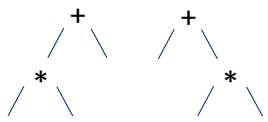
Jouette+

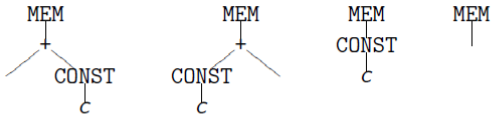
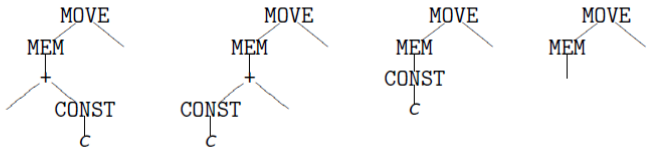
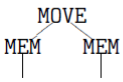
Set of instructions of a hypothetic microprocessor named as *Jouette+*:

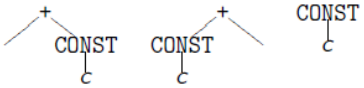
ADD rd = rs1 + rs2 ADDI rd = rs + c MUL rd = rs1* rs2 MADD rd = rs1*rs2+rs3 SUB rd = rs1 − rs2 SUBI rd = rs – c DIV rd = rs1/rs2	LOAD rd = M[rs + c] STORE M[rs1 + c] = rs2 MOVEM M[rs1] = M[rs2] MOVER rd = rs	GOTO goto label/address BNE if(rs1 != rs2) goto label/address BEQ if(rs1 == rs2) goto label/address BGE if(rs1 >= rs2) goto label/address BLE if(rs1 <= rs2) goto label/address BGT if(rs1 > rs2) goto label/address BLT if(rs1 < rs2) goto label/address

In which *rd* and *rs* identify 32-bit registers of the microprocessor (from *r0* to *r31*, with *r0* always with value 0), *c* identifies a constant (literal) and *M[x]* indicates an access to the memory address given by x.

The following are the tree patterns for some of the instructions:

Instruction	Effect	IR Tree Pattern	Instruction	Effect	IR Tree Pattern
MADD	ri ← r_j*r_k+r_t		MOVER	$r_i \leftarrow r_j$	

Instruction	Effect	IR Tree Pattern
load	$r_i \leftarrow M[r_j + c]$	
store	$M[r_j + c] \leftarrow r_i$	
movem	$M[r_j] \leftarrow M[r_i]$	

Instruction	Effect	IR Tree Pattern
—	r_i	
add	$r_i \leftarrow r_j + r_k$	
mul	$r_i \leftarrow r_j * r_k$	
sub	$r_i \leftarrow r_j - r_k$	
div	$r_i \leftarrow r_j / r_k$	
addi	$r_i \leftarrow r_j + c$	
subi	$r_i \leftarrow r_j - c$