Regular Expressions

CSE 105 Week 1 Discussion

Deadlines and Logistics

- Make sure you have access to the class website, canvas, piazza and gradescope
- Make sure you can see all our OH in the class calendar
- Make sure you have access to PrairieLearn and PrairieTest.
- Read the grading scheme on the class website !
- Schedule your tests asap on PrairieTest !
- HW1 due next week on 8th (Tuesday) at 5 PM
- Link for the slides : https://docs.google.com/presentation/d/1rYhfMTnWjxi5GmIOC0Oj13gXquWENJHr5Azs4E_DpOA/edit?usp=sharing

What can you expect from the discussion section

- Recap* of key concepts from lectures
- Lots of informal interaction
- Practice problems from notes, review quizzes, Sipser and HW
- Ask us any questions and get your doubts answered !

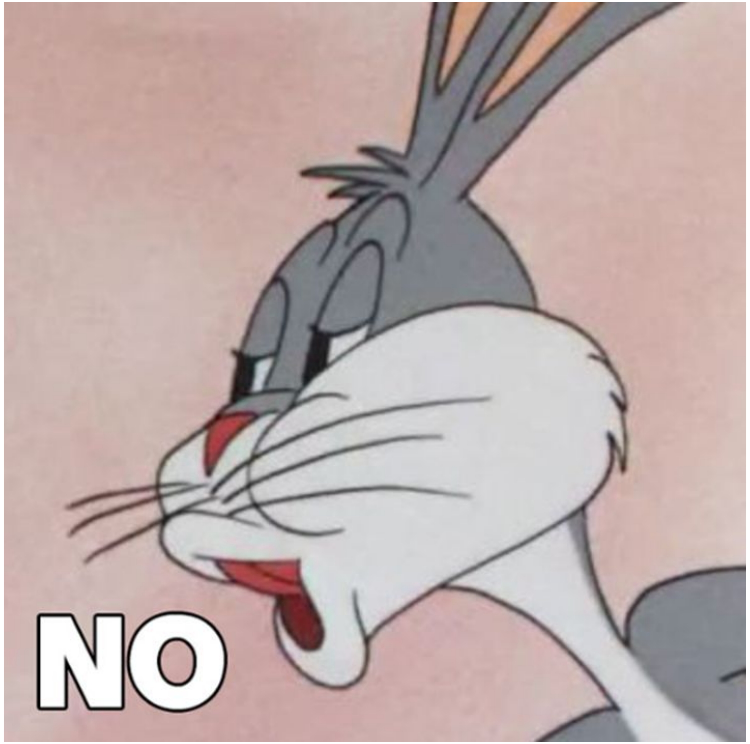
PS - All screenshots taken either from Sipser or class notes unless specified otherwise

*See next slide for the extent of the recap

What NOT to expect from the discussion section

- A self contained “lecture” covering CSE 105. The discussion does NOT replace M/W/F lectures
- Answers to “Graded for correctness” HW questions
- Self contained notes to study from

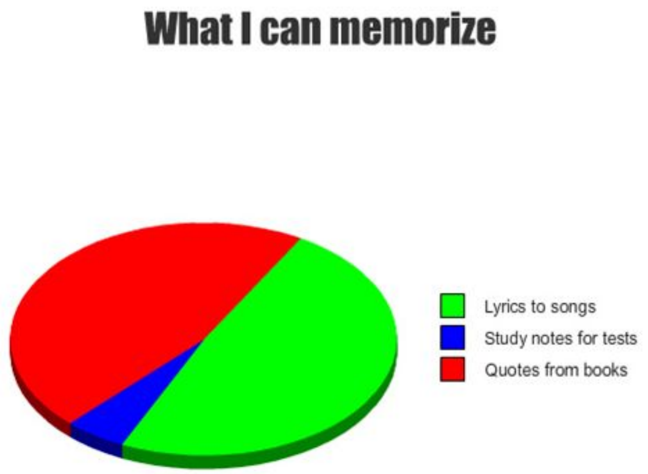
Are the discussion sections podcasted/online/recorded ?



Ask me why

General Information

Do not memorize anything for the sake of completion of HWs, exams etc without first understanding underlying concepts !



Motivation

1. What are the capabilities and limitations of computers ?
2. Can we answer how “difficult” (or if even possible) a certain computation task is ?
3. Can we mathematically model computational problems ?

Important notation

Alphabet	A non-empty finite set, usually denoted as Σ
Symbol	An element of the alphabet
String over Σ	A finite list of symbols from Σ
Language over Σ	A set of strings over Σ
Σ*	Set of all possible strings formed from symbols in Σ

Key takeaways

- A language is a **SET** of strings
- Any time you see a language contain anything BUT strings, alarm bells should go off.
- You can have an empty language denoted by Ø. This is a language with NO strings

Is this statement correct, given Language L = {w|w is a string over {0} such that |w| = 1 }

L = 0 ?

