Today:
- Sets of strings (languages)
- Regular expressions

Distinguished Lecture after class:
"Some Hash-Based Data Structures and Algorithms Everyone Should Know"
Prof. Michael Mitzenmacher, Harvard

Sets of STRINGS (elements of formal language theory)

Define and give examples:

**Alphabet** – a finite nonempty set (of “characters”) \( \Sigma \)

**Strings** - a finite sequence of characters drawn from some alphabet.
- operation: **concatenation**, \( xy \) or \( x \circ y \)

**Language** – a set of strings, all of them over some alphabets
- extend concatenation to languages

**Empty string** \( (\varepsilon) \)

General set operations: union, intersection, complement; and concatenation

\( A^* \) -- Kleene closure – **define it** - \( \bigcup_{i \geq 0} A^i \)

\( A^0 = \{ \varepsilon \} \) (why? For \( A^i \ A^i = A^{i+1} \))

\( \Sigma^* \)

BYTES = \( \{0,1\}^8 \)

BYTES*

ENGLISH-WORDS = \{a, aah, aardvark, aardwolf, aba,..., zymotic, zymurgy\}

PRIMES = ...

SAT = ...

The relationship between languages and **decision questions**.

![Decision Diagram](attachment:image.png)

**Regular expression over \( \Sigma \):**
1) $a$ is a regular expression, for every $a \in \Sigma$. Also, symbols $\varepsilon$ and $\emptyset$ are regular expressions.

2) If $\alpha$ and $\beta$ are regular expressions, then so are $(\alpha \circ \beta)$, $(\alpha \cup \beta)$, $(\alpha^*)$

We routinely omit parenthesis, understanding it as a shorthand, with * binding most tightly, then concatenation, then union.

Example: a number

$$(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^* (0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*$$

A real number in decimal notation

$$(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*$$.  $(0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9)^*$. 

An even number in binary

$(0 \cup 1)^0$

Bit strings that start and top with the same bit (having at least one bit)

$00^* \cup 11^* \cup 0(0 \cup 1)^*0 \cup 1(0 \cup 1)^*1$

The complement of that set

$\varepsilon \cup 0(0 \cup 1)^*1 \cup 1(0 \cup 1)^*0$

Exercise 1: write a regular expression for all strings over $\{0,1\}$ that contain _some_ $'111'$. 

$(0u1)^* 111 (0u1)^*$

Exercise 2: write a regular expression for all strings over $\{a,b\}$ whose length is divisible by 3.

$(aub)(aub)(aub))^*$

Exercise 3: write a regular expression for all strings over $\{a,b\}$ whose length is NOT divisible by 3.

$(aub)(aub)(aub))^*(aub) u (aub)(aub)(aub))^*(aub)(aub)$

Exercise 4: write a regular expression for all strings over $\{0,1\}$ that contain an even # of 0's and an even # of 1's.

Kind of hard

Exercise 5: write a regular expression for all strings over $\{0,1\}$ that contain the same number of 0’s and 1’s.

CAN’T BE DONE. Why? Take ECS120!

---

**Relations**
(Change of topics. But do define some relations on strings, regular
languages, and DFAs to tie the two topics together.)

**DEF:** A and B sets. Then a *relation* R is subset of $A \times B$.

$R \subseteq A \times B$

Variant notation: $x \mathbin{R} y$ for $(x,y) \in R$

May use a symbol like ~ or < for a relations

$x \sim y$ if $(x,y) \in \sim$

Relations in arithmetic, where A and B are both natural numbers:

$= < <= > =$

| | divides

what about succ, +, *   **NO:** function symbols, not relations

In set theory:

$\in$  

what about $\emptyset$   **NO:** constant symbol

Relations are useful for things other than numbers and sets and the like:

$S =$ all UCD students for F13

$C =$ all UCD classes for F13

$P =$ all UCD professors for F13

$E: \text{enrolled relation} \subseteq S \times C$

$s \mathbin{E} c$ (ie, $(s,c) \in E)$ - $x$ is taking class $y$

$T: \text{teaches relation} \subseteq C \times P$

$c \mathbin{T} p$ (ie, $(c,p) \in T)$ - professor $p$ is teaching class $c$ this term

You can *compose* relations

what _should_  

$E \circ T$

mean, do you think

$E \circ T \subseteq S \times P$  

$S \times C \quad C \times P \quad \rightarrow \quad S \times P$

$s \mathbin{E \circ T} p$ if there exists $c$ in $C$ such that $s \mathbin{E} c$ and $c \mathbin{T} p$ -- student $s$ is taking some course that $p$ is teaching -- $p$ is $s$'s teacher this term

What I've just given is the general definition

$R \subseteq X \times Y$

$S \subseteq Y \times Z$ then $R \circ S \subseteq X \times Z$ is \{(x,z): \exists y \text{ in } Y \mathbin{Ry} \text{ and } y \mathbin{Sz}\}

What _should_ $R^{-1}$ should be?

formalize
if \( R \subseteq X \times Y \) is a relation that \( R^{-1} \) is the relation on \( Y \times X \) where \((y,x) \in R^{-1} \iff x \in R \ y\).

