FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
Mestrado Integrado em Engenharia Informática e Computação

Teoria da Computação

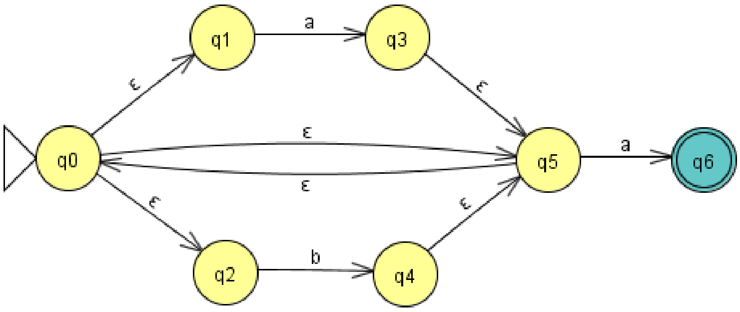
Exame de Época Normal, 12 de janeiro de 2012

DURAÇÃO MÁXIMA: 2 horas e 30 minutos

É apresentada aqui uma possível resolução parcial do exame.

Problema 1: Autómatos Finitos e Expressões Regulares (5 valores)

Considere o autómato finito seguinte:



1.a) Indique o fecho-ε para cada estado do autómato.

R:

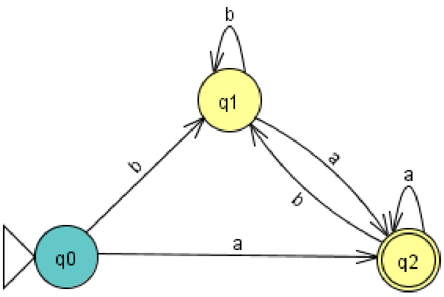
Fecho-ε(q_0) = {q_0, q_1, q_2, q_5}
Fecho-ε(q_1) = {q_1}
Fecho-ε(q_2) = {q_2}
Fecho-ε(q_3) = {q_3, q_5, q_0, q_1, q_2}
Fecho-ε(q_4) = {q_4, q_5, q_0, q_1, q_2}
Fecho-ε(q_5) = {q_5, q_0, q_1, q_2}
Fecho-ε(q_6) = {q_6}

1.b) Converta-o para um DFA e desenhe o DFA completo resultante.

R:

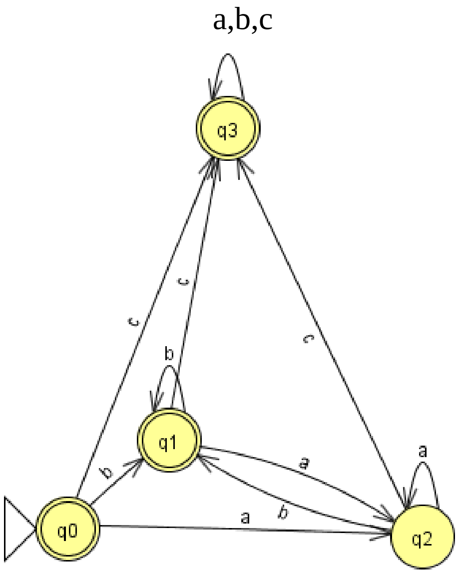
Usar o método dos subconjuntos para fazer a conversão...

DFA resultante:



1.c) Apresente o diagrama do DFA que represente o complemento da linguagem representada pelo autómato obtido em 1b), mas tendo em conta o alfabeto Σ={a,b,c}.

R:

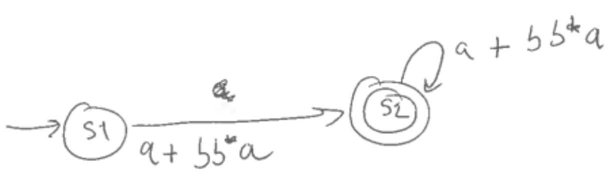


1.d) Converta o DFA resultante de 1b) para uma expressão regular usando o método de eliminação de estados. Apresente os passos efetuados para a conversão.

