Simple TAC Instruction Set

CSCI 434T
Spring, 2019

TAC

This handout summarizes a simple TAC intermediate language. There are many choices as to the exact instructions to include in such a language, and you will probably want to modify and extend this variant when we translate IC programs into TAC.

Instruction Forms

- **Arithmetic and Logic Instructions.**

  The basic instruction forms are:

  ```
  a = b OP c          a = OP b
  ```

  where `OP` can be

an arithmetic operator:	`ADD`, `SUB`, `DIV`, `MUL`
a logic operator:	`AND`, `OR`, `XOR`
a comparison operator:	`EQ`, `NEQ`, `LE`, `LEQ`, `GE`, `GEQ`
a unary operator:	`MINUS`, `NEG`

- **Data Movement Instructions.**

Copy:	`a = b`	
Load/store:	`a = *b`	`*a = b`
Array load/store:	`a = b[i]`	`a[i] = b`
Field load/store:	`a = b.f`	`a.f = b`

- **Branch Instructions.**

Label:	`label L`	
Unconditional jump:	`jump L`	
Conditional jump:	`cjump a L`	(jump to `L` if `a` is true)

- **Function Call Instructions.**

Call with no result:	`call f(`a_1`, ..., `a_n`)`
Call with result:	`a = call f(`a_1`, ..., `a_n`)`

  (Note: there is no explicit TAC representation for parameter passing, stack frame setup, etc.)