Análise Léxica

Considere em se definir os “tokens” da linguagem. Para isso usamos transdutores.

Transdurtor

$$(p) \; ---\frac{\alpha}{\sigma}--> (q)$$

sendo:
α: Transição
σ: Ação

Considere uma linguagem com os seguintes elementos:

- variáveis
- números
- palavras reservadas
    - if
    - else
    - then
    - goto
    - print
    - read
    - let
    - end
    - of
- Sinais
    - >
    - <
    - =
    - !
    - :=
    - (
    -)
    - % (comentário)
- outros

Um exemplo de entrada possível seria:
let A:=10: print A: A:= A+1 : end %ABC

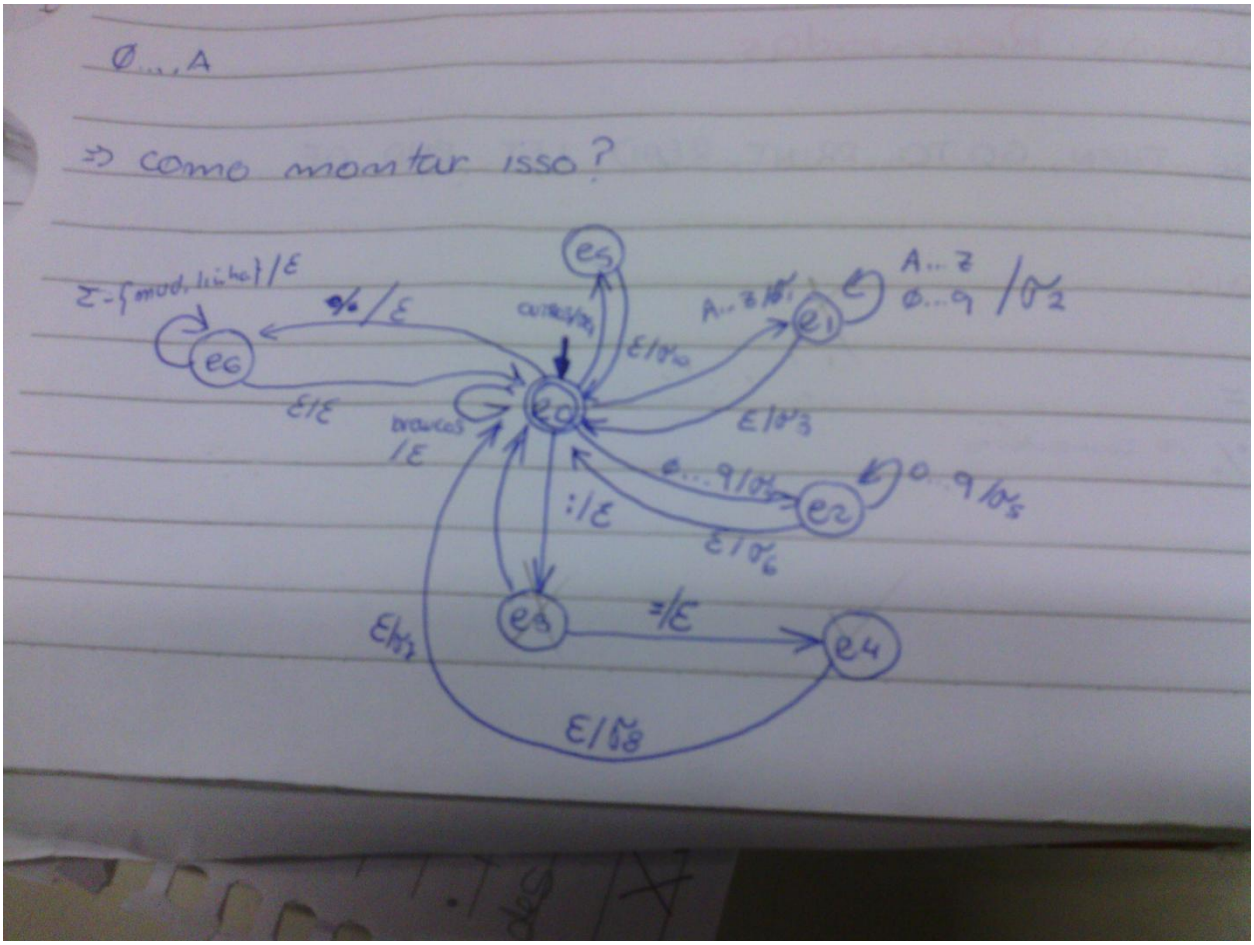
Saída
P(1) V(0) := N(10): P(3)V(0): V(0) := V(0) +N(1): P(8)

Palavras reservadas:
0. if
1. let
2. then
3. print
4. of
5. else
6. goto
7. read
8. end

Variáveis
0….A

ε

Como montar isso?



Considerar que:
TempS…. é um vetor char
TempN…. é um inteiro
TempC…. é um char

Há uma tabela de variáveis inicialmente vazia.

Assim:

σ_1....

TempS <- Simbolo

σ_2....

Anexa Simbolo em TempS

σ_3....

Finaliza TempS
se TempS está na tabela de palavras reservadas **então**
produz P([n° da posição na tabela])
senão
se TempS está na tabela de variáveis **então**
produz P([n° da posição na tabela]) na saída
senão
Cadastra variável na tabela
Produz P([n° da posição na tabela]) na saída

σ_4....

TempN <- símbolo - '0'

σ_5....

TempN <- TempN * 10 + Simbolo '0'

σ_6....

Produz N([valor de TempN]) na saída

σ_7....

Produz ":"

σ_8....

Produz ";"

σ_9....

TempC <- símbolo

σ_{10}....

Imprime conteúdo de TempC na saída

Exemplo:
Entrada: AB:= AB + C : print AB

TempS -> AB TempC
Saída: V(0) := V(0) + V(1) : P(3) V(0)

Entrada: let X := X1 + X2

Projeto 2: Construção do analisador léxico
O programa irá operar da seguinte forma

Digite uma linha:
if(A>10) then B:=B+A %ABC
Saída:
P(2)(V(0) > N(10))P(4)V(1):=V(1)+V(0)

Palavras Reservadas:
[lista]

Variáveis:
[lista]

Prazo para entrega 13/09