



ALGORITMO do predict:

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

ALGORITMO do first:

```
first(α) {
    if (α = ε) then
        return {ε}
    h = head (α)      # com |h| = 1
    ω = tail (α)      # tal que α = h ω
    if (h ∈ T) then
        return {h}
    else
        return ∪(h → βi) ∈ P first(βi ω)
}
```

ALGORITMO do follow:

1. \$ ∈ follow(S)
2. if (A → αB ∈ P) then  
follow(B) ⊇ follow(A)
3. if (A → αBβ ∈ P) ∧ (ε ∉ first(β)) then  
follow(B) ⊇ first(β)
4. if (A → αBβ ∈ P) ∧ (ε ∈ first(β)) then  
follow(B) ⊇ ((first(β) - {ε}) ∪ follow(A))

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

$$\begin{array}{lcl} P & \rightarrow & \varepsilon \mid X \ I \ t \ P \mid X \ b \ P \ z \ P \\ X & \rightarrow & \varepsilon \mid w \ C \\ I & \rightarrow & \varepsilon \mid a \qquad \qquad \tau \text{ (totot)} \\ C & \rightarrow & T \mid C \ o \ T \xrightarrow{\tau} C \rightarrow \tau X \\ T & \rightarrow & v \mid n \ T \qquad \qquad X \rightarrow o TX \qquad \qquad \downarrow \varepsilon \end{array}$$

$$\begin{aligned} \text{first}(XItP) &= \text{first}((\varepsilon \mid wC)ItP) \\ &= \text{first}(I \mid P) \cup \text{first}(w \dots) \\ &= \text{first}(a \mid P) \cup \text{first}(t \mid P) \cup \text{first}(z \mid P) \\ &= \{a, t, w\} \end{aligned}$$

$$\begin{aligned} \text{Assim: } \{w, t \in C \mid a, t, w\} &= \text{first}(XItP) \\ &\uparrow \\ &\text{não é igual pois } a \in \text{first}(XItP) \\ &\text{mas } a \notin \{w, t\} \end{aligned}$$

- [ 1,5 ] (a) Mostre que  $atwnvzbz \in L_1$ .  
 [ 1,5 ] (b) Avalie a veracidade da afirmação:  $\{w, t\} \subset \text{first}(XItP)$ .  
           Apresente os passos intermédios e/ou o raciocínio adequados para suportar a sua resposta.  
 [ 1,5 ] (c) Avalie a veracidade da afirmação:  $t \in \text{follow}(T)$ .  
           Apresente os passos intermédios e/ou o raciocínio adequados para suportar a sua resposta.  
 [ 2,0 ] (d) Calcule o conjunto  $\text{predict}(P \rightarrow XbPzP)$ .  
           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  $A = \{a, b, c\}$  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  $T_2 = A \cup \{(), *, +\}$ , em que \* representa o operador de fecho e + o operador de escolha; operação de concatenação tem o operador implícito. Em termos de precedência, 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  $L_2$ .

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

```

draw   →   seq
seq    →   ε
      |   seq item
item   →   COLOR NUM
      |   THICKNESS NUM
      |   CIRCLE point NUM
      |   BOX point '{' seq '}' ,
point  →   NUM NUM
  
```

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

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

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

	NUM	BOX	CIRCLE	THICKNESS	COLOR	{	}	\$	draw	seq	item	point
$Z_0$		R1	R1	R1	R1			R1	R1		$Z_1$	
$Z_1$		Shift, $Z_6$	Shift, $Z_6$	Shift, $Z_1$	Shift, $Z_3$				ACCEPT			$Z_2$
$Z_2$		R3	R3	R3	R3			R3	R3			
$Z_3$	Shift $Z_7$											
$Z_4$	Shift $Z_8$											

Reduções:  
R1: seq → ε  
R3: seq → seq item  
follow(seq) = {BOX, CIRCLE, THICKNESS, COLOR, {, }, \$}

