

Augusto Grai Dal'Asta

Questão 1 - Letra E

Questão 2 -

I - (V)

II - (F) Um token, por vezes, pode não ser representado por estas 3 informações de valores através do uso de informações de classe e posição para fazer sua representação.

III - (F) É possível utilizar ER, GR e AF para gerar os tokens.

IV - (F) Na análise léxica ocorre apenas a geração de Tokens. A geração de árvore ocorre apenas na análise sintática.

Questão 3 -

	id	-	*	^	()	\$
id		>	>	>		>	>
-	<		<	<	<	>	>
*	<	>		<	<	>	>
^	<	>	>		<	>	>
(<	<	<	<		=	
)		>	>	>		>	>
\$	<	<	<	<	<		OK

DS/IO/JS

Stack (top) skip

Pilha	Rel.	Entrada	Ação	Handle
\$	<	(id-id*id^(id))\$	E(
\$(<	id-id*id^(id))\$	E id -	
\$(id	>	-id*id^(id))\$	R	E → id
\$(E	<	-id*id^(id))\$	E -	
\$(E-	<	id*id^(id))\$	E id	
\$(E-id	>	*id^(id))\$	R	E → id
\$(E-E	<	*id^(id))\$	E *	
\$(E-E*	<	id^(id))\$	E id	
\$(E-E*id	>	^(id))\$	R	E → id
\$(E-E*E	<	^(id))\$	E ^	
\$(E-E*E^	<	(id))\$	E(
\$(E-E*E^(<	id))\$	E id -	
\$(E-E*E^(id	>))\$	R	E → id
\$(E-E*E^(E	=))\$	E)	
\$(E-E*E^(E)	>)\$	R	(E) → E
\$(E-E*E^E	>)\$	R	E ^ E → E
\$(E-E*E	>)\$	R	E * E → E
\$(E-E	>)\$	R	E - E → E
\$(E	=)\$	E)	
\$(E)	>	\$	R	(E) → E
\$E	OK	\$	AC	

Questão 4 -

a)

PILHA	ENTRADA	AÇÃO
0	ccccccdd\$	E3
0c3	ccccdd\$	E3
0c3c3	ccccdd\$	E3
0c3c3c3	ccccdd\$	E3
0c3c3c3c3	ccdd\$	E3
0c3c3c3c3c3	cdd\$	E3
0c3c3c3c3c3c3	dd\$	E4
0c3c3c3c3c3c3d\$	d\$	R3
0c3c3c3c3c3c3C8	d\$	R2
0c3c3c3c3c3C8	d\$	R2
0c3c3c3c3C8	d\$	R2
0c3c3c3C8	d\$	R2
0c3c3C8	d\$	R2
0c3C8	d\$	R2
0c2	d\$	E7
0c2d7	\$	R3
0c2c5	\$	R1
0s1	\$	AC

b)

PILHA	ENTRADA	AÇÃO
0	c d c c c d c \$	E3
0 c 3	d c c c c d c \$	E4
0 c 3 d 4	c c c c d c \$	R3
0 c 3 c 8	c c c c d c \$	R2
0 c 2	c c c c d c \$	E6
0 c 2	c c c d c \$	E6
0 c 2 c 6	c c d c \$	E6
0 c 2 c 6 c 6	c d c \$	E6
0 c 2 c 6 c 6 c 6	d c \$	E7
0 c 2 c 6 c 6 c 6 d 7	c \$	ERRO!

c) $\{dd, cdd, dcd, ccdd, cdcd, dccd, cccdd, ccdcd, cdccd, dccccd\}$

Descreva uma linguagem recursiva a direita que aceita uma quantidade qualquer de c e exatamente 2 d, tendo d como seu caractere final.