Lecture 27:

- Exam-2 Overview
- Language Paradigms
- Models of Computation

Announcements:

- HW-6 out
- Proj Part 2 due
- Extra Credit proposal due Fri
- Exam 2 Mon

Exam 2 Overview

Basics:

- Closed notes, etc.
- Worth 60 points (10% of final grade)
- Four multipart questions

Notes:

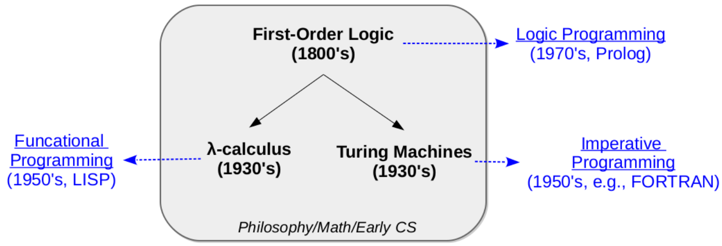
- AST creation and traversal (HW-3)
- Semantic checking (HW-4)
- The MyPL VM and instructions
- Code Generation
- There will be coding questions (in Java)
- Use lecture notes, extended notes, quizzes, homeworks

Programming Language "Paradigms"

There are many ways languages are categorized ...

- script-based, object-oriented, dynamic vs static typing, memory-safety

The three main language "**paradigms**" ... another categorization



Imperative: Say how to solve problem, system carries out steps

Declarative: Say what solution achieves, system determines how to compute it

Logic and Functional languages are considered (more) declarative

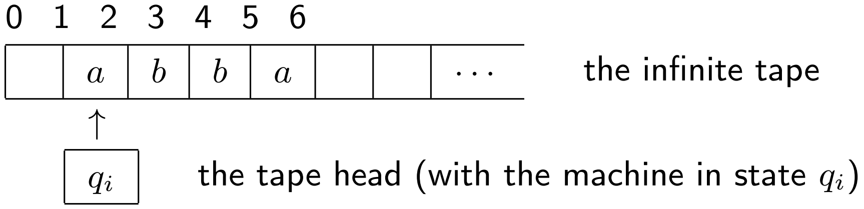
- largely based on underlying **models of computation** used

Turing Machines

Basic Idea: ... even lower level than RAM Machines!

1. ***Infinite tape*** of memory cells, each cell holds one symbol
2. ***Read/write head*** that can move left/right (L/R) one cell at a time
3. ***State register*** that stores the current state of the machine
4. ***Transition table***: Curr State & Symbol $\rightarrow$ New State, Symbol, & Direction

Example: replace a's with b's



Special blank (␣) symbol, and halting states q_{halt}, q_{yes}, q_{no}

Many variants (e.g., multitape machines, non-deterministic)