R:

No diagrama em baixo S1 corresponde a q0 de 1b), S2 a q2, e S3 a q1:

Eliminamos S3:



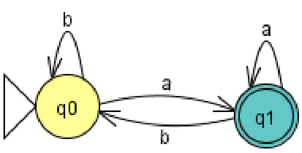
$RE = (a + bb^*a)(a + bb^*a)^*$.

1.e) Apresente o DFA resultante após a aplicação do método de minimização de estados ao DFA de 1b). Apresente a tabela de estados distinguíveis.

R:

Apresentar tabela de estados distinguíveis...

DFA resultante:



Problema 2: **Linguagens (3 valores)**

Mostre, usando o lema da bombagem para as linguagens regulares, que a linguagem $L=\{a^{n!} \mid n \geq 0\}$ não é uma linguagem regular.

R: *Possível resolução:*

Vamos assumir que L é regular.

Escolhemos $w = a^{n!}$*, que pertence a L e* $|w| \geq n$

Podemos decompor w em 3 partes x,y,z tal que w=xyz, com $y \neq \varepsilon$

Assim $1 \leq |y| \leq n!$[1]

Para k = 2 temos: xyyz que terá de pertencer a L segundo o lema.

Como $1 \leq |y| \leq n!$ *então* $n!-1+2\times 1 \leq |xyyz| \leq 2n!$

I.e., $n!+1 \leq |xyyz| \leq 2n!$

Como $|xyz|=n!$*, a palavra de comprimento logo a seguir será* $|xyyz|$ *e terá de corresponder a um dos próximos factoriais. O factorial logo a seguir a n! é* $(n+1)!=(n+1)n!$

que é maior do que 2n! para $n \geq 2$ *e por isso xyyz não pertence a L qualquer que seja a decomposição w=xyz, com* $y \neq \varepsilon$*.*

Temos assim uma contradição e por isso L é não regular.

Problema 3: **Gramáticas e Autómatos de Pilha (5 valores)**

Considere a linguagem $L = \{a^i b^i c^j \mid i, j \geq 0\}$

3.a) Apresente uma CFG (gramática sem contexto) para esta linguagem.

R:

CFG $G_1 = (\overset{V}{\{S, X, Y\}}, \overset{\Sigma}{\{a, b, c\}}, P, S)$

Produções P:

$S \rightarrow XY$

$X \rightarrow aXb \mid \varepsilon$

$Y \rightarrow cY \mid \varepsilon$

3.b) A gramática apresentada é ambígua? Justifique a resposta.

R:

A gramática não é ambígua pois não se consegue arranjar uma palavra que implique duas ou mais árvores de sintaxe.

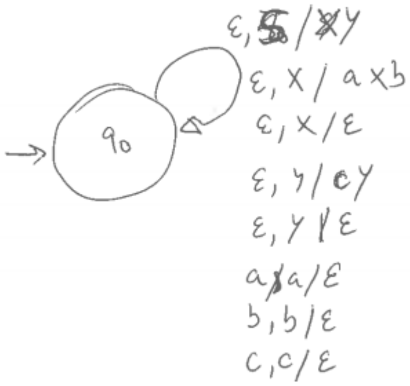
---

[1] *Notem que neste caso não estamos a considerar a restrição:* $|xy| \leq n$*. Poderíamos realizar a prova tendo em conta essa restrição. Como?*

3.c) Apresente um PDA para a linguagem L.

R:

$$\text{PDA: } P = (\underbrace{\{q_0\}}_{Q}, \underbrace{\{a,b,c\}}_{\Sigma}, \underbrace{\{a,b,c,S,X,Y\}}_{\Gamma}, \delta, q_0, S, \{\})$$



3.d) O PDA é determinista ou não determinista? Justifique a resposta dada.

