

Compsci 334 - Mathematical Foundations of CS  
Dr. Susan Rodger  
Section: Introduction (Ch. 1) (handout)

**Announcements:**

- This is a math course with applications. Prereq: Compsci 230 or equivalent, CompSci 201.
- Course web page:  
www.cs.duke.edu/courses/spring22/compsci334  
Familiarize yourself with all parts of the web page. All lecture notes, assignments and resources will be on this page.
- Read Chapter 1 in the Linz book for next time.
- Complete the reading quiz on Sakai. We will have reading quizzes due almost every class period. They must be completed before the next class period starts and will be based on assigned reading for that date. They turn off automatically when class starts. Sakai will be used mostly for reading quizzes and also for posting grades.
- Course bulletin board: Ed Discussion.  
You will be automatically joined to the course bulletin board where you can ask questions about the course.

**What will we do in Compsci 334?**

**Questions**

- Can you write a program to determine if a string is an integer?  
9998.89                                  Sabab                                  789342
- Can you do this if your machine had no additional memory other than the program? (can't store any values and look at them again)
- Can you write a program to determine if the following are correct arithmetic expressions?

$$((34 + 7 * (18/6)))$$

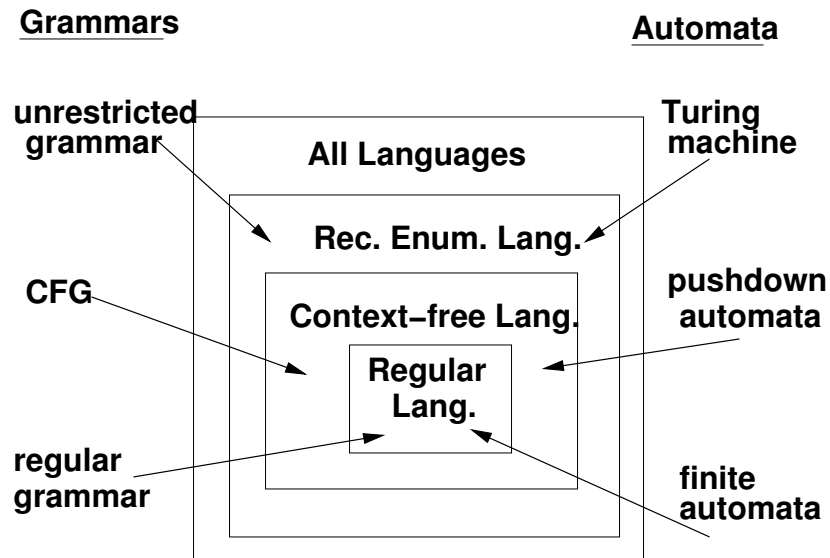
$$(((((((a + b) + c) * d(e + f))))))$$

- Can you do this if your machine had no additional memory other than the program?
- Can you write a program to determine the **value of** the following expression?

$$((34 + 7 * (18/6)))$$

- Can you write a program to determine if a file is a valid Java program?
- Can you write a program to determine if a Java program given as input will ever halt?

## Language Hierarchy



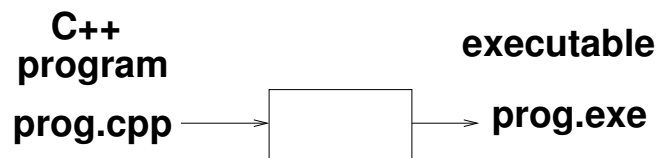
## Power of Machines

automata	Can do?	Can't do?
finite automata (FA) (no memory)	integers	arith expr
pushdown automata (PDA) (only memory is stack)	arith expr	compute expr
Turing machines (TM) (infinite memory)	compute expr	decide if halts

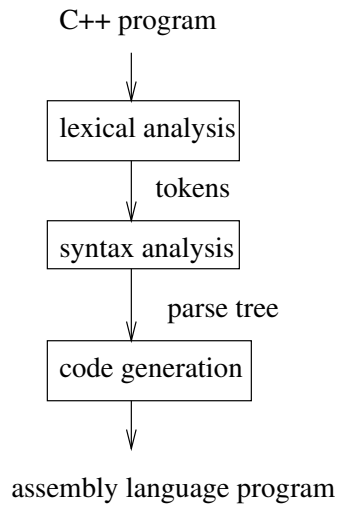
## Application

### Compiler

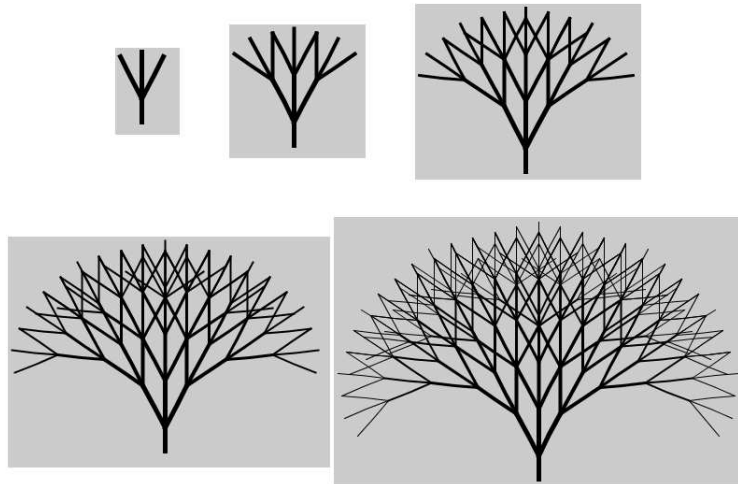
- Our focus - Question: Given a program in some language (say Java or C++) - is it valid?
- Question: language L, program P - is P valid?
- Other things to consider, how is the compilation process different for different programming languages? (Java vs C++?)



