San José State University
Department of Computer Science

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

Grammars

(Part 1)

Lecture 20

Day 24/31

CS 154

Formal Languages and Computability

Spring 2018

Agenda of Day 24

- Solution and Feedback of Quiz 7
- Solution and Feedback of Quiz ++
- Summary of Lecture 19
- Couple of slides from previous lecture
- Lecture 20: Teaching ...
  - Grammars (Part 1)

Solution and Feedback of Quiz 7 (Out of 20)

Metrics	Section 1	Section 2	Section 3
Average	17	17	18
High Score	20	19	20
Low Score	6	11	12

Solution and Feedback of Quiz ++ (Out of 60)

Metrics	Section 1	Section 2	Section 3
Average	50	49	50
High Score	59	54	59
Low Score	39	39	28

Summary of Lecture 19: We learned ...

REGEXs

- We defined REGEXs formally as:

1. $\emptyset$, λ, and $a \in \Sigma$ are all REGEXs.
2. If r_1 and r_2 are REGEXs, then the following expressions are REGEXs too:

   $r_1 + r_2$

   $r_1 . r_2$

   r_1^*

   (r_1)
3. A string is REGEX iff it can be derived from the primitive REGEXs by a finite number of applications of the rule #2.

- Between REGEXs and languages, there are the following correspondence:

1. $L(\emptyset) = \{ \}$
2. $L(\lambda) = \{\lambda\}$
3. $L(a) = \{a\}$ for all $a \in \Sigma$
4. $L(r_1 + r_2) = L(r_1) \cup L(r_2)$
5. $L(r_1 . r_2) = L(r_1) L(r_2)$
6. $L((r_1)) = L(r_1)$
7. $L(r_1^*) = (L(r_1))^*$

- We learned how to find the language represented by a REGEX by using the above correspondences.

Any Question?

Summary of Lecture 19: We learned ...

REGEXs

- Two regular expressions r_1 and r_2 are equivalent iff ...
- ... both represent the same language.

$$r_1 = r_2 \leftrightarrow L(r_1) = L(r_2)$$

Identities

- If r, s, and t are REGEXs, and a, b ∊ Σ, then:

1. $r(s + t) = rs + rt$
2. $(s + t)r = sr + tr$
3. $(a^*)^* = a^*$
4. $(a \ldots a)^* a = a (a \ldots a)^*$
5. $a^* (a + b)^* = (a + b)^* a^* = (a + b)^*$

Theorem

- A language is regular iff at least one REGEX represents it.
- The limitation of REGEXs ...
  - They just represent regular languages.
- While more interesting languages are non-regular!

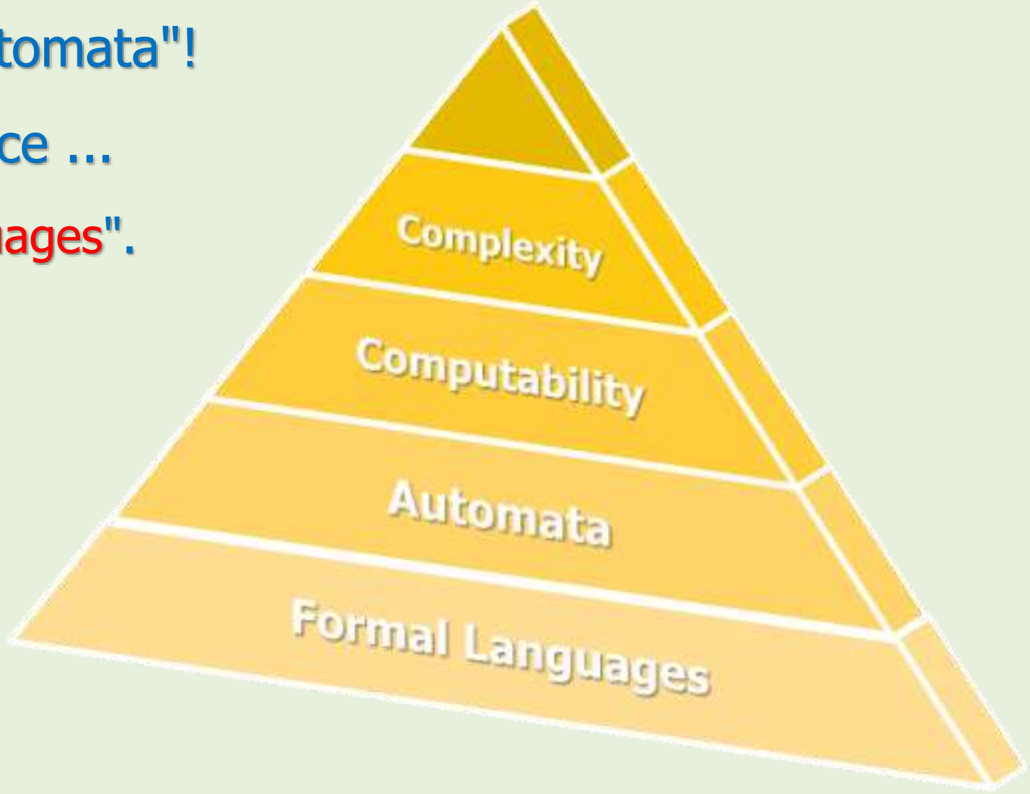
Any Question?

