

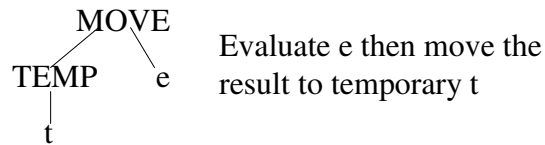
IR Trees – Expressions 1/2

$\begin{array}{c} \text{CONST} \\ \\ i \end{array}$	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

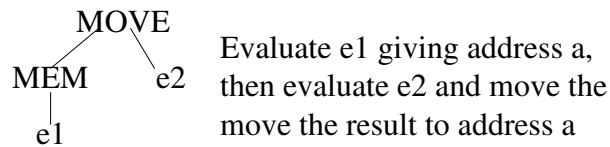
IR Trees – Expressions 2/2

$\begin{array}{c} \text{BINOP} \\ / \quad \quad \backslash \\ \text{op} \quad e1 \quad e2 \end{array}$	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

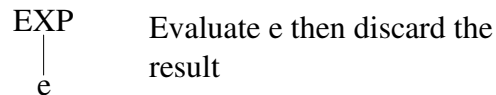
IR Trees – Statements 1/2



Evaluate e then move the result to temporary t

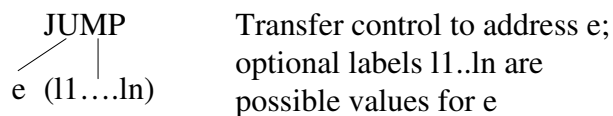


Evaluate e1 giving address a, then evaluate e2 and move the result to address a

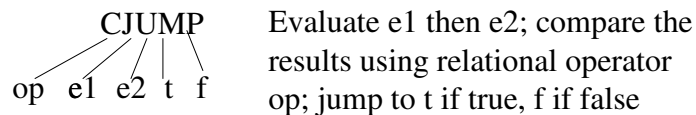


Evaluate e then discard the result

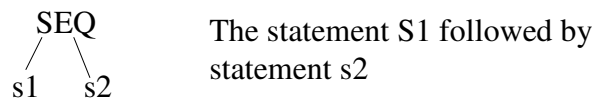
IR Trees – Statements 2/2



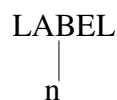
Transfer control to address e; optional labels l1..ln are possible values for e



Evaluate e1 then e2; compare the results using relational operator op; jump to t if true, f if false



The statement S1 followed by statement s2



Define constant value of name *n* as current code address; NAME(*n*) can be used as target of jumps, calls, etc.