San José State University
Department of Computer Science

Ahmad Yazdankhah
ahmad.yazdankhah@sjsu.edu
www.cs.sjsu.edu/~yazdankhah

Grammars

(Part 2)

Lecture 22

Day 26/31

CS 154

Formal Languages and Computability

Spring 2019

Agenda of Day 26

- Solution and Feedback of Quiz 8
- Summary of Lecture 21
- Lecture 22: Teaching ...
  - Grammars (Part 2)

Solution and Feedback of Quiz 8 (Out of 20)

Section	Average	High Score	Low Score
01 (TR 3:00 PM)	16.37	20	12
02 (TR 4:30 PM)	16.65	20	12
03 (TR 6:00 PM)	16.57	20	9.5

Summary of Lecture 21: We learned ...

Grammars

- We were looking for a more powerful and practical tool to represent formal languages.
- Roughly speaking, a set of production rules is called grammar.
- A sentence is well-formed if ...
  - ... we can derive it from the grammar.
- Associated language to the grammar G is ...
  - ... the set of all strings generated by it.
  - ... denoted by L(G).

Any Question

Definitions

Formal Definition of Grammar

- A grammar G is defined by the quadruple:

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

- Where:
  - V is a nonempty finite set of variables.
  - T is a nonempty finite set of symbols (aka terminals) called terminal alphabet.
  - $S \in V$ is a special symbol called start variable.
  - P is a finite set of production rules (or simply rules) of the form

    $xAy \rightarrow z$

    where:

    $A \in V$ and $x, y, z \in (T \cup V)^*$

- Note that in this course, we'd always have only one variable in LHS.

Formal Definition of Grammar

$$G = (\overset{\text{Set of Variables}}{V}, \underset{\text{Set of Terminals}}{T}, \overset{\text{Start Variable}}{S}, \underset{\text{Set of Production Rules}}{P})$$

Formal Definition of Grammar: Example

Example 13

- As we saw before, the following grammar

  $S \to aSb \mid \lambda$

  generates the language $L = \{a^n b^n : n \geq 0\}$.

- Write V, T, Starting variable, and P.

Solution

$V = \{S\}$

$T = \{a, b\}$

Start variable: $S \in V$

$P = \{S \to aSb \; , \; S \to \lambda\}$

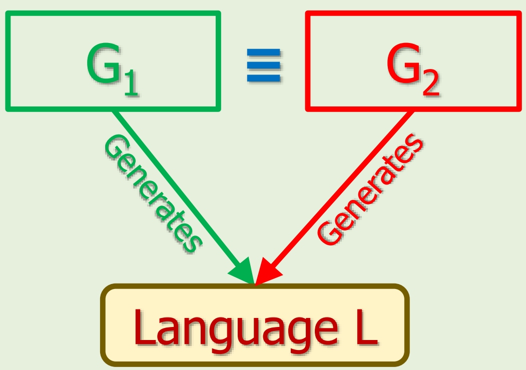
Equivalency of Grammars

- A given language can be generated by many grammars.

Definition

- Two grammars G_1 and G_2 are equivalent iff both has the same associated language.

$$G_1 \equiv G_2 \leftrightarrow L(G_1) = L(G_2)$$



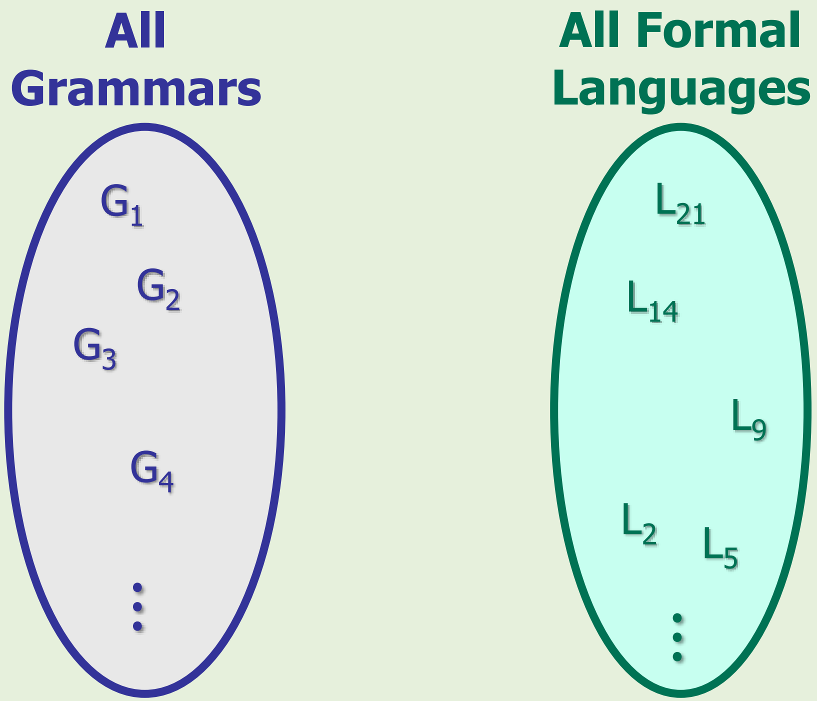
Grammars and Languages Association

Grammars and Languages Association

- What is the relationship between:

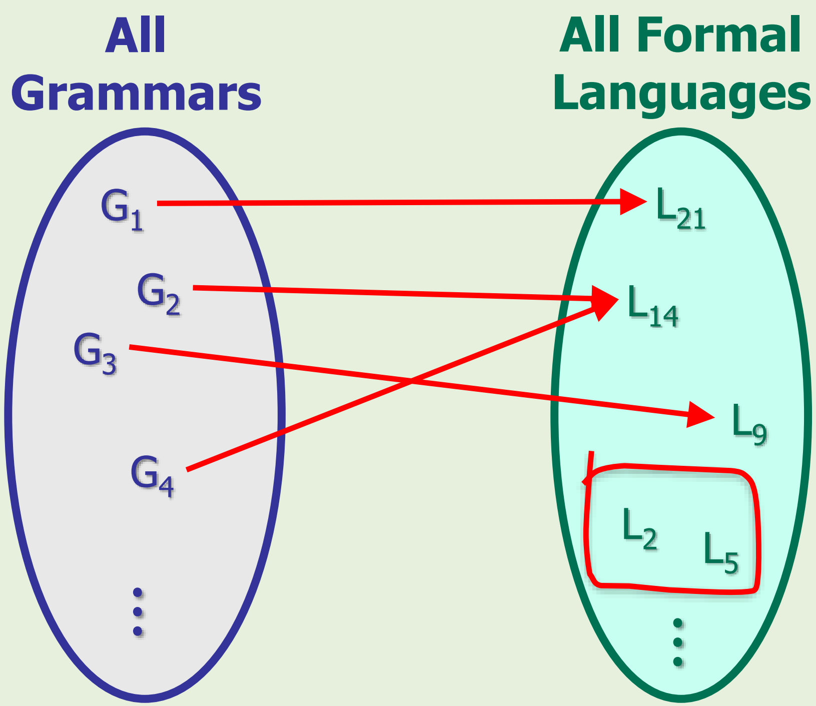
  the set of Grammars, and

  the set of all formal languages?



Grammars and Languages Association

- You agree that "every grammar represents a language".
- BUT we don't know yet whether we can represent every language, by a grammar or not!
  - Our knowledge is not enough yet.



Types of Grammars

Linear Grammars

Definition

- A grammar G is linear if the right hand side of every production rule has at most one variable.

$A \to w \mid w\,B\,u$

Where $A, B \in V$ and $w, u \in T^*$

Example 14

- Is the following grammar linear?

$S \to A$

$A \to baBb \mid \lambda$

$B \to Abb$

- Yes, because all production rules have at most one variable in the RHS.

- Note that in this course, we'd always have only one variable in LHS.

Right-Linear Grammars

Definition

- A linear grammar is said to be right-linear if all production rules are of the form:

$A \rightarrow w \mid u\,B$

Where $A, B \in V$ and $w, u \in T^*$

- In the case of $A \rightarrow w$, we consider $A \rightarrow wB^0$.

Example 15

- Is the following grammar right-linear?

  $S \rightarrow abS \mid a$

- Yes, it is right-linear.

Left-Linear Grammars

Definition

- A linear grammar is said to be left-linear if all production rules are of the form:

$A \rightarrow w \mid B\,u$

Where $A, B \in V$ and $w, u \in T^*$

- In the case of $A \rightarrow w$, we consider $A \rightarrow B^0 w$.

Example 16

- Is the following grammar left-linear?

  $S \rightarrow Aab$

  $A \rightarrow Bab \mid B$

  $B \rightarrow a$

- Yes, it is left-linear.

Regular Grammars

Definition

- A grammar is said to be regular if it is either right-linear or left-linear.

Example 17

- Is the following grammar regular?

  $S \rightarrow A$

  $A \rightarrow aB \mid \lambda$

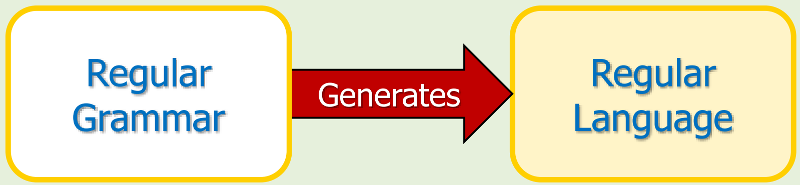
  $B \rightarrow Ab$

- It is NOT regular because it is neither right-linear nor left-linear.

Regular Grammars and Regular Languages

Theorem

- Let G be a regular grammar, then $L(G)$ is a regular language over T.

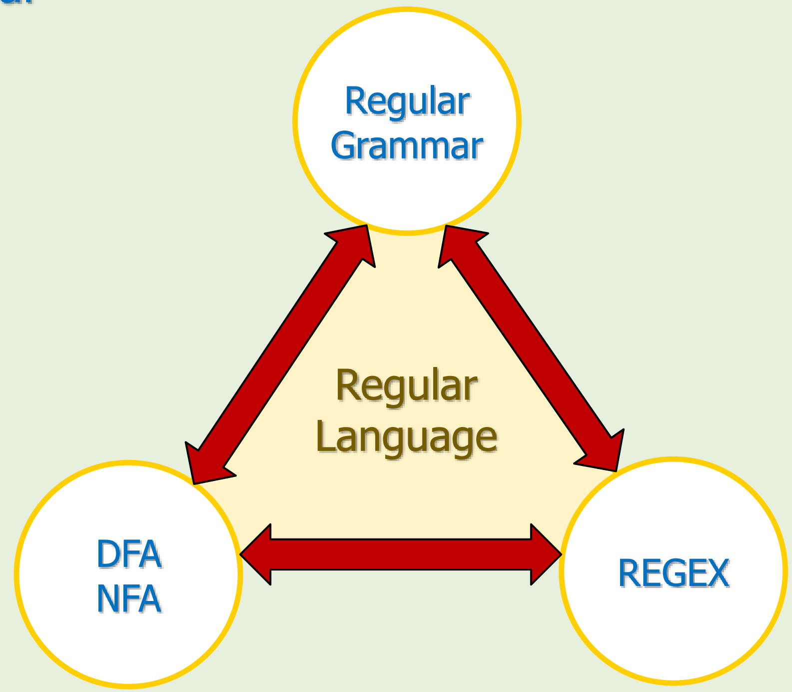


Theorem

- Let L be a regular language over Σ.
  Then there exists a regular grammar G such that $L = L(G)$.

Regular Languages Representations

- Now, we have three ways for representing Regular Languages:
  - DFA / NFA
  - REGEX
  - Regular Grammar



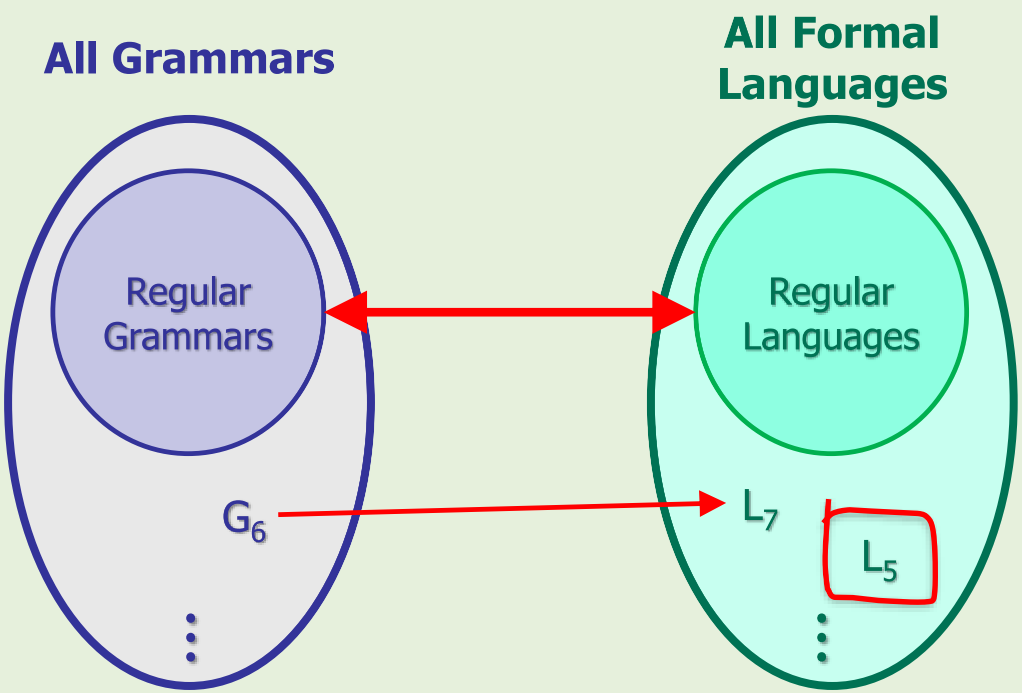
Grammars and Languages Association

Revisited

- We've already known that "every grammar represents a language".
- At this moment we know that:

Regular grammars represent regular languages.

Every regular language can be represented by a regular grammar.



Context-Free Grammars (CFG)

Context-Free Grammars (CFGs)

Definition

- A grammar G is said to be context-free (CFG) if all production rules are of the form:

- Note again that:
  In this course, LHS has always one variable.

$A \rightarrow v$

Where $A \in V$ and $v \in (V \cup T)^*$

Example 18

- Is the following grammar context-free?

  $S \rightarrow a S b \mid \lambda$

- Yes, it is a context free grammar.

CFGs Examples

Example 19

- Let the grammar G be:

  $S \rightarrow a\,S\,a \mid b\,S\,b \mid \lambda$

1. Is G context-free?
2. L(G) = ? // show it by a set-builder.

Solution

CFGs Examples

Example 20

- Let the grammar G be:

  $S \rightarrow S\,S \mid a\,S\,b \mid b\,S\,a \mid \lambda$

1. Is G context-free?
2. L(G) = ? // show it by a set-builder.

Solution

- What would happen if:

  a = (

  b =)

Context-Free Languages (CFL)

Definition

- A language L is said to be context-free (CFL) iff there exists a context-free grammar G such that $L = L(G)$.
  - In other words, CFGs generates CFLs, and
  - ... for every CFL, we can create a CFG.

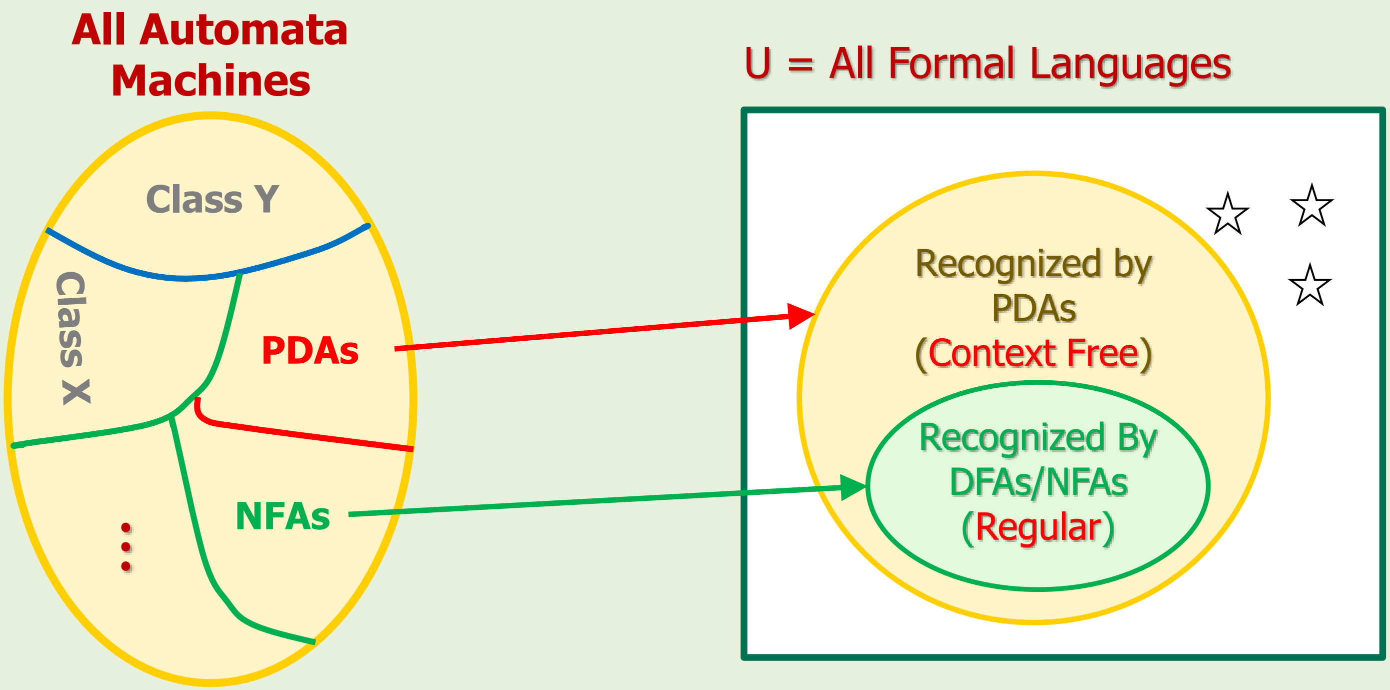
- Therefore, all of the following languages are context-free:
- $L = \{a^n b^n : n \geq 0\}$
- $L = \{ww^R : w \in \Sigma^*\}$
- $L = \{w : n_a(w) = n_b(w), w \in \{a, b\}^*\}$

- Note that a regular grammar is a CFG but NOT vice versa!

PDAs and Languages Association

Flashback

- We mentioned CFLs when we introduced PDAs.
  - We were supposed to explain them later.



- PDAs can recognize CFLs.

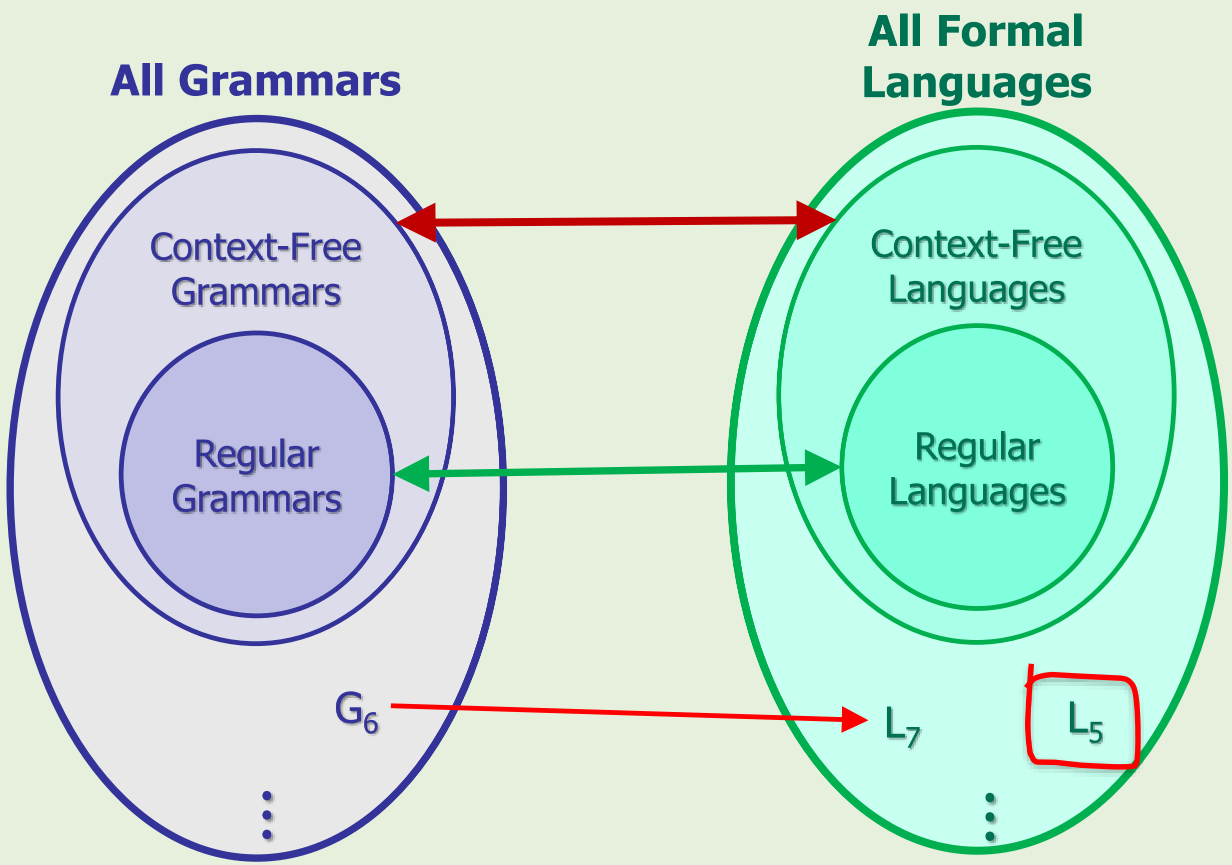
CFLs Representations

- So, now we have two ways for representing CFLs:
  - PDAs
  - CFGs

- Let's revisit the grammars and languages association.

Grammars and Languages Association

Revisited



Application of CFGs in Programming Languages

Example 21

- Consider $L_1 = \{a^n b^n : n \geq 0\}$ over $\Sigma = \{a, b\}$.
- Let's take a different look at this language.

- For example, consider this language:
- $L_2 = \{(^n\ )^n : n \geq 0\}$ over $\Sigma = \{(\ ,\ )\}$ //parentheses are just symbols!

1. What strings would this language contain?
2. What strings do not belongs to this language?

- What is L_2 representing?

