PCS2056 - Linguagens e Compiladores

P2

Alunos:
Renan Martins Zomignani Mendes
Tiago Schelp Lopes

Professor:
Ricardo Luis de Azevedo da Rocha

08.12.2011

PRIMEIRA PARTE DA P2

1. (2,5 pontos) Construa um reconhecedor determinístico, baseado no autômato de pilha estruturado, que aceite como entrada válida um arquivo contendo descrições e comandos em Kipple. Não é necessário colocar toda a descrição da linguagem (incluindo bibliotecas), basta utilizar os elementos descritos explicitamente na sintaxe.

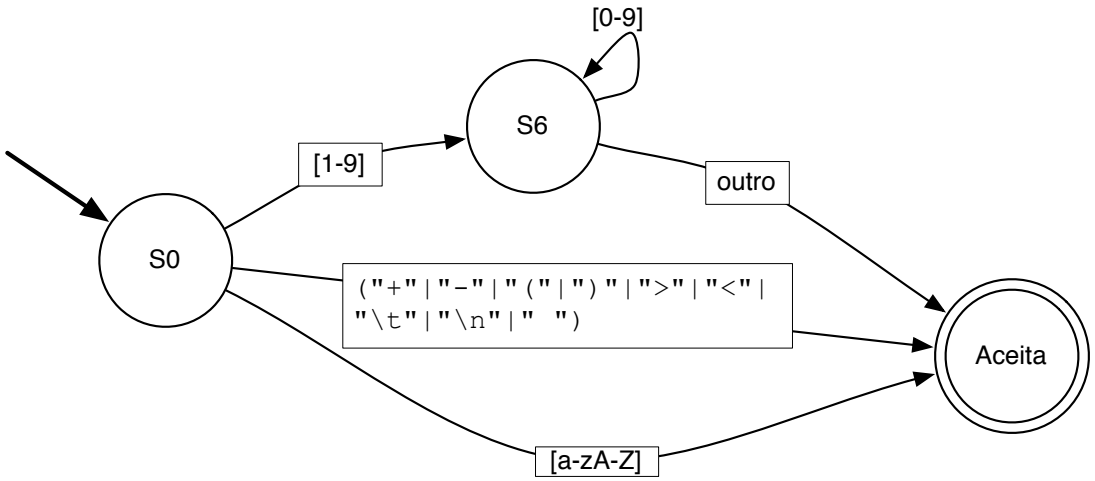
a) Descrição Wirth da linguagem:

```
STACK_ID =
"a"|"b"|"c"|"d"|"e"|"f"|"g"|"h"|"i"|"j"|"k"|"l"|"m"|"n"|
"o"|"p"|"q"|"r"|"s"|"t"|"u"|"v"|"w"|"x"|"y"|"z"|"A"|"B"|
"C"|"D"|"E"|"F"|"G"|"H"|"I"|"J"|"K"|"L"|"M"|"N"|"O"|"P"|
"Q"|"R"|"S"|"T"|"U"|"V"|"W"|"X"|"Y"|"Z"|"@".

NUM = "0"|(("1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9")
          {"0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"}).

OPERATION = OP1 | OP2.
OP_UM = NUM [REST_PR].
OP_DOIS = STACK_ID ( [REST_PR | REST_PL | REST_ADD |
                      REST_SUB | REST_CLR] ).
REST_PR  = ">" OP_DOIS.
REST_PL  = "<" (OP_UM | OP_DOIS).
REST_ADD = "+" (OP_UM | OP_DOIS).
REST_SUB = "-" (OP_UM | OP_DOIS).
REST_CLR = "?".

LOOP = "(" OP_DOIS CODE ")".
CODE = {LOOP | OPERATION}.
```