Key takeaways

- A language is a **SET** of strings
- Any time you see a language contain anything BUT strings, alarm bells should go off.
- You can have an empty language denoted by Ø. This is a language with NO strings

Is this statement correct, given Language L = {w|w is a string over {0} such that |w| = 1 }

L = 0 ?

No : A language cant be just a string, it can has to be a set of (0 or more) strings!
L = {0} is the correct notation

Review Quiz Question

Consider the language $\{w \mid w \text{ is a string over } \{0,1\} \text{ and } |w| \text{ is an integer multiple of } 3\}$. Which of the following are elements of this language? (Select all and only that apply)

- ☐ 000
- ☐ 0
- ☐ The empty set
- ☐ $(1, 0, 1)$
- ☐ $\{000\}$
- ☐ The empty string

Select all possible options that apply.

What option(s) can you eliminate right away by virtue of the definition of strings and languages ?

Review Quiz Question

Consider the language $\{w \mid w \text{ is a string over } \{0,1\} \text{ and } |w| \text{ is an integer multiple of } 3\}$. Which of the following are elements of this language? (Select all and only that apply)

- [ ] 000
- [ ] 0
- [ ] ~~The empty set~~
- [ ] ~~(1, 0, 1)~~
- [ ] ~~{000}~~
- [ ] The empty string

Select all possible options that apply.

Can eliminate all non string options ! (since only strings can be elements of a language

What option(s) can you eliminate right away by virtue of the definition of strings and languages ?

Regular Expressions

Remember that at the end of the day, Languages are sets. How can we define a set?

1. List out all the elements
2. Use set builder notation and describe membership condition

Or

3. Use recursive definitions (Regular Expressions !)

The language described by a regular expression *R* is *L(R)*

Recap

Let A and B be languages. We define the regular operations ***union***, ***concatenation***, and ***star*** as follows:

- **Union**: $A \cup B = \{x | x \in A \text{ or } x \in B\}$.
- **Concatenation**: $A \circ B = \{xy | x \in A \text{ and } y \in B\}$.
- **Star**: $A^* = \{x_1 x_2 \ldots x_k | k \geq 0 \text{ and each } x_i \in A\}$.

Pop quiz : If ∑ = {0,1} and language A = {0, 01, 1} and B = {ε, 1} What is

- A U B
- A○B
- B*
- Is (A*)* = A* ?

Recap

Let A and B be languages. We define the regular operations ***union***, ***concatenation***, and ***star*** as follows:

- **Union:** $A \cup B = \{x | x \in A \text{ or } x \in B\}$.
- **Concatenation:** $A \circ B = \{xy | x \in A \text{ and } y \in B\}$.
- **Star:** $A^* = \{x_1 x_2 \ldots x_k | k \geq 0 \text{ and each } x_i \in A\}$.

Pop quiz : If ∑ = {0,1} and language A = {0, 01, 1} and B = {ε, 1} What is

- A ∪ B : {ε, 0, 1, 01}
- A○B : {0, 01, 1, 011, 11}
- B* : {1, 11, 111 ....} or $\{1_0 \ldots 1_k |$ k is a non negative integer}
- Is (A*)* = A* ? : Yes ! Both expressions produce the set of all strings that can be formed by concatenating strings in A with one another as many times as we want

R is a regular expression over the alphabet Σ

1. $R = a$, where $a \in \Sigma$
2. $R = \varepsilon$
3. $R = \emptyset$
4. $R = (R_1 \cup R_2)$, where R_1, R_2 are themselves regular expressions
5. $R = (R_1 \circ R_2)$, where R_1, R_2 are themselves regular expressions
6. (R_1^*), where R_1 is a regular expression.

Context is super important !

For the following examples assume the alphabet is $\Sigma_1 = \{0, 1\}$:

The language described by the regular expression 0 is $L(0) = \{0\}$

The language described by the regular expression 1 is $L(1) = \{1\}$

The language described by the regular expression ε is $L(\varepsilon) = \{\varepsilon\}$

The language described by the regular expression $\emptyset$ is $L(\emptyset) = \emptyset$

The language described by the regular expression $(\Sigma_1\Sigma_1\Sigma_1)^*$ is $L(\ (\Sigma_1\Sigma_1\Sigma_1)^*\ ) =$

Do both the Σ_1 refer to the same thing ?

Context is super important !

Refers to the alphabet set Σ_1 containing symbols 1 and 0 !

For the following examples assume the alphabet is $\Sigma_1 = \{0, 1\}$:

The language described by the regular expression 0 is $L(0) = \{0\}$

The language described by the regular expression 1 is $L(1) = \{1\}$

The language described by the regular expression ε is $L(\varepsilon) = \{\varepsilon\}$

The language described by the regular expression $\emptyset$ is $L(\emptyset) = \emptyset$

The language described by the regular expression $(\Sigma_1\Sigma_1\Sigma_1)^*$ is $L(\ (\Sigma_1\Sigma_1\Sigma_1)^*\ ) =$

Refers to one occurrence of any symbol (0 or 1 in this case) from Σ_1.