Grammars Hierarchy

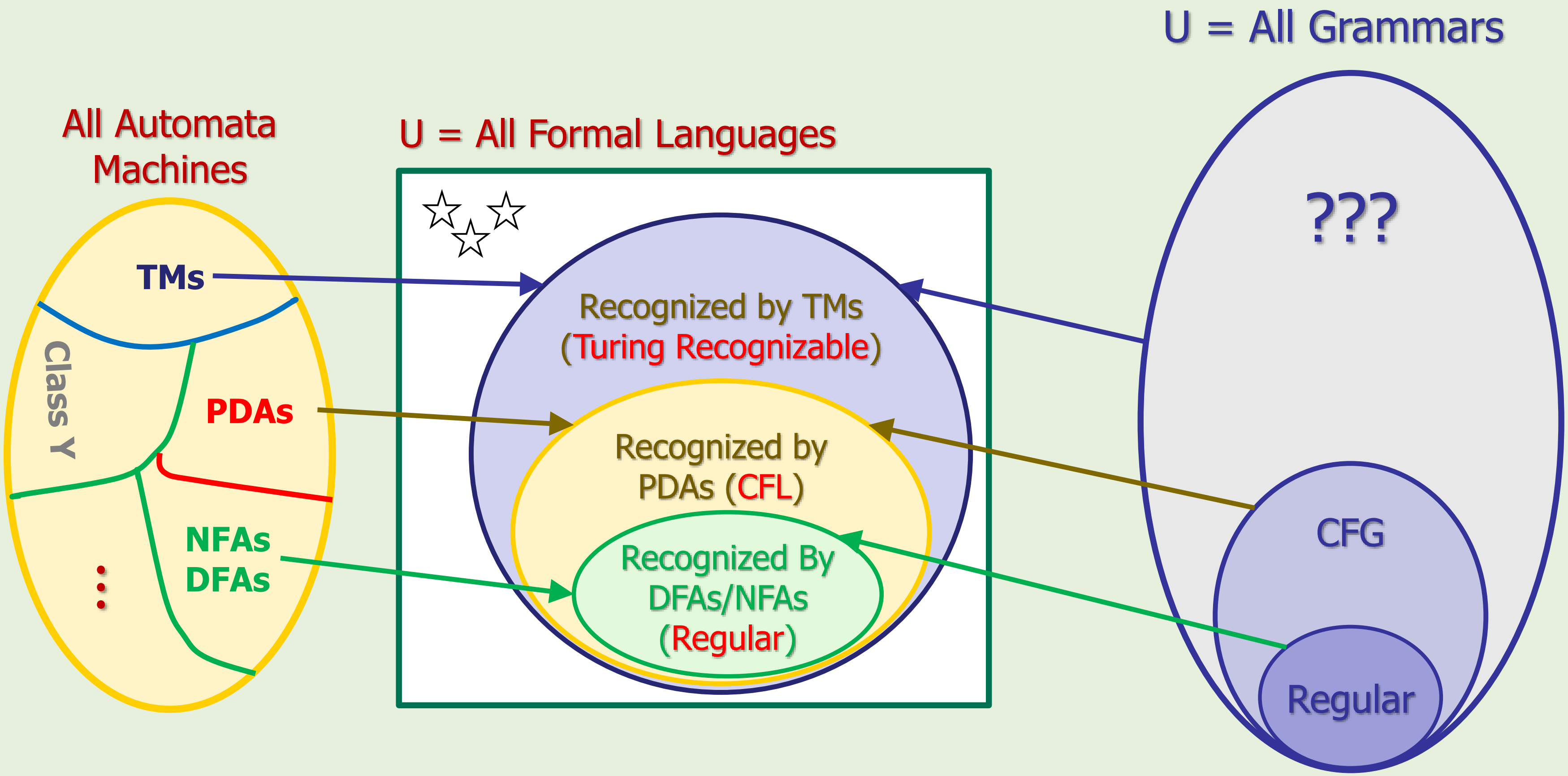
CFG for More Complex Languages

- Find a grammar for each of the following languages:
  1. $L = \{a^n b^n c^n : n \geq 0\}$ over $\Sigma = \{a, b, c\}$
  2. $L = \{ww : w \in \Sigma^*\}$ over $\Sigma = \{a, b\}$

Solution

- ... !
- Struggling?!
- After some struggling, you realize that you cannot find any grammar for these languages! Why?
- Recall that we could not construct PDAs for these languages too!

Machines, Languages, and Grammars Association



- Is there any other grammar that can produce more complex languages like "Turing-recognizable"?

