\}) \ \cup\ \texttt{follow}(A))$.

ALGORITMO do `predict`:

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