Notes for COM1026 in semester test 1, condensed from lecture slides

Jim Lam

November 27, 2023

Contents

1 Set Theory

1.1 Introduction

Quick recap on Naive set theory, meaning:

1. Introducing the basic concept of sets;
2. Introduce notation;
3. Illustrate Union, Intersection, and Set Difference operations;
4. Venn diagrams as proof;
5. Power sets;
6. How to proof with more rigor.

1.2 Set Definition

Question: Why are sets relevant to computing?

We have to represent data to compute it. To group data, we put it into sets.

Some sets that I have seen before:

$$\text{The set of natural numbers: } \mathbb{N} = \{1, 2, 3, ...\}$$
$$\text{The set of integers: } \mathbb{Z} = \{..., -3, -2, -1, 0, 1, 2, 3, ...\}$$

Question: What are sets?

A set is a collection of objects, called elements of the set. The elements of a set can be anything, but they must be distinct.

1.3 Notation

Sets are denoted with capital letters, e.g. A, B, C. The elements of a set are listed inside curly brackets:

$$A = \{1, 2, 3\}$$
$$B = \{a, b, c, d, e, f, g, h\}$$

1.3.1 Cardinality

The cardinality of a set is the number of elements in the set. The cardinality of a set is denoted with vertical bars, with the hash, or alternatively, the function *card*:

$$\text{Card}(A) = |A| = 3$$
$$\#\{1, 2, 3, 4, 5\} = 5$$
$$\#A = 3$$
$$\#\mathbb{N} = \infty$$

1.3.2 Abstraction axiom

GIven a property $P(x)$, we can define a set A as:

$$A = \{x | P(x)\}$$

In other words, whatever property P, there exists a set A containing the objects that satisfy P and only these objects.

1.3.3 Set builder notation

Other than enumerating the elements of a set, there are other ways to describe a set. Verbal descriptions and adding an inclusion rule to the set builder notation are two more examples:

$$C = \{x | x \in \mathbb{N}, 0 \leq x \leq 5\}$$
$$= \{x | x \text{ is in the appropriate set}, 0 \leq x \leq 5\}$$

1.3.4 Empty set

The empty set is a set with no elements. It is denoted $\emptyset$ or $\{\}$.

$$\emptyset = \{\}$$
$$\emptyset \neq \{\emptyset\}$$

1.4 Basic set operations

1.4.1 Membership

The membership relation is denoted $\in$ and is used to indicate that an element is in a set. Conversely, $\notin$ is used to indicate that an element is not in a set.

$$\text{Let the set } A = \{1, 2, 3\}$$
$$1 \in A = \text{True}$$
$$3 \notin A = \text{False}$$

To denote a subset, we use $\subseteq$. Since a subset can include the set itself, we use $\subset$ to denote a proper subset.

$$\text{Let the set } A = \{1, 2, 3\}$$
$$\{1, 2\} \subseteq A = \text{True}$$
$$\{1, 2, 3\} \subset A = \text{False}$$

1.4.2 Union, intersection, and set difference

The union of two sets A and B is denoted $A \cup B$ and contains all elements of both sets.

The intersection $A \cap B$ contains elements common to both.

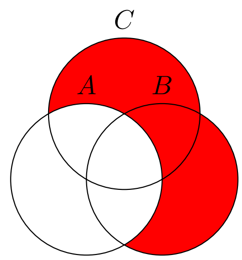
Set diffrence $A \setminus B$ contains elements in A but not in B. (Note: A must come first, otherwise the result is different.)

$$A \cup B = \{1, 2, 3, a, b, c\}$$
$$A \cap B = \{1, 2, 3\}$$
$$A \setminus B = \{1, 2, 3\}$$

1.5 Venn diagrams

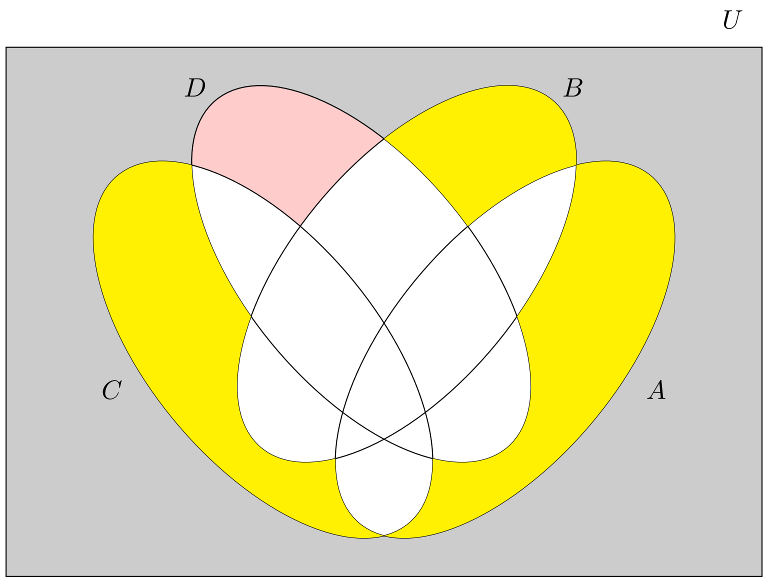
Venn diagrams are a way to visualise sets and their operations. Use circles to represent sets, and shading to distinguish areas of interest.

$$\begin{aligned} \text{Let the set } &= \{1, 2, 3\} \\ \text{Let the set } &= \{2, 3, 4\} \\ \text{Let the set } &= \{3, 4, 5\} \\ (C \cup B) \cap A &= \{4, 5\} \end{aligned}$$



Here is a more complicated example involving 4 sets:

Refrenced from `https://www.overleaf.com/latex/examples/example-venn-diagram-with-isolated-a xjptmqsjfdlc`



This covers the basics of set notation and operations.

1.6 Power sets

The power set of a set is the set of all subsets of that set.

The power set of a set A is denoted in various ways, including 2^A, $\mathcal{P}(A)$, $\mathbb{P}(A)$, or $\wp(A)$.

$$\begin{aligned} &\text{Let the set } A = \{1, 2, 3\} \\ &\wp(A) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\} \\ \text{More generally: } &\mathbb{P}(A) = \{B | B \subseteq A\} \end{aligned}$$

1.7 Proofs

1.7.1 Proof by property

Proposition (example from lecutre notes): For any sets A, B, and C:

$$A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C)$$

Recapping some of the basic properties of sets, using sets S and T, and element x:

$$\begin{aligned} &\text{Property 1:} && S \subseteq T \text{ and } T \subseteq S \iff S = T \\ &\text{Property 2:} && (\text{For any x} \in S \implies x \in T) \iff S \subseteq T \\ &\text{Property 3:} && x \in S \text{ and } x \in T \iff x \in S \cap T \\ &\text{Property 4:} && x \in S \text{ or } x \in T \iff x \in S \cup T \\ &\text{Property 5:} && x \in S \text{ and } x \notin T \iff x \in S \setminus T \\ &\text{Property 6:} && x \notin S \text{ and } x \notin T \iff x \notin T \cup S \end{aligned}$$

Using property 1, we can prove that two sets are equal by proving that each is a subset of the other.

$$\begin{aligned} &\text{Let } x \in A \setminus (B \cup C) \\ &\iff x \in A \text{ and } x \notin B \cup C \\ &\iff x \in A \text{ and } x \notin B \text{ and } x \notin C \\ &\iff x \in A \setminus B \text{ and } x \in A \setminus C \\ &\iff x \in (A \setminus B) \cap (A \setminus C) \\ &\therefore A \setminus (B \cup C) \subseteq (A \setminus B) \cap (A \setminus C) \end{aligned}$$

Doing this for the other direction:

$$
\begin{aligned}
&A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C) \\
&\text{Let } x \in (A \setminus B) \cap (A \setminus C) \\
&\iff x \in A \setminus B \text{ and } x \in A \setminus C \\
&\iff x \in A \text{ and } x \notin B \text{ and } x \in A \text{ and } x \notin C \\
&\iff x \in A \text{ and } x \notin B \cup C \\
&\iff x \in A \setminus (B \cup C) \\
&\therefore (A \setminus B) \cap (A \setminus C) \subseteq A \setminus (B \cup C)
\end{aligned}
$$

1.8 Von Neumann Ordinals

The von Neumann ordinals are a way of representing the natural numbers using sets.

let $0 = \emptyset$, $n + 1 = n \cup \{n\}$:

$$
\begin{aligned}
&0 = \emptyset \\
&1 = \{0\} = \{\emptyset\} \\
&2 = \{0, 1\} = \{\emptyset, \{\emptyset\}\} \\
&3 = \{0, 1, 2\} = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\} \\
&n = \{0, 1, 2, ..., n - 1\} \text{ you get the idea}
\end{aligned}
$$

2 Relations

2.1 Cartesian product and relations

The cartesian product of two sets A and B is the set of all ordered pairs (x, y), where: $x \in A$ and $y \in B$. The cartesian product of A and B is denoted as $A \times B$.

