

## IR Trees – Expressions 1/2

---

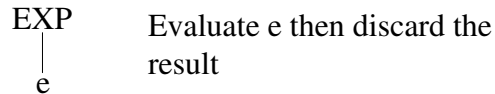
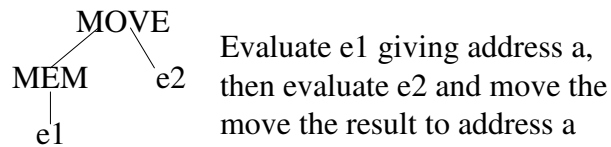
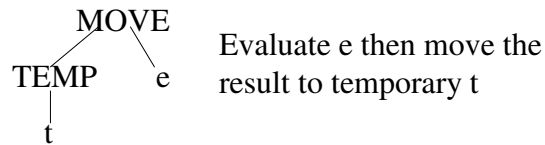
$\begin{array}{c} \text{CONST} \\   \\ i \end{array}$	Integer constant $i$
$\begin{array}{c} \text{NAME} \\   \\ n \end{array}$	Symbolic constant $n$
$\begin{array}{c} \text{TEMP} \\   \\ t \end{array}$	Temporary $t$ - a register
$\begin{array}{c} \text{MEM} \\   \\ m \end{array}$	Contents of a word of memory starting at $m$

## IR Trees – Expressions 2/2

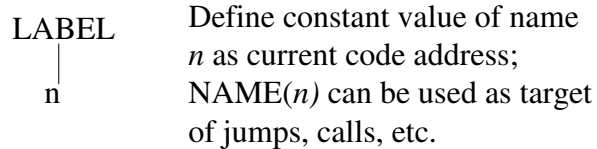
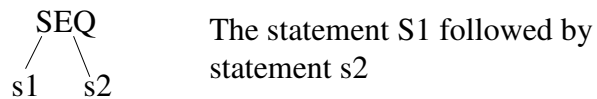
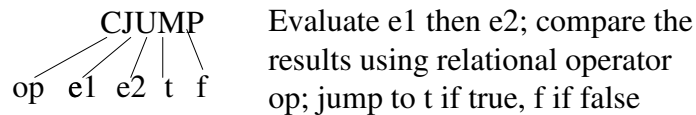
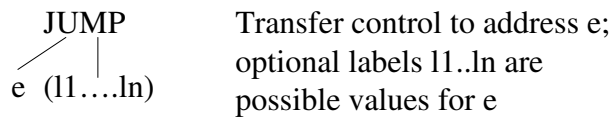
---

$\begin{array}{c} \text{BINOP} \\ / \quad   \quad \backslash \\ \text{op} \quad e1 \quad e2 \end{array}$	$e1$ op $e2$ - Binary operator Evaluate $e1$ , then $e2$ , then apply op to $e1$ and $e2$
$\begin{array}{c} \text{CALL} \\ / \quad   \\ f \quad (e1 \dots en) \end{array}$	Procedure call: evaluate $f$ then the arguments in order, then call $f$
$\begin{array}{c} \text{ESEQ} \\ / \quad \backslash \\ s \quad e \end{array}$	Evaluate $s$ for side effects then $e$ for the result

## IR Trees – Statements 1/2



## IR Trees – Statements 2/2



## Simplification Rules

---

- $ESEQ(s1, ESEQ(s2, e)) \Rightarrow$ 
    - $ESEQ(SEQ(s1, s2), e)$

---

  - $BINOP(op, ESEQ(s, e1), e2) \Rightarrow$ 
    - $ESEQ(s, BINOP(op, e1, e2))$
  - $MEM(ESEQ(s, e1)) \Rightarrow$ 
    - $ESEQ(s, MEM(e1))$
  - $JUMP(ESEQ(s, e1)) \Rightarrow$ 
    - $SEQ(s, JUMP(e1))$
  - $CJUMP(op, ESEQ(s, e1), e2, l1, l1) \Rightarrow$ 
    - $SEQ(s, CJUMP(op, e1, e2, l1, l2))$
  - $MOVE(ESEQ(s, e1), e2)$ 
    - $= SEQ(s, MOVE(e1, e2))$

---

  - $BINOP(op, e1, ESEQ(s, e2)) \Rightarrow$ 
    - $ESEQ(MOVE(TEMP t, e1), ESEQ(s, BINOP(op, TEMP t, e2)))$
  - $CJUMP(op, e1, ESEQ(s, e2), l1, l2) \Rightarrow$ 
    - $SEQ(MOVE(TEMP t, e1), SEQ(s, CJUMP(op, TEMP t, e2, l1, l2)))$

---

  - $CALL(f, a) =$ 
    - $ESEQ(MOVE(TEMP t, CALL(f, a)), TEMP(t))$
-