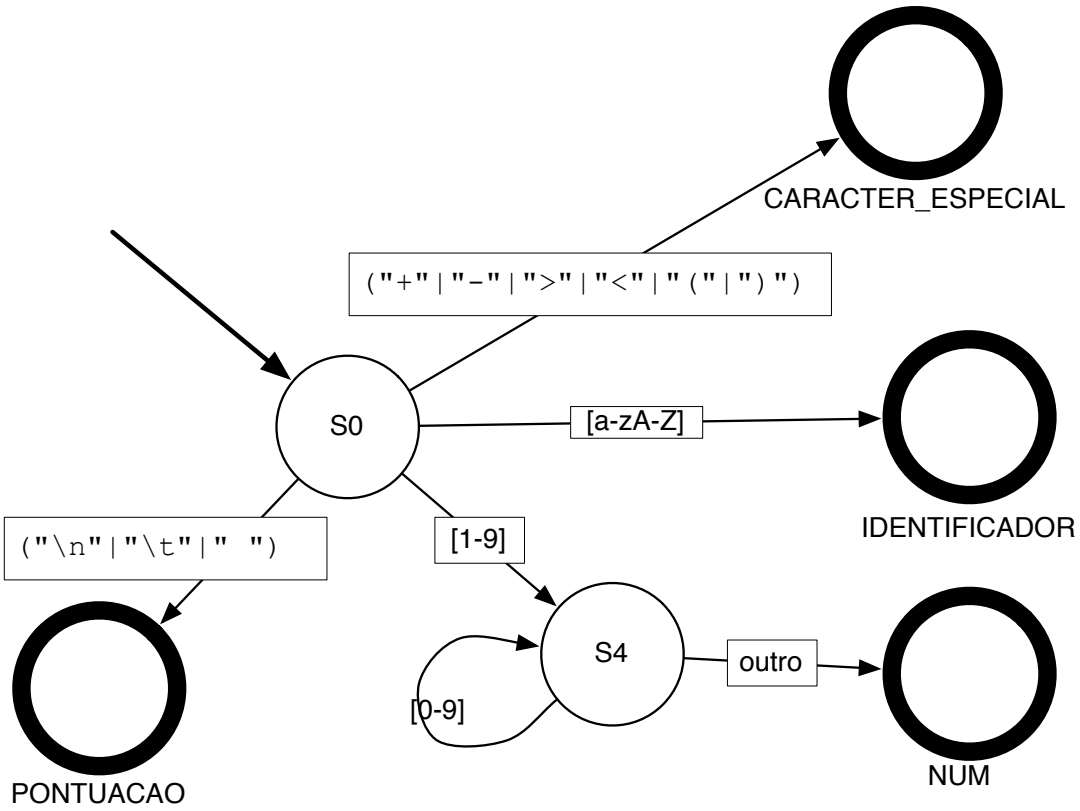
b) Tabela de tokens:

Expressão Regular	Valor	Tipo
["a"|"b"|…|"z"|"A"|"B"|…|"Z")	Lexema Lido	ID
(">"|"<"|"+"|"-"|"("|")")	Lexema Lido	SPECIAL_CARACTER
(0|[1-9][0-9]*)	Lexema lido	NUM
(" "|"\n"|"\t")	-	PONTUACAO

c) Autômato finito que reconhece os lexemas da linguagem:

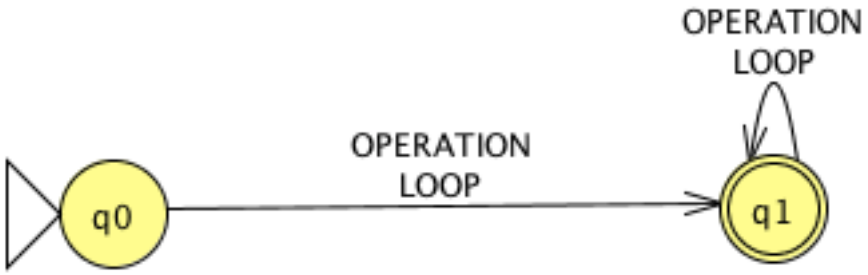


d) Transdutor:



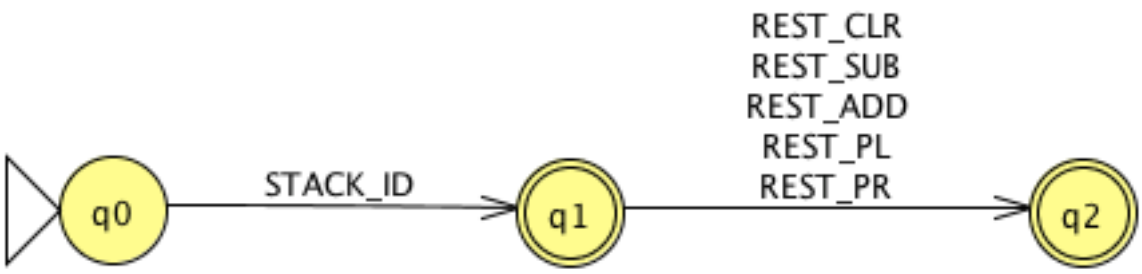
e) Autômato de pilha estruturado para o reconhecimento da linguagem Kipple.