Example:

$$
\begin{aligned}
Books &= \{1984, \text{Concrete}, \text{Incerto}\} \\
Rating &= \{\text{Good}, \text{Evil}, \text{Unimportant}\} \\
Books \times Rating &= \{(1984, \text{Good}), (1984, \text{Evil}), (1984, \text{Unimportant})\} \\
&\cup \{(\text{Concrete}, \text{Good}), (\text{Concrete}, \text{Evil}), (\text{Concrete}, \text{Unimportant})\} \\
&\cup \{(\text{Incerto}, \text{Good}), (\text{Incerto}, \text{Evil}), (\text{Incerto}, \text{Unimportant})\}
\end{aligned}
$$

A relation R from A to B is a subset of the cartesian product of A and B. i.e. $R \subseteq A \times B$. Taking the example above, we can define a relation R from Books to Rating as:

$$R = \{(1984, \text{Good}), (\text{Concrete}, \text{Evil}), (\text{Incerto}, \text{Unimportant})\}$$

Note that R is a subset of the cartesian product of Books and Rating.

2.2 Relation notation

There are three main ways to denote a relation R from A to B:

- Ordered pairs
- Table
- Mapping

Should you want to confuse yourself even further, the infix notation is a good option.

Let the relation "likes" be defined as: $\mathbf{L} = \{(x, y)|x \text{ likes } y\}$

You can denote (x, y) as a subset of the likes by writing: x**L**y

2.3 Representing relations

Since winter is approaching, let's define a relation "likes" from people to clothing.

Let the set of people $P = \{\text{Jim}, \text{Bob}, \text{Alice}, \text{Eve}, \text{Mallory}\}$

Let the set of clothing $C = \{\text{Jacket}, \text{Scarf}, \text{Gloves}, \text{Hat}, \text{Socks}\}$

Let the relation $\mathbf{L} = \{(x, y)|x \text{ likes } y\}$

In table form:

P	L
Jim	Scarf
Bob	Jacket
Alice	Scarf
Eve	Gloves
Mallory	Hat
Mallory	Socks

Since a table popped into your view, it is as good a time as any to introduce it's application in databases.

Given this example relation of students and their various attributes, we can represent it in a table.

$$\text{Let the set of students } S = \{\text{Jim}, \text{Bob}, \text{Alice}, \text{Eve}, \text{Mallory}\}$$
$$\text{Let the set of attributes } A = \{\text{name}, \text{Age}, \text{Height}, \text{Weight}, textHappiness, \text{Net Worth}\}$$
$$\text{Let the relation } \mathbf{R} = \{(x, y) | x \text{ has attribute } y\}$$
$$\text{In table form:}$$

Name	Age	Height	Weight	Happiness	Net Worth
Jim	20	180	80	0.5	0.1
Bob	21	170	70	0.6	0.2
Alice	19	160	60	0.7	0.3
Eve	18	150	50	0.8	0.4
Mallory	17	140	40	0.9	0.5

We can extract information by getting a subset of relations.
For example: $(\text{Jim}, 180) \in \text{Height}$

2.4 Domain and range

The domain of a relation is the set of all first elements of the ordered pairs in the relation.

$$Dom(\mathbf{L}) = \{x \in A \mid \exists x \in A : (x, y) \in \mathbf{L}\}$$

The range of a relation is the set of all second elements of the ordered pairs in the relation.

$$Ran(\mathbf{L}) = \{y \in B \mid \exists y \in B : (x, y) \in \mathbf{L}\}$$

2.5 Relational composition

The relational composition of two relations $\mathbf{R}$ and $\mathbf{S}$ is the relation $\mathbf{R} \circ \mathbf{S}$ (or $\mathbf{R}; \mathbf{S}$, used to avoid confusion with function composition) defined as:

$$\mathbf{R}; \mathbf{S} = \{(x, z) \mid \exists y \in B : (x, y) \in \mathbf{R} \text{ and } (y, z) \in \mathbf{S}\}$$

