Theory of Computation
Week 1
Supplementary slides for self study and week 1 video

Much of the material on this slides comes from the recommended textbook by Elaine Rich

Detailed content

Weekly program

- **Week 1 – Background knowledge revision: logic, sets, proof techniques**

- Week 2 – Languages and strings. Hierarchies. Computation. Closure properties
- Week 3 – Finite State Machines: non-determinism vs. determinism
- Week 4 – Regular languages: expressions and grammars
- Week 5 – Non regular languages: pumping lemma. Closure
- Week 6 – Context-free languages: grammars and parse trees
- Week 7 – Pushdown automata
- Week 8 – Non context-free languages: pumping lemma and decidability. Closure
- Week 9 – Decidable languages: Turing Machines
- Week 10 – Church-Turing thesis and the unsolvability of the Halting Problem
- Week 11 – Decidable, semi-decidable and undecidable languages (and proofs)
- Week 12 – Revision of the hierarchy. Safety-critical systems
- Week 13 – Extra revision (if needed)

Week 01

Set Theory

- ❑ Sets
- ❑ Functions
- ❑ Relations
- ❑ Closure

Sets

- A ***set*** is simply a collection of objects.
- We call the objects ***elements*** or ***members*** of the set
- May contain any type of object: numbers, symbols, other sets,…
  - Set membership: $\in$
  - Non-membership: $\notin$
  - Subset: $\subseteq$
    - Proper subset
  - Empty set: $\emptyset$
  - Infinite set contains infinitely many elements
    - E.g., set of integers {…-2, -1, 0, 1, 2,…}

Defining a Set

- By ***enumerating*** the elements of *S*.

- By using the ***characteristic function*** of *S*.
  - Often when S is defined as a subset of other set.

- If we use a program to define a set, it can
  - return an enumeration of all elements of S
  - ***True*** if run on some element that is in *S*, and ***False*** if run on an element that is not in *S*.

Sets example

- S_1 = {13, 11, 8, 23}.
- S_2 = {8, 23, 11, 13}.
- S_3 = {8, 8, 23, 23, 11, 11, 13, 13}.
- S_4 = {apple, pear, banana, grape}.
- S_5 = {January, February, March, April, May, June, July, August, September, October, November, December}.
- $S_6 = \{x : x \in S_5$ and x has 31 days}.
- S_7 = {January, March, May, July, August, October, December}.

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Set Cardinality

- How many elements does S contain?
- If $S=\{2,7,11\}$ then $|S| = |\{2,7,11\}| = 3$.
- We can have three different kinds of answers
  - If S is finite then a natural number
  - If S has the same number of elements as there are integers then it is ‘countably infinite’
  - If S has more elements than there are integers then ‘uncountably infinite’ or ‘uncountable’

