**MINISTÉRIO DA EDUCAÇÃO**

INSTITUTO FEDERAL DO ESPÍRITO SANTO

REITORIA

Avenida Rio Branco, 50 – Santa Lúcia – 29056-255 – Vitória – ES

27 3357-7500

# CONCURSO PÚBLICO
# EDITAL Nº 03 / 2014

## Professor do Magistério do Ensino Básico, Técnico e Tecnológico

| | |
|---|---|
| **ÍNDICE DE INSCRIÇÃO** | 309 |
| **CAMPUS** | Serra |
| **ÁREA/SUBÁREA/ESPECIALIDADE** | Ciência da Computação – Teoria da Computação |

## PROVA DE CONHECIMENTOS ESPECÍFICOS | DISCURSIVA
## MATRIZ DE CORREÇÃO

**QUESTÃO 01**

**Item (a)**

Função OrdenaçãoPorSeleção: $S(n)=S(n-1)+(n-1)$

Função OrdenaçãoPorFusão: $F(n) = 2F\left(\frac{n}{2}\right)+(n-1)$

**Item (b)**

Usando as fórmulas de recorrência:

A Ordenação Por Seleção é uma recorrência de primeira ordem com:

$S(1)=0$

$c=1$

$g(n)=n-1$

$$S(n)=1^{n-1}\cdot(0)+\sum_{i=2}^{n} 1^{n-i}\cdot(i-1)$$

$$S(n)=\frac{(n-1)n}{2}$$

A Ordenação Por Fusão é uma recorrência do tipo dividir para conquistar com:

$S(1)=0$

$c=2$

$g(n)=n-1$

$$S(n)=2^{\log n}\cdot(0)+\sum_{i=1}^{\log n} 2^{(\log n)-i}\cdot(2^i-1)$$

$S(n)=n(\log n)-n+1$

**Item (c)**

As complexidades dos algoritmos OrdenaçãoPorSeleção e OrdenaçãoPorFusão são $O(n^2)$ e $O(n\log n)$, respectivamente. Logo a Ordenação por Fusão é mais eficiente.

**Item (d)**

Obtenção da fórmula da recorrência de primeira ordem.

$$\begin{aligned}S(n)&=cS(n-1)+g(n)\\&=c[cS(n-2)+g(n-1)]+g(n)\\&=c^2S(n-2)+cg(n-1)+g(n)\\&=c^2[cS(n-3)+g(n-2)]+cg(n-1)+g(n)\\&\vdots\\&=c^kS(n-k)+c^{(k-1)}g(n-(k-1))+\cdots+cg(n-1)+g(n)\end{aligned}$$

Para o término da recursão temos $n-k=1$ ou $k=n-1$

$$S(n)=c^{(n-1)}S(1)+c^{(k-2)}g(2)+\cdots+cg(n-1)+g(n)$$

$$S(n)=c^{n-1}\cdot S(1)+\sum_{i=2}^{n}c^{n-i}\cdot g(i)$$

Prova por indução:

Caso base: $S(2)=cS(1)+g(2)$

Suponha que: $S(n)=c^{n-1}\cdot S(1)+\sum_{i=2}^{n}c^{n-i}\cdot g(i)$

Mostrar que: $S(n+1)=c^{(n+1)-1}\cdot S(1)+\sum_{i=2}^{n+1}c^{(n+1)-i}\cdot g(i)$

$$\begin{aligned}S(n+1)&=cS(n)+g(n+1)\\&=c\left[c^{n-1}\cdot S(1)+\sum_{i=2}^{n}c^{n-i}\cdot g(i)\right]+g(n+1)\\&=c^{(n+1)-1}\cdot S(1)+\sum_{i=2}^{n+1}c^{(n+1)-i}\cdot g(i)\end{aligned}$$

## QUESTÃO 02

**Item (a)**

| | |
|---|---|
| 1. $q\rightarrow(p\rightarrow r)$ | hip |
| 2. $\neg r$ | hip |
| 3. $q$ | hip |
| 4. $p\rightarrow r$ | 1, 3, $\rightarrow e$ |
| 5. $\neg p$ | 4, 2, MT |

