Universidade Federal do Pará
Instituto Tecnológico
Faculdade de Engenharia de Computação
Teoria da Computação

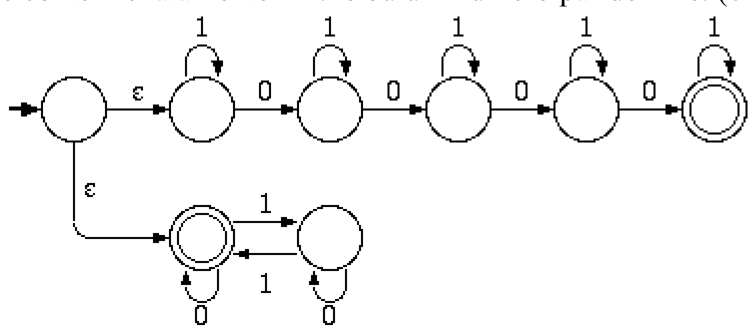
1ª Avaliação – Data: 07/05/2012

Aluno: Matrícula:

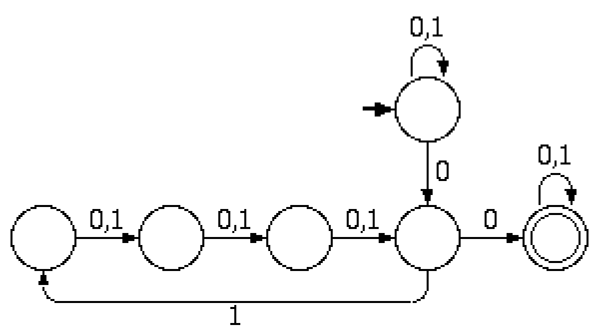
1. Desligue o celular ou coloque-o no modo silencioso. Não é permitido o seu uso durante a prova.
2. Só serão aceitas as questões justificadas.
3. Seja organizado.

1. (Valor 1,5 ponto) Desenhe AFND com o número de estados especificado que aceitam as seguintes linguagens:

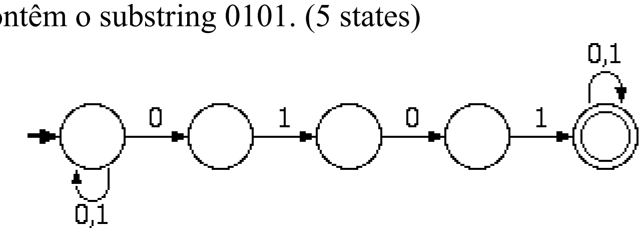
a) Todos os strings que contem exatamente 4 "0"s ou um número par de "1"s. (8 states)



b) Todos os strings em que ocorrem dois zeros separados por um string de comprimento 4i para algum i >= 0. (6 states)



c) Todos os strings que contêm o substring 0101. (5 states)

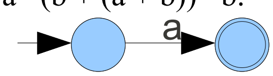


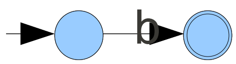
2. (Valor 0,5 ponto) Numere a primeira coluna de acordo com a segunda:

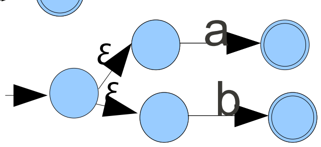
(3) GLD 1. G =({S, A}, {a, b}, {S → Aaa | Abb, A → Aa | Ab | ε}, S).
(1) GLE 2. G =({S, A}, {a, b}, {S → Aa | a, A → Sb}, S).
(4) GLUD 3. G =({S, A}, {a, b}, {S → aS | bS | A, A → aa | bb}, S).
(2)GLUE 4. G =({S, A, B}, {a, b}, {S → aA, A → bB | ε, B → aA}, S).

