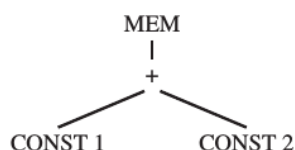




Trabalho III
Compiladores
Prof. Roberto Cabral

1. Implemente um algoritmo que recebe como entrada um conjunto de instruções no formato linear e o imprime em formato de árvore. Por exemplo, se a entrada for MEM(+ (CONST 1, CONST 2)), sua saída deve ser algo parecido com a figura a seguir.



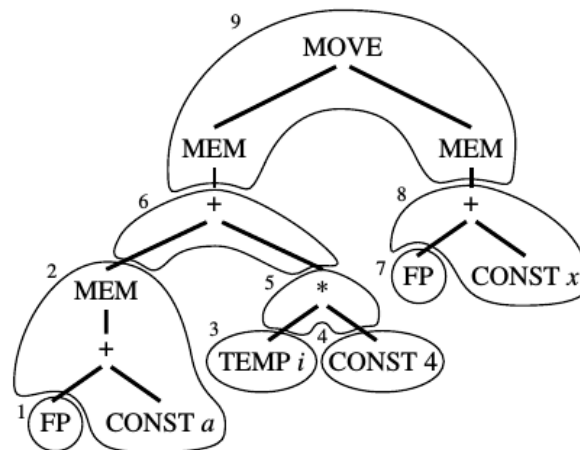
2. Implemente a fase de Seleção de Instrução de um compilador considerando os padrões da arquitetura *Jouette*, conforme tabela abaixo.

—	r_i	TEMP
ADD	$r_i \leftarrow r_j + r_k$	<pre> graph TD Plus1[+] --> Plus1L[] Plus1 --> Plus1R[] Plus1L --- Plus1LL[] Plus1R --- Plus1RR[] </pre>
MUL	$r_i \leftarrow r_j \times r_k$	<pre> graph TD Star[*] --> StarL[] Star --> StarR[] StarL --- StarLL[] StarR --- StarRR[] </pre>
SUB	$r_i \leftarrow r_j - r_k$	<pre> graph TD Minus1[-] --> Minus1L[] Minus1 --> Minus1R[] Minus1L --- Minus1LL[] Minus1R --- Minus1RR[] </pre>
DIV	$r_i \leftarrow r_j / r_k$	<pre> graph TD Slash[/] --> SlashL[] Slash --> SlashR[] SlashL --- SlashLL[] SlashR --- SlashRR[] </pre>
ADDI	$r_i \leftarrow r_j + c$	<pre> graph TD Plus2[+] --> Plus2L[] Plus2 --> Plus2R[] Plus2L --- Plus2LL[] Plus2R --- Plus2RR[] Plus2LL --- CONST1[CONST] Plus2RR --- CONST2[CONST] </pre>
SUBI	$r_i \leftarrow r_j - c$	<pre> graph TD Minus2[-] --> Minus2L[] Minus2 --> Minus2R[] Minus2L --- Minus2LL[] Minus2R --- Minus2RR[] Minus2LL --- CONST3[CONST] </pre>
LOAD	$r_i \leftarrow M[r_j + c]$	<pre> graph TD MEM1[MEM] --> Plus3[+] Plus3 --> Plus3L[] Plus3 --> Plus3R[] Plus3L --- Plus3LL[] Plus3R --- Plus3RR[] Plus3LL --- CONST4[CONST] Plus3RR --- MEM2[MEM] MEM2 --> Plus4[+] Plus4 --> Plus4L[] Plus4 --> Plus4R[] Plus4L --- Plus4LL[] Plus4R --- Plus4RR[] Plus4LL --- MEM3[MEM] Plus4RR --- CONST5[CONST] MEM3 --> Plus5[+] Plus5 --> Plus5L[] Plus5 --> Plus5R[] Plus5L --- Plus5LL[] Plus5R --- Plus5RR[] Plus5LL --- MEM4[MEM] Plus5RR --- CONST6[CONST] </pre>
STORE	$M[r_j + c] \leftarrow r_i$	<pre> graph TD MOVE1[MOVE] --> MEM5[MEM] MOVE1 --> Plus6[+] Plus6 --> Plus6L[] Plus6 --> Plus6R[] Plus6L --- Plus6LL[] Plus6R --- Plus6RR[] Plus6LL --- Plus6LLL[] Plus6RR --- Plus6RRR[] Plus6LLL --- CONST7[CONST] Plus6RRR --- CONST8[CONST] MEM5 --> Plus7[+] Plus7 --> Plus7L[] Plus7 --> Plus7R[] Plus7L --- Plus7LL[] Plus7R --- Plus7RR[] Plus7LL --- Plus7LLL[] Plus7RR --- Plus7RRR[] Plus7LLL --- MEM6[MEM] Plus7RRR --- CONST9[CONST] MEM6 --> Plus8[+] Plus8 --> Plus8L[] Plus8 --> Plus8R[] Plus8L --- Plus8LL[] Plus8R --- Plus8RR[] Plus8LL --- MEM7[MEM] Plus8RR --- CONST10[CONST] MEM7 --> Plus9[+] Plus9 --> Plus9L[] Plus9 --> Plus9R[] Plus9L --- Plus9LL[] Plus9R --- Plus9RR[] Plus9LL --- MEM8[MEM] Plus9RR --- CONST11[CONST] </pre>
MOVEM	$M[r_j] \leftarrow M[r_i]$	<pre> graph TD MOVE2[MOVE] --> MEM9[MEM] MOVE2 --> MEM10[MEM] MEM9 --> Plus10[+] Plus10 --> Plus10L[] Plus10 --> Plus10R[] Plus10L --- Plus10LL[] Plus10R --- Plus10RR[] Plus10LL --- MEM11[MEM] Plus10RR --- MEM12[MEM] MEM11 --> Plus11[+] Plus11 --> Plus11L[] Plus11 --> Plus11R[] Plus11L --- Plus11LL[] Plus11R --- Plus11RR[] Plus11LL --- MEM13[MEM] Plus11RR --- MEM14[MEM] </pre>



Nota: você deve utilizar o algoritmo baseado em Programação de Dinâmica e o algoritmo baseado no Guloso. Para computar os custos de cada instrução, considere os seguintes:

- I. A instrução TEMP (a primeira) tem custo zero, ou seja, um simples carregamento para um registrador, por exemplo, $R_i \leftarrow \text{TEMP } X$, tem custo zero.
 - II. O custo de uma instrução MOVEM é três, ou seja, carregar da memória e atribuir à memória, por exemplo, $M[R_1] \leftarrow M[R_2]$, tem custo dois.
 - III. O custo de um acesso a memória tem custo dois.
 - IV. Os custos das demais instruções são unitários.
3. Implemente uma função que recebe um conjunto de padrões (Questão 2) e exibe o código equivalente. Por exemplo:



2	LOAD	$r_1 \leftarrow M[\mathbf{fp} + a]$
4	ADDI	$r_2 \leftarrow r_0 + 4$
5	MUL	$r_2 \leftarrow r_i \times r_2$
6	ADD	$r_1 \leftarrow r_1 + r_2$
8	ADDI	$r_2 \leftarrow \mathbf{fp} + x$
9	MOVEM	$M[r_1] \leftarrow M[r_2]$



INFORMAÇÕES IMPORTANTES

Trabalho individual. A data máxima de entrega é 13/06/2018. A nota final do trabalho levará em conta o código fonte, o relatório e a apresentação para o professor.