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

$$\begin{aligned}
Z_5 &= \{ \text{item} \rightarrow \text{CIRCLE} \bullet \text{ point NUM} \} \cup \{ \text{point} \rightarrow \bullet \text{ NUM NUM} \} \\
Z_6 &= \{ \text{item} \rightarrow \text{BOX} \bullet \text{ point } \{ \text{ seq } \} \} \cup \{ \text{point} \rightarrow \bullet \text{ NUM NUM} \} \\
Z_9 &= \delta(Z_5, \text{point}) = \{ \text{item} \rightarrow \text{CIRCLE point} \bullet \text{ NUM} \} \\
Z_{10} &= \delta(Z_5, \text{NUM}) = \{ \text{point} \rightarrow \text{NUM} \bullet \text{ NUM} \} \\
Z_{11} &= \delta(Z_6, \text{point}) = \{ \text{item} \rightarrow \text{BOX point} \bullet \text{ seq } \{ \text{ } \}
\end{aligned}$$

$Z_0 = \{\text{draw} \rightarrow \bullet \text{ seq}, \text{seq} \rightarrow \bullet, \text{seq} \rightarrow \bullet \text{ seq item}\}$ $Z_1 = \delta(Z_0, \text{seq}) = \{\text{draw} \rightarrow \text{seq} \bullet, \text{seq} \rightarrow \text{seq} \bullet \text{ item}, \text{item} \rightarrow \bullet \text{ COLOR NUM}, \text{item} \rightarrow \bullet \text{ THICKNESS NUM},$ $\text{item} \rightarrow \bullet \text{ CIRCLE point NUM}, \text{item} \rightarrow \bullet \text{ BOX point } \{, \text{seq } \}'\}$ $Z_2 = \delta(Z_1, \text{item}) = \{\text{seq} \rightarrow \text{seq item} \bullet\}$ $Z_3 = \delta(Z_1, \text{COLOR}) = \{\text{item} \rightarrow \text{COLOR} \bullet \text{ NUM}\}$ $Z_4 = \delta(Z_1, \text{THICKNESS}) = \{\text{item} \rightarrow \text{THICKNESS} \bullet \text{ NUM}\}$ $Z_5 = \delta(Z_1, \text{CIRCLE}) = \{\dots\}$ $Z_6 = \delta(Z_1, \text{BOX}) = \{\dots\}$ $Z_7 = \delta(Z_3, \text{NUM}) = \{\text{item} \rightarrow \text{COLOR NUM} \bullet\}$ $Z_8 = \delta(Z_4, \text{NUM}) = \{\text{item} \rightarrow \text{THICKNESS NUM} \bullet\}$
---

draw	→	seq
seq	→	$\varepsilon$
seq	→	seq item
item	→	COLOR NUM
		THICKNESS NUM
		CIRCLE point NUM
point	→	BOX point '{ seq }'
		NUM NUM

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 círculo centrado 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 caracterizada 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  $t$ .

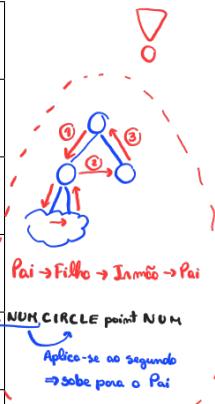
- [ 1,5 ] (a) Trace a árvore de derivação da palavra

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

Se quiser, ao traçar a árvore, pode abreviar a designação dos símbolos, usando n, cr, co, t, b, s, i e p em vez de NUM, CIRCLE, COLOR, THICKNESS, BOX, seq, item e point, respectivamente.

- [ 3,0 ] (b) Complete a gramática de atributos abaixo tal que ela adequadamente invoque a função `drawCircle` para cada circunferência incluída numa descrição em  $L_3$ .