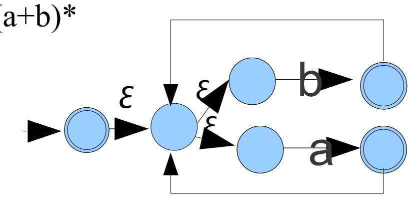
3. (Valor 2,0 pontos) Construa os diagramas dos AFε que reconhecem as linguagens denotadas por:

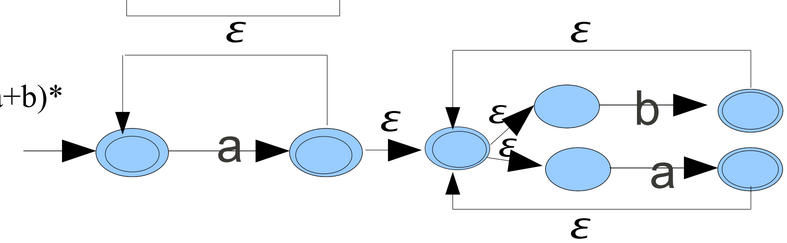
a) a* (b + (a + b))* b.

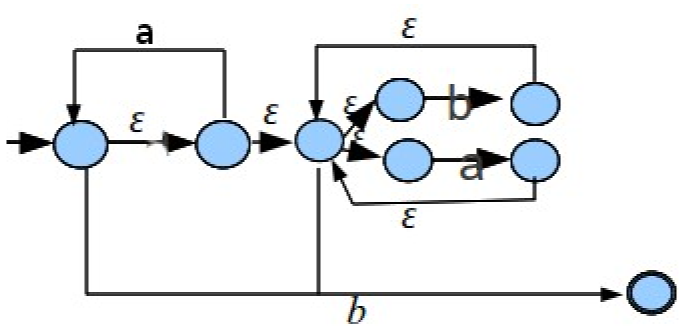
a 

b 

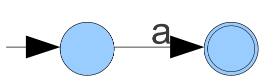
a+b = b+(a+b)


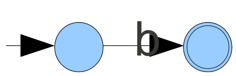
(a+b)*


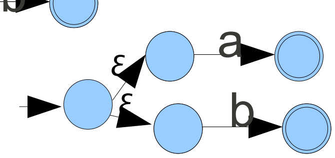
a*(a+b)*


a*(a+b)*b


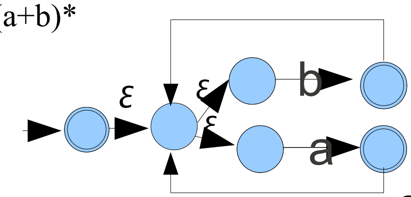
b) ((a + ε) + ab*) (a + b)*

a (a+ ε = a) 

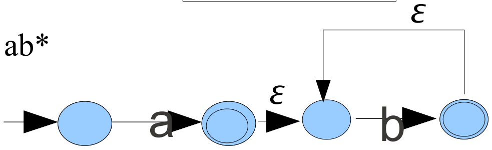
b 

a+b = b+(a+b)


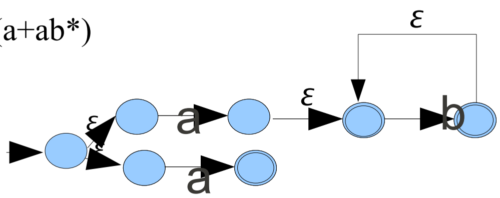
(a+b)*



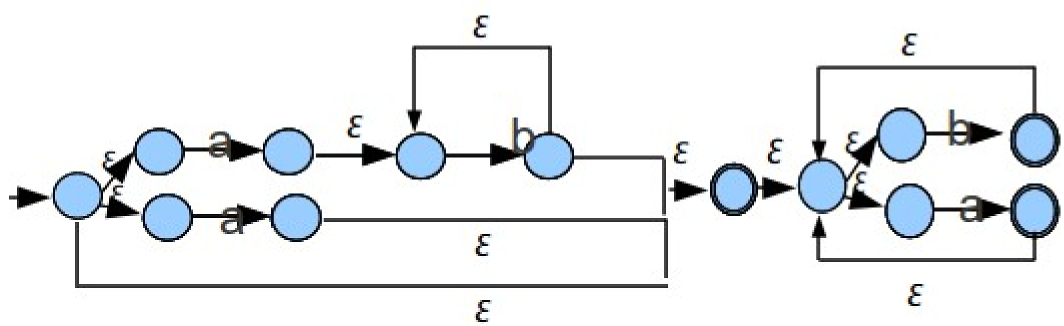
ab*



(a+ab*)

