

Analísador Léxico

Tudo começa pela leitura do programa fonte: uma sequência de caracteres. Ocorre aqui a **análise léxica**: seqüências de caracteres significativos são organizadas em unidades chamadas marcas (ou em inglês: token). Uma marca aqui deve ser entendida como uma palavra em um texto em português. Por exemplo, a linha de código

alfa[indice] := 22 + 7;

Nesta linha as marcas são:

alfa um identificador (uma variável)

[colchete à esquerda

indice uma variável usada como índice em variável indexada

] colchete à direita

:= atribuição

22 um número composto de 2 dígitos

+ sinal de operação, no caso adição

7 número de um dígito

; terminador da linha

Note que o processo não é trivial, uma vez que ambos os comandos abaixo devem gerar exatamente a mesma lista de marcas:

alfa[indice]:=22+7;
alfa [indice] := 22+ 7 ;

Uma coisa que o sistema de análise léxica já pode fazer é criar uma tabela de símbolos (contendo todas variáveis referenciadas no código) e uma tabela de constantes (idem).

Expressões regulares

Uma ferramenta para expressar casamento de padrões. Composta por uma linguagem (Σ =conjunto de todos os caracteres válidos)+ ϵ . Um conjunto de metacaracteres, a saber:

alternativa (ou) No formato $r|s$ que significa r OU s .

concatenação (e) No formato rs que significa primeiro r e depois s .

qualquer coisa No formato r^* que significa r seguido de qualquer coisa.

uma ou mais r^+ , indica que haverá uma ou mais cópias de r

zero ou uma $r^?$, indica que haverá uma cópia de r ou não haverá nada.

qualquer caractere Σ . indica qualquer caracter de Σ .

intervalo $[a - z]$ resume $a|b|c|...|z$.

negação $[^abc]$ qualquer caractere que não seja a , nem b , nem c .

Usando ER, eis como serão as descrições de **número** e de **identificador** na nossa linguagem:

natural = [0-9]+
natcomsin = (+|-)? natural
numero = natcomsin ("." natural)? (E natcomsin)?

Veja-se agora a definição de um identificador:

letra = [a-zA-Z]
digito = [0-9]
identificador = letra(letra|digito)*

Autômatos Finitos

Um formalismo gráfico capaz de expressar (entre outros) expressões regulares: veja-se os 2 conceitos acima na forma de autômatos:

Trabalho esperado Espera-se um programa objeto que seja um programa fonte em ANSI C plenamente compilável e isento de quaisquer erros. Tal programa deve ler as entradas como aqui descritas e ao final deve imprimir a soma das últimas 6 linhas de cada entrada. É muito parecido com o exercício anterior, mas há diferenças: Primeiro, não existe mais a operação ELEVADO (que era representada pela letra E); Depois, não há mais as operações I, S e T exatamente para não complicar a análise dos comandos de entrada. Por outro lado, há necessidade de criar um separador de tokens, já que nomes e constantes podem ter comprimento variável.

Notem que a ausência das funções I, S e T permite um atalho simples para quem não conseguiu fazer o trabalho passado (que seria um pré-requisito para este). Se não houver jeito de operaciobalizer o exercício anterior (o U04), este aqui pode ser resolvido facilmente: basta entregar ao compilador C o comando da entrada devidamente acertado. Entretanto, quem usar este plano B não fará jus à nota integral.

Exemplos

A seguir, 3 exemplos do que se pede:

funcao E1
Y4GR=357!
C35=185.729!
K4000=53.63!
J57=5662!
LR7=32.63!
(3283+C35)-(1119-C35)!
(K4000+737)-(K4000+3598)!
(C35+8535)+(796-LR7)!
(C35+Y4GR)-(71.93-LR7)!
((LR7-354)+(K4000-C35))+(1384-LR7)!
(K4000+C35)-((283+317)+(J57+LR7))!
fimfuncao

Cujo resultado é 70267.14.

funcao E2
PT6N=15.68!
D2N58=72.55!
NKJ=741.125!
Z5443=2911!
CL2SZ=2648!
(D2N58+PT6N)+(8426+1009)!
(557.285+Z5443)-(4789-79.36)!
(CL2SZ+PT6N)-(663+CL2SZ)!
(NKJ-D2N58)+(CL2SZ-93.68)!
((7190+CL2SZ)+(PT6N+5704))+(PT6N+90.63)!
(CL2SZ+871)-((Z5443-9421)+(PT6N+71.91))!
fimfuncao

Cujo resultado é -1715850.15.

funcao E3
T3P2=1909!
UZ781=92.81!
U38=74.42!
W4R=270.557!
L11=5320!
(L11+3657)+(UZ781+U38)!
(L11-9796)+(U38-T3P2)!
(4383*962.614)-(UZ781*89.23)!
(UZ781+575.238)+(50.91*264)!
((95.36-L11)+(T3P2-7710))+(902.143-879.825)!
(W4R+T3P2)-((L11-W4R)+(UZ781+W4R))!
fimfuncao

Cujo resultado é 4220300.779.

🔗 Para você fazer

funcao P1
CF98=57.27!
L421=8752!

L6ML=9454!
L317Y=149!
J6PKW=938.041!
(L317Y-428)+(3849+L317Y)!
(771-614.147)-(885*L421)!
(L421-3631)-(L317Y+455)!
(789+22.84)+(L421+35.85)!
((235.975*625.494)+(L317Y+3091))+(L317Y+L421)!
(L6ML-24.58)-((L421+470.547)+(J6PKW+8199))!
fimfuncao

funcao P2
C246=799.636!
H229=385!
LBT=440!
SJ33=7867!
G7842=367!
(433-203)-(218.643-G7842)!
(H229-786.386)+(H229*288.619)!
(H229-2790)-(C246+29.36)!
(G7842+970.636)+(C246-H229)!
((H229-SJ33)+(C246+848))+(511*588.893)!
(G7842+LBT)-((H229-LBT)+(24.03+9822))!
fimfuncao

funcao P3
F24P4=105.749!
V598=388.232!
D5DON=74.12!
D9Y15=8438!
MV633=236!
(MV633+F24P4)+(D5DON-111)!
(D5DON+MV633)-(F24P4*84.95)!
(24.66-239.375)-(D9Y15-513)!
(V598+D9Y15)+(D5DON+D9Y15)!
((157.172+2890)-(V598*145))+(V598+673)!
(911.961-8386)-((MV633-D9Y15)+(536.155*640))!
fimfuncao

funcao P4
UF2N2=85.63!
G90B=96.28!
R75G6=41.57!
BJLRH=32.51!
D3A=310.178!
(UF2N2*3047)-(930*483.485)!
(3539*7734)-(R75G6-D3A)!
(BJLRH-554.631)+(G90B*8727)!
(G90B-D3A)-(273-404)!
((866.572-27.67)+(UF2N2+D3A))+(UF2N2+G90B)!
(432-7723)-((R75G6+BJLRH)+(2017+BJLRH))!
fimfuncao

Avaliação

A avaliação deste trabalho se fará em duas partes:

1. Valendo 20% a correção numérica obtida na execução dos 4 programas acima, e cujos valores achados foram

P1	P2	P3	P4

2. Valendo 80% da nota, a entrega do programa fonte e do programa executável do compilador. Esta entrega deverá se fazer em mídia magnética no pendrive do professor. Os programas fonte e executável deverão ter o nome de U5A00001

Por favor, coloque como comentário logo no início do seu programa fonte o seu nome. O programa fonte será examinado quanto a ocorrência de contrafação. O programa executável será certificado quanto à correspondência com o programa fonte entregue e receberá uma carga de teste inédita, cujos valores esperados são conhecidos pelo avaliador.



- 1 - /