R:

PDA NÃO DETERMINISTA. PARA A MESMA ENTRADA E MESMO SÍMBOLO NO TOPO DA PILHA EXISTEM DUAS OU MAIS TRANS.

Ex. $\delta(q_0, \varepsilon, X) = \{(q_0, aXb), (q_0, \varepsilon)\}$

3.e) Apresente uma sequência de descrições instantâneas que conduz à aceitação quando o PDA obtido processa a *string* **abcc**.

R:

$(q_0, abcc, S) \vdash (q_0, abcc, XY) \vdash (q_0, abcc, aXbY) \vdash (q_0, bcc, XbY) \vdash (q_0, bcc, bY) \vdash (q_0, cc, Y) \vdash (q_0, cc, cY) \vdash (q_0, c, Y) \vdash (q_0, c, cY) \vdash (q_0, \varepsilon, Y) \vdash (q_0, \varepsilon, \varepsilon)$

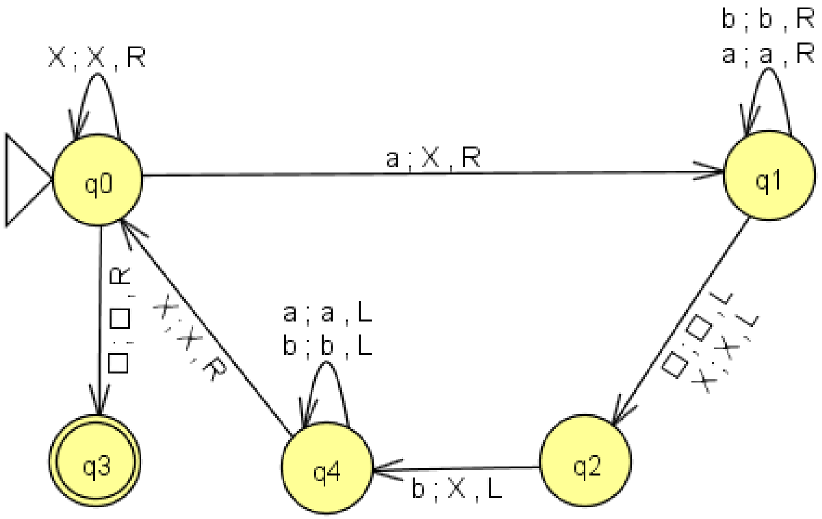
Problema 4: Máquina de Turing (4 valores)

4.a) Desenhe o diagrama de transições de estado de uma Máquina de Turing que aceite as palavras da linguagem $L = \{a^n b^n \mid n \geq 0\}$.

R:

Possível Máquina de Turing:

Máquina de Turing (TM) M= ({q0,q1,q2,q3, q4}, {a,c}, {a,c,X,□}, δ, q_0, □, {q3})



□ representa o símbolo branco da fita (célula sem nada).

R representa movimento para a direita, e

L representa movimento para a esquerda

4.b) Apresente o traço de computação da Máquina de Turing quando a entrada na fita é **ab**.

R:

$q_0ab \vdash Xq_1b \vdash Xbq_1 \vdash Xq_2b \vdash q_4XX \vdash Xq_0X \vdash XXq_0□ \vdash XX□q_3□$

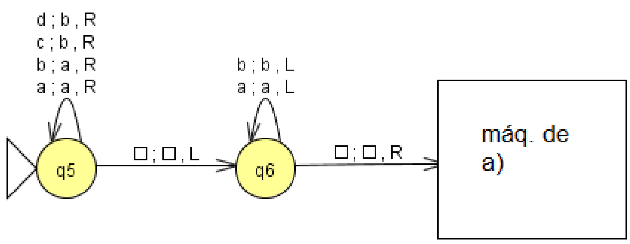
4.c) Apresente o traço de computação da Máquina de Turing quando a entrada na fita é **aab**.

