

1) Dada a GLC:
Símbolo inicial: {S}
Variáveis ou Não Terminais: {S, M, T}
Terminais: {+, n}
Produções abaixo:

$S \rightarrow TM$
 $M \rightarrow +TM \mid \lambda$
 $T \rightarrow n$

A) mostre uma expressão regular para a linguagem produzida pela GLC acima.

$S \Rightarrow TM \Rightarrow nM \Rightarrow n$
 $S \Rightarrow TM \Rightarrow nM \Rightarrow n+TM \Rightarrow n+nM \Rightarrow n+n+TM \Rightarrow n+n+nM \Rightarrow n+n+n$

$n (+ n)^*$

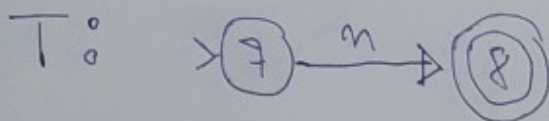
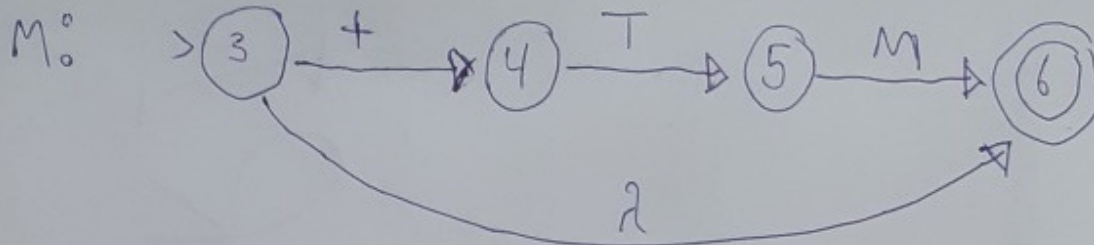
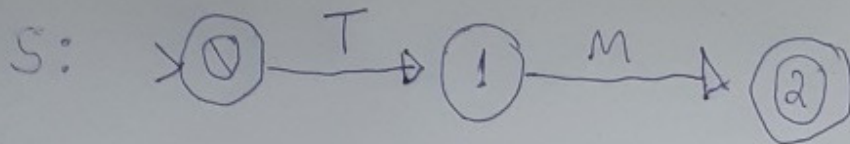
B) construa um programa em portugol para a GLC acima. De forma similar ao que foi feito para a análise com retrocesso.

```
Bool S (){  
    If( ( T() ) && ( M() ) )  
        Return true;  
    Else  
        Return false;  
}
```

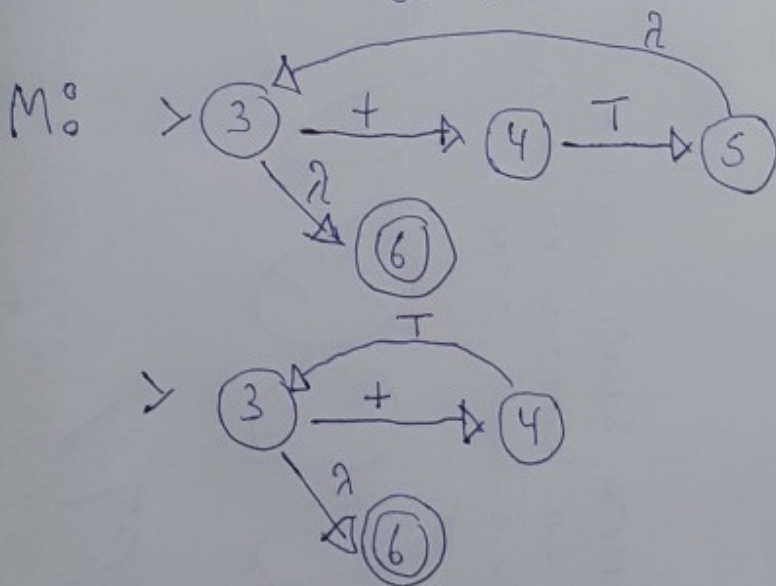
```
Bool M(){  
    If (lookahead == '+'){  
        Match('+');  
        S();  
    }  
    If (lookahead == '$'){  
        Return True;  
    }  
    Else{  
        Return False;  
    }  
}
```

```
Bool T (){  
    If( lookahead == 'n' ){  
        Match('n' );  
        Return true;  
    } else{  
        Return false;  
    }  
}
```

C) para a GLC acima construa o diagrama de transição, simplifique o mesmo e mostre o diagrama resultante de sua simplificação.