The Big Picture of the Course

Recap

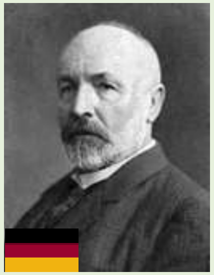
Computer Science Foundation

- We started the semester with "Formal Languages" but we said:
  - we'd get back to it during the semester.
- We are done with the "Automata"!
- So far, we've got back twice ...
  - Introduced "Regular Languages".
  - Introduced "REGEXs".



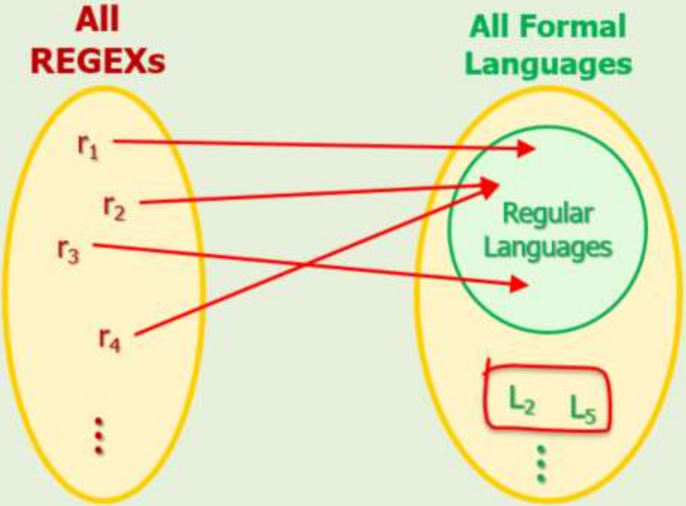
Motivation

- So far, we've represented formal languages by two mathematical objects:
  - Sets (set builder, roster method, Venn diagrams)
  - Regular Expressions (REGEXs)



What is the problem with them?

- Sets was introduced by a mathematician.
  - German mathematician, George Cantor (1845-1918)
- Sets are NOT practical in computer science!
- REGEXs are limited to regular languages.



Objective of This Lecture

- We need a more powerful and practical tool to represent all formal languages!

That is Grammar!

- So, our target is to represent all formal languages!
- But we'll see the limitations of grammars later.

Grammars

Introduction

- What would be your reaction if you encounter the following English sentences:
  - dog the runs.
  - dog runs the.
  - runs dog the.

- Even though all words are correct English words but the combinations do not make sense.
- The words' positions are not as they should be!

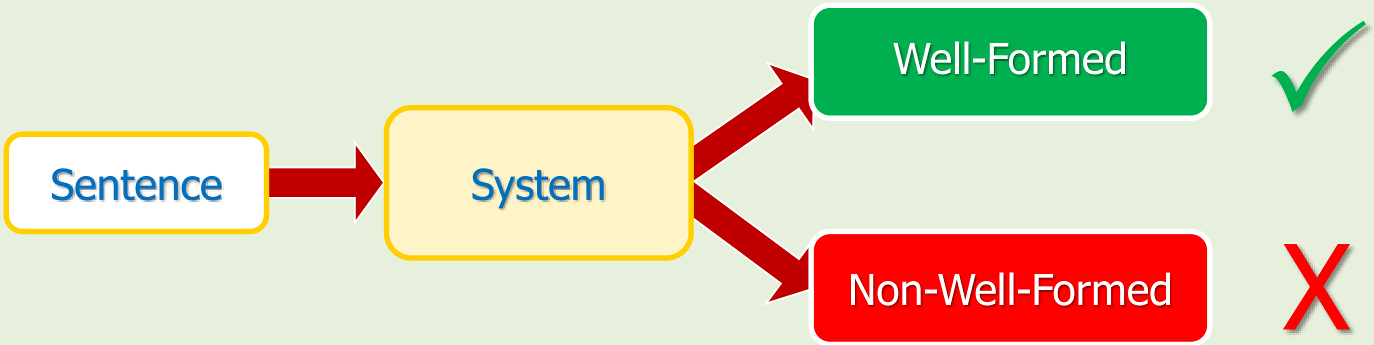
- In computer science and linguistic terminology, we say:

These sentences are not "well-formed".

What Are We Looking For?

- We need a system to distinguish between:

  "well-formed" and "non-well-formed"



- Let's take some examples from a "natural language" like English.
- Then we'll generalize the idea to formal languages.

A Simple English Grammar

- A simple rule for constructing a "sentence" in English grammar is:

  <sentence> → <noun-phrase> <predicate>

  – Read "→" symbol as: "is defined as"

- This expression is called a "production rule".
  – We can produce an English "sentence" by this production rule.

- The problem is:
  We defined a sentence but we introduced two new "variables":
  – <noun-phrase> and <predicate>!

- So, we need to define <noun-phrase> and <predicate>.

A Simple English Grammar

<sentence> → <noun-phrase> <predicate>

- We can define these new variables by the following production rules:

<noun-phrase> → <article> <noun>

<predicate> → <verb>

- Again, we introduced new variables <article>, <noun>, and <verb> that should be defined.

- So, we keep going ...

A Simple English Grammar

<sentence> → <noun-phrase> <predicate>

<noun-phrase> → <article> <noun>

<predicate> → <verb>

- We can define these new variables by the following production rules:

<article> → a

<article> → the

<noun> → dog

<noun> → boy

<verb> → runs

<verb> → walks

A Simple English Grammar: Notes & Definitions

- Variables are defined either by other variables,
- Or they are assigned "values" (aka terminals),
- Or mix of the two (we'll see in the next slides)

Draft Definition of Grammar

- The set of production rules is called "grammar".
  - Later, we define it formally.
- Every grammar has a "starting variable".
  - In this example, <sentence> is the starting variable.

Simple English Grammar

1. <sentence> → <noun-phrase> <predicate>
2. <noun-phrase> → <article> <noun>
3. <predicate> → <verb>
4. <article> → a
5. <article> → the
6. <noun> → dog
7. <noun> → boy
8. <verb> → runs
9. <verb> → walks

Repeated

A Simple English Grammar: Examples

Example 1

- Is the following sentence "well-formed"?

"the dog runs"

Solution

- A sentence is well-formed if we can "derive" it from the production rules.
- We start from the "starting variable":

<sentence> ⇒ <noun-phrase> <predicate>

⇒ <article> <noun> <verb>

⇒ the <noun> <verb>

⇒ the dog <verb>

⇒ the dog runs

Simple English Grammar

1. <sentence> → <noun-phrase> <predicate>
2. <noun-phrase> → <article> <noun>
3. <predicate> → <verb>
4. <article> → a
5. <article> → the
6. <noun> → dog
7. <noun> → boy
8. <verb> → runs
9. <verb> → walks

Repeated

A Simple English Grammar: Examples

Example 1 (cont'd)

- We could derive the sentence "the dog runs" from the set of production rules (grammar), so, it is well-formed.

- We used "⇒" notation for "derivation".
  - Read "⇒" as: "derives"
- We can also represent the whole process as:

$$\langle\text{sentence}\rangle \overset{*}{\Rightarrow} \text{the dog runs}$$

to represent "multiple derivations" when we don't care about the middle ones.

- Now, let's take a failure example.

A Simple English Grammar: Examples

Example 2

- Is the following sentence "well-formed"?

"the runs dog"

Solution

- <sentence> ⇒ <noun-phrase> <predicate>

  ⇒ <article> <noun> <verb>

  ⇒ the <noun> <verb>
- "runs" is NOT a "noun"!
- It fails, so, the sentence is not well-formed.
- What else can we derive from this grammar?

Simple English Grammar

1. <sentence> → <noun-phrase> <predicate>
2. <noun-phrase> → <article> <noun>
3. <predicate> → <verb>
4. <article> → a
5. <article> → the
6. <noun> → dog
7. <noun> → boy
8. <verb> → runs
9. <verb> → walks

Repeated

A Simple English Grammar: Examples

Example 3

- The set of all sentences derivable from the grammar:

{ "a dog runs",
"a dog walks",
"a boy runs",
"a boy walks",
"the dog runs",
"the dog walks",
"the boy runs",
"the boy walks" }

- What can we call this set of strings?