*Q.E.D.*

**Item (b)**

| | |
|---|---|
| 1. $p\rightarrow r$ | hip |
| 2. $q\rightarrow r$ | hip |
| 3. $p\wedge q$ | sup |
| 4. $p$ | 3, $\wedge e_1$ |
| 5. $r$ | 1, 4, $\rightarrow e$ |

## QUESTÃO 02

| | |
|---|---|
| 6. $p \wedge q \rightarrow r$ | 3..5, $\rightarrow i$ |

*Q.E.D.*

**Item (c)**

| | |
|---|---|
| 1. $p \rightarrow (q \rightarrow r)$ | hip |
| 2. $p \rightarrow q$ | hip |
| 3. $p$ | sup |
| 4. $q \rightarrow r$ | 1, 3, $\rightarrow e$ |
| 5. $q$ | 2, 3, $\rightarrow e$ |
| 6. $r$ | 4, 5, $\rightarrow e$ |
| 7. $p \rightarrow r$ | 3..6, $\rightarrow i$ |

*Q.E.D.*

**Item (d)**

| | |
|---|---|
| 1. $p \rightarrow q$ | sup |
| 2. $r \rightarrow s$ | sup |
| 3. $p \wedge r$ | sup |
| 4. $p$ | 3, $\wedge e_1$ |
| 5. $q$ | 1, 4, $\rightarrow e$ |
| 6. $r$ | 3, $\wedge e_2$ |
| 7. $s$ | 2, 4, $\rightarrow e$ |
| 8. $q \wedge s$ | 5, 7, $\wedge i$ |
| 9. $p \wedge r \rightarrow q \wedge s$ | 3..8, $\rightarrow i$ |
| 10. $(r \rightarrow s) \rightarrow (p \wedge r \rightarrow q \wedge s)$ | 2..9, $\rightarrow i$ |
| 11. $(p \rightarrow q) \rightarrow ((r \rightarrow s) \rightarrow (p \wedge r \rightarrow q \wedge s))$ | 1..10, $\rightarrow i$ |

*Q.E.D.*

## QUESTÃO 03

**Item (a)**

Seja $V = \{\langle sttmt \rangle, \langle sttmt-seq \rangle, \langle if-sttmt \rangle, \langle while-sttmt \rangle, \langle a-expr \rangle, \langle factor \rangle, \langle b-expr \rangle, \langle term \rangle\}$ o conjunto de símbolos não-terminais da gramática dada, e seja $A = \langle Q, \Sigma, \Gamma, \delta, q_0, \oslash, F \rangle$ o autômato pedido, onde:

$Q = \{q_0, q_1, q_2\}$ é o conjunto de estados;

$\Sigma = \{\text{skip}, :=, ;, \text{if}, \text{then}, \text{else}, \text{while}, \text{do}, +, -, (, ), \text{and}, \text{not}, \leq, \text{INT}, \text{BOOL}, \text{IDENT}\}$ é o conjunto de símbolos terminais;

$\Gamma = V \cup \Sigma \cup \{\oslash\}$ é o conjunto de símbolos não terminais;

$q_0$ é o estado inicial;

$\oslash$ é o símbolo de pilha vazia;

$F = \{q_2\}$ é o conjunto de estados finais;

A função de transição $\delta$ é dada por:

$\delta(q_0, \epsilon, \oslash) = \{(q_1, \langle sttmt \rangle \oslash)\}$

$\delta(q_1, \varepsilon, \langle sttmt \rangle) = \{(q_1, \text{skip}), (q_1, \text{IDENT} := \langle \text{a-expr} \rangle), (q_1, \langle \text{sttmt-seq} \rangle), (q_1, \langle \text{if-sttmt} \rangle), (q_1, \langle \text{while-sttmt} \rangle)\}$

