

”

E-fólio B | Folha de resolução para E-fólio



UNIDADE CURRICULAR: Linguagens e Computação

CÓDIGO: 21078

DOCENTES: Jorge Morais (professor) e Rúdi Gualter (tutor)

A preencher pelo estudante

NOME: Luís Carlos Crispim Pereira

N.º DE ESTUDANTE: 2300163

CURSO: LEI – Licenciatura em Engenharia Informática

DATA DE ENTREGA: 05/01/25

TRABALHO / RESOLUÇÃO:

1.) Para esta questão devemos começar por analisar o enunciado, onde é descrito que Zé Caracol e Ana Faísca tem o mesmo número de bolas (douradas e prateadas, respetivamente) e que Ana Faísca termina sempre primeiro. Independentemente de quem começa, Ana Faísca termina sempre primeiro. Com base neste contexto, podemos estabelecer todas as condições necessárias para criar uma CFG (gramática independente de contexto).

Condições:

O número de bolas douradas é igual ao número de bolas prateadas.

A sequência termina sempre com uma bola dourada.

A sequência pode começar com uma bola dourada ou uma bola prateada.

Nota: Na gramática uso "d" como bola dourada e "p" como bola prateada.

Na construção desta CFG, considerei diferentes fases. Inicialmente, analisei as possíveis formas de iniciar e terminar a sequência. Assim, identifiquei duas opções principais, a sequência pode começar com "d" e terminar com "d" ou começar com "p" e terminar com "d". Cada uma destas opções leva a um estado intermédio, o estado A, quando começa e termina com "d", e o estado B, quando começa com "p" e termina com "d".

No estado A, garanti que a quantidade de "p" equilibra a quantidade de "d" iniciais. Para isso, utilizei a sequência "pp" como saída deste estado. Este estado assegura a correção da paridade entre "d" e "p", podendo transitar de volta para o estado S ou para o estado B. Já no estado B, adicionei um "d" após um "p" ou vice-versa, criando combinações como "pd" e "dp". Este estado também permite a sequência terminal "ε", representando o final do processo. O estado B pode voltar a S ou terminar diretamente. Assim após testes e integrações cheguei a seguinte CFG:

$$S \rightarrow dAd \mid pBd \mid ddAAdd \mid \varepsilon$$
$$A \rightarrow pBp \mid dAp \mid pAd \mid dpApd \mid pdAdp$$
$$B \rightarrow BB \mid pSd \mid dSp \mid pBd \mid dBp \mid \varepsilon$$

Com esta construção, assegurei que todas as condições do problema são respeitadas. A paridade entre bolas douradas e prateadas é mantida, a sequência pode começar com "d" ou "p" e termina sempre com "d". As transições definidas garantem a inclusão de todas as combinações possíveis, refletindo corretamente o comportamento descrito no enunciado.

2.) Para esta questão construí uma tabela de combinações de tamanho seis e contendo três "d" e três "p", onde a coloquei na UAbALL e apresento aqui o seu resultado:

dpddpp	Insucesso!
ppddpd	Sucesso!
pddpdp	Insucesso!
ddpppd	Sucesso!
dpppdd	Sucesso!
dddppp	Insucesso!
pddppd	Sucesso!
dppddp	Insucesso!
pdppdd	Sucesso!
pdppdp	Insucesso!

ddpdpp	Insucesso!
ppdpdd	Sucesso!
pppddd	Sucesso!
dpdpdp	Insucesso!
ddppdp	Insucesso!
ppdddp	Insucesso!
pdddpp	Insucesso!
dppdpd	Sucesso!
pdppdp	Sucesso!
dpdppd	Sucesso!

Das 20 combinações possíveis, apenas 10 deram resultado "Sucesso!". Todas as combinações terminadas em "p" deram "Insucesso". Analisando verificamos que a CFG criada e testada esta correta, onde só deu "Sucesso!" quando Ana Faísca terminou primeiro, independentemente de quem começou, pois a paridade estava sempre garantida.

Derivações das strings "Sucesso!":

ppddpd:

Regra	Aplicação	Resultado
Início → S	Início	S
S → pBd	S	pBd
B → BB	pBd	pBBd
B → pSd	pBBd	ppSdBd
S → ε	ppSdBd	ppdBd
B → dSp	ppdBd	ppddSpd
S → ε	ppddSpd	ppddpd

ddpppd:

Regra	Aplicação	Resultado
Início → S	Início	S
S → dAd	S	dAd
A → dAp	dAd	ddApd
A → pBp	ddApd	ddpBppd
B → ε	ddpBppd	ddpppd

dpppdd:

Regra	Aplicação	Resultado
Início → S	Início	S
S → dAd	S	dAd
A → pAd	dAd	dpAdd
A → pBp	dpAdd	dppBpdd
B → ε	dppBpdd	dpppdd

pddppd:

Regra	Aplicação	Resultado
Início → S	Início	S
S → pBd	S	pBd
B → dBp	pBd	pdBpd
B → dSp	pdBpd	pddSpdp
S → ε	pddSpdp	pddppd

pdppdd:

Regra	Aplicação	Resultado
Início → S	Início	S
S → pBd	S	pBd
B → BB	pBd	pBBd
B → dSp	pBBd	pdSpBd
S → ε	pdSpBd	pdpBd
B → pSd	pdpBd	pdppSdd
S → ε	pdppSdd	pdppdd

ppdpdd:

Regra	Aplicação	Resultado
Início → S	Início	S
S → pBd	S	pBd
B → pBd	pBd	ppBdd
B → dSp	ppBdd	ppdSpdd
S → ε	ppdSpdd	ppdpdd

pppddd:

Regra	Aplicação	Resultado
Início → S	Início	S
S → pBd	S	pBd
B → pSd	pBd	ppSdd
S → pBd	ppSdd	pppBddd
B → ε	pppBddd	pppddd

dppdpd:

Regra	Aplicação	Resultado
Início → S	Início	S
S → dAd	S	dAd
A → pBp	dAd	dpBpd
B → pSd	dpBpd	dppSdpd
S → ε	dppSdpd	dppdpd

pdpdpd:

Regra	Aplicação	Resultado
Início → S	Início	S
S → pBd	S	pBd
B → dSp	pBd	pdSpd
S → pBd	pdSpd	pdpBdpd
B → ε	pdpBdpd	pdpdpd

dpdpdpd:

Regra	Aplicação	Resultado
Início → S	Início	S
S → dAd	S	dAd
A → pBp	dAd	dpBpd
B → dSp	dpBpd	dpdSpdp
S → ε	dpdSpdp	dpdpdpd

Nota: Todas estas strings tem derivações ambíguas, só apresentei uma das derivações.

3.) Nesta alínea descobre-se que Ana Fáisca nunca esta em segunda podendo estar empatada ou a frente do irmão Zé Caracol sendo assim as condições a cumprir são:

Condições:

O número de bolas douradas é igual ao número de bolas prateadas.

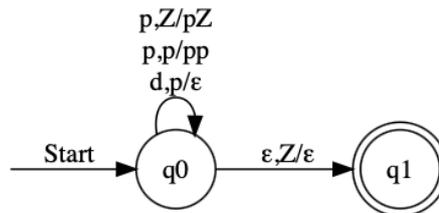
A sequência termina sempre com uma bola dourada.

A sequência só pode começar com uma bola prateada.

Criei então um PDA que verifica as condições onde o pensamento foi que quando recebe "p" empilha "p" e quando recebe diferente ou seja "d" desempilha "p" sendo que não recebe "d" em caso de não haver "p" para desempilhar terminando quando a pilha ficar vazia. Por exemplo, "pppddd" contando o topo da pilha como o elemento mais à esquerda, teremos sucessivamente:

Z, pZ, ppZ, pppZ, ppZ, pZ, Z e finalmente pilha vazia.

O PDA a que cheguei foi:



O PDA comporta-se de seguinte maneira:

Se a pilha tiver vazia:

- Receber p empilha p
- Não recebe nada, não faz nada e passa para o estado final.

Se a pilha tiver p:

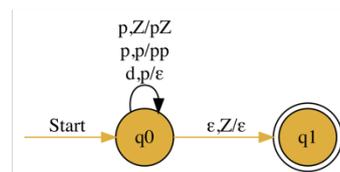
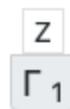
- Receber p empilha p
- Receber d desempilha p

Com esta construção simples, assegurei que todas as condições do problema são respeitadas. As transições definidas garantem a inclusão de todas as combinações possíveis, refletindo corretamente o comportamento descrito no enunciado.

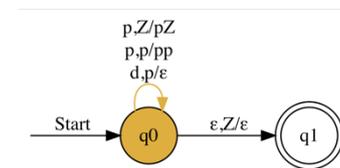
4.) Usando a ferramenta UAbALL, verifique o que acontece com as sequências "pdppdd" e "pddppd" no PDA.

Simulação passo a passo "pdppdd":

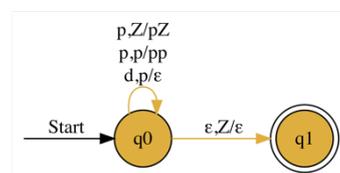
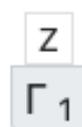
Estado	Por Começar
Entrada	pdppdd



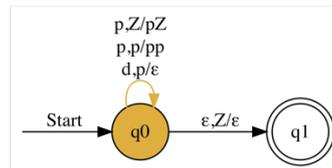
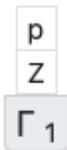
Estado	Em simulação!
Entrada	pdppdd



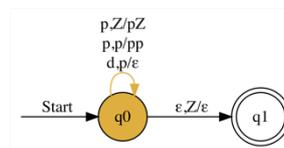
Estado	Em simulação!
Entrada	pdppdd



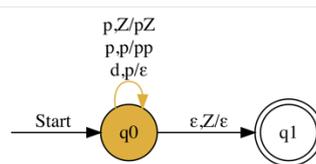
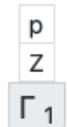
Estado	Em simulação!
Entrada	pd pp dd



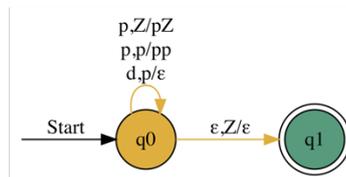
Estado	Em simulação!
Entrada	pd pp dd



Estado	Em simulação!
Entrada	pd pp dd

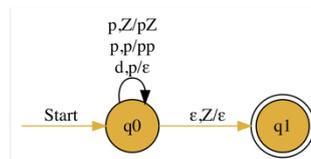
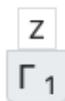


Estado	Sucesso!
Entrada	pd pp dd

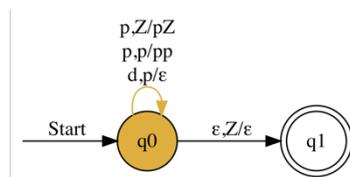
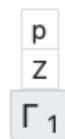


Simulação passo a passo "pddppd":

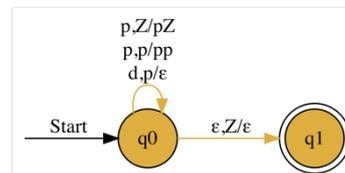
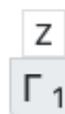
Estado	Por Começar
Entrada	pddppd



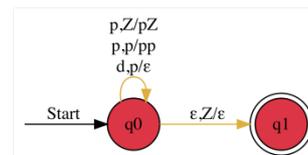
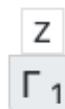
Estado	Em simulação!
Entrada	p d dppd



Estado	Em simulação!
Entrada	p d dppd



Estado	Insucesso!
Entrada	p d dppd



Como seria de esperar "pdppdd" dá sucesso, sendo que Ana Faísca nunca fica para trás, e "pddppd" falha (ou seja insucesso) quando Zé Caracol ficaria a frente, algo que nunca acontece. Sendo assim esta comprova-se o bom funcionamento do PDA.