Production	Semantic rule
draw → seq	$seq.c = 0, seq.t = 1, seq.x = 0, seq.y = 0$
seq → $\varepsilon$	①      ②      ③
seq0 → seq1 item	$seq1.c = seq0.c$ $item.c = seq1.c$ $seq0.c = item.c$ $seq1.t = seq0.t$ $item.t = seq1.t$ $seq0.t = item.t$ $seq1.x = seq0.x, seq1.y = seq0.y$ $item.x = seq1.x, item.y = seq1.y$
item → COLOR NUM	$item.c = NUM.v$
item → THICKNESS NUM	$item.t = NUM.v$
item → CIRCLE point NUM	$point.x = point.x + item.x, point.y = point.y + item.y$ $drawCircle(point.x, point.y, NUM.v, item.c, item.t)$
item → BOX point { seq }	$Seq.x = point.x + item.x, Seq.y = point.y + item.y$ $Seq.c = item.c, Seq.t = item.t$
point → NUM1 NUM2	$point.x = NUM1.v, point.y = NUM2.v$

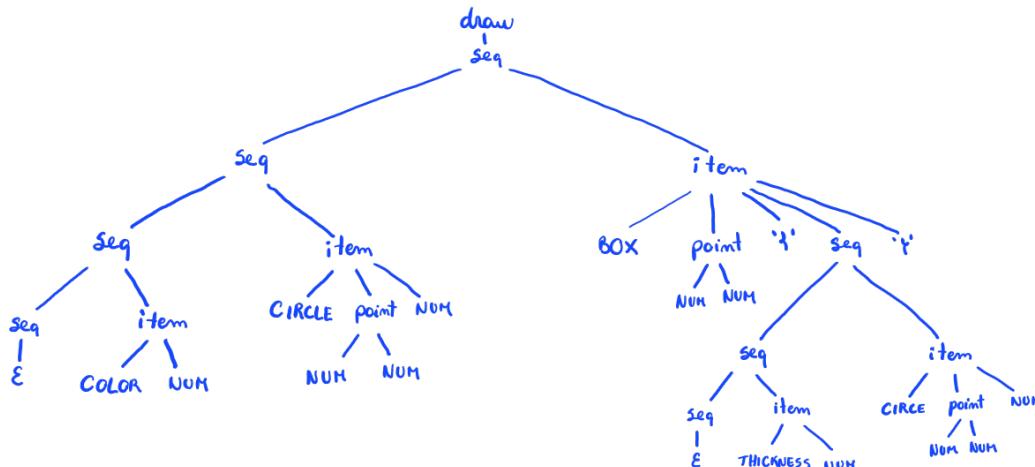


a)

- (a) Trace a árvore de derivação da palavra

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

Se quiser, ao traçar a árvore, pode abreviar a designação dos símbolos, usando n, cr, co, t, b, s, i e p em vez de NUM, CIRCLE, COLOR, THICKNESS, BOX, seq, item e point, respectivamente.



$T_2 = \{a, b, c\} \cup \{C, \), *, +\}$ 
$$\begin{aligned} S &\rightarrow E \\ E &\rightarrow E^* \\ | &E^+ E \\ | &EE \\ | &(' E ') \\ | &T \\ T &\rightarrow 'a' \\ | &'b' \\ | &'c' \end{aligned}$$
$$\begin{aligned} S &\rightarrow E \\ E &\rightarrow E^+ + T \\ | &T \\ T &\rightarrow TF \\ | &F \\ F &\rightarrow F^* \\ | &O \\ O &\rightarrow (' E ') \\ | &'a' \\ | &'b' \\ | &'c' \end{aligned}$$

Da  $\oplus$  baixa para  
 $a \oplus$  alta!



ALGORITMO do predict:

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

ALGORITMO do first:

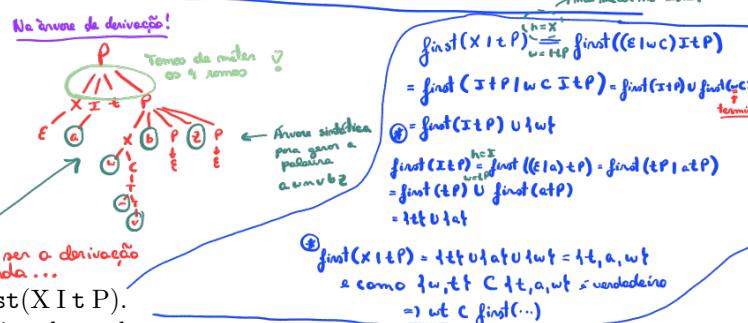
```
first(α) {
    if (α = ε) then
        return {ε}
    h = head (α)      # com |h| = 1
    ω = tail (α)      # tal que α = h ω
    if (h ∈ T) then
        return {h}
    else
        return ∪  
        (h → βi) ∈ P  
        first(βi ω)
}
```