R:

$q_0aab \vdash Xq_1ab \vdash Xaq_1b \vdash Xabq_1 \vdash Xaq_2b \vdash Xq_4aX \vdash q_4XaX \vdash Xq_0aX \vdash XXq_1X \vdash X q_2XX$

4.d) Desenhe uma Máquina de Turing que aceite as palavras da linguagem L ={x^ny^n | n ≥ 0, x∈{a,b}, y∈{c,d}} e que use a Máquina de Turing de a).

R:



4.e) É possível converter qualquer DFA numa máquina de Turing? Em caso afirmativo, explique os passos necessários para transformar DFA's em máquinas de Turing.

R:

Sim. Uma possibilidade é partirmos do autómato que representa o DFA completo e modificarmos as transições do autómato de $\delta(q,x) = p$ para $\delta(q,x) = (p,x,R)$, com $x \in \Sigma$, $q \in Q$ e $p \in Q$. Os estados finais do DFA passam a ser estados finais da Máquina de Turing, o conjunto de símbolos na entrada Σ é o mesmo, e o conjunto de símbolos na fita $\Gamma = \Sigma \cup \{B\}$.

Problema 5: Afirmações sobre Linguagens (3 valores)

Para cada uma das afirmações seguintes, diga se é verdadeira ou falsa e dê uma justificação sucinta para a resposta dada.

5.a) Um PDA não-determinista não consegue implementar mais linguagens do que um PDA determinista.

F. Existem CFL's que podem ser implementadas por um PDA não determinista mas que não podem ser implementadas por um PDA determinista. Um exemplo é a linguagem $L=\{ww^R \mid w \in \{0,1\}\}$. O não determinismo é necessário pois não é possível saber com apenas um caminho no traço de computação quando é que o símbolo seguinte na entrada é o símbolo de início de w^R.

5.b) A concatenação de uma linguagem regular com uma linguagem não regular pode produzir uma linguagem regular.

V. Por exemplo quando a linguagem regular é $\emptyset$ a concatenação com outra linguagem dá $\emptyset$.

5.c) A união de uma linguagem regular com uma linguagem não regular produz sempre uma linguagem não regular.

*F. Exemplo: a união da linguagem regular dada por a*b* com a linguagem não regular $L=\{a^nb^n \mid n\geq 0\}$ origina a linguagem a*b* que é regular.*

5.d) Dado um DFA é possível programar um algoritmo que indique se esse DFA representa todas as palavras formadas no alfabeto utilizado.

V. Sendo n o número de estados do DFA, basta verificar se ele aceita todas as palavras com comprimento de 0 a 2n-1.

Outra opção: minimizar o DFA e verificar se tem apenas um estado (que deve ser de aceitação) com transição para ele próprio a envolver todos os símbolos do alfabeto.

5.e) O PDA resultante da implementação de uma CFG ambígua pelo método apresentado em TCOM resulta sempre num PDA não determinista.

V. Se a CFG é ambígua então existe pelo menos uma palavra da linguagem com pelo menos duas derivações. Como o método de conversão de CFG para PDA "emula" as derivações então o PDA resultante é forçosamente não determinista, pois vai haver casos em que para o mesmo símbolo na entrada e o mesmo símbolo no topo da pilha (uma variável da gramática) pode haver mais do que uma transição (que originam as diferentes derivações).

5.f) Se L é uma linguagem não regular então o complemento de L também é não regular.

V. O complemento de uma linguagem não regular é sempre uma linguagem regular.
Vamos supor que existe uma linguagem não regular L tal que o complemento de L, L^C, é regular e por isso pode ser representada por uma expressão regular R. Considere a linguagem dada por L^C(R). Esta linguagem terá de ser regular devido à propriedade de fecho das linguagens regulares em relação ao complemento. Mas nós sabemos que o complemento do complemento de uma linguagem é a própria linguagem e por isso temos uma contradição.

(Fim.)