PROVA DE LINGUAGENS E COMPUTAÇÃO

Época normal 2021/22

PROPOSTA DE RESOLUÇÃO

e-fG/Ex 1)

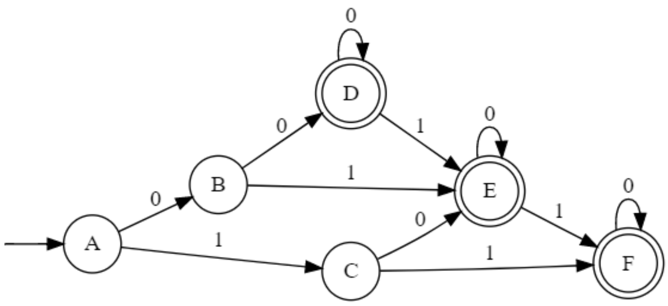
Distribuição: descrição da linguagem – 1; transformação em DFA – 1.

L = {w ∈ Σ* | |w| ≥ 2 e w contém, no máximo, dois 1's}

δ	0	1
→A = {q0}	{q0,q1} = B	{q1} = C
B	{q0,q1,q2} = D	{q1,q2} = E
C	E	{q2} = F
*D	D	E
*E	E	F
*F	F	Ø

Não podemos simplificar, dado que os pares (B,D) e (C,E), apesar de terem, dois a dois, transações iguais para o mesmo par (estado,símbolo), um deles é não-final e outro é final.

Assim, temos:



Basicamente, os estados B e C são os estados em que ainda só foi lido um símbolo do alfabeto, 0 ou 1, respetivamente, D, E e F são estados finais em que já foram lidos, pelo menos, dois símbolos, sendo que em D não foi lido nenhum 1, em E já foi lido um 1, e em F já foram lidos dois 1's, pelo que, tendo sido atingido o número limite de 1's, a partir desse momento só pode receber 0's, não saindo mais desse estado.

e-fG/Ex 2)

Distribuição: expressão regular – 2.

1*01*0(0+1)*

Possível alternativa: (0+1)*0(0+1)*0(0+1)*

Ex 3)

Distribuição: cada uma das linhas – 1.

(1+3+5+7+9)(04+12+2(0+8)+36+44+52+6(0+8)+76+84+92) +

(2+4+6+8)(0(0+8)+16+24+32+4(0+8)+56+64+72+8(0+8)+96)

Ex 4)

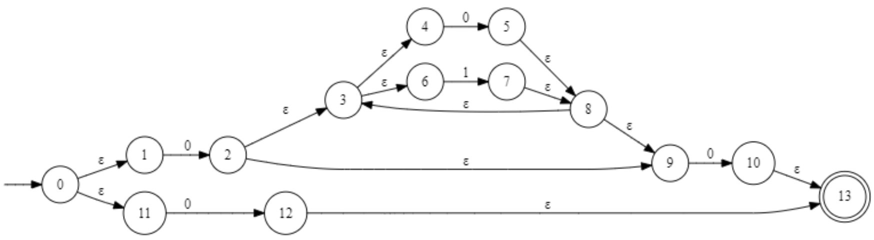
Distribuição: expressão regular – 0,5; NFA-ε – 1,5.

Foram consideradas duas respostas, uma considerando que 0 também começa e termina em 0 (mais correta) e a outra considerando que o 0 de início e de fim são distintos.

Hipótese A:

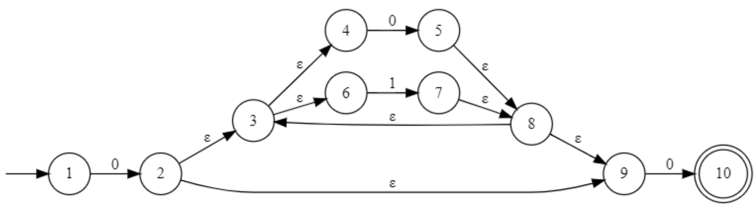
Expressão regular:

0(0+1)*0 + 0



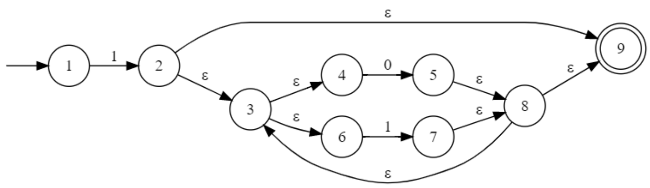
Hipótese B:

Expressão regular: 0(0+1)*0



e-fG 3/Ex 5)

Distribuição: diagrama – 1; transformação em DFA – 1.