More examples:
Often \( X = Y \) is the *same* set
Relations on natural numbers, real numbers, strings, etc.

\( X = \) set of strings
\( x \leq y \) "is a substring of y"

\( \alpha \) and \( \beta \) are regular expressions.
\( \alpha \sim \beta \) if \( L(\alpha) = L(\beta) \)

T/F: \( (0u1)^* (0u1)^* \sim (00 \ u \ 01 \ u \ 10 \ u \ 11)^* \) TRUE
\( e \sim e^* \) TRUE
\( 0(0u1)0 \sim 1(0u1)1 \) FALSE

Relations, continued. Let \( R \) be a relation on \( A \times A \)
We say that \( R \) is

Reflexive: if \( x \in R \ x \) for all \( x \)
Symmetric: if \( x \in R \ y \rightarrow y \in R \ x \) for all \( x, y \)
Transitive: if \( x \in R \ y \) and \( y \in R \ z \rightarrow x \in R \ z \) for all \( x, y, z \)

If \( R \) has all three properties, \( R \) is said to be an equivalence relation

<table>
<thead>
<tr>
<th></th>
<th>Reflexive</th>
<th>Symmetric</th>
<th>Transitive</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>= on Integers</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>(or anything else)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \leq ) , integers</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>antisymmetric</td>
</tr>
<tr>
<td>( \subseteq ) , sets</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>antisymmetric</td>
</tr>
<tr>
<td>( x \in E \ y ) if ( x ) and ( y ) are regular expressions and the regular ( L(x) = L(y) )</td>
<td>Yes</td>
<td>Yes</td>
<td>blocks are languages</td>
<td></td>
</tr>
<tr>
<td>( x \in S \ y ) if ( x ) is a substring of ( y )</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>( x \in R \ y ) where ( x ) and ( y ) are strings and ( M ) is a some DFA and you go to the same state on processing</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>
x and y
x | y  if 3 | x-y
-  y         Yes            Yes       Carefully
prove this one
out its blocks.

n | m

We only got to here – and then I jumped ahead to defining functions. We’ll take up equivalence classes and
quotients next time, as well as properties of functions, like injectivity and surjectivity.

4. Functions

Definition: A function f is a relation on A x B such that
there is one and only one pair a R b for every in A.

We write b=f(a) to mean that (a,b) in f.

(Just one way to do it: we could have defined functions as the primitive
and used the function to define the relation, putting in a pair
(a,f(a)) for every a in A.)

- We call A the domain of f,  Dom(f).
- We call B the *codomain* (or *target*) of f.
  Note that this does not mean the set {b: f(a)=b for some a in A}!
  That is a different (and important) st called the *Range* (or *image*)
  of f. Denote it f(A).

Example 1:
Domain={1,2,3}
f(a) = a^2.
Dom(f) = {1,2,3}
f(A) = {1,4,9}

co-domain: unclear, might be \N, might be \R, ....

Example 2:
Domain = students in this class
b(x) = birthdays, encoded as {1,..,12} x {1..31}.

b(phil) = (7,31)
b(ellen) = (4,1)

Example 3:
f: \R -> \R defined by f(x) = x^2
is it a function?
Represent it as a graph

Two functions f and g are equal, f=g, if their domains and ranges are equal
and f(x) = g(x) for all x in Dom(f)

Function composition
\[ f \circ g \]
\[ f: A \rightarrow B, \quad g: B \rightarrow C \]
the \((g \circ f): A \rightarrow C\) is defined by
\[(g \circ f)(x) = g(f(x))\]

Kind of "backwards" notation, but fairly tradition. Some algebrists will reverse it, \((x) \ (f \circ g)\) "function operates on the left"

Some computer scientists like to denote functions by "lambda expressions"
To say that \(f\) is the function that maps \(x\) to \(x^2\) we write
\[ f = \lambda x. x^2 \]
Here \(x\) is just a formal variable;
\[ \lambda x. x^2 = \lambda y. y^2 \]
The domain is not explicitly

Functions don't have to be defined on numbers, of course

\[ |x| = \text{maps } \Sigma^* \rightarrow \mathbb{N} \]
\[ \text{hd}(x) = \text{the first character of the string } x, \ x \neq \text{emptystring} \]
\[ \text{tl}(x) = \text{all but the first character of } x \ (\text{define how when } x=\text{emptystring})? \]
\[ \text{dim}(A) = \text{the dimensions of the matrix } A, \text{ regarded as a pair of natural numbers} \]