$\delta(q_1, \varepsilon, \langle \text{sttmt-seq} \rangle) = \{(q_1, \langle \text{sttmt} \rangle), (q_1, \langle \text{sttmt} \rangle ; \langle \text{sttmt-seq} \rangle)\}$

$\delta(q_1, \varepsilon, \langle \text{if-sttmt} \rangle) = \{(q_1, \text{if} \langle \text{b-expr} \rangle \text{then} \langle \text{sttmt} \rangle \text{else} \langle \text{sttmt} \rangle)\}$
$\delta(q_1, \varepsilon, \langle \text{while-sttmt} \rangle) = \{(q_1, \text{while} \langle \text{b-expr} \rangle \text{do} \langle \text{sttmt} \rangle)\}$
$\delta(q_1, \varepsilon, \langle \text{a-expr} \rangle) = \{(q_1, \langle \text{factor} \rangle), (q_1, \langle \text{factor} \rangle + \langle \text{factor} \rangle), (q_1, \langle \text{factor} \rangle - \langle \text{factor} \rangle)\}$
$\delta(q_1, \varepsilon, \langle \text{factor} \rangle) = \{(q_1, \text{INT}), (q_1, \text{IDENT}), (q_1, -\langle \text{factor} \rangle), (q_1, (\langle \text{a-expr} \rangle))\}$
$\delta(q_1, \varepsilon, \langle \text{a-expr} \rangle) = \{(q_1, \langle \text{term} \rangle), (q_1, \langle \text{term} \rangle \text{and} \langle \text{term} \rangle)\}$
$\delta(q_1, \varepsilon, \langle \text{factor} \rangle) = \{(q_1, \text{BOOL}), (q_1, \text{IDENT}), (q_1, \text{not} \langle \text{term} \rangle), (q_1, (\langle \text{b-expr} \rangle)), (q_1, \langle \text{a-expr} \rangle \leq \langle \text{a-expr} \rangle)\}$
$\delta(q_1, \text{skip}, \text{skip}) = \{(q_1, \varepsilon)\}$
$\delta(q_1, :=, :=) = \{(q_1, \varepsilon)\}$
$\delta(q_1, ;, ;) = \{(q_1, \varepsilon)\}$
$\delta(q_1, \text{if}, \text{if}) = \{(q_1, \varepsilon)\}$
$\delta(q_1, \text{then}, \text{then}) = \{(q_1, \varepsilon)\}$
$\delta(q_1, \text{else}, \text{else}) = \{(q_1, \varepsilon)\}$
$\delta(q_1, \text{while}, \text{while}) = \{(q_1, \varepsilon)\}$
$\delta(q_1, \text{do}, \text{do}) = \{(q_1, \varepsilon)\}$
$\delta(q_1, +, +) = \{(q_1, \varepsilon)\}$
$\delta(q_1, -, -) = \{(q_1, \varepsilon)\}$
$\delta(q_1, (, () = \{(q_1, \varepsilon)\}$
$\delta(q_1, ), )) = \{(q_1, \varepsilon)\}$
$\delta(q_1, \text{and}, \text{and}) = \{(q_1, \varepsilon)\}$
$\delta(q_1, \text{not}, \text{not}) = \{(q_1, \varepsilon)\}$
$\delta(q_1, \leq, \leq) = \{(q_1, \varepsilon)\}$
$\delta(q_1, \text{INT}, \text{INT}) = \{(q_1, \varepsilon)\}$
$\delta(q_1, \text{BOOL}, \text{BOOL}) = \{(q_1, \varepsilon)\}$
$\delta(q_1, \text{IDENT}, \text{IDENT}) = \{(q_1, \varepsilon)\}$
$\delta(q_1, \varepsilon, \varnothing) = \delta(q_2, \varnothing)$

**Item (b)**

O autômato determinístico apresentado tem função de transição total, portanto não é necessário acrescetar um estado de *deadlock*.