The language generated by the grammar

Simple English Grammar

1. <sentence> → <noun-phrase> <predicate>
2. <noun-phrase> → <article> <noun>
3. <predicate> → <verb>
4. <article> → a
5. <article> → the
6. <noun> → dog
7. <noun> → boy
8. <verb> → runs
9. <verb> → walks

Repeated

Grammar Terminologies

Starting Variables

Left-Hand Side

Right-Hand Side

<sentence> → <noun-phrase> <predicate>

<noun-phrase> → <article> <noun>

<article> → the

Production Rule

Variables

Terminal

"is defined as" symbol

A Side Note About Natural Languages

- Unfortunately, natural languages were not developed by mathematicians.
- That's why, there are thousands of exceptions in their definition!
- So, learning them perfectly as second languages is almost impossible!
- The only language that was developed by a scientist is ...

  Esperanto!
- For more info, please look at the appendix A at the end of this lecture note.
- Note that it is just for your information and is NOT part of this course.

Formal Grammars

Formal Grammars

- We can generalize the natural languages grammars to formal grammars.

Example 4

- Consider the following set of "production rules":

$$\begin{cases} S \rightarrow aB \\ B \rightarrow baB \\ B \rightarrow \lambda \end{cases}$$

- These rules are an example of "formal grammar".
- Let's explain in detail what they mean?

Ingredients of the Production Rules

$$\begin{cases} S \rightarrow aB \\ B \rightarrow baB \\ B \rightarrow \lambda \end{cases}$$

- S and B are "variables".
  - Variables are represented by capital letters.
- The "starting variable" by default is 'S' unless we mention something else.
- 'a', 'b', and λ are called "terminal symbols".
  - Terminals are represented by lower-case letters.
  - λ is our familiar "empty string".
  - Terminals can be any sequence of terminal symbols or λ.
- "aB" and "baB" contain both variable and terminal and is called "sentential form".

How a string can be derived from a grammar?

Example 5

- Consider the following grammar:
  1. $S \rightarrow a\ S\ b$
  2. $S \rightarrow \lambda$
- Derive string "ab"

Solution

$$S \overset{1}{\Rightarrow} a\ S\ b \overset{2}{\Rightarrow} a\ \lambda\ b \Rightarrow ab$$

- Could we derive the string "ab" if we had started with rule #2?

Derivation of Strings

Example 6

- Consider the following grammar:
  1. $S \rightarrow a\,S\,b$
  2. $S \rightarrow \lambda$
- Derive string "aabb".

Solution

$$S \overset{1}{\Rightarrow} a\,S\,b \overset{1}{\Rightarrow} aa\,S\,bb \overset{2}{\Rightarrow} aabb$$

- We can summarize the above derivation like this:

$$S \overset{*}{\Rightarrow} aabb$$

- As we said before, this notation is used when we don't care about the detail of the derivations.

A Convention

- When the left-hand sides of two or more production rules are the same, we can combine the right-hand sides by separating them with a vertical bar "|".

Example 7

- Consider the following grammar:

$S \rightarrow a\,S\,b$

$S \rightarrow \lambda$

- We can represent it as:

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

Associated Language to Grammars

- We can apply the production rules "recursively" in any arbitrary orders.
- Therefore, a grammar can generate many strings.

Definition

- The set of all strings produced (aka generated) by the grammar G is called the "associated language to G" and is denoted by L(G).

Associated Language to Grammars

Example 8

- Consider the following grammar:

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

- What language does it produce?
  - Show it by set-builder.

Solution

- How about this one:

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

- Is there any difference?

Associated Language to Grammars

Example 9

- Consider the following grammar:

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

- What language does it produce?

Solution

Conclusion

- After this example, we know that grammars can represents more languages than regular languages!

Associated Language to Grammars

Example 10

- Consider the following grammar:
  1. $S \rightarrow AB$
  2. $A \rightarrow aA \mid \lambda$
  3. $B \rightarrow bB \mid \lambda$
- What language does it produce?

Solution

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/

San José State University
Department of Computer Science

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

Esperanto

A World Without War!

Appendix A

Spring 2018

Esperanto Creator's Motivation

- Constructed in 1873 by Polish medical doctor, inventor, and writer, Ludwik L. Zamenhof (1859-1917).