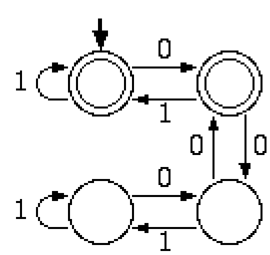


((a + ε) + ab*) (a + b)*



4. (Valor 2,0 pontos) Quais das seguintes linguagens são Regulares? (Você deve provar sua resposta, em cada caso.)

a) O conjunto dos strings que possuem número par de 00. (Note que três zeros seguidos contam como dois 00).
REGULAR. Essa linguagem é reconhecida pelo seguinte DFA:



b)O conjunto dos strings sobre o alfabeto {0} da forma 0^n onde n não é primo.

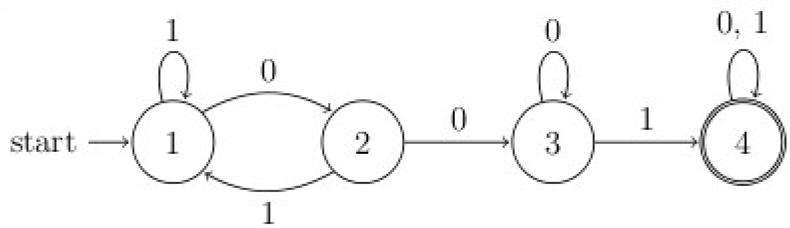
NÃO. Para provar que essa linguagem não é regular, examinamos o complemento dessa linguagem, uma vez que o conjunto das linguagens regulares é fechado em relação ao complemento.

Suponha que esse conjunto seja regular. Seja p o pumping lenght da linguagem. Então, de acordo com o pumping lemma, o string $s=0^p$ pode ser dividido em s=xyz onde y tem comprimento maior que 0. Então, $s=xy^iz = 0^{p + (i-1) |y|}$ deve pertencer à linguagem, para qualquer i. Em particular, seja i = p+1. Entãp $xy^{p+1}z=0^{p+p|y|}$ deve pertencer à linguagem, ou seja, p + p|y| = p(1 +|y|) deve ser primo. Como esse número claramente não é primo, temos uma contradição e a linguagem, portanto, não é regular.

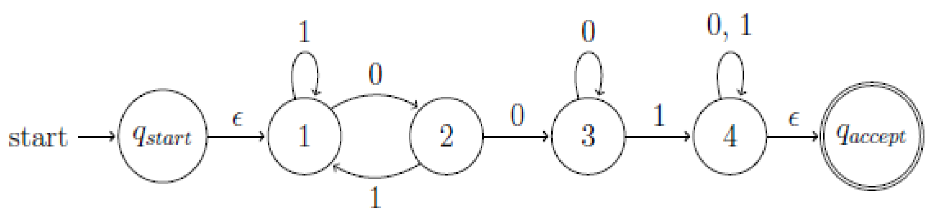
5. (Valor 1,0 ponto) Traduza o formalismo seguinte em Autômato Finito

a) (ab+ba)*(aa+bb)*: Zero ou mais sequências das substrings 'ab' ou 'ba' seguidas de zero ou mais repetições das substrings 'aa' ou bb'.
b) ab(abb*+baa*)*ba: Palavras onde as substrings 'ab' e 'ba' são separadas por zero ou mais repetições da substring 'ab'seguida de zero ou mais 'b's' ou repetições da substring 'ba' seguida de zero ou mais 'a's'.