Set Cardinality

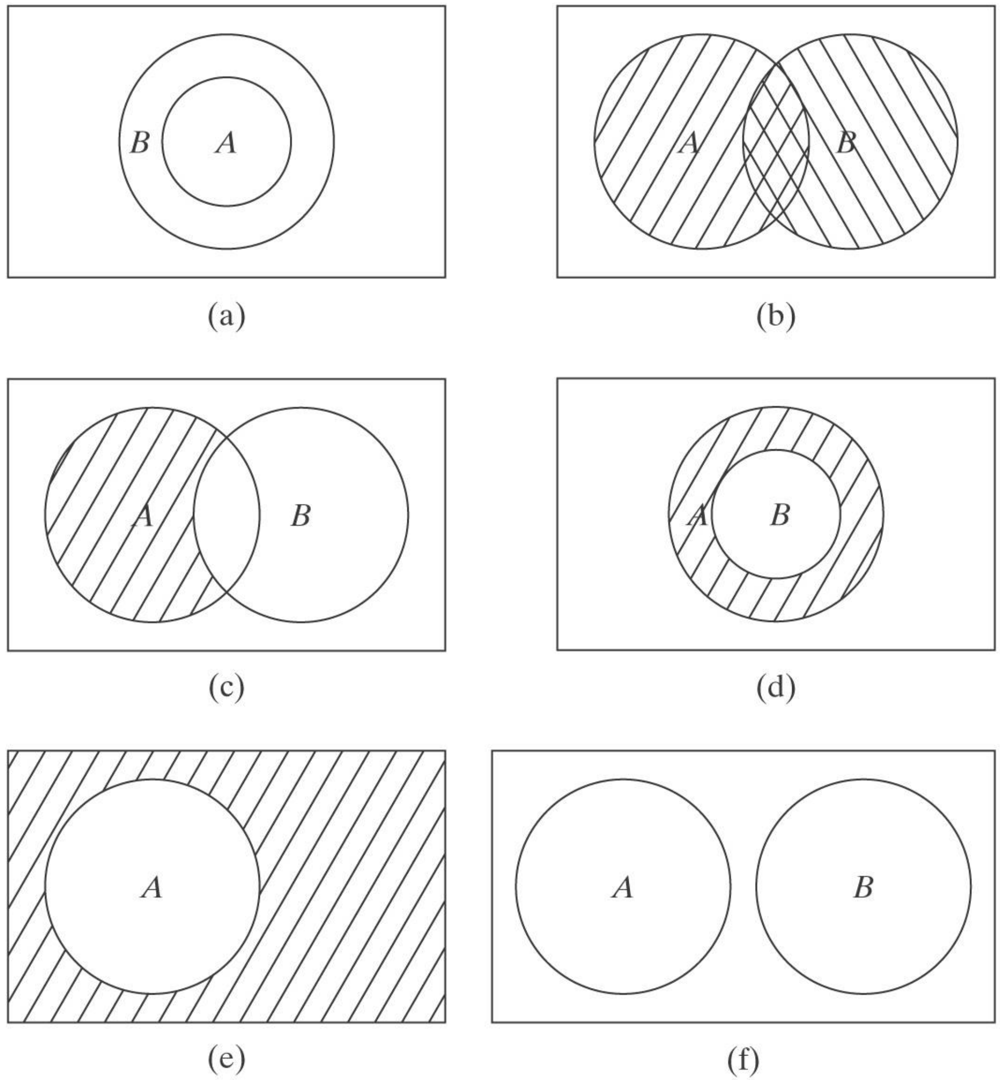
- The Infinite Hotel Paradox - Jeff Dekofsky

https://www.youtube.com/watch?v=Uj3_KqkI9Zo

Sets - What you need to know

- Be able to determine the **union** ($\cup$), **intersection** ($\cap$), **difference** (-), **complement** (overbar), and **power set** $|P(X)| = 2^X$ of any given set.
- Proving **equality of sets** X and Y by:
  - Showing every element of X is an element of Y and vice versa (i.e., show that $X \subseteq Y$ and $Y \subseteq X$ to show $X = Y$.)
  - Mathematical induction
  - Case enumeration

Set Operations: relating sets to each other



Sets of Sets

- The ***power set*** of A is the set of all subsets of A.

  Let $A = \{1, 2, 3\}$. Then:

  $$\mathcal{P}(A) = \{\varnothing, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}.$$

- $\Pi \subseteq \mathcal{P}(A)$ is a ***partition*** of a set A iff:
  - no element of Π is empty,
  - all pairs of elements of Π are disjoint , and
  - the union of all the elements of Π equals A.

  Partitions of A:

  $$\{\{1\}, \{2, 3\}\} \quad \text{or} \quad \{\{1, 3\}, \{2\}\} \quad \text{or} \quad \{\{1, 2, 3\}\}.$$

Sets of Sets

- If S is a set where $|S| = n$, then the number of elements in the power set of S, $\mathcal{P}(S)$ is 2^n

Relations

- An ***ordered pair*** is a sequence of two objects, written: (x, y)
- The ***Cartesian product*** of two sets A and B is the set of all ordered pairs (a, b) such that $a \in A$ and $b \in B$. We write it as: $A \times B$
- A ***binary relation*** over two sets A and B is a subset of $A \times B$.
- An ***n-ary relation*** over sets $A_1, A_2, \ldots A_n$ is a subset of $A_1 \times A_2 \times \ldots \times A_n$

Relations: Examples

Let A be: {Dave, Sara, Billy}

Let B be: {cake, pie, ice cream}

Cartesian Product

$A \times B$ = {(Dave, cake), (Dave, pie), (Dave, ice cream), (Sara, cake), (Sara, pie), (Sara, ice cream), (Billy, cake), (Billy, pie), (Billy, ice cream)}.

A Binary Relation

Dessert= {(Dave, cake), (Dave, ice cream), (Sara, pie), (Sara, ice cream)}

Relations: Examples

Let A be: {Dave, Sara, Billy, Beth, Mark, Cathy, Pete}

A Ternary Relation

Child-of= {(Sara, Dave, Billy), (Beth, Mark, Cathy), (Cathy, Billy, Pete)}

Let $\mathbb{Z}^+$ = set of positive integers

A Ternary Relation

Reminder-of = {(3,2,1), (5,3,2), (7,4,3) ….}

Properties of Relations

$R \subseteq A \times A$ is ***reflexive*** iff, $\forall x \in A\ ((x, x) \in R)$.

Examples:

- *Address* defined as "lives at same address as".
- $\leq$ defined on the integers. For every integer x, $x \leq x$.

1		
	1	
		1

Properties of Relations

$R \subseteq A \times A$ is ***symmetric*** iff $\forall x, y\ ((x, y) \in R \rightarrow (y, x) \in R)$.

Examples:

- *Address* is symmetric.
- $\leq$ is not symmetric.

(a)

	*	*		
1				1
1				
	*			

(b)

Properties of Relations

$R \subseteq A \times A$ is ***transitive*** iff:

$$\forall x, y, z\ (((x, y) \in R \wedge (y, z) \in R) \rightarrow (x, z) \in R).$$

Examples:

- $<$
- *Address*
- *Mother-of* ✗
- *Ancestor-of*

Equivalence Relations

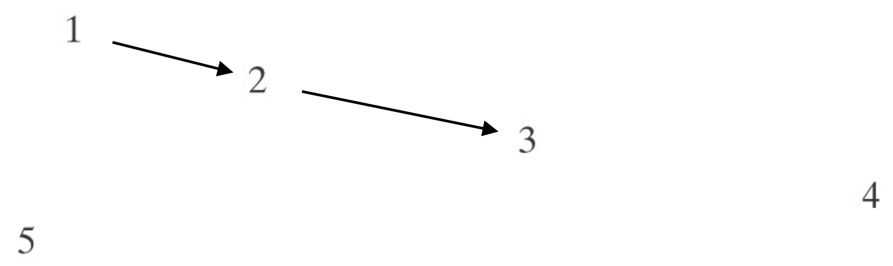
A relation $R \subseteq A \times A$ is an ***equivalence relation*** iff it is:

- reflexive,
- symmetric, and
- transitive.

Examples:

- Equality
- Lives-at-Same-Address-As
- Same-Length-As

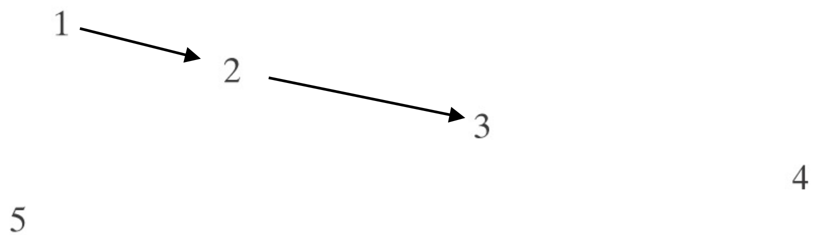
Equivalence Relation/Classes



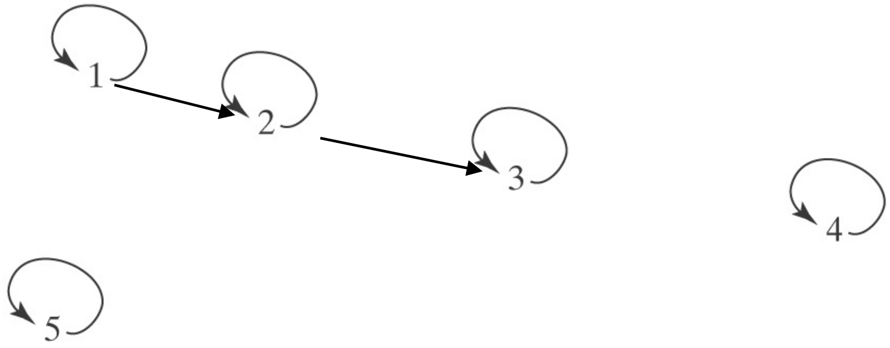
{{1,2}, {2,3}}

Make it reflexive, symmetric and transitive

Equivalence Relation/Classes



(a)

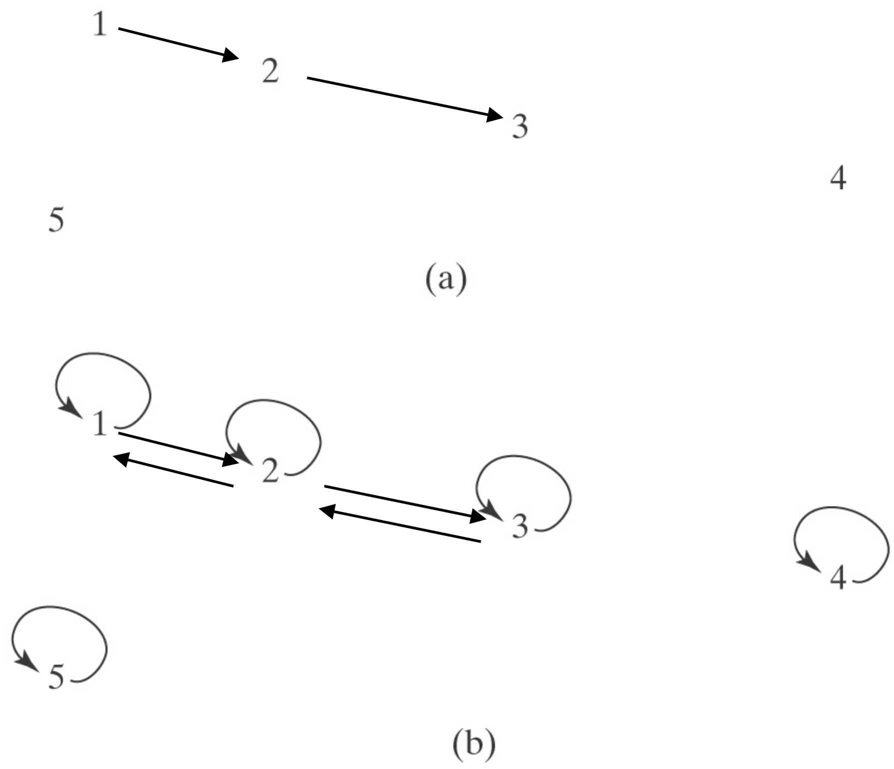


(b)

{{1,2}, {2,3}, {1,1}, {2,2}, {3,3}, {4,4}, {5,5} }

Make it reflexive, symmetric and transitive

Equivalence Relation/Classes

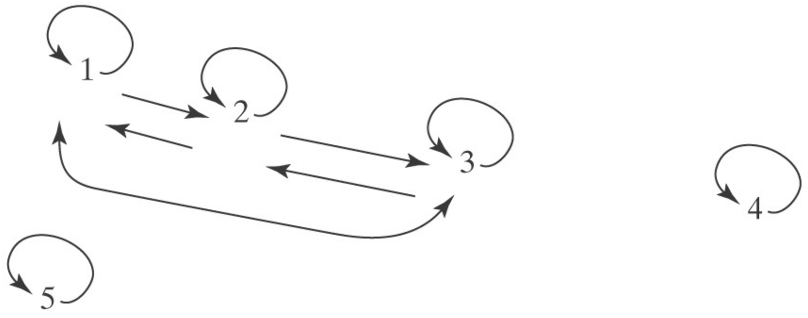


{{1,2}, {2,3}, {1,1}, {2,2}, {3,3}, {4,4}, {5,5}, {2,1}, {3,2} }

Make it reflexive, symmetric and transitive

Equivalence Relation/Classes

(c)



(d)

{{1,2}, {2,3}, {1,1}, {2,2}, {3,3}, {4,4}, {5,5}, {2,1}, {3,2}, {1,3}, {3,1 }

Make it reflexive, symmetric and transitive

Equivalence Classes

- An equivalence relation *R* on a set *S* carves *S* up into a set of clusters or islands, which we'll call ***equivalence classes***. This set of equivalence classes has the following key property:

$$\forall s, t \in S \; ((s \in class_i \wedge (s, t) \in R) \rightarrow t \in class_i).$$

- If *R* is an equivalence relation on a nonempty set *A,* then the set of equivalence classes of *R* is a partition Π of *A*. Because Π is a partition:

  (a) no element of Π is empty;

  (b) all members of Π are disjoint; and

  (c) the union of all the elements of Π equals *A*.

Closure

- A binary relation *R* on a set *A* is ***closed under*** property *P* if and only if *R* ***possesses*** *P*.

  Examples

  $<$ on the integers, *P* = transitivity

  $\leq$ on the integers, *P* = reflexive

- The ***closure*** of *R* under *P* is a smallest set that includes *R* and that is closed under *P*.

Closure

- Let $R = \{(1, 2), (2, 3), (3, 4)\}$ defined on a set $A=\{1,2,3,4\}$.
- The reflexive closure of R is:
- The transitive closure of R is:

Closure

- Let R = {(1, 2), (2, 3), (3, 4)} defined on a set A={1,2,3,4}.

- The reflexive closure of R is:
  {(1, 2), (2, 3), (3, 4), (1, 1), (2, 2), (3, 3), (4, 4)}

- The transitive closure of R is:
  {(1, 2), (2, 3), (3, 4), (1, 3), (1, 4), (2, 4)}

Functions

- A ***function*** *f* from a set *A* to a set *B* is a mapping of elements of A to elements of B such that <u>each element</u> of A maps to <u>exactly one element</u> of B.

$$\forall x \in A\ ((((x, y) \in f \wedge (x, z) \in f) \rightarrow y = z) \wedge \exists y \in B\ ((x, y) \in f\ ))$$

Let *A* be: {Dave, Sara, Billy}

Let *B* be: {cake, pie, ice cream}

- *Dessert* = {(Dave, cake), (Dave, ice cream), (Sara, pie), (Sara, ice cream)} is not a function.
- $succ(n) = n + 1$ is a function.

Types of Functions

- $f : A \rightarrow B$ is a ***unary function*** that maps from a single elements to another single element.

  Ex: succ(n) = n+1

- $f : A \text{ X B} \rightarrow C$ is a ***binary function*** that maps from an ordered pair to a value.

  Ex: +(2,3) = 5

- $f : A_1 X A_2 X \ldots X A_n \rightarrow B$ is a ***n-ary function*** that maps from a n-tuple to another single element.

  Ex: volume(h,l,w) = h l * w*

Properties of Functions

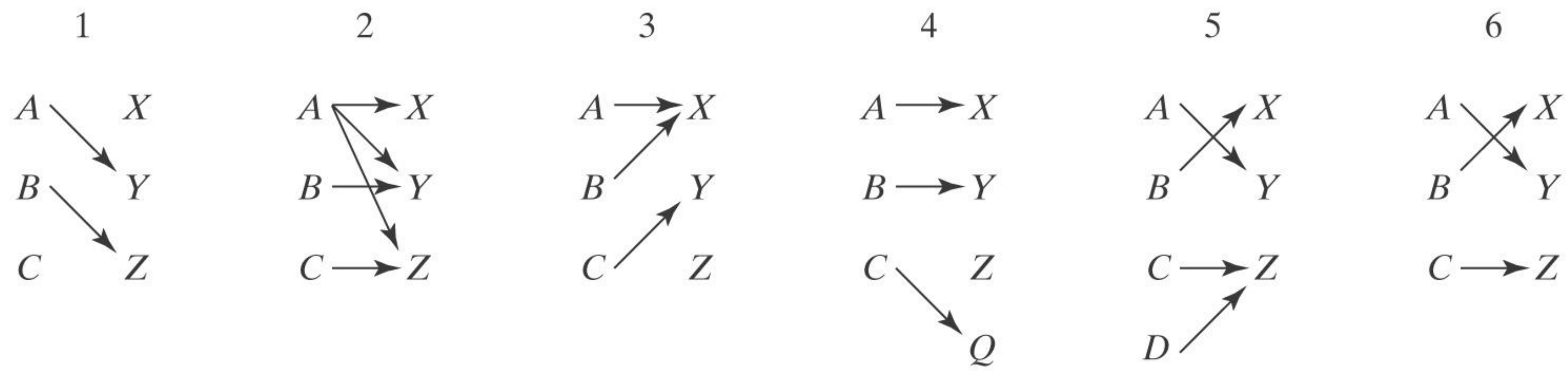
- $f : A \to B$ is a ***total function*** on A iff it is a function that is defined on all elements of A.

- $f : A \to B$ is a ***partial function*** on A iff f is a subset of $A \times B$ and f is defined on zero or more elements of A.

- $f : A \to B$ is ***one-to-one*** iff no two elements of A map to the same element of B.

- $f : A \to B$ is ***onto*** iff every element of B is the value of some element of A.

Properties of Functions



Properties of Functions on Sets

- Commutativity: $A \cup B = B \cup A$, $A \cap B = B \cap A$
- Associativity: $(A \cup B) \cup C = A \cup (B \cup C)$

  $(A \cap B) \cap C = A \cap (B \cap C)$
- Idempotency: $A \cup A = A$, $A \cap A = A$
- Distributivity: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

  $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
- Absorption: $(A \cup B) \cap A = A$, $(A \cap B) \cup A = A$
- Identity: $A \cup \varnothing = A$
- Zero: $A \cap \varnothing = \varnothing$
- Self Inverse: $\neg\neg A = A$
- *De Morgan's:* $\neg(A \cup B) = \neg A \cap \neg B$ *and*

  $\neg(A \cap B) = \neg A \cup \neg B$