A tabela abaixo relacionas os estados do autômato em pares na forma (linha, coluna) – as células em cinza escuro não são utilizadas. As células correspondentes aos pares de estados trivialmente não equivalentes estão representadas por um X.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $q_1$ | | | | | | | |
| $q_2$ | X | X | | | | | |
| $q_3$ | | | X | | | | |
| $q_4$ | | | X | | | | |
| $q_5$ | | | X | | | | |
| $q_6$ | | | X | | | | |
| $q_7$ | X | X | | X | X | X | X |
| | $q_0$ | $q_1$ | $q_2$ | $q_3$ | $q_4$ | $q_5$ | $q_6$ |

Analisando os demais pares de estados, temos:

$\delta(q_0, a) = q_1$, $\delta(q_0, b) = q_4$, $\delta(q_1, a) = q_5$, $\delta(q_1, b) = q_2$, não equivalentes.

$\delta(q_0, a) = q_1$, $\delta(q_0, b) = q_4$, $\delta(q_3, a) = q_3$, $\delta(q_3, b) = q_3$, indefinido.

$\delta(q_0, a) = q_1$, $\delta(q_0, b) = q_4$, $\delta(q_4, a) = q_1$, $\delta(q_b, b) = q_4$, equivalentes.

$\delta(q_0, a) = q_1$, $\delta(q_0, b) = q_4$, $\delta(q_5, a) = q_1$, $\delta(q_5, b) = q_4$, equivalentes.

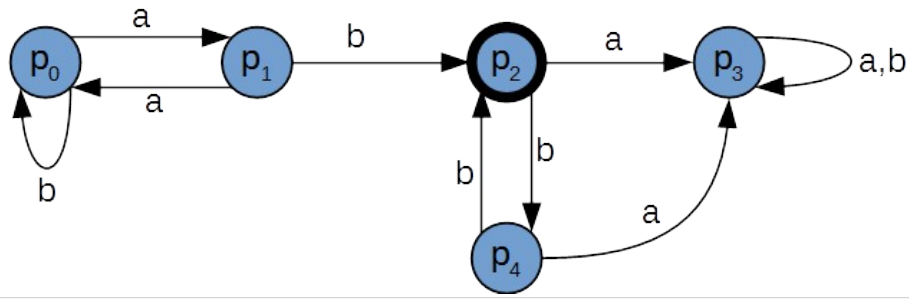
| | |
|---|---|
| $\delta(q_0,a)=q_1$, $\delta(q_0,b)=q_4$<br>$\delta(q_6,a)=q_3$, $\delta(q_6,b)=q_7$ , não equivalentes. | $\delta(q_1,a)=q_5$, $\delta(q_1,b)=q_2$<br>$\delta(q_3,a)=q_3$, $\delta(q_3,b)=q_3$ , não equivalentes. |
| $\delta(q_1,a)=q_5$, $\delta(q_1,b)=q_2$<br>$\delta(q_4,a)=q_1$, $\delta(q_4,b)=q_4$ , não equivalentes. | $\delta(q_1,a)=q_5$, $\delta(q_1,b)=q_2$<br>$\delta(q_5,a)=q_1$, $\delta(q_5,b)=q_4$ , não equivalentes. |
| $\delta(q_1,a)=q_5$, $\delta(q_1,b)=q_2$<br>$\delta(q_6,a)=q_3$, $\delta(q_6,b)=q_7$ , indefinido. | $\delta(q_2,a)=q_3$, $\delta(q_2,b)=q_6$<br>$\delta(q_7,a)=q_3$, $\delta(q_7,b)=q_6$ , equivalentes. |
| $\delta(q_3,a)=q_3$<br>$\delta(q_4,a)=q_1$ , não equivalentes. | $\delta(q_3,a)=q_3$<br>$\delta(q_5,a)=q_1$ , não equivalentes. |
| $\delta(q_3,a)=q_3$, $\delta(q_3,b)=q_3$<br>$\delta(q_6,a)=q_3$, $\delta(q_6,b)=q_7$ , não equivalentes. | $\delta(q_4,a)=q_1$, $\delta(q_4,b)=q_4$<br>$\delta(q_5,a)=q_1$, $\delta(q_5,b)=q_4$ , equivalentes. |
| $\delta(q_4,a)=q_1$<br>$\delta(q_6,a)=q_3$ , não equivalentes. | $\delta(q_5,a)=q_1$<br>$\delta(q_6,a)=q_3$ , não equivalentes. |

