COMP9020 Week 1
Session 1, 2017
Numbers, Sets, Alphabets

- Textbook (R & W) - Ch. 1, Sec. 1.1-1.5, 1.7
- Problem set 1
- Supplementary Exercises Ch. 1 (R & W)

COMP9020 17s1 Course Convenor

Name:	Michael Thielscher
Email:	mit