Before we start

If you or someone you know is suffering from food and/or housing insecurities there are UCSD resources here to help:

Basic Needs Office: https://basicneeds.ucsd.edu/

Triton Food Pantry (in the old Student Center) is free and anonymous, and includes produce:

https://www.facebook.com/tritonfoodpantry/

Mutual Aid UCSD: https://mutualaiducsd.wordpress.com/

If you find yourself in an uncomfortable situation, ask for help. We are committed to upholding University policies regarding nondiscrimination, sexual violence and sexual harassment.

Counseling and Psychological Services (CAPS) at 858 5343755 or http://caps.ucsd.edu

OPHD at (858) 534-8298, ophd@ucsd.edu , http://ophd.ucsd.edu. CARE at Sexual Assault Resource Center at 858 5345793 sarc@ucsd.edu http://care.ucsd.edu

Spring quarter philosophy

Spring 2022 is a transition quarter so please be patient with us as we do our best to serve the needs of all students while adhering to the university guidelines. First and foremost is the health and safety of everyone. Please do not come to class if you are sick or even think you might be sick. Please reach out (minnes@eng.ucsd.edu) if you need support with extenuating circumstances.

Masks are required in class. All students who attend class must also be fully vaccinated against COVID-19 unless they have a university-approved exemption. Campus policy requires masks and daily "symptom screeners" for everyone and we expect all students to follow these rules.

Welcome to CSE 105: Introduction to Theory of Computation in Spring 2022!

Themes and applications for CSE 105

- **Technical skepticism**: Know, select and apply appropriate computing knowledge and problem-solving techniques. Reason about computation and systems. Use mathematical techniques to solve problems. Determine appropriate conceptual tools to apply to new situations. Know when tools do not apply and try different approaches. Critically analyze and evaluate candidate solutions.
- **Multiple representations**: Understand, guide, shape impact of computing on society/the world. Connect the role of Theory CS classes to other applications (in undergraduate CS curriculum and beyond). Model problems using appropriate mathematical concepts. Clearly and unambiguously communicate computational ideas using appropriate formalism. Translate across levels of abstraction.

Applications: Numbers (how to represent them and use them in Computer Science), Recommendation systems and their roots in machine learning (with applications like Netflix), "Under the hood" of computers (circuits, pixel color representation, data structures), Codes and information (secret message sharing and error correction), Bioinformatics algorithms and genomics (DNA and RNA).

Introductions

Class website: http://cseweb.ucsd.edu/classes/fa21/cse20-a

Pro-tip: the URL structure is your map to finding your course website for other CSE classes.

Pro-tip: you can use MATH109 to replace CSE20 for prerequisites and other requirements.

Instructor: Prof. Mia Minnes "Minnes" rhymes with Guinness, minnes@eng.ucsd.edu, http://cseweb.ucsd.edu/ minnes

Our team: Four TAs and 10 tutors + all of you

Fill in contact info for students around you, if you'd like:

On a typical week: **MWF** Lectures + review quizzes, **T** HW due, **W** Discussion, office hours, Piazza. Project parts will be due some weeks.

All dates are on Canvas (click for link) and details are on course calendar (click for link).

Monday March 28

Alphabet e.g. Σ, Γ	non-empty finite set
Symbol over Σ	element of alphabet Σ
String over Σ	finite list of symbols from Σ
Language over Σ	set of strings over Σ
Empty set $\emptyset$	the empty language
Pages 3, 4, 13, 14	

With $\Sigma_1 = \{0,1\}$ and $\Sigma_2 = \{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z\}$ and $\Gamma = \{0,1,x,y,z\}$

An example of a string of length 3 over Σ_1 is

An example of a string of length 1 over Σ_2 is

The number of distinct strings of length 2 over Γ is

An example of a language over Σ_1 of size 1 is

An example of an infinite language over Σ_1 is

An example of a finite language over Γ is

Empty string ε	the string of length 0
Reverse of a string w, $w^{\mathcal{R}}$	write w in the opposite order, if $w = w_1 \cdots w_n$ then $w^{\mathcal{R}} = w_n \cdots w_1$
Concatenating strings x and y	take $x = x_1 \cdots x_m$, $y = y_1 \cdots y_n$ and form $xy = x_1 \cdots x_m y_1 \cdots y_n$
String z is a substring of string w	there are strings u, v such that $w = uzv$
String x is a prefix of string y	there is a string z such that $y = xz$
String x is a proper prefix of string y	x is a prefix of y and $x \neq y$
Pages 13, 14	

$\varepsilon \in \Sigma_1$	True	False
ε is a string over Σ_1	True	False
ε is a language over Σ_1	True	False
ε is a prefix of some string over Σ_1	True	False
There is a string over Σ_1 that is a proper prefix of ε	True	False

String order over alphabet Σ: Order strings over Σ first by length and then according to the dictionary order, assuming symbols in Σ have an ordering.