A tabela abaixo mostra a situação das equivalências entre os pares de estados após a análise par-a-par. Os pares de estados não equivalentes aparecem marcados por um '#'. Abaixo da tabela são mostradas as listas de dependências construídas durante a análise.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $q_1$ | # | | | | | | |
| $q_2$ | X | X | | | | | |
| $q_3$ | # | # | X | | | | |
| $q_4$ | | # | X | # | | | |
| $q_5$ | | # | X | # | | | |
| $q_6$ | # | # | X | # | # | # | |
| $q_7$ | X | X | | X | X | X | X |
| | $q_0$ | $q_1$ | $q_2$ | $q_3$ | $q_4$ | $q_5$ | $q_6$ |

$(q_1,q_5) = [(q_0,q_1),(q_1,q_4)]X$
$(q_1,q_3) = [(q_0,q_3),(q_0,q_6)]X$
$(q_2,q_7) = [(q_1,q_6)]$
$(q_3,q_4) = [(q_0,q_3)]X$
$(q_3,q_5) = [(q_1,q_3),(q_1,q_6)]X$

Analisando a tabela vemos que foram formadas as seguintes equivalência: $q_0$–$q_4$–$q_5$ e $q_2$–$q_7$. Assim o autômato mínimo formado é mostrado no diagrama abaixo:




Onde os estados do autômato mínimo representam conjuntos de estados dos autômato original segundo o seguinte mapeamento: $p_0=\{q_0,q_4,q_5\}$ , $p_1=\{q_1\}$ , $p_2=\{q_2,q_7\}$ , $p_3=\{q_3\}$ e $p_4=\{q_6\}$ .

## QUESTÃO 04

**Item (a.1)**

Uma gramática formal $G$ é definida como $G=\langle V,\Sigma,P,S\rangle$ , onde:

$V$ é um conjunto finito de símbolos não terminais;

$\Sigma$ é um conjunto finito de símbolos terminais;

$P$ é um conjunto de regras de produção, tal que $P=\{\alpha \rightarrow \beta : \alpha, \beta \in (V \cup \Sigma)^\star\}$ ; e a notação $A^\star$ designa o conjunto de todas as cadeias formadas pela justaposição de símbolos do conjunto $A$ .

$S \in V$ é o símbolo inicial da gramática.

**Item (a.2)**

A Hierarquia de Chomsky é uma classificação para gramáticas formais em quatro (4) níveis proposta por Noam Chomsky. O quatro níveis propostos por Chomsky são:

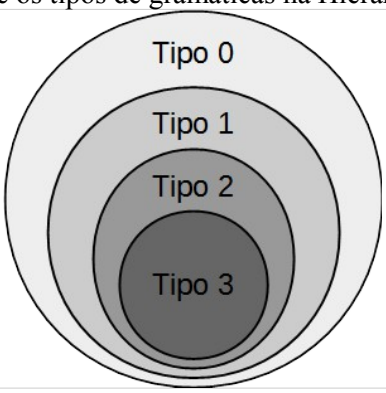
Tipo 0 – Gramáticas irrestritas

Tipo 1 – Gramáticas sensíveis ao contexto

Tipo 2 – Gramáticas livres de contexto

Tipo 3 – Gramáticas regulares

O diagrama abaixo ilustra a relação entre os tipos de gramáticas na Hierarquia de Chomsky:




**Item (a.3)**

| **Tipo #** | **Autômato/Máquina** |
|---|---|
| Tipo 3 | Autômato de Estados Finito |
| Tipo 2 | Autômato com Pilha |
| Tipo 1 | Autômato Linearmente Limitado |
| Tipo 0 | Máquina de Turing |

**Item (b)**

Uma máquina de Turing é definida como uma tupla $M=\langle Q, \Sigma, \Gamma, s, b, F, \delta \rangle$ , onde:

- $Q$ é um conjunto finito de estados;
- $\Sigma$ é um conjunto fintio de símbolos;
- $\Gamma$ é o alfabeto da fita (conjunto finito de símbolos);
- $s \in Q$ é o estado inicial da máquina;
- $b \in \Gamma$ é o símbolo “em branco”/vazio;
- $F \in Q$ é o conjunto de estados finais;
- $\delta : Q \times \Gamma \Rightarrow Q \times \Gamma \times \{L, R\}$ é a função (parcial) de transição;

A função de transição é definida como um conjunto de quíntuplas da forma $(s,i,i',s',d)$ onde $s, s' \in Q$ ; $i, i' \in \Gamma$ e $d \in \{L, R\}$ tal que não há mais de uma quíntupla começando pelos mesmos símbolos $s$ e $i$.

**Item (c)**

Seja $x = |z(x)|$ com $z(x) \in \{1\}^\star$ , ou seja, o número será representado pela quantidade de dígitos 1.

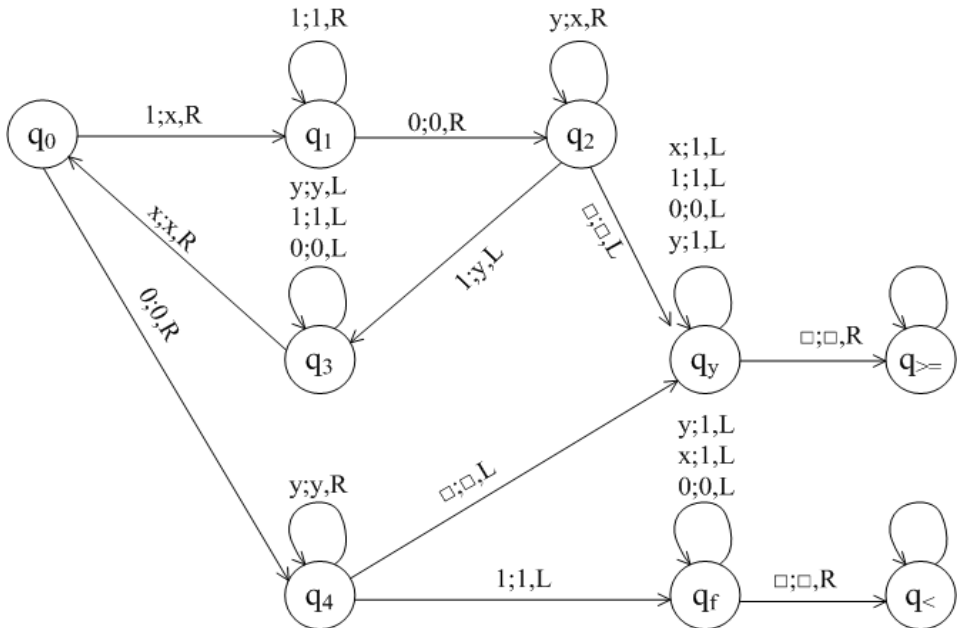
A Máquina de turing deverá:

- Chegar ao estado $q_{>=}$ se $x \geq y$ , ou seja, $q_0$ w = $q_0$z(x) 0 z(y) $\vdash^*$ $q_{>=}$ z(x) 0 z(y)
- Chegar ao estado $q_<$ se $x < y$ , ou seja, $q_0$ w = $q_0$z(x) 0 z(y) $\vdash^*$ $q_<$ z(x) 0 z(y)

Assim, $M=\langle \{q_0, q_1, q_2, q_3, q_4, q_f, q_v, q_{\geq}, q_<\}, \{1,0\}, \{1,0,x,y,\square\}, \square, q_0, \{q_{\geq}, q_<\} \rangle$ com:

$\delta(q_0,1) = (q_1,x,R)$,
$\delta(q_0,0) = (q_4,0,R)$,
$\delta(q_1,1) = (q_1,1,R)$,