Recursively enumerable Grammar

Definition

- A grammar G is said to be Recursively enumerable (aka unrestricted) if all production rules are of the form:

$$xAy \rightarrow z$$

Where $A \in V$, $x, y, z \in (V \cup T)^*$

Example 22

$S \rightarrow bcA \mid aAbB \mid A$

$aAbB \rightarrow bcA \mid A \mid \lambda$

$A \rightarrow a \mid \lambda$

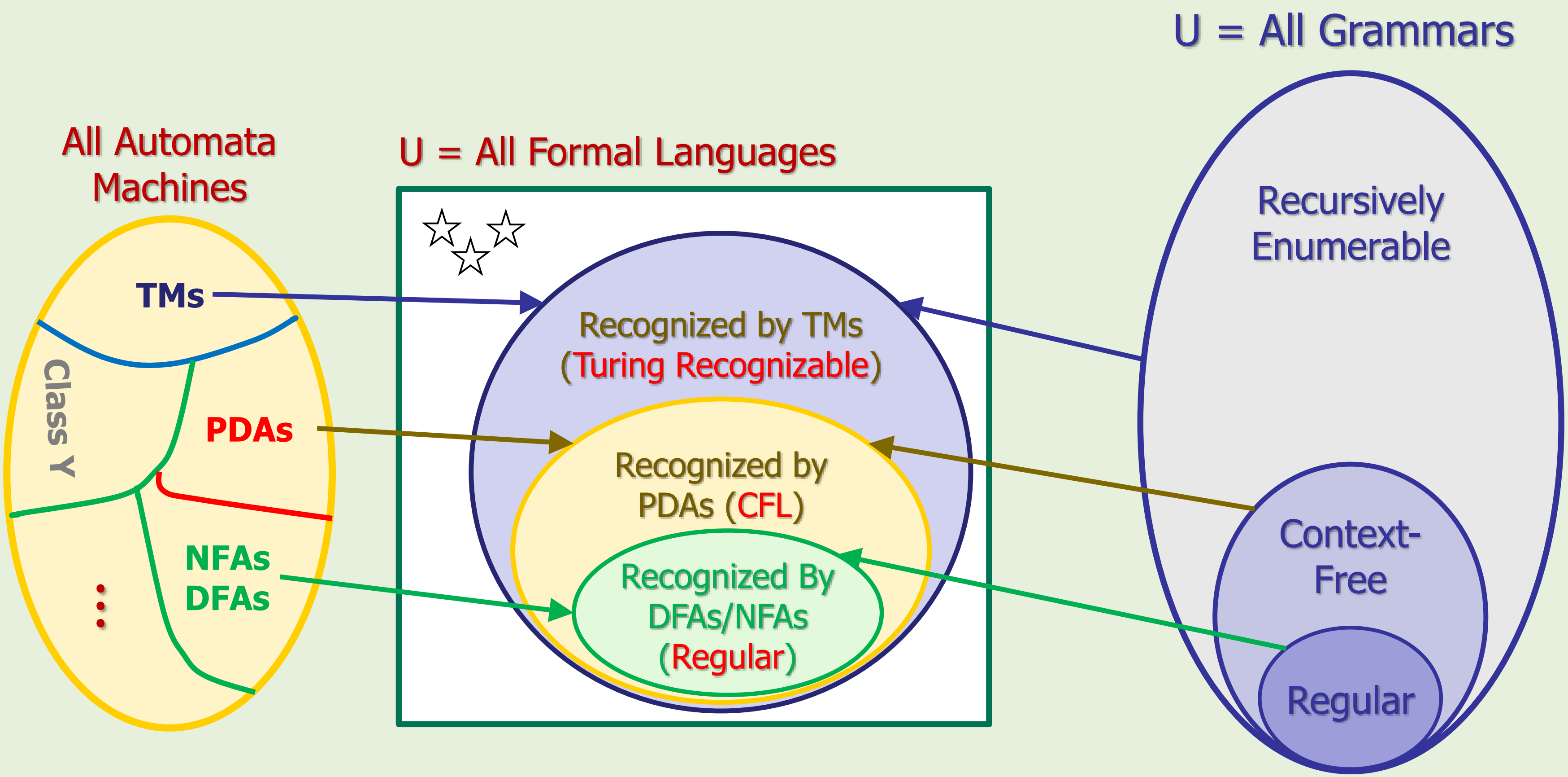
$bcA \rightarrow bbA \mid \lambda$

...

- This type of grammar can produce Turing-recognizable languages.

Machines, Languages, and Grammars Association

Revisited



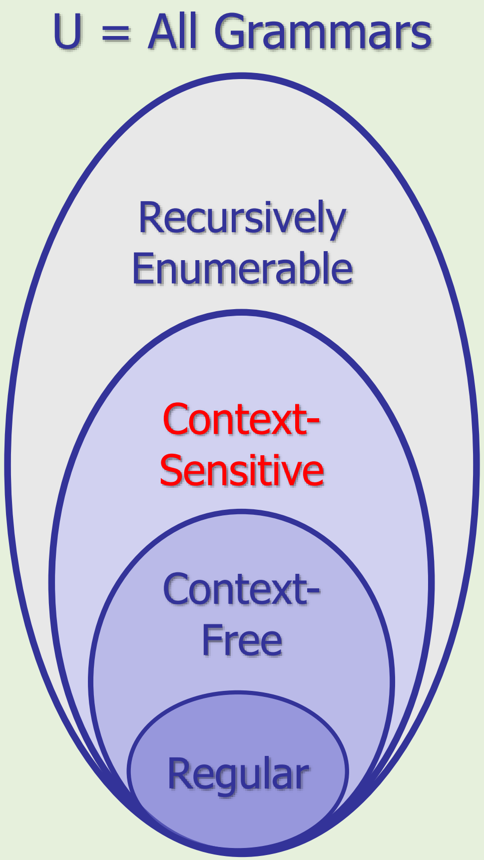
- Note that recursively enumerable grammars include all formal grammars.