- He had the dream of "a world without war".
- He believed this language can help international people to communicate easily.
- And the communications could prevent wars!

Alphabet (5 Vowels + 23 Consonants)

Letter	English Example
a	father
b	
c	cats
ĉ	chip
d	
e	bet
f	
g	go
ĝ	gem

Letter	English Example
h	
ĥ	kh in Persian
i	see
j	yes
ĵ	measure
k	
l	
m	
n	

Letter	English Example
o	so
p	
r	rolled "R"
s	
ŝ	share
t	
u	soon
ŭ	cow
v	
z	

Vowels & Consonants

- 5 Vowels: A, E, I, O, U
- 23 Consonants: the rest of alphabets

- Name the vowels by their pronunciation
- Name the consonants by: letter + o
- For example, we call "b" as "bo".

- The number of syllables of a word is the number of vowels.
- For example, "domo" (means house) has two syllables.

- The accented syllable is the second to last syllable.

Rules

- All "nouns" are ended with 'o'.
- For example, "domo" means house.

- All "adjectives" are ended with 'a'.
- For example, "doma" means domestic.

- Adjectives can be placed before or after nouns.
- For example, "dolĉa pomo" or "pomo dolĉa" means "sweet apple".

- To make a negative adjective, just prefix it with "mal".
- For example, "bona" means "good" and "malbona" means "bad".

Rules

- All "adverbs" are ended with 'e'.
- For example, "rapide" means "quickly".

- Adverbs can be placed before or after the verb they modify.
- For example, "Ĝi flugas rapide" means "it flies quickly".
- Or we can say, "Ĝi rapide flugas".

- To make "plural", add 'j' at the end of the nouns and adjectives.
- For example, "seĝoj" means chairs.
- Example for plural, "rapidaj aŭtoj" means fast cars.
- Note that both adjective and noun should be plural.

Rules

- There is only one "definite article" "la" in Esperanto.
- For example, "la domo" means "the house".

- There is no indefinite article in Esperanto.
- So, if you don't use "la", it means the article is indefinite.

Rules

- To make "possessive personal pronoun", just add 'a' to the end of personal pronoun.
- For example, "mi" means "I", "mia" means "my".
- "vi" means "you", "via" means "your".

- To make a verb negative, just add "ne" before it.
- For example, "Ŝi estas alta." means "She is tall".
- "Ŝi ne estas alta." means "She is not tall".

Tenses

- Infinitives are ended with 'i'.
- For example, "flugi" means "to fly".

- Present tense is ended with "as".
- For example, "Mi flugas" means "I fly".

- Past tense is ended with "is".
- For example, "Mi flugis" means "I flied".

- Future tense is ended with "os".
- For example, "Mi flugos" means "I will fly".

Tenses

- Progressive Present is ended with "anta" but needs "estas" as well.
- For example, "Mi estas fluganta" means "I am flying".

Miscellaneous

- To make a question, add "Ĉu" at the beginning of the sentences.
- For example, "Mi flugas" means "I fly".
- "Ĉu mi flugas" means "Do I fly".
- It doesn't matter what tense the sentence has.

- Over all, Esperanto has 16 constants rules.
- These were some of them, I selected for your information.

General Info About Esperanto

- "Esperanto" means "one who hopes".
- The flag of Esperanto ...

- It is the most successful constructed language in the world.
- Almost two million people speak in Esperanto.
- The first World Congress of Esperanto was organized in 1905 in France.
- Esperanto was recognized as a language by UNESCO in 1954.
  - It was recommended as international non-governmental organizations language in 1985.
- In 2012, Google Translate added Esperanto into its list.
- Language codes in computer: ISO 639-1, ISO 639-2, ISO 639-3

Did You Know That …

- Learning Esperanto as the second language will speed up learning 3rd and more languages.
- 8 Nobel laureates have been Esperantists
- Esperantists make up the largest non-political grouping in the British parliament.
- Leo Tolstoy helped found the Esperantist Vegetarian Association in 1908.
- In 1993, more than 4900 people (mainly non-Esperantists) visited the International Esperanto Museum in Vienna.
- Esperantists have continually suffered oppression from totalitarian governments

Reference: http://esperanto.org/us/USEJ/world/index.html

References

1. Wikipedia, Esperanto available at: https://en.wikipedia.org/wiki/Esperanto
2. Font for Windows: Tajpi http://www.zz9pza.net/tajpi/en/installation/
3. Esperanto lessons: https://www.youtube.com/watch?v=bLx5hLag6WQ