$\delta(q_1,0) = (q_2,0,R)$,
$\delta(q_2,1) = (q_3,y,L)$,
$\delta(q_2,y) = (q_2,y,R)$,
$\delta(q_2,\square) = (q_v,\square,L)$,
$\delta(q_3,y) = (q_3,y,L)$,
$\delta(q_3,0) = (q_3,0,L)$,
$\delta(q_3,1) = (q_3,1,L)$,
$\delta(q_3,x) = (q_0,x,R)$,
$\delta(q_4,y) = (q_4,y,R)$,
$\delta(q_4,1) = (q_f,1,L)$,
$\delta(q_4,\square) = (q_v,\square,L)$,
$\delta(q_f,0) = (q_f,0,L)$,
$\delta(q_f,y) = (q_y,1,L)$,
$\delta(q_f,x) = (q_f,1,L)$,
$\delta(q_f,\square) = (q_<,\square,R)$,
$\delta(q_v,y) = (q_v,1,L)$,
$\delta(q_v,x) = (q_v,1,L)$,
$\delta(q_v,0) = (q_v,0,L)$,
$\delta(q_v,1) = (q_v,1,L)$,
$\delta(q_v,\square) = (q_{>=},\square,R)$.




## QUESTÃO 05

**Item (a.i)**

Verificação de modelos é uma tecnica automática para verificação de sistemas concorrentes de estados finitos, e tipicamente se refere ao seguinte problema: dado um modelo de um sistema, verificar de modo exaustivo e automático se se o modelo está de acordo com uma dada especificação.

**Item (a.ii)**

Enquanto os modelos dos sistemas são, geralmente, representados de modo abstrato como estruturas de Kripke, as propriedades que se deseja verificar para esses modelos são expressas na forma de fórmulas de alguma lógica temporal.

**Item (a.iii)**

A técnica de *symbolic model checking* utiliza fórmulas booleanas para representar conjuntos de estados e as relações de transição de estados, comumente utilizando estruturas de dados tais como *binary decision diagrams* (BDD) para atingir uma representação mais compacta das fórmulas.

Historicamente, symbolic model checking esteve associado ao uso de BDDs, entretanto é possível desenrolar, ou expandir, as fórmulas em lógica temporal correspondentes às propriedades que se deseja verificar, desde que se limite a expansão a um certo número limitado de transições, gerando-se uma representação booleana de todo o problema de verificação "limitada" de modelos. Essa representação booleana pode então ser entregue a um SAT-solver para que este procure uma solução para o problema de verificação do modelo sem a necessidade de se empregar os algoritmos clássicos de verificação de modelos ou de se manipular BDDs. Esta tecnica é denominada *bounded model checking*.