EXAMPLE:

$$\mathbf{R} = \{(1, 2), (2, 3), (3, 4)\}$$
$$\mathbf{S} = \{(2, 3), (3, 4), (4, 5)\}$$
$$\mathbf{R}; \mathbf{S} = \{(1, 3), (2, 4), (3, 5)\}$$

2.6 Closures and equivalence classes

2.6.1 Property of relations

A relation $\mathbf{R}$ is reflexive if $\forall x \in A : (x, x) \in \mathbf{R}$.

A relation $\mathbf{R}$ is symmetric if $\forall x, y \in A : (x, y) \in \mathbf{R} \implies (y, x) \in \mathbf{R}$.

A relation $\mathbf{R}$ is transitive if $\forall x, y, z \in A : (x, y) \in \mathbf{R}$ and $(y, z) \in \mathbf{R} \implies (x, z) \in \mathbf{R}$.

2.6.2 Closures

The closure of a relation $\mathbf{R}$ is the smallest relation containing $\mathbf{R}$ that is transitive.

Example of constructing a reflexive closure:

$$\begin{aligned}\text{Let the relation } \mathbf{R} &= \{(1,2),(2,3),(3,4)\}\\ \text{The reflexive closure of } \mathbf{R} &= \{(1,1),(1,2),(2,3),(3,4),(2,2),(3,3),(4,4)\}\end{aligned}$$

Example of constructing a symmetric closure:

$$\begin{aligned}\text{Let the relation } \mathbf{R} &= \{(1,2),(2,3),(3,4)\}\\ \text{The symmetric closure of } \mathbf{R} &= \{(1,2),(2,3),(3,4),(2,1),(3,2),(4,3)\}\end{aligned}$$

Example of constructing a transitive closure:

$$\begin{aligned}\text{Let the relation } \mathbf{R} &= \{(1,2),(2,3),(3,4)\}\\ \text{The transitive closure of } \mathbf{R} &= \{(1,2),(2,3),(3,4),(1,3),(2,4),(1,4)\}\end{aligned}$$

2.6.3 Equivalence classes

let $\rho \subseteq A \times A$ be an equivalence relation on A, given $A \neq \emptyset$, $a \in A$ be an arbitrary element of A.

NOTE: MUST CONSTRUCT EQUIVALENCE RELATION BEFORE CONSTRUCTING EQUIVALENCE CLASSES.

$$[a]_\rho = \{x \in A \mid (a, x) \in \rho\}$$

2.6.4 More properties of relations

- A relation $\mathbf{R}$ is antisymmetric if $\forall x, y \in A : (x, y) \in \mathbf{R}$ and $(y, x) \in \mathbf{R} \implies x = y$.
- A relation $\mathbf{R}$ is a partial order if it is reflexive, antisymmetric, and transitive.
- A connex relation is a relation $\mathbf{R}$ such that $\forall x, y \in A : (x, y) \in \mathbf{R}$ or $(y, x) \in \mathbf{R}$.
- Total order means that a relation is a partial order and a connex relation.

3 Functions

3.1 Definition

What is a function in the context of discrete mathematics?

A function is a relation **f** from A to B such that every element in A is mapped to exactly one element in B, i.e.:

$\forall x \in A. \forall y, z \in B : ((x, y) \in f \land (x, z) \in f \implies y = z)$.

3.2 Notation

A function **f** from A to B is denoted as $\mathbf{f} : A \to B$.

3.3 Injective, surjective, bijective

- A function $\mathbf{f} : A \to B$ is injective if $\forall x, y \in A : \mathbf{f}(x) = \mathbf{f}(y) \implies x = y$.
- A function $\mathbf{f} : A \to B$ is surjective if $\forall y \in B : \exists x \in A : \mathbf{f}(x) = y$.
- A function $\mathbf{f} : A \to B$ is bijective if it is both injective and surjective.