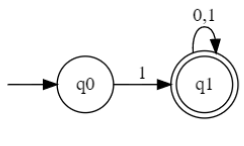


Fecho- ε(1) = {1}

δ	0	1
→A = {1}	Ø	{2,3,4,6,9} = B
*B	{3,4,5,6,8,9} = C	{3,4,6,7,8,9} = D
*C	C	D
*D	C	D

Como B, C e D têm as mesmas transições para cada par (estado,símbolo) e são todos estados finais, podemos simplificar B = C = D. Assim, ficamos com

δ	0	1
→A	Ø	B
*B	B	B



e-fG 4/Ex 6)

Distribuição: gramática – 1,5; verificação da sequência: 0,5.

n=1: 001; n=2: 0**001**11; n=3:00**001**1111; …; n=k: $0^{k+1}1^{2k-1}$ = 0^{k-1}**001**$1^{2(k-1)}$(isolando 001). Assim, temos:

G=({L}, {0,1}, {L → 0L11 | 001},L)

Alternativa:

L → 00M1, M → 0M11 | ε

L => 0L11 (L → 0L11)

=> 00L1111 (L → 0L11)

=> 000011111 (L → 001)

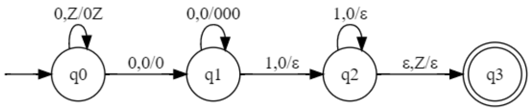
Ex 7)

Distribuição: dois 0's iniciais – 0,5; restantes 0's – 0,5; tratamento dos 1's – 0,5; reconhecimento da sequência – 0,5.

Se tivéssemos 0^n1^{2n-1}, colocaríamos inicialmente apenas um 0 para o 0 inicial e dois 0's para cada um dos outros, de modo a termos na pilha, antes do primeiro 1, 2n-1 0's. Como temos 0^{n+1}, teremos de ignorar um dos 0's. Assim, uma solução possível será:

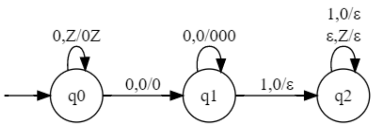
Versão com estado final

P=({q0,q1,q2,q3}, {0,1}, {0,Z}, δ, q0, Z, {q3})



Versão por pilha vazia

P=({q0,q1,q2}, {0,1}, {0,Z}, δ, q0, Z)



e-fG 5/Ex 8)

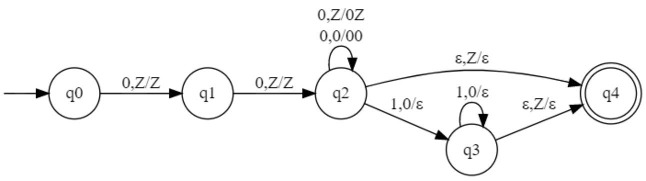
Distribuição: dois 0's iniciais – 0,5; restantes 0's – 0,5; tratamento dos 1's – 0,5; reconhecimento da sequência – 0,5.

n=1: 00; n=2: 0001; …; n=k: $0^{k+1}1^{k-1}$; n=k+1: $0^{k+2}1^k$

Neste caso, existem mais dois 0's do que 1's, pelo que bastará não fazermos nada com os dois primeiros 0's. No entanto, é preciso ter cuidado, porque para n=1 não existem 1's. Assim, uma possível solução será:

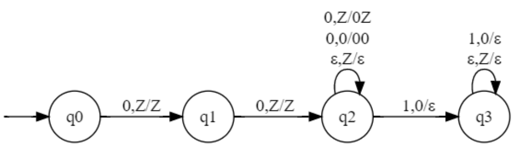
Versão com estado final

P=({q0,q1,q2,q3,q4}, {0,1}, {0,Z}, δ, q0, Z, {q4})



Versão por pilha vazia

P=({q0,q1,q2,q3}, {0,1}, {0,Z}, δ, q0, Z)



eFG 6/Ex 9)

Distribuição: 0 inicial e dois finais – 0,5; ciclo geral – 1; reconhecimento da sequência – 0,5.

n=1: 000; n=2:**0**01**00**; n=3: **0**00110**00**;…; n=k: $0^k1^{k-1}0^{k+1}=0(0^{k-1}1^{k-1}0^{k-1})00$. Assim, se isolarmos o primeiro e os dois últimos 0's, no resto da sequência temos correspondência de k-1 elementos.

