# COMS W4115
# Programming Languages and Translators
# Lecture 19: Code Generation
# April 8, 2013

## Lecture Outline

1. Issues in code generation
2. Memory hierarchy
3. Categories of target machines
4. Primary tasks of a code generator
5. Basic blocks and flow graphs

## 1. Issues in Code Generation

- Role of the code generator
  - Input: IR of the source program produced by the font end
  - Output: good target machine code
- Challenges of code generation
  - The target program must preserve the semantic meaning of the source program.
  - The target program should make efficient use of the target machine's resources.
  - The code generator itself should be efficient.
  - The problem of optimal code generation is undecidable.
  - Many subproblems in code generation, such as optimal register allocation, are computationally intractable.
  - The design of a good code generator often reverts to the problem of designing good heuristics.

## 2. Memory Hierarchy

- Processor registers are the fastest devices in a computer's memory hierarchy. Here are some typical access times and sizes of the components of a computer's memory hierarchy.

```
                Access Time                  Size
registers        0.2 - 0.5 ns               256 -  1024 B
L1 cache        0.4 -   1 ns                32 -   256 KB
L2 cache          4 -  10 ns               512 KB -  2 MB
main memory         50 - 500 ns                256 MB - 16 GB
disk                  5 -  15 ms                 80 GB
tape                  1 -  50 s                infinite
```

• As we can see, there are several orders of magnitude difference in the access time between registers and main memory. As a consequence, compiler code generators attempt to use registers to hold the most frequently used objects in the target program. But since there are a small number of registers, the compiler will have to make choices as to what quantities it keeps in registers and what quantities it must spill into main memory.

## 3. Categories of Target Machines

- Reduced instruction set machines (RISC)
  - many registers
  - three-address instructions
  - simple addressing modes
  - simple instruction set architecture
- Complex instruction set machines (CISC)
  - few registers
  - two-address instructions
  - variety of addressing modes
  - several register classes
  - variable-length instructions
  - instructions with side effects
- Stack-based machines
  - push operands onto stack
  - perform operations on operands at top of stack

    - stack kept in registers
    - model for Java Virtual Machine
- Multicore machines

## 4. Primary Tasks of a Code Generator

- Instruction selection
    - Determining factors:
        - level of IR
        - nature of ISA
        - desired quality of generated code
- Register allocation and assignment
    - Register allocation determines the set of variables that will reside in registers at each point in the program.
    - Register assignment determines the specific register in which a variable will reside.
- Evaluation order
    - Some computation orders require fewer registers to hold intermediate results than others.
    - Picking an optimal order in the general case is NP-complete.

## 5. Basic Blocks and Flow Graphs

- A basic block is a maximal sequence of consecutive three-address instructions such that

1. The flow of control can only enter the basic block throught the first instruction in the block.
2. Control will leave the block without halting or branching except possibly at the last instruction in the block.

- A flow graph for the basic blocks of an intermediate program can be constructed as follows:
- The basic blocks are the nodes of the flow graph.
- There is an edge from block B to block C iff it is possible for the first instruction in C to immediately follow the last instruction in B.
- A set of nodes L in a flow graph is a loop if

1. There is a node in L called the loop entry with the property that no other node in L has a predecessor outside L.
2. Every node in L has a nonempty path completely within L to the entry of L.

## 6. Practice Problem

1. ALSU, Exercise 8.4.1, p. 531.

## 7. Reading

- ALSU, Sections 8.1-8.5

---

aho@cs.columbia.edu