3.4 Composition

The composition of two functions $\mathbf{f} : A \to B$ and $\mathbf{g} : B \to C$ is the function $\mathbf{g} \circ \mathbf{f} : A \to C$ defined as:

$$\mathbf{g} \circ \mathbf{f} = \{(x, z) \mid \exists y \in B : (x, y) \in \mathbf{f} \text{ and } (y, z) \in \mathbf{g}\}$$

3.5 Inverse

The inverse of a function $\mathbf{f} : A \to B$ is the function $\mathbf{f}^{-1} : B \to A$ defined as:

$$\mathbf{f}^{-1} = \{(y, x) \mid (x, y) \in \mathbf{f}\}$$

4 Language and regular expressions

4.1 Alphabet

We specify an Alphabet using the symbol Σ.

Examples:

$$\begin{aligned}
&\Sigma_1 = \{a, b, c, d, e, f\} \\
&\Sigma_2 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \\
&\Sigma_3 = \{S \mid S \subseteq \Sigma_1\} && \text{(The power set of } \Sigma_1, \#\Sigma_3 = 2^6 = 64) \\
&\Sigma_4 = \{(x, y) \mid x \in \Sigma_1 \land y \in \Sigma_2\} && \text{Ordered pairs of characters work as well}
\end{aligned}$$

An alphabet must be a set that contain finite elements, hence sets like $\mathbb{N} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, ...\}$ cannot be the alphabet of a language.

4.2 Strings

A string is a finite sequence of characters from an alphabet.

A string of length n is denoted as the n-tuple $w = a_1a_2a_3...a_n$, written without punctuation.

The set of all finite strings are denoted as Σ^*, and we can say that string s is in Σ^* if $s \in \Sigma^*$.

4.2.1 Empty string

Might be jarring to you, Jim, but you have the option to denote an empty string as ϵ. Will be useful later on.

4.2.2 String Operations

- Concatenation: w_1w_2 is the concatenation of strings w_1 and w_2.
- Length: $|w|$ is the length of string w. The length of concatenated strings are simply the sum of the lengths of the individual strings.

4.3 Language

A language is a set of strings.

4.4 Regular expressions

A regular expression is a string that denotes a language. Here are the rules:

- let Σ be an alphabet set.
- a denotes the language $\{a\}$, where $a \in \Sigma$, which is on its own a regular expression.
- ϵ and $\emptyset$ denote the languages $\{\epsilon\}$ and $\emptyset$, which are also regular expressions.
- Given r and s as regular expressions, the following are also regular expressions:
  - rs, $r|s$, r^*

4.4.1 Examples

The following are rules for matching strings to RegEx, let s be a string and r be a regular expression:

- s matches a when $s = a$
- ϵ matches ϵ when $s = \epsilon$
- $\emptyset$ matches nothing.
- $r|s$ matches r or s.
- rs matches r followed by s.
- r^* matches if $s = \epsilon$ or $s = s_1 s_2 ... s_n$, where s_i matches r for all i.

4.4.2 Regular languages

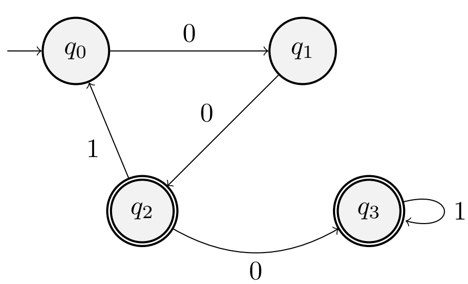
A language is regular if it is denoted by a regular expression.

$$\begin{aligned} &\text{For alphabet } \Sigma \text{ and regular expressions r:} \\ &L(r) = \{s \in \Sigma^* \mid s \textbf{ matches } r\} \end{aligned}$$

5 Finite automata

5.1 Deterministic finite automata

$$\begin{aligned} \text{States: } & Q = \{q_0, q_1, q_2, q_3\} \\ \text{Symbol: } & \Sigma = \{0, 1\} \\ \text{Transition Function} & \delta : Q \times \Sigma \to Q \\ \text{Start: } & = q_0 \in Q \\ \text{Accepting: } & = \{q_2, q_3\} \subseteq Q \\ \text{DFA}(^\epsilon)\text{: } & M = (Q, \Sigma, \delta, q_0, \{q_2, q_3\}) \end{aligned}$$



5.1.1 Criteria for a DFA

- DFAs have exactly one start state.
- May have one or more accepting states.
- For each state, there must be at most one outgoing transition **for each symbol in the alphabet.**