## Stages of a Compiler



## L-Systems - Model the Growth of Plants



**Set Theory** - Read Chapter 1 Linz.

A Set is a collection of elements.

$A = \{1, 4, 6, 8\}$ ,  $B = \{2, 4, 8\}$ ,  $C = \{3, 6, 9, 12, \dots\}$ ,  $D = \{4, 8, 12, 16, \dots\}$

- (union)  $A \cup B =$
- (intersection)  $A \cap B =$
- $C \cap D =$
- (member of)  $42 \in C?$
- (subset)  $B \subset C?$
- $B \cap A \subseteq D?$
- $|B| =$

- (product)  $A \times B =$
- $|A \times B| =$
- $\emptyset \in B \cap C?$
- (powerset)  $2^B =$

**Example** What are all the subsets of  $\{3, 5\}$ ?

How many subsets does a set  $S$  have?

$ S $	number of subsets
0	
1	
2	
3	
4	

How do you prove? Set  $S$  has  $2^{|S|}$  subsets.

Technique: Proof by Induction

1. Basis:  $P(1)$ ? Prove smallest instance is true.
2. Induction Hypothesis - I.H.  
Assume  $P(n)$  is true for  $1, 2, \dots, n$
3. Induction Step - I.S.  
Show  $P(n+1)$  is true (using I.H.)

**Proof of Example:**

1. Basis:
2. I.H. Assume
3. I.S. Show

**Ch. 1: 3 Major Concepts**

- languages
- grammars
- automata

## Languages

- $\Sigma$  - set of symbols, alphabet
- string - finite sequence of symbols
- language - set of strings defined over  $\Sigma$

alphabet  $\Sigma$

## Examples

- $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$   
 $L = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, \dots\}$
- $\Sigma = \{a, b, c\}$   
 $L = \{ab, ac, cabb\}$
- $\Sigma = \{a, b\}$   
 $L = \{a^n b^n \mid n > 0\}$

## Notation

- symbols in alphabet: a, b, c, d, ...
- string names: u, v, w, ...

## Definition of concatenation

Let  $w = a_1 a_2 \dots a_n$  and  $v = b_1 b_2 \dots b_m$

Then  $w \circ v$  OR  $wv =$

See book for formal definitions of other operations.

## String Operations

strings:  $w = abbc$ ,  $v = ab$ ,  $u = c$

- size of string  
 $|w| + |v| =$
- concatenation  
 $v^3 = vvv = vovov =$
- $v^0 =$
- $w^R =$
- $|vv^R w| =$
- $ab \circ \lambda =$

## Definition

$\Sigma^*$  = set of strings obtained by concatenating 0 or more symbols from  $\Sigma$

## Example

$$\Sigma = \{a, b\}$$

$$\Sigma^* =$$

$$\Sigma^+ =$$

## Examples

$$\Sigma = \{a, b, c\}, L_1 = \{ab, bc, aba\}, L_2 = \{c, bc, bcc\}$$

- $L_1 \cup L_2 =$
- $L_1 \cap L_2 =$
- $\overline{L_1} =$
- $\overline{L_1 \cap L_2} =$
- $L_1 \circ L_2 = \{xy \mid x \in L_1 \text{ and } y \in L_2\} =$

## Definition

$$L^0 = \{\lambda\}$$

$$L^2 = L \circ L$$

$$L^3 = L \circ L \circ L$$

$$L^* = L^0 \cup L^1 \cup L^2 \cup L^3 \dots$$

$$L^+ = L^1 \cup L^2 \cup L^3 \dots$$

## Grammars

### Grammar for english

<sentence>  $\rightarrow$  <subject><verb><d.o.>

<subject>  $\rightarrow$  <noun> | <article><noun>

<verb>  $\rightarrow$  hit | ran | ate

<d.o.>  $\rightarrow$  <article><noun> | <noun>

<noun>  $\rightarrow$  Fritz | ball

<article>  $\rightarrow$  the | an | a

### Examples (derive a sentence)

Fritz hit the ball.

```

<sentence> -> <subject><verb><d.o>
            -> <noun><verb><d.o>
            -> Fritz <verb><d.o.>
            -> Fritz hit <d.o.>
            -> Fritz hit <article><noun>
            -> Fritz hit the <noun>
            -> Fritz hit the ball

```

Can we also derive the sentences?

The ball hit Fritz.

The ball ate the ball

Syntactically correct?

Semantically correct?

### Grammar

$G=(V,T,S,P)$  where

- V - variables (or nonterminals)
- T - terminals
- S - start variable ( $S \in V$ )
- P - productions (rules)
  - $x \rightarrow y$  “means” replace x by y
  - $x \in (V \cup T)^+, y \in (V \cup T)^*$
  - where V, T, and P are finite sets.

### Definition

$w \Rightarrow z$             w derives z  
 $w \xRightarrow{*} z$             derives in 0 or more steps  
 $w \xRightarrow{+} z$             derives in 1 or more steps

### Definition

$G=(V,T,S,P)$

$L(G)=\{w \in T^* \mid S \xRightarrow{*} w\}$

**Example**

$G = (\{S\}, \{a,b\}, S, P)$

$P = \{S \rightarrow aaS, S \rightarrow b\}$

$L(G) =$

**Example**

$L(G) = \{a^n ccb^n \mid n > 0\}$

$G =$

**Example**

$G = (\{S\}, \{a,b\}, S, P)$

$P = \{S \rightarrow aSb, S \rightarrow SS, S \rightarrow ab\}$

Which of these strings *aabb*, *abab*, *abba*, *babab* can be generated by this grammar? Show the derivations.

$L(G) =$

**Automata** Abstract model of a digital computer