Grammars Hierarchy

- There are still another type of grammars between CFGs and recursively enumerable called "context-sensitive grammars".

- Note that both "recursively enumerable" and "context-sensitive" grammars are beyond the scope of this course.

- These categorizations was defined by Noam Chomsky (next slide).

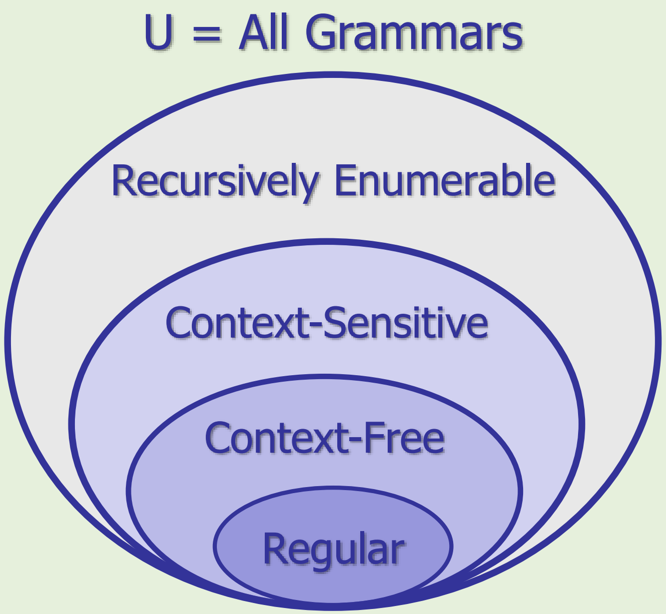


Chomsky's Hierarchy

- Avram Noam Chomsky, the American linguist, philosopher, and historian (1928 - ?), has categorized formal languages that is called "Chomsky's Hierarchy".

- He categorized formal grammars into 4 types as:

- Type 0: Recursively-enumerable
- Type 1: Context-sensitive
- Type 2: Context-free
- Type 3: Regular



Derivations Techniques

Derivations Techniques

- Consider a production rule that has two or more variables.

  $S \rightarrow a\ A\ B$

  $A \rightarrow \ldots$

  $B \rightarrow \ldots$

- To derive a string, we should substitute A and B with some other production rules.
- But in what order?
  - We can substitute them randomly.
  - Or we can pick a specific order. (e.g. left var first or right var first ...)
- Note that we are looking into this question from a software angle, not a human.

Derivations Techniques Example

Example 23

- Derive string "aab" from the following grammar:

$S \rightarrow AB$
$A \rightarrow aaA \mid \lambda$
$B \rightarrow Bb \mid \lambda$

- **Approach 1**: Substitute the leftmost variables first

$$S \overset{1}{\Rightarrow} AB \overset{2}{\Rightarrow} aaAB \overset{3}{\Rightarrow} aaB \overset{4}{\Rightarrow} aaBb \overset{5}{\Rightarrow} aab$$

- **Approach 2**: Substitute the rightmost variables first

$$S \overset{1}{\Rightarrow} AB \overset{4}{\Rightarrow} ABb \overset{5}{\Rightarrow} Ab \overset{2}{\Rightarrow} aaAb \overset{3}{\Rightarrow} aab$$

- Both derivations yielded the same results.

Leftmost / Rightmost Derivations

Definition

- A derivation is said to be leftmost if in each step the leftmost variable in the sentential form is substituted.

Definition

- A derivation is said to be rightmost if in each step the rightmost variable in the sentential form is substituted.

- The default method would be "leftmost" if we don't mention specifically.

Homework

- Derive string "abbbb" from the following grammar:
  1. $S \rightarrow aAB$
  2. $A \rightarrow bBb$
  3. $B \rightarrow A \mid \lambda$

- Leftmost derivation:

- Rightmost derivation:

References

1. Linz, Peter, "An Introduction to Formal Languages and Automata, 5th ed.," Jones & Bartlett Learning, LLC, Canada, 2012
2. Michael Sipser, "Introduction to the Theory of Computation, 3rd ed.," CENGAGE Learning, United States, 2013
ISBN-13: 978-1133187790
3. The ELLCC Embedded Compiler Collection, available at: http://ellcc.org/