# Compsci 334 - Mathematical Foundations of CS <br> Dr. Susan Rodger <br> 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/spring21/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: Piazza

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

8abab
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?

$$
\begin{gathered}
((34+7 *(18 / 6))) \\
(((((((a+b)+c) * d(e+f)))))
\end{gathered}
$$

- 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?


Power of Machines

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

## Application

## Compiler

- Our focus - Question: Given a program in some language (say Java or $\mathrm{C}++$ ) - 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 $\mathrm{C}++$ ?)



## Stages of a Compiler



L-Systems - Model the Growth of Plants


Set Theory - Read Chapter 1 Linz.
A Set is a collection of elements.
$\mathrm{A}=\{1,4,6,8\}, \mathrm{B}=\{2,4,8\}, \mathrm{C}=\{3,6,9,12, \ldots\}, \mathrm{D}=\{4,8,12,16, \ldots\}$

- (union) $\mathrm{A} \cup \mathrm{B}=$
- (intersection) $\mathrm{A} \cap \mathrm{B}=$
- $\mathrm{C} \cap \mathrm{D}=$
- (member of) $42 \in \mathrm{C}$ ?
- (subset) $\mathrm{B} \subset \mathrm{C}$ ?
- $\mathrm{B} \cap \mathrm{A} \subseteq \mathrm{D}$ ?
- $|\mathrm{B}|=$
- (product) $\mathrm{A} \times \mathrm{B}=$
- $|\mathrm{A} \times \mathrm{B}|=$
- $\emptyset \in \mathrm{B} \cap \mathrm{C}$ ?
- $($ 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: $\mathrm{P}(1)$ ? Prove smallest instance is true.
2. Induction Hypothesis - I.H.

Assume $\mathrm{P}(\mathrm{n})$ is true for $1,2, \ldots, \mathrm{n}$
3. Induction Step - I.S.

Show $\mathrm{P}(\mathrm{n}+1)$ 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

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


## Notation

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


## Definition of concatenation

Let $\mathrm{w}=a_{1} a_{2} \ldots a_{n}$ and $\mathrm{v}=b_{1} b_{2} \ldots b_{m}$
Then $w \circ v$ OR wv=
See book for formal definitions of other operations.

## String Operations

strings: $w=a b b c, v=a b, u=c$

- size of string

$$
|w|+|v|=
$$

- concatenation
$v^{3}=\mathrm{vvv}=\mathrm{vovov}=$
- $v^{0}=$
- $w^{R}=$
- $\left|v v^{R} w\right|=$
- $\mathrm{ab} \circ \lambda=$


## Definition

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

## Example

$\Sigma=\{a, b\}$
$\Sigma^{*}=$
$\Sigma^{+}=$

## Examples

$\Sigma=\{a, b, c\}, L_{1}=\{a b, b c, a b a\}, L_{2}=\{c, b c, b c c\}$

- $L_{1} \cup L_{2}=$
- $L_{1} \cap L_{2}=$
- $\overline{L_{1}}=$
- $\overline{L_{1} \cap L_{2}}=$
- $L_{1} \circ L_{2}=\left\{x y \mid x \in L_{1}\right.$ and $\left.y \in L_{2}\right\}=$


## Definition

$L^{0}=\{\lambda\}$
$L^{2}=L \circ L$
$L^{3}=L \circ L \circ L$
$L^{*}=L^{0} \cup L^{1} \cup L^{2} \cup L^{3} \ldots$
$L^{+}=L^{1} \cup L^{2} \cup L^{3} \ldots$

## Grammars

## Grammar for english

```
<sentence> -> <subject><verb><d.o.>
<subject> -> <noun> | <article><noun>
<verb> }->\mathrm{ hit | ran | ate
<d.o.> -> <article> <noun> | <noun>
<noun> }->\mathrm{ Fritz | ball
<article> }->\mathrm{ 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

$\mathrm{G}=(\mathrm{V}, \mathrm{T}, \mathrm{S}, \mathrm{P})$ where

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


## Definition

$\mathrm{w} \Rightarrow \mathrm{z} \quad \mathrm{w}$ derives z
$\mathrm{w} \stackrel{*}{\Rightarrow} \mathrm{z} \quad$ derives in 0 or more steps
$\mathrm{w} \stackrel{+}{\Rightarrow} \mathrm{z} \quad$ derives in 1 or more steps
Definition
$\mathrm{G}=(\mathrm{V}, \mathrm{T}, \mathrm{S}, \mathrm{P})$
$\mathrm{L}(\mathrm{G})=\left\{\mathrm{w} \in T^{*} \mid \mathrm{S} \xlongequal{*} \mathrm{w}\right\}$

## Example

$\mathrm{G}=(\{\mathrm{S}\},\{\mathrm{a}, \mathrm{b}\}, \mathrm{S}, \mathrm{P})$
$\mathrm{P}=\{\mathrm{S} \rightarrow \mathrm{aaS}, \mathrm{S} \rightarrow \mathrm{b}\}$
$\mathrm{L}(\mathrm{G})=$

## Example

$\mathrm{L}(\mathrm{G})=\left\{a^{n} c c b^{n} \mid n>0\right\}$
$G=$

## Example

$\mathrm{G}=(\{\mathrm{S}\},\{\mathrm{a}, \mathrm{b}\}, \mathrm{S}, \mathrm{P})$
$\mathrm{P}=\{\mathrm{S} \rightarrow \mathrm{aSb}, \mathrm{S} \rightarrow \mathrm{SS}, \mathrm{S} \rightarrow \mathrm{ab}\}$
Which of these strings $a a b b, a b a b, a b b a, b a b a b$ can be generated by this grammar? Show the derivations.
$\mathrm{L}(\mathrm{G})=$

Automata Abstract model of a digital computer