5.1.2 Language definition with DFAs

For automaton M, the language $L(M)$ consists of all strings s over its alphabet of input symbols statisfying:

$$q_0 \xrightarrow{\text{s}} *q \tag{1}$$
$$s = q_0, q_1, q_2, ..., q_n \text{ for the states: } q_0, q_1, q_2, ..., q_n \tag{2}$$

If (1) is the case, s is accepted by M. More formally:

$$L(M) = \{u \mid u \textbf{ is accepted by } M\}$$

5.2 Non-deterministic finite automata

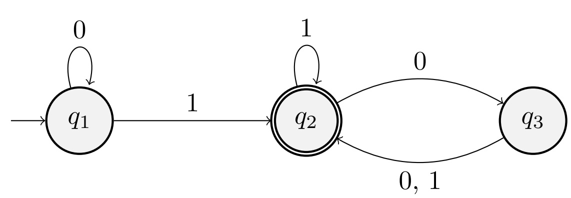
5.2.1 What is the difference?

- NDFAs can have multiple outgoing transitions for a given symbol.
- NDFAs can have ϵ-transitions, which are transitions that can be taken without consuming an input symbol.
- NDFAs can have multiple start states.
- NDFAs can have no accepting states.
- NDFAs can have multiple accepting states.

5.3 Examples

Referenced from `https://www3.nd.edu/~kogge/courses/cse30151-fa17/Public/other/tikz_tutorial.pdf`.

5.3.1 DFA 1



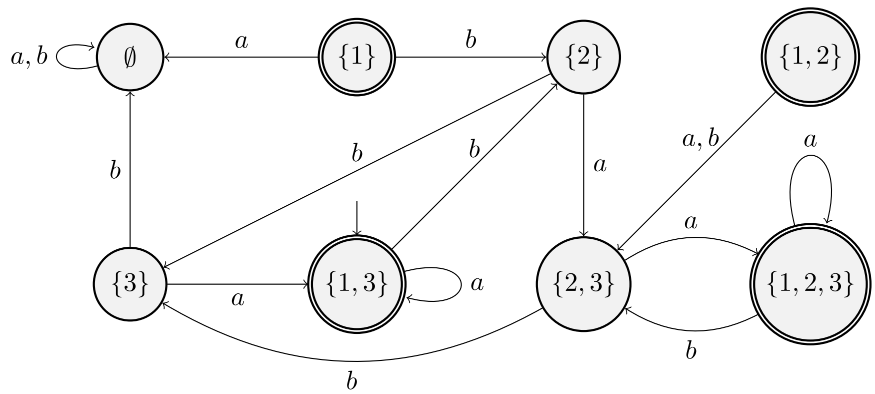
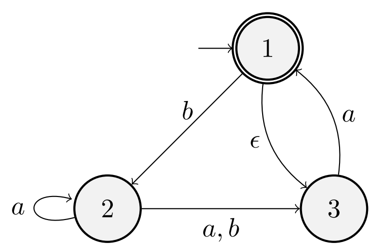


Figure 1: DFA

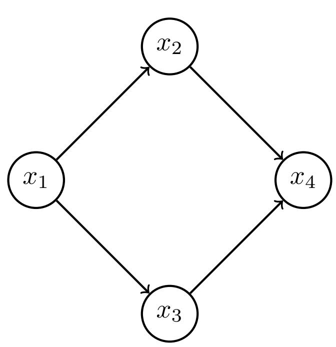
5.3.2 NFA 1



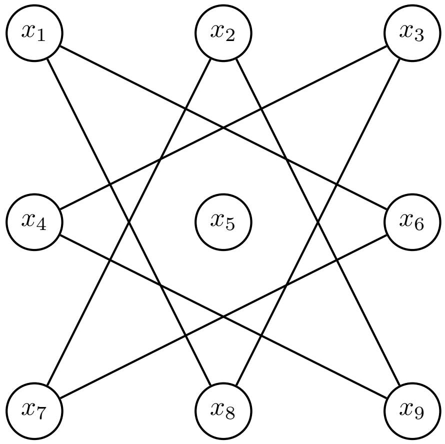
5.3.3 DFA 2

6 logic

7 Graphs and trees



Ayyy lmao.



Now onto the 3×4 example.