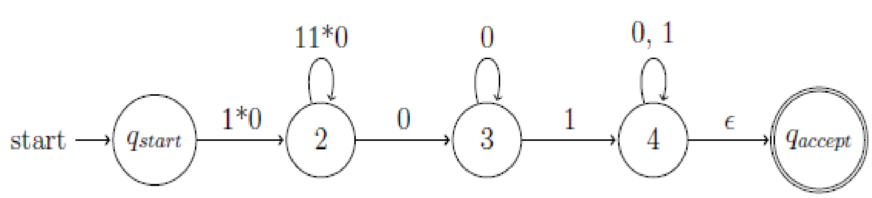
6. (Valor 1,0 ponto) Considere o seguinte AFD:



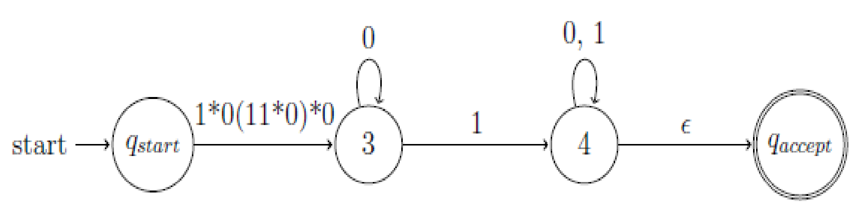
a) Converta o AFD em expressão regular usando a definição de Autômato finito não-determinístico generalizado (AFNDG). A ordem de remoção dos estados **deve** ser 1, 2, 3 e 4.
1º – Adicionar estado inicial e estado aceito. As transições vazias entre q_start e os outros estados, bem como de todos os estado para q_accept foram omitidas por questão de espaço:



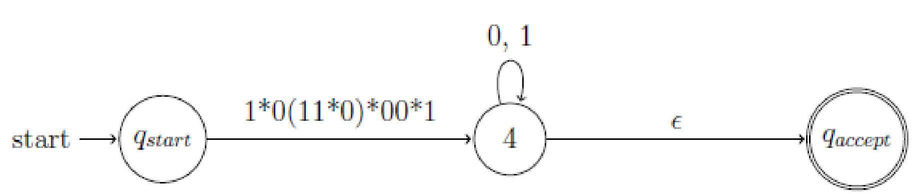
2º – Remova os estados na ordem E1



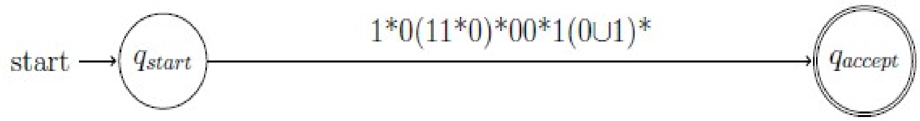
E2



E3

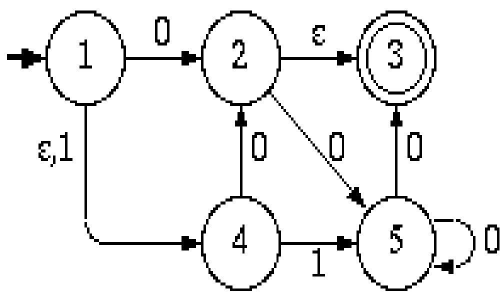


E4



Expressão regular resultante: 1*0(11*0)*00*1(0𝑈1)*.

7. (Valor 1,0 ponto) Encontre o autômato minimizado do AF seguinte:



8. (Valor 1,0 ponto) Prove (justifique em poucas palavras), ou dê um contra-exemplo, para mostrar que cada para cada um dos seguintes pares expressões regulares representa, ou não, a mesma linguagem. Suponha que r, s e t representam expressões regulares sobre o alfabeto {0,1}.

a) r(s + t) and rs + rt são equivalentes, porque a primeira descreve o conjunto de strings formado por um a string de r seguido de um string de s ou um string de t, e a segunda descreve o conjunto dos strings que, ou são formados por um string de r seguido de uma string de s ou são formados por um string de r seguido de um string de t e esses conjuntos são, claramente, o mesmo.
b) (r*)*and r* são equivalentes, porque a primeira descreve a concatenação de um número arbitrário de termos que são, eles próprios, concatenações de um número arbitrário de strings de r. Isso é o mesmo que r*, que é a concatenação de um número arbitrário de strings de r.
c) (r + s)* and r*s* NÃO são equivalentes, porque se s_1 é um string de s e r_1 é um string de r, então s_1r_1 é um string de (r+s)*, mas não de r*s*.