SIMPLIFICANDO "M"



ANTES:

$S \rightarrow TM$

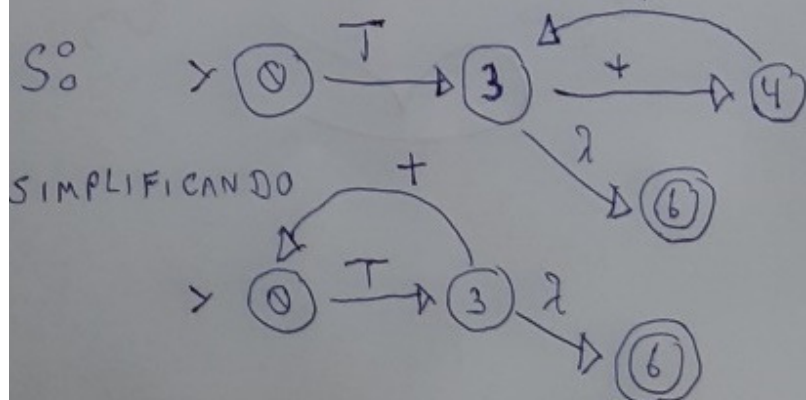
$M \rightarrow +TM \mid \lambda$

$T \rightarrow n$

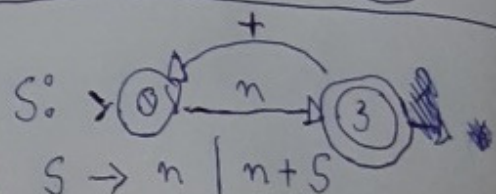
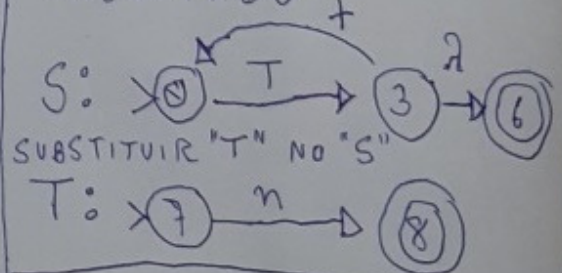
DEPOIS:

$S \rightarrow n+S \mid n$

SUBSTITUINDO "M" EM "S"



RESULTADO +



2) Com base na gramática e na tabela de parsing abaixo, mostre os movimentos do analisador tabular preditivo para a sequência de tokens: (id * id) + id

Gramática:

$E \rightarrow TE'$

$E' \rightarrow +TE' \mid \lambda$

$T \rightarrow FT'$

$T' \rightarrow *FT' \mid \lambda$

$F \rightarrow (E) \mid id$

Tabela de Parsing

	id	+	*	()	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

Execução

Pilha	Entrada	Saída
\$ E	(id * id) + id \$	$E \rightarrow TE'$
\$ E' T	(id * id) + id \$	$T \rightarrow FT'$
\$ E' T' F	(id * id) + id \$	$F \rightarrow (E)$
\$ E' T') E ((id * id) + id \$	
\$ E' T') E	id * id) + id \$	$E \rightarrow TE'$
\$ E' T') E' T	id * id) + id \$	$T \rightarrow FT'$
\$ E' T') E' T' F	id * id) + id \$	$F \rightarrow id$
\$ E' T') E' T' id	id * id) + id \$	
\$ E' T') E' T'	* id) + id \$	$T' \rightarrow *FT'$
\$ E' T') E' T' F *	* id) + id \$	
\$ E' T') E' T' F	id) + id \$	$F \rightarrow id$
\$ E' T') E' T' id	id) + id \$	
\$ E' T') E' T') + id \$	$T' \rightarrow \lambda$
\$ E' T') E') + id \$	$E' \rightarrow \lambda$
\$ E' T')) + id \$	
\$ E' T'	+ id \$	$T' \rightarrow \lambda$
\$ E'	+ id \$	$E' \rightarrow +TE'$
\$ E' T +	+ id \$	
\$ E' T	id \$	$T \rightarrow FT'$
\$ E' T' F	id \$	$F \rightarrow id$
\$ E' T' id	id \$	
\$ E' T'	\$	$T' \rightarrow \lambda$
\$ E'	\$	$E \rightarrow \lambda$
\$	\$	