

Algorithms Overview

Thomas Schwarz, SJ

Algorithms

- Algorithms
 - Named after Persian abu-Ja'far Mohammed ibn-Mūsa al-Khūwārīzmi
 - Description of a problem solving procedure
 - On a generic computation machine
 - Described at various levels:
 - Mathematical procedure
 - Pseudo-code
 - almost implementation: actual OS-independent code

Computability

- Based on models of computation
 - All modern computing systems (including quantum computing) can be simulated by a “normal” computer with “infinite” storage
 - What can be computed currently does not depend on computing technology
 - What can be computed *effectively* does depend on computing technology

Computability

- Turing's Theorem: There are problems that cannot be solved with computation
 - Example: Given a program, can we decide whether the program will stop?
 - Can do it for some programs, but there is no algorithm that will do it for all.

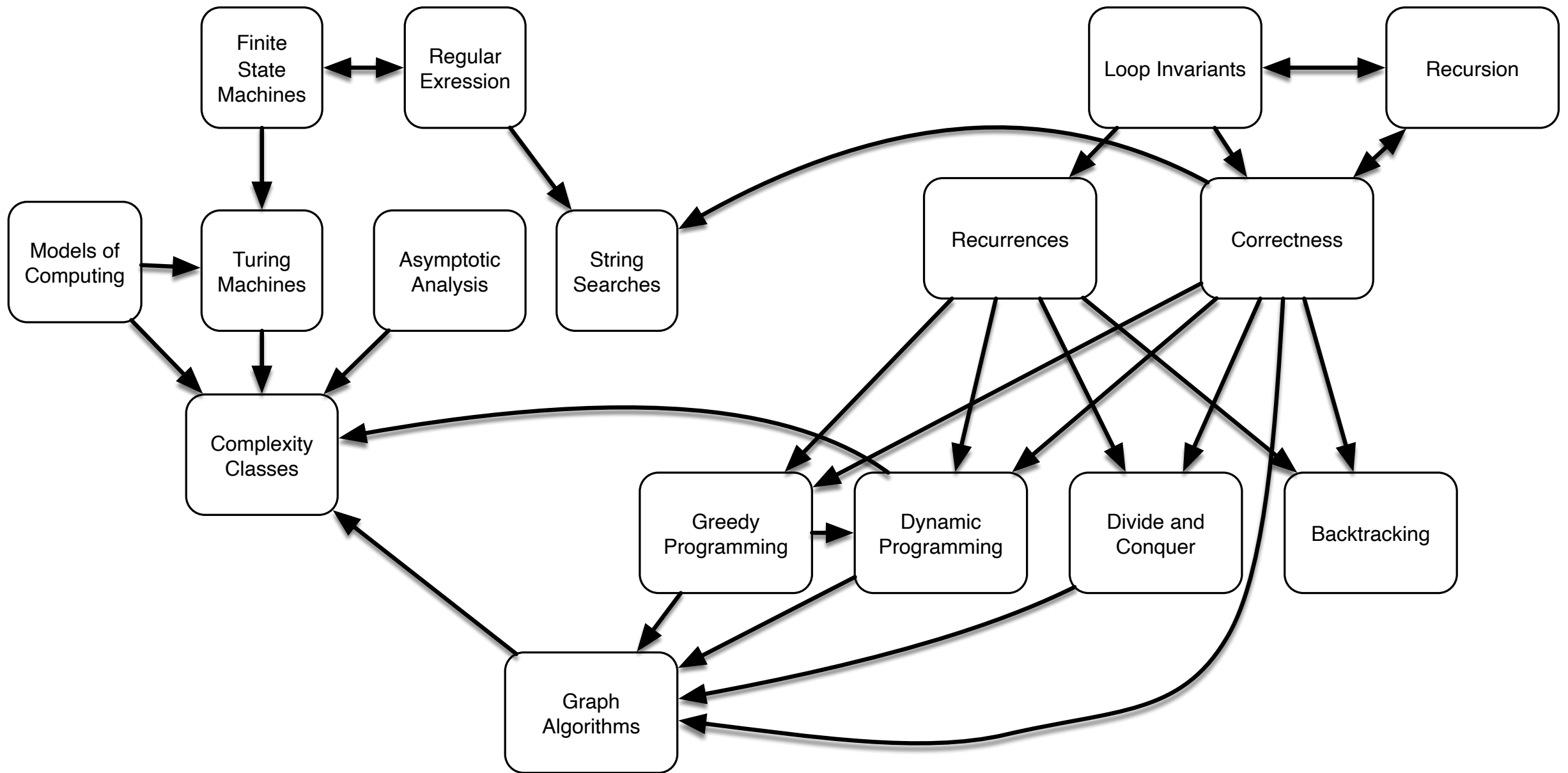
Computability

- Complexity classes
 - Decision or optimization problems
 - Difficulty of a problem measured in instance sizes
 - Two principle measures: space and time (number of instructions)
- Fundamental Unsolved Problem:
 - There are problems whose solution can be checked in a time \leq polynomial in the instance size
 - But cannot be solved in a time \leq polynomial in the instance size

Course Objects

- Capability to design and apply algorithms
 - Done by using programming interview questions
- Capability to analyze algorithms
- Capability to apply discrete mathematics and calculus

Contents



Evaluation

- Group quizzes: Based on previous reading at beginning of class
- Individual quizzes: Based on class contents
- Homeworks & Programming Assignments
 - One correction of submitted homework at a discount
- Midterm
- Final