ALGORITMO do follow:

1. \$ ∈ follow(S)
2. if (A → αB ∈ P) then  
follow(B) ⊇ follow(A)
3. if (A → αBβ ∈ P) ∧ (ε ∉ first(β)) then  
follow(B) ⊇ first(β)
4. if (A → αBβ ∈ P) ∧ (ε ∈ first(β)) then  
follow(B) ⊇ ((first(β) - {ε}) ∪ follow(A))

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

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



- [1,5] (a) Mostre que  $\underline{atwnvbz} \in L_1$ . *(Não importa com método, pode ser a derivação à esquerda...)*
- [1,5] (b) Avalie a veracidade da afirmação:  $\{w, t\} \subset \text{first}(XItP)$ . Apresente os passos intermédios e/ou o raciocínio adequados para suportar a sua resposta.
- [1,5] (c) Avalie a veracidade da afirmação:  $t \in \text{follow}(T)$ . Apresente os passos intermédios e/ou o raciocínio adequados para suportar a sua resposta.
- [2,0] (d) Calcule o conjunto  $\text{predict}(P \rightarrow X \ b \ P \ z \ P)$ . *(sem terminais ou regras E g → predict(P → xbPzP) = first(xbPzP) = lb, wt)* 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.

$$\begin{array}{l} t \in \text{first}(xbPzP) \quad P \rightarrow \varepsilon \mid X \ (ItP \mid bPeP) \quad P \rightarrow \varepsilon \mid X \ Y \quad Y \rightarrow ItP \mid bPeP \quad \text{Poros prefixos comuns em ambiguidade!} \\ wt \in \text{first}(XItP) \quad Y \rightarrow ItP \mid bPeP \quad \text{Poros prefixos comuns em ambiguidade!} \end{array}$$

2. Considere o alfabeto  $A = \{a, b, c\}$  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  $T_2 = A \cup \{(), *, +\}$ , em que \* representa o operador de fecho e + o operador de escolha; operação de concatenação tem o operador implícito. Em termos de precedência, 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  $L_2$ . *(Mandar email ao star com a solução!)*

Uma expressão regular → \* tem de ter prioridade ...

continua na página seguinte

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

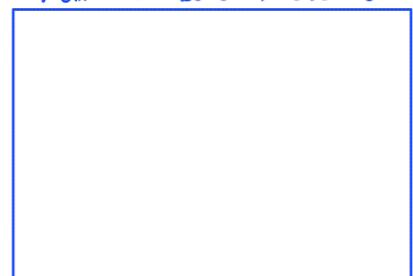
```

draw   →   seq
seq    →   ε
|     seq item
item   →   COLOR NUM
|     THICKNESS NUM
|     CIRCLE point NUM
|     BOX point '{', seq '}'
point  →   NUM NUM
    
```

*Sequência de Círculos!*

*Ajuste da zona ...*

Normalmente ele dá a tabela dos follows!



Considere ainda o conjunto de estados (conjuntos de itens) usado na construção de um reconhecedor ascendente parcialmente apresentada a seguir, onde  $\delta(Z_i, a)$  representa a função de transição de estado.