The first five strings over Σ_1 in string order, using the ordering $0 < 1$:

The first five strings over Σ_2 in string order, using the usual alphabetical ordering for single letters:

Assuming A and B are languages over alphabet Σ	
The union A and B	$A \cup B = \{x \mid x \in A \text{ or } x \in B\}$
The concatenation of A and B	$A \circ B = \{xy \mid x \in A \text{ and } y \in B\}$
The star of A	$A^* = \{x_1 x_2 \cdots x_k \mid k \geq 0 \text{ and each } x_i \in A\}$
Definition 1.23 page 44	

Assuming Σ is the alphabet, recursive definition of regular expressions is		
a is a regular expression	for $a \in \Sigma$	$L(a) = \{a\}$
ε is a regular expression		$L(\varepsilon) = \{\varepsilon\}$
$\emptyset$ is a regular expression		$L(\emptyset) = \{\} = \emptyset$
$(R_1 \cup R_2)$ is a regular expression	for R_1, R_2 regular expressions	$L(\ (R_1 \cup R_2)\ ) = L(R_1) \cup L(R_2)$
$(R_1 \circ R_2)$ is a regular expression	for R_1, R_2 regular expressions	$L(\ (R_1 \circ R_2)\ ) = L(R_1) \circ L(R_2)$
(R_1^*) is a regular expression	for R_1 a regular expression	$L(\ (R_1^*)\ ) = (\ L(R_1)\ )^*$
Definition 1.52 page 64		

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
Pages 63 - 65	

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

Regular expression, R	Language described by the regular expression, $L(R)$
0	$\{0\}$
1	$\{1\}$
ε	$\{\varepsilon\}$
$\emptyset$	$\emptyset$
$((0 \cup 1) \cup 1)$	
1^+	
$\Sigma_1^* 1$	
$(\Sigma_1 \Sigma_1 \Sigma_1 \Sigma_1 \Sigma_1)^*$	
$1^* \emptyset 0$	
	$\{00, 01, 10, 11\}$
	$\{0^n 1 \mid n \text{ is even}\}$

Review: Week 1 Monday

1. Please complete the beginning of the quarter survey https://forms.gle/gvibFnNixxqcWbaU8
2. We want you to be familiar with class policies and procedures so you are ready to have a successful quarter. Please take a look at the class website http://cseweb.ucsd.edu/classes/fa21/cse20-a and answer the questions about it on Gradescope.

Pre class reading for next time: Figure 1.4, Definition 1.5

Week 1 Wednesday

Alphabet e.g. Σ, Γ	non-empty finite set
Symbol over Σ	element of alphabet Σ
String over Σ	finite list of symbols from Σ
Language over Σ	set of strings over Σ
Empty set $\emptyset$	the empty language
Regular expression over Σ e.g. R	syntactic expression built up recursively
Language described by R, $L(R)$	set of strings matching pattern given by regular expression
Pages 3, 4, 13, 14, 64, 65	

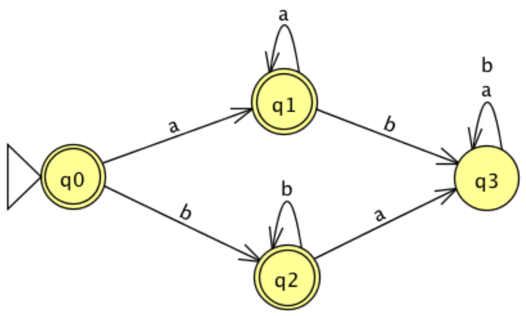
For the following True/False questions assume the alphabet is $\Sigma = \{a, b, c\}$:

$a \in L(a \cup b \cup c)$	True	False
$ab \in L(\ (a \cup b)^*\ )$	True	False
$ba \in L(\ a^*b^*\ )$	True	False
$\varepsilon \in L(a \cup b \cup c)$	True	False
$\varepsilon \in L(\ (a \cup b)^*\ )$	True	False
$\varepsilon \in L(\ a^*b^*\ )$	True	False

Deterministic finite automaton	$M = (Q, \Sigma, \delta, q_0, F)$
Finite set of states Q	Can be labelled by any collection of distinct names. Default: $q0, q1, \ldots$
Alphabet Σ	Each input to the automaton is a string over Σ.
Transition function δ	Gives the next state based on current state of machine next input symbol
Start state q_0	Element of Q. Each computation of the machine starts at the start state.
Accept (final) states F	$F \subseteq Q$. Used to flag if the machine accepts or rejects an input string.
Computation	The computation of a machine on an input string is a sequence of states in the machine, starting with the initial state, determined by transitions of the machine as it reads successive input symbols.
M accepts the input string	The computation of M on the input string ends in an accept state.
M rejects the input string	The computation of M on the input string ends in a nonaccept state.
Language of M, $L(M)$ aka language recognized by M	The set of all strings that are each accepted by the machine M.
Pages 34-36	

What is **finite** about a deterministic finite automaton? (Select all that apply)

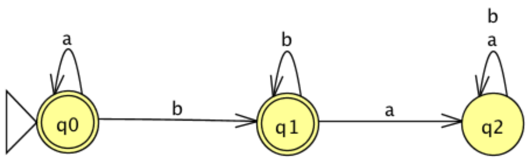
- The size of the machine (number of states, number of arrows)
- The number of strings that are accepted by the machine
- The length of the computations of the machine



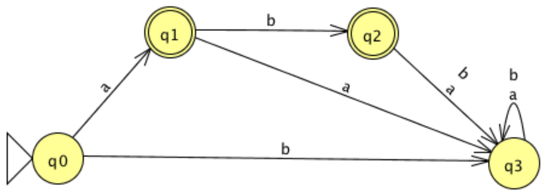
The formal definition of this DFA is

Input string	**Result**: this string is ...	
a	accepted by the DFA	rejected by the DFA
aa	accepted by the DFA	rejected by the DFA
ab	accepted by the DFA	rejected by the DFA
ba	accepted by the DFA	rejected by the DFA
bb	accepted by the DFA	rejected by the DFA
ε	accepted by the DFA	rejected by the DFA

The language recognized by this DFA is



The language recognized by this DFA is



The language recognized by this DFA is