Do both the Σ_1 refer to the same thing ? : NO !

Other conventions

Assuming Σ is the alphabet, we use the following conventions

Σ	regular expression describing language consisting of all strings of length 1 over Σ
$*$ then $\circ$ then $\cup$	precedence order, unless parentheses are used to change it
$R_1 R_2$	shorthand for $R_1 \circ R_2$ (concatenation symbol is implicit)
R^+	shorthand for $R^* \circ R$
R^k	shorthand for R concatenated with itself k times, where k is a (specific) natural number

Remember

A regular expression should describe a SET of strings over an alphabet since it is descriptive of a Language, which is a SET of strings over an alphabet.

All “operations” and “conventions” you see in a regular expression boil down to some fundamental operation(s) on set(s).

What operations and what sets ?

Remember

A regular expression should describe a SET of strings over an alphabet since it is descriptive of a Language, which is a SET of strings over an alphabet.

All "operations" and "conventions" you see in a regular expression boil down to some fundamental operation(s) on set(s).

What operations and what sets ?

U, * and ○

As a basis step :

- Element ∈ alphabet or
- ε
- Ø

Inductively:

On any regular expression(s)

When you see... The language described by the regular expression $1^* \circ 1$ is $L(1^* \circ 1) =$

You need to think : sets and operations !

Solve :

When you see... The language described by the regular expression $1^* \circ 1$ is $L(1^* \circ 1) =$

You need to think : sets and operations !

Solve :

$L(1^* \circ 1)$
$= L(1^*) \circ L(1)$
$= L(1)^* \circ L(1)$
$= \{1\}^* \circ \{1\}$
$= \{1^k | k >= 0\} \circ \{1\}$
$= \{1^k 1 | k >= 0\}$ (sufficient to leave it at this step)
$= \{1^k | k >= 1\}$

From Regex to Language

The language over $\Sigma_1 = \{0, 1\}$ described by the regular expression Σ_1^*1 is $L(\Sigma_1^*1) =$ BLANK

Describe this set in (a) Simple english and (b) Set builder notation

(a) Set of all strings over Σ_1 ending in a 1

(b) $\{x1 \mid x \in \Sigma_1^*\}$

Describe the language generated by these Regex(over ∑ = {0,1}) : Basics

1. (0U1)
2. (0U1)*
3. ∑*
4. (0)U(1)
5. (01) U (1)
6. (01)* U (1)
7. (01 U 1)*

No, Seriously, try them yourself first !

Describe the language generated by these Regex(over Σ = {0,1}) : Basics

1. (0U1) : {0,1}
2. $(0U1)^*$: $\{x \mid x \in \Sigma^*\}$
3. Σ^* : Set of all strings that can be created from elements in Σ (Notice similarity to (2)
4. (0)U(1) : {0, 1}
5. (01) U (1): {01, 1}
6. $(01)^*$ U (1): $\{(01)^k \mid k>=0\}$ U {1}
7. $(01 \text{ U } 1)^*$: $\{x \mid x \in \{01,1\}^*\}$

Describe the language generated by these Regex(over Σ = {0,1}) : Level up

1. $(01)(01)^*(01)$
2. $(01)^+(01)$
3. $1 \cup 11 \cup 111 \cup 1111 \cup 11111^*$
4. $\varepsilon^*\Sigma\Sigma\Sigma\varepsilon^*$
5. $(\varepsilon \cup \Sigma)(\varepsilon \cup \Sigma)(\varepsilon \cup \Sigma)$
6. $\Sigma^*1\Sigma^*0\Sigma^* \cup \Sigma^*0\Sigma^*1\Sigma^*$
7. $(1\Sigma^*1) \cup (0\Sigma^*0) \cup (\varepsilon) \cup (1) \cup (0)$

No, Seriously, try them yourself first !

Describe the language generated by these Regex(over $\Sigma = \{0,1\}$) : Level up

1. $(01)(01)^*(01)$: Set of strings containing repeating units of (01) with at least 2 repeats
2. $(01)^+(01)$: Set of strings containing repeating units of (01) with at least 2 repeats (notice similarity to (1))
3. $1 \cup 11 \cup 111 \cup 1111 \cup 11111^*$: Same language as 11^* or 1^+
4. $\varepsilon^*\Sigma\Sigma\Sigma\varepsilon^*$: All strings of length 3
5. $(\varepsilon \cup \Sigma)(\varepsilon \cup \Sigma)(\varepsilon \cup \Sigma)$: All strings of length at most 3
6. $\Sigma^*1\Sigma^*0\Sigma^* \cup \Sigma^*0\Sigma^*1\Sigma^*$: All strings containing at least one 1 and one 0
7. $(1\Sigma^*1) \cup (0\Sigma^*0) \cup (\varepsilon) \cup (1) \cup (0)$: All strings starting and ending with the same symbol

Questions?

Good luck for HW 1 !