$$\begin{aligned}
Z_0 &= \{\text{draw} \rightarrow \bullet \text{ seq}, \text{ seq} \rightarrow \bullet, \text{ seq} \rightarrow \bullet \text{ seq item}\} \\
Z_1 &= \delta(Z_0, \text{seq}) = \{\text{draw} \rightarrow \text{seq} \bullet, \text{ seq} \rightarrow \text{seq} \bullet \text{ item}, \boxed{\text{item} \rightarrow \bullet \text{ COLOR NUM}, \text{ item} \rightarrow \bullet \text{ THICKNESS NUM}}, \\
&\quad \boxed{\text{item} \rightarrow \bullet \text{ CIRCLE point NUM}, \text{ item} \rightarrow \bullet \text{ BOX point } \{, \text{ seq } \}}\} \\
Z_2 &= \delta(Z_1, \text{item}) = \{\text{seq} \rightarrow \text{seq item} \bullet\} \\
Z_3 &= \delta(Z_1, \text{COLOR}) = \{\text{item} \rightarrow \text{COLOR} \bullet \text{ NUM}\} \\
Z_4 &= \delta(Z_1, \text{THICKNESS}) = \{\text{item} \rightarrow \text{THICKNESS} \bullet \text{ NUM}\} \\
Z_5 &= \delta(Z_1, \text{CIRCLE}) = \{\dots\} \quad \rightarrow \text{O que falta...} \text{ (para termos linhas independentes)} \\
Z_6 &= \delta(Z_1, \text{BOX}) = \{\dots\} \\
Z_7 &= \delta(Z_3, \text{NUM}) = \{\text{item} \rightarrow \text{COLOR NUM} \bullet\} \\
Z_8 &= \delta(Z_4, \text{NUM}) = \{\text{item} \rightarrow \text{THICKNESS NUM} \bullet\}
\end{aligned}$$

*Operação do fecho!*

- [ 2,0 ] (a) Preencha as linhas da tabela de reconhecimento (*parsing*) para um reconhecedor ascendente relativamente aos estados  $Z_0$  a  $Z_4$ .  $\text{follow}(\text{seq}) = \{\$, \text{c0}, \text{c1}, \text{c2}, \text{b4}\}$

*Podia ser descendente!»*

*Como é que o Z0 evolui?*

	NUM	?	BOX	CIRCLE	THICKNESS	COLOR	{	}	\$	draw	seq	item	point
$Z_0$		R1	R1	R1	R1	R1			R1		$Z_1$		
$Z_1$		Shift <sub>21</sub>	Shift <sub>25</sub>	Shift <sub>24</sub>	Shift <sub>23</sub>							$Z_2$	
$Z_2$													
$Z_3$													
$Z_4$	Shift <sub>1,28</sub>												

*Não termina!*

*Não são preenchidos*

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

$Z_5$  e  $Z_6$  vão evoluir!

(ao surgir novos estados escolheremos 3)

*↳ Os mais fáceis!*

$$Z_5 = \{ \text{item} \rightarrow \text{c1} \cdot \text{point NUM} \} \cup \{ \text{point} \rightarrow \cdot \text{NUM NUM} \}$$

*Não é terminal*

$$Z_6 = \{ \text{item} \rightarrow \text{b} \cdot \text{point } ' \epsilon ' \text{ seq } ' \{ ' \} \cup \{ \text{point} \rightarrow \cdot \text{NUM NUM} \}$$

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 caracterizada 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  $t$ .

*um passo de cada vez !!! (não use 2 símbolos!)*

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

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

Se quiser, ao traçar a árvore, pode abreviar a designação dos símbolos, usando n, cr, co, t, b, s, i e p em vez de NUM, CIRCLE, COLOR, THICKNESS, BOX, seq, item e point, respectivamente.

- [ 3,0 ] (b) Complete a gramática de atributos abaixo tal que ela adequadamente invoque a função `drawCircle` para cada circunferência incluída numa descrição em  $L_3$ .

Production	Semantic rule
$draw \rightarrow seq$	
$seq \rightarrow \epsilon$	
$seq_0 \rightarrow seq_1 item$	
$item \rightarrow COLOR NUM$	
$item \rightarrow THICKNESS NUM$	
$item \rightarrow CIRCLE point NUM$	<code>drawCircle(point.x, point.y, NUM.v, ...)</code>
$item \rightarrow BOX point \{ seq \}$	
$point \rightarrow NUM_1 NUM_2$	