Deste modo, tratamos primeiro esses 0's e depois o resto da sequência.

M=({p,q,r,s,t,u,v,w,x,y}, {0,1}, {0,1,X,Y,Z,B}, δ, p, B, {y}), onde δ é dado pela seguinte tabela:

δ	0	1	X	Y	Z	B
→p	(q,X,R)	------------	------------	------------	------------	------------
q	(q,0,R)	(q,1,R)	------------	------------	------------	(r,B,L)
r	(s,Z,L)	------------	------------	------------	------------	------------
s	(t,Z,L)	------------	------------	------------	------------	------------
t	(t,0,L)	(t,1,L)	(u,X,R)	(t,Y,L)	(t,Z,L)	------------
u	(v,X,R)	------------	------------	(x,Y,R)	(y,Z,R)	------------
v	(v,0,R)	(w,Y,R)	------------	(v,Y,R)	------------	------------
w	(t,Z,L)	(w,1,R)	------------	------------	(w,Z,R)	------------
x	------------	------------	------------	(x,Y,R)	(x,Z,R)	(y,B,L)
*y	------------	------------	------------	------------	------------	------------

Assim, começando no símbolo mais à esquerda, em p lemos o primeiro 0, colocando um X, em q andamos tudo para a direita até encontrarmos B, em r lemos o último 0, colocando um Z,. e em s lemos o penúltimo 0, colocando um Z, finalmente em t vamos andar tudo para a esquerda até encontrarmos o X, e transitarmos para u. Neste momento, temos a sequência na fita na forma $X0^{k-1}1^{k-1}0^{k-1}ZZ$, estando a cabeça da máquina no elemento à direita de X.

Em u, podem acontecer três coisas:

- se lermos Z, é sinal que temos o caso n=0, isto é, temos na fita XZZ; neste caso avançamos para o estado final y e a sequência foi reconhecida;
- se temos Y, quer dizer que já não temos mais 0's, pelo que só nos falta andar tudo para a direita e verificar se só temos Y's e Z's;
- se temos 0, vamos fazer o ciclo em que colocamos X, seguimos para v, onde colocamos Y no primeiro 1, e depois para w onde colocamos Z no primeiro 0 depois dos 1's, voltando a t, que vai fazer o mesmo processo anterior de andar tudo para a esquerda até encontrar X.

Finalmente, y é o estado final, sem transições, serve apenas para reconhecer a sequência, caso tenha sido toda lida.

Ex 10)

Distribuição: 0,5 para cada estado.

Num número binário, multiplicar por 2 corresponde a adicionar um 0 à direita da sequência. Subtrair 1, corresponde a começar no símbolo mais à direita e, da direita para a esquerda, enquanto tiver 0, substituo por 1, quando encontrar 1, substituo por 0 e paro de substituir.

Exemplo: 1010 multiplicando por 2 dá 10100, subtraindo 1 fica 10011.

No enunciado diz que devo começar no símbolo mais à direita e terminar no símbolo mais à esquerda. Temos assim:

M=({p,q,r,s }, {0,1}, {0,1,B}, δ, p, B, {s}), onde δ é dado pela seguinte tabela:

δ	0	1	B
→p	(p,0,R)	(p,1,R)	(q,1,L)
q	(q,1,L)	(r,0,L)	------------
r	(r,0,L)	(r,1,L)	(s,B,R)
*s	------------	------------	------------

Em p andamos para o símbolo à direita que deverá ser B (NOTA: neste caso, a cabeça da máquina até pode estar em cima de qualquer outro símbolo, também funciona). Ao encontrar B, substitui por 1, anda para a esquerda e muda para o estado q (multiplicou por 2 e fez logo a substituição do 0 acrescentado à direita por 1). Em q, da direita para a esquerda, substitui os 0's por 1, até encontrar um 1, substituir por 0 e mudar para o estado r. Em r só tem de andar tudo para a esquerda até encontrar B. Nesse momento, move-se para a direita, ficando sobre o dígito mais à esquerda do número (NOTA: pode ser 0), e transita para o estado s, que é o estado de aceitação final.