**Dupla:**

EXERCÍCIOS - Análise Sintática Recursiva Descendente

Autômatos a Pilha

Linguagens Formais e Autômatos - 2020

Prof. Roberto C. de Araujo

Luan Damato - 31817051

1. Codificar um **ANALISADOR SINTÁTICO RECURSIVO DESCENDENTE** para a GLC abaixo

**S ** **X$**

**X ** **aXc**

**X ** **bY**

1. **** **d**
2. **** **ac**

**Y ** ****

/\*

\* Luan Rocha Damato - 31817051

\*

\*/

#include<stdio.h>

#include<stdlib.h>

#include <ctype.h> // isdigit, isalpha

/\*

\* S -> X$

\* X -> aXc

\* X -> bY e: palavra vazia

\* X -> d

\* Y -> ac

\* Y -> e

\*/

char lookahead; /\* Excepcionalmente variavel global \*/

int match(char t, char palavra[], int \*pos){

  if (lookahead == t){

    lookahead= palavra[++(\*pos)];

    return(1);

  }

  return(0);

}

/\* Y -> ac , Y -> e \*/

int Y(char palavra[], int \*pos){

  if (lookahead == 'a'){ /\* lookahead e� o FIRST!!! \*/

    if (match('a', palavra, pos) && match('c', palavra, pos))

return(1);

   else

return(0);

  }

else{

return(1); /\* caso com palavra vazia \*/

}

}

/\* X -> aXc , X -> bY , X -> d \*/

int X(char palavra[], int \*pos){

  if (lookahead == 'a'){ /\* lookahead e� o FIRST!!! \*/

    if (match('a', palavra, pos) && X(palavra, pos) && match('c', palavra, pos)){

return(1);

}

   else{

return(0);

}

  }

else if(match('b', palavra, pos)){

return (Y(palavra, pos));

}

else if(match('d', palavra, pos)) return (1);

else return(0);

}

int S(char palavra[], int \*pos){

if (X(palavra, pos) &&

  match('$', palavra, pos) )

return(1);

  else return(0);

}

void trataErro(){

  printf("\n\nERRO DE SINTAXE\n\n");

}

int main(){

  char palavra[]= "aaabacccc$";

  int pos=0;

  lookahead= palavra[pos];

  if (S(palavra, &pos))

   printf("\nPalavra %s reconhecida\n", palavra);

  else

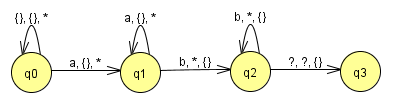
   trataErro();

  return(0);

}

1. Construa um autômato a pilha que reconheça a linguagem livre de contexto L = {anbn+1 | n  1}.

Palavra: aaabbbb



|  |  |  |
| --- | --- | --- |
| **Pilha** | **Entrada** | **Estado** |
| Base [ | aaabbbb | Q0 |
| Base [\* | aaabbbb | Q0 |
| Base [\* \* | aabbbb | Q1 |
| Base [\* \* \* | abbbb | Q1 |
| Base [\* \* \* \* | bbbb | Q1 |

|  |  |  |
| --- | --- | --- |
| Base [\* \* \* | bbb | Q2 |
| Base [\* \* | bb | Q2 |
| Base [\* | b | Q2 |
| Base [ |  | Q2 |
| Base [ |  | Q3 |

**Linguagens Formais e Autômatos - 2020**