CSc 135 Course overview
Computing Theory and Programming Languages

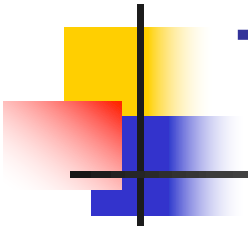
Course overview
Three basic concepts
Set theory: a brief review
Languages: set notation and operations

Course Goals

Course Goals

To provide students with:

1. familiarity with the theoretical foundations of Computer Science.
2. facility with the concepts, notations, and techniques of the theories of automata, formal languages, and Turing machines.
3. understanding of selected programming language features and their implementation.
4. experience using and implementing a recursive-descent parser.
5. experience writing programs using functional and logic language paradigms.

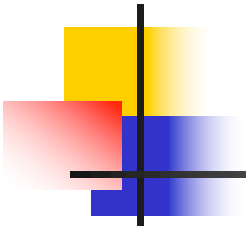


Tests and Exam

- **3 tests** – 25 - 40 min. each, focus on computing theory and key concepts in programming languages

- **exam** – combining Computing Theory (CT) and Programming Languages (PL)

CT Home works and PL Assignments



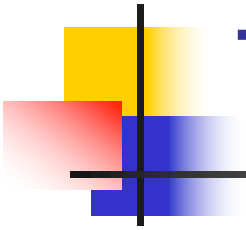
- **Computing Theory Home works**
  - Each CT homework will be discussed and presented in class activity session
- **Programming Language Assignments**
  - Each PL assignment contains a set problems on:
    - (1) Parsing and Recursive-descent parser algorithm
    - (2) functional programming
    - (3) logic programming
    - (4) Web programming Recursive-descent parser
  - **Subset of CT hw and key concepts of PL assignments will be included in tests/exam**

Flipped Classroom Idea -- Students do the learning

- **Students are responsible to:** read textbook, lecture notes, and other course related materials, HW
  - 1 required CT exercises presentation
  - Attend class regularly
  - Participate group discussion
- **Instructor will support** class activities with:
  - Tests
  - Study Guideline and examples solutions of problems
  - Lecture and notes

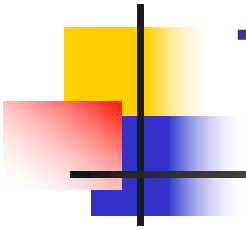
Computing theory: Three Major Concepts - 1

- Languages
  - 4 formal languages
- Language generators
  - 4 grammars (language generators) for 4 languages
- Languages recognizers
  - 4 automata (language recognizers) for 4 languages



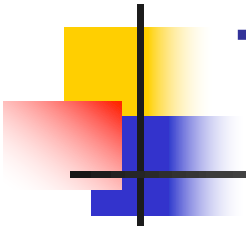
Three Major Concepts - 2

- 4 formal languages
  - Communication vehicle between man and machine
    - Regular Language (RL)
    - Context-Free Language (CFL)
    - Context-Sensitive Language (CSL)
    - Recursively Enumerable Language (REL)



Three Major Concepts - 3

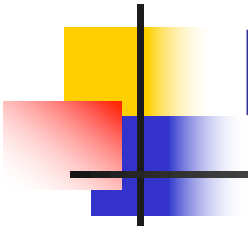
- Language generators – 4 Grammars for 4 languages
  - RG for RL
  - CFG for CFL
  - CSG for CSL
  - Unrestricted Grammar for REL



Three Major Concepts - 4

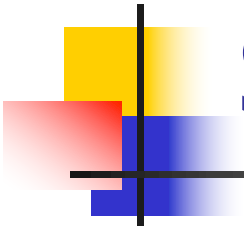
- Language recognizers – 4 automata for 4 languages
  - Finite Automata (FA) for RL
  - Push Down Automata (PDA) for CFL
  - Bounded TM for CSL
  - Turing Machine (TM) for REL

Automata: an abstract model of a digital computer



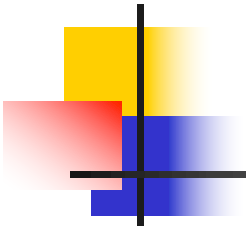
Languages and Abstractions

- Why do we need to study formal languages?
  - Formal language is a powerful abstraction that we can use it to design and implement new programming languages and study other computation problems
- What is an abstraction?
  - A method/language for working with one aspect of a design
  - All abstractions have limitations



Set Theory – a brief review

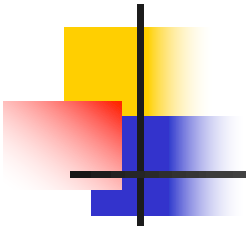
- Sets, subsets, empty set, multiset
- Cartesian product of A and B
- Power set of A:
  - 2^A -- set of all subsets of A
- Cardinality of A: size of set A
- Sets operations:
  - Union, intersection, difference, universal set



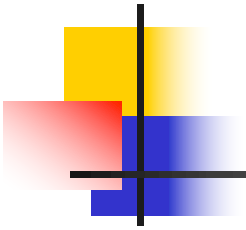
Languages in Set Notation - 1

- **Alphabet**: a finite, nonempty set S of symbols
- **Strings**: finite sequences of symbols from S
- **Language**: a set of strings from S

Languages in Set Notation - 2



- **S^*** - set of strings obtained by concatenating zero or more symbols from S
- Empty string λ and empty set
- **S^+** = S^* - {λ}
- **Language**: a subset of S^*
- Example: S = {a, b}. What are S^* and S^+

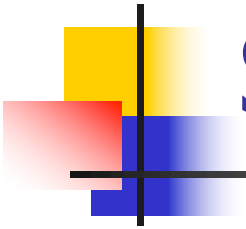


Language Operations - 1

- A language L is a set
- set operations => language operations
- Complement of L, S^* - L
- Concatenation (product) of languages
  - $L_1 L_2 = \{xy: x \text{ from } L_1, y \text{ from } L_2\}$
  - Example of product when
    - $L_1 = \{0, 1\}$ and $L_2 = \{ a, b\}$
    - $L_1 L_2$ =?

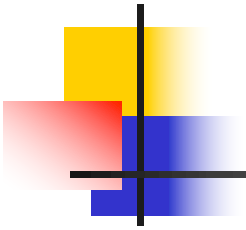
Language Operations - 2

- Power of L
  - $L^0 = \{\lambda\}$
  - $L^n = L \ \ L^{n-1}$
  - Example of power: given $L = \{a^n b^n : n \geq o?\}$
  - What is L^2?



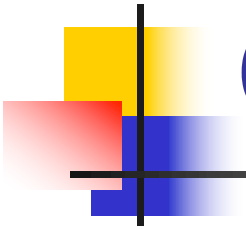
Summary

- See Class website for latest details:
  - http://gaia.ecs.csus.edu/~mei/135/index.html
- To do well in this class, you need to
  - Read before/after class and attend class
  - Review lecture notes and HW before activities and tests
  - Do homework and participate in discussions
  - Ask questions in class and office hours



Homework

- Review set theory,
  - a quiz on set theory **next class**
- Read chapter 1 and 2 of (text or notes)
- Free PDF is available online



Quiz scope on set theory

- Subsets, notation and concepts
- Member of a set
- Empty set
- Power set
- Set operations: Product, union, intersection, difference