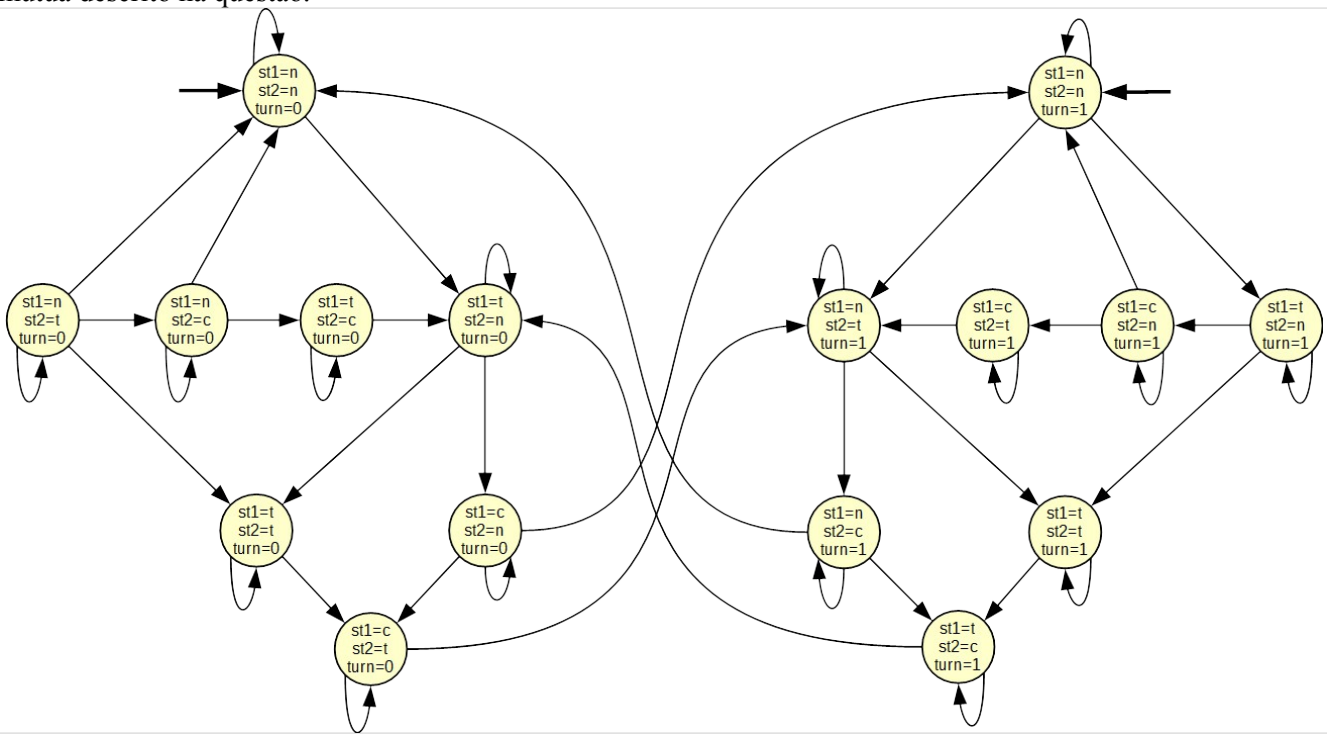
Assim como o *symbolic model checking*, o *bounded model checking* também representa conjuntos de estados e a relação de transição de estados por fórmulas booleanas. Diferentemente do *symbolic model checking*, o *bounded model checking,* em princípio, não é completo, ou seja, é possível que existam contra-exemplos da propriedade sendo verificada que esteja além do limite estabelecido para a verificação; neste caso tais contra-exemplos não serão detectados pelo processo de verificação. Entretanto o bounded model checking pode ser tornado completo desde que o limite para a verificação seja igual ou superior a um certo limite chamado "*completeness threshould*", que é dependente do modelo e da propriedade sendo verificada.

**Item (a.iv)**

Uma bissimulação é uma relação binária entre sistemas de transição de estados, associando sistemas que se comportam da mesma maneira, no sentido de que um simula o outro e vice-versa. Intuitivamente, dois sistemas são bissimilares se há uma correspondências entre os movimento de um e do outro. Neste sentido os sistemas não podem ser distinguidos um do outro por um observador externo.

**Item (b.i)**

O diagrama abaixo ilustra o sistema de transição de estados que descreve a solução para o problema de exclusão mútua descrito na questão.




Em cada estado do sistema ilustrado acima, as variáveis "st1" e "st2" representam a situação dos processos 1 e 2 respectivamente; os valores de "st1" e "st2" podem ser "n" (normal), "t" (tentando entrar na região crítica) ou "c" (na região crítica). A variável "turn" controla qual processo tem a vez na região crítica.

**Item (b.ii.1)**

Em LTL: $G\neg(\text{st1}=c\wedge\text{st2}=c)$

**Item (b.ii.2)**

Em CTL: $AG\big((\text{st1}=n\rightarrow EX\,\text{st1}=t)\wedge(\text{st2}=n\rightarrow EX\,\text{st2}=t)\big)$

**Item (b.ii.3)**

Em CTL: $AG\big((\text{st1}=t\rightarrow AF\,\text{st1}=c)\wedge(\text{st2}=t\rightarrow AF\,\text{st2}=c)\big)$

Assinatura Presidente | Assinatura Membro | Assinatura Membro