CODE:

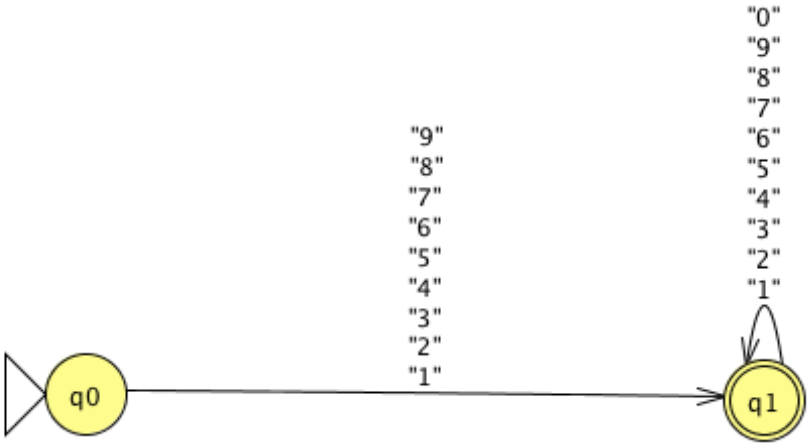


LOOP:

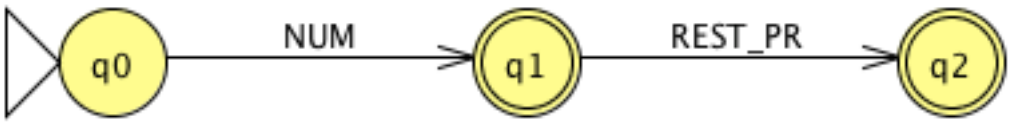
OP_DOIS:



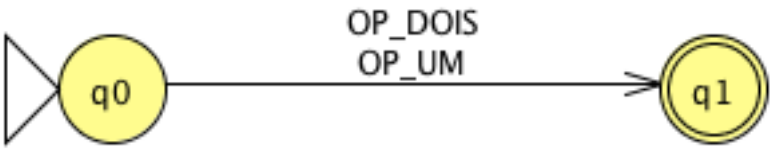
NUM:



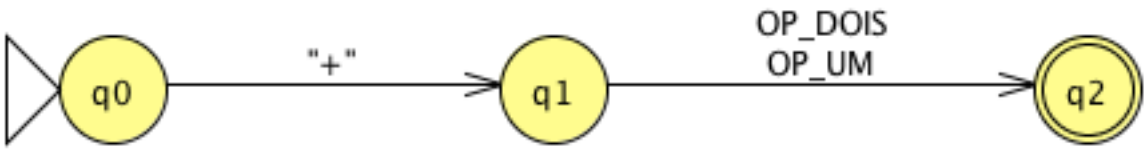
OP_UM:



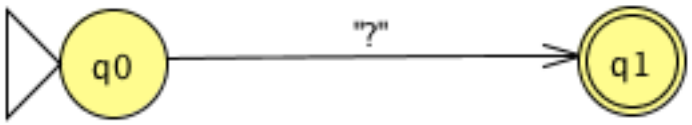
OPERATION:



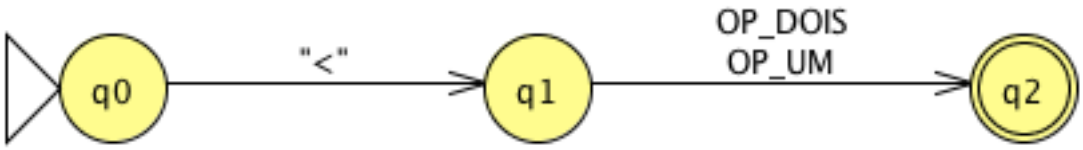
REST_ADD:



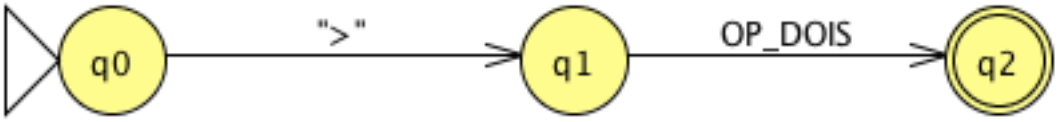
REST_CLR:



REST_PL:



REST_PR:



REST_SUB: