Announcements:

- **This is a math course with systems applications. Prereq: CompSci 201, Compsci 230 or equiv.**
- **Course web page:**
  www.cs.duke.edu/courses/
  fall24/compsci334
  Familiarize yourself with all parts of the web page.
- **Flipped class - reading/quizzes BEFORE**
- **Read Chapter 1 in the Linz/Rodger book for next time.**
- **Complete the reading quizzes on Canvas before class.**
  (Due to Drop/add, QZ01-QZ05 turn off Sept 10, 11:45am!!)
- **Course bulletin board: Ed Discussion (get to from Canvas)**

- **Course participation required!**

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

  yes

- **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)**

  yes

- **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?**

NO

- **Can you write a program to determine the value of the following expression?**

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

yes

- **Can you write a program to determine if a file is a valid Java program?**

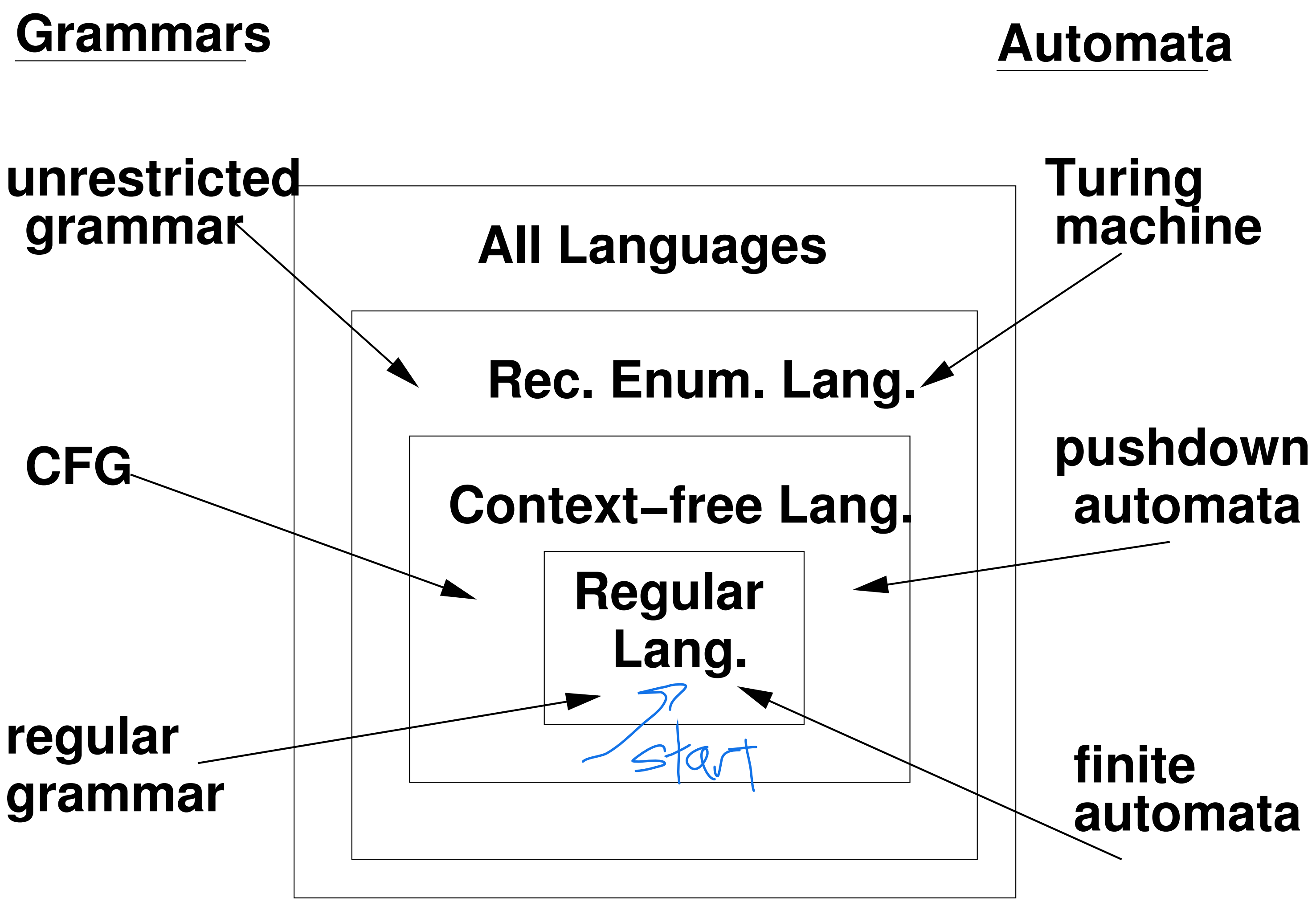
yes, what a compiler does?

or interpreter

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

no, hard!

Language Hierarchy



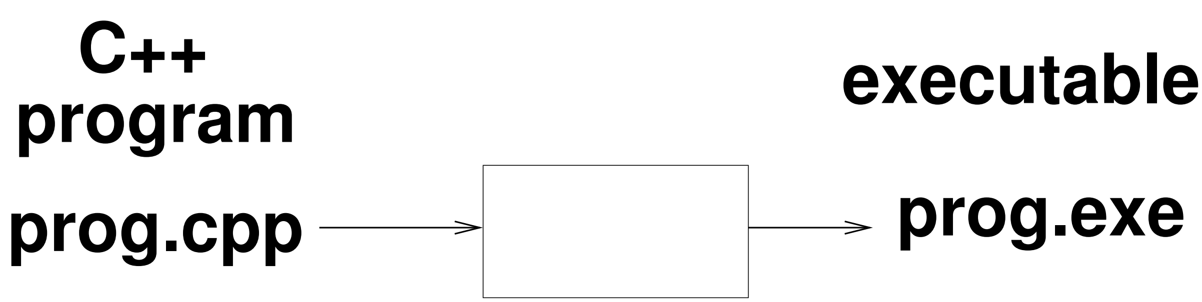
Power of Machines

automata	Can do?	Can't do?
FA (no memory)	recognize integer	recognize arith expr
PDA (stack)	recognize arith expr	compute arith expr
TM (infinite) memory	compute arith expr	decide if prog will halt

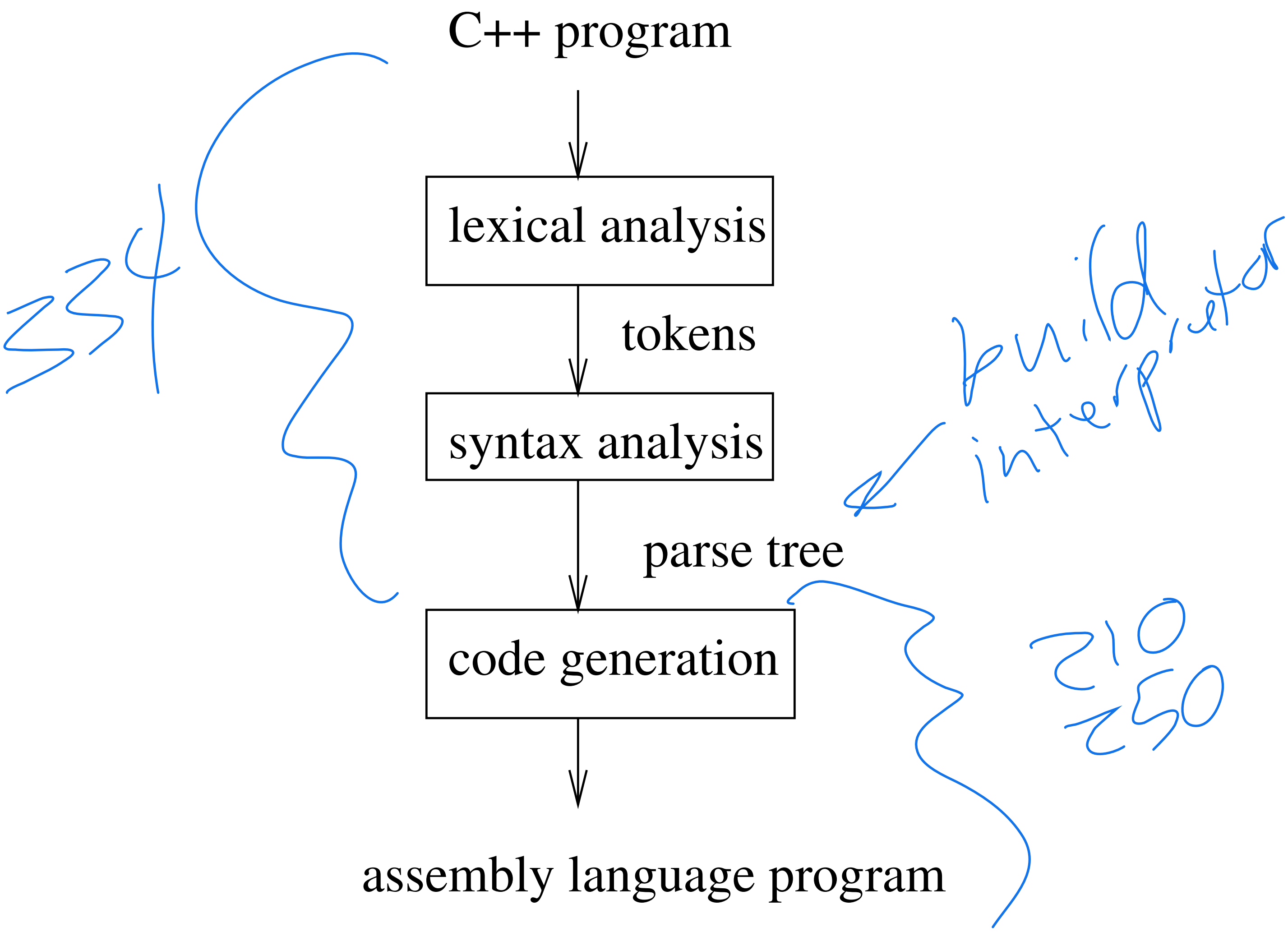
Application

Compiler

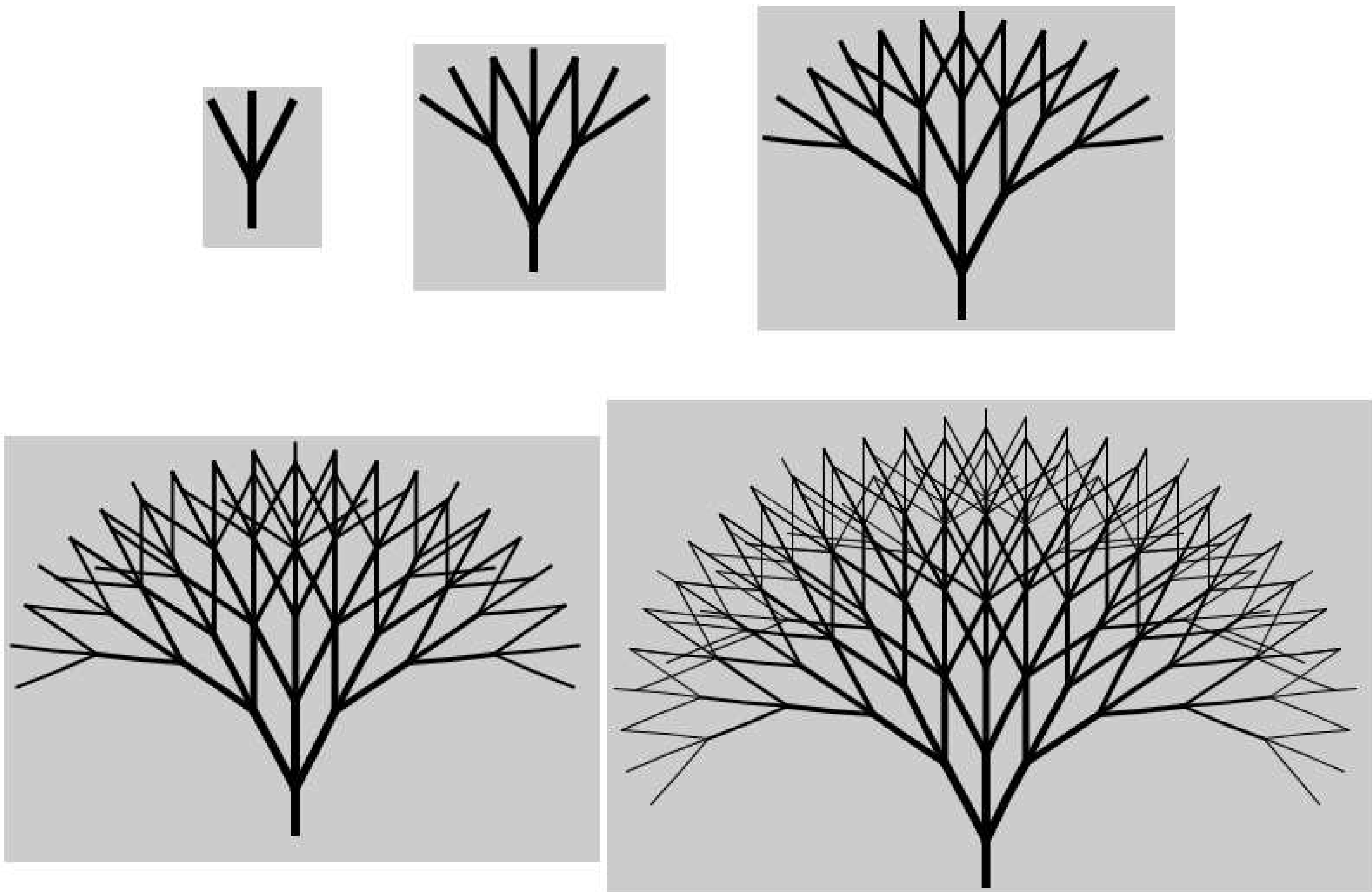
- **Our focus - Question: Given a program in some language (say Java or C++) - is it valid?**
- **Question: language L, program P - is P valid?**
- **Other things to consider, how is the compilation process different for different programming languages? (Java vs C++?)**

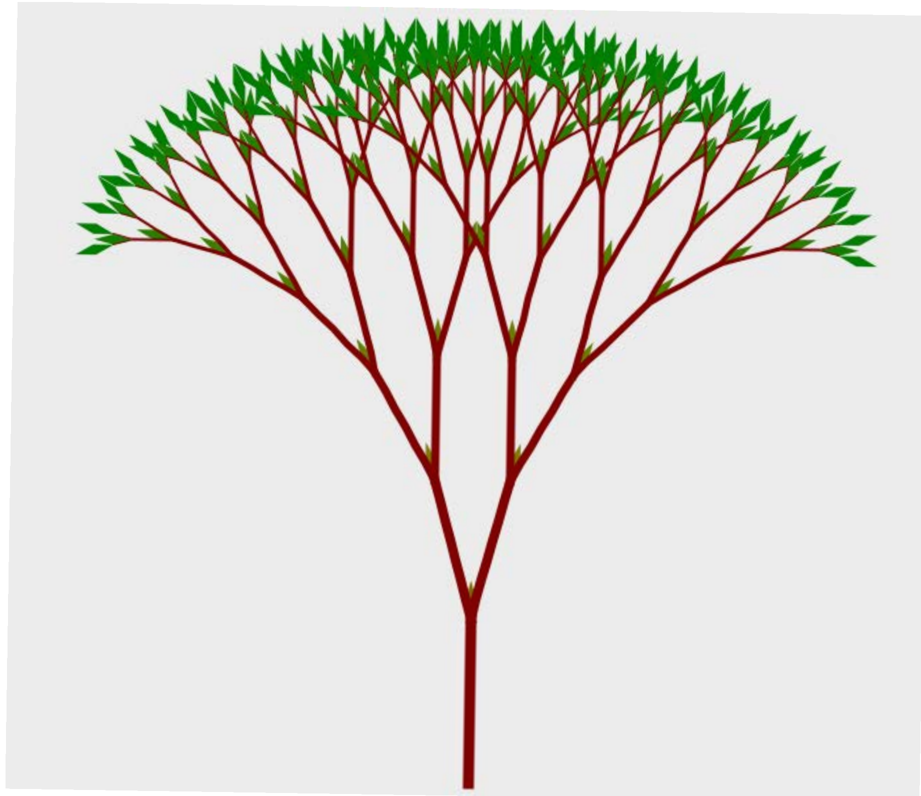


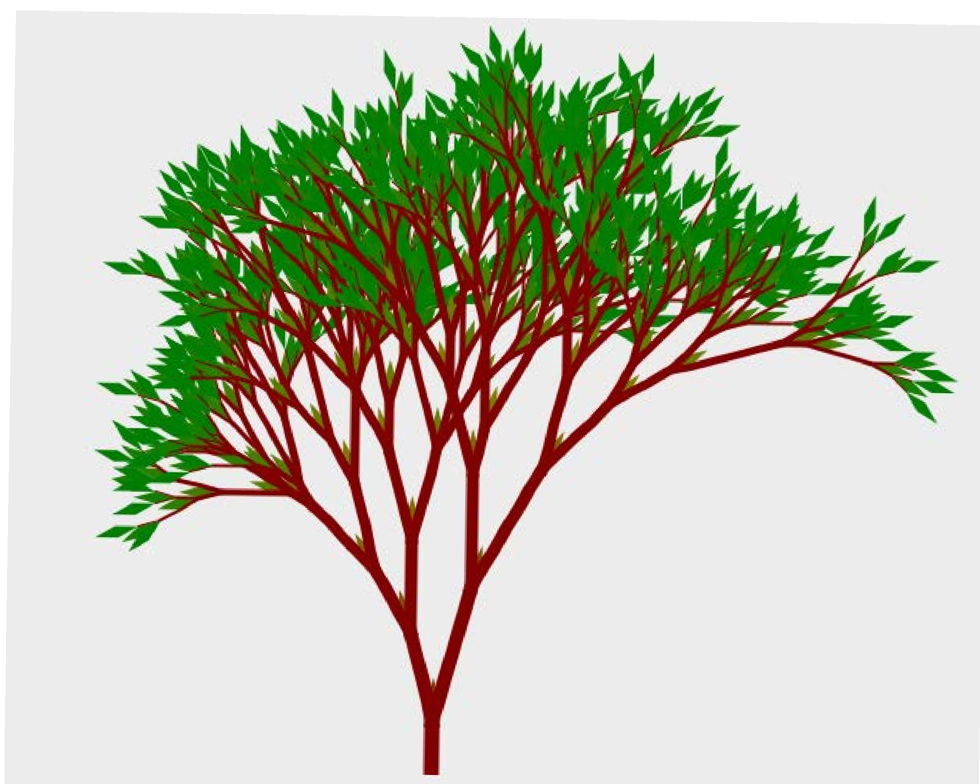
Stages of a Compiler



L-Systems - Model the Growth of Plants







Chapter 1 - Set Theory

A Set is a collection of elements.

A={1,4,6,8}, B={2,4,8},

C={3,6,9,12,...}, D={4,8,12,16,...}

- **(union) A∪B=** {1,2,4,6,8}
- **(intersection) A∩B=** {4,8}
- **C∩D=** {12,24,...} or {12*x | x>0}
- **(member of) 42 ∈ C?** yes
- **(subset) B⊂C?** no
- **B∩A ⊆D?** yes
- **|B|=** 3
- **(product) A×B=** {(1,2),(1,4)(1,8),(4,2), (4,4),(4,8) ... (8,8)} — 12 tuples
- **|A×B|=** 12
- **∅ ∈ B∩C?** NO — B∩C = { } ← nothing in there
- **(powerset)** 2^B**=**

2^B = {∅, {2},{4},{8}, {2,4},{2,8}, {4,8},{2,4,8}}

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

$\{\emptyset, \{3\}, \{5\}, \{3,5\}\}$

How many subsets does a set S have?

$\lvert S \rvert$	**number of subsets**
0	1
1	2
2	4
3	8
4	16

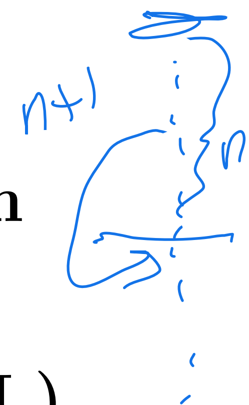
How do you prove? Set S has $2^{|S|}$ subsets.

Technique: Proof by Induction

1. **Basis: P(1)?**
2. **I.H.**

   Assume P(n) is true for 1,2,...,n
3. **I.S.**

   Show P(n+1) is true (using I.H.)



Set S has $2^{|S|}$ subsets.

Proof:

1. Basis: $|S|=0$ has 1 element
$2^0=1$ checked

2. I.H. Assume $2^{|S|}$ is equal to the number of subsets in S for all $|S|\leq n$

3. I.S. Show for $|S|=n+1$ that there are 2^{n+1} subsets

Take one element out of S
$S=T\cup\{a\}$
by I.H. T has 2^n subsets
S has all the subsets in T plus a copy of each subset with a
or 2 * number of subsets T has
$2*2^n=2^{n+1}$ ✓

Ch. 1: 3 Major Concepts

- **languages**
- **grammars**
- **automata**

Languages

- Σ - set of symbols, alphabet
- string - finite sequence of symbols
- language - set of strings defined over Σ

alphabet Σ

Examples

- $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
  $\mathbf{L}=\{0, 1, 2, ..., 12, 13, 14, ...\}$
- $\Sigma=\{a, b, c\}$
  $\mathbf{L}=\{ab, ac, cabb\}$
- $\Sigma=\{a, b\}$
  $\mathbf{L}=\{a^n b^n \mid n > 0\}$ $= \{ab, aabb, a^3b^3, ...$

  $\{ (^n)^n \}$ kinda like that

Stopped here

Notation

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

Definition of concatenation

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

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

String Operations

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

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

Definition

Σ^* concatenate 0 or more

Example

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

$\Sigma^* =$

$\Sigma^+ =$

Examples

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

- $L_1 \cup L_2 =$
- $L_1 \cap L_2 =$
- $\overline{L_1} =$
- $\overline{L_1 \cap L_2} =$
- $L_1 \circ L_2 = \{xy \mid x \in L_1$ **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> →
<subject><verb><d.o.>

<subject> → <noun> |
<article><noun>

<verb> → hit | ran | ate

<d.o.> → <article><noun> | <noun>

<noun> → Fritz | ball

<article> → 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 x$\in$(V$\cup$ T)$^+$, y$\in$(V$\cup$T)*

Definition

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

Definition of Language of a grammar - L(G)

G=(V,T,S,P)

L(G)={w$\in T^*$ | S $\stackrel{*}{\Rightarrow}$ w}

Example

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

P={S→ aaS, S→ b}

L(G)=

Example

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

G =

Example

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

P={S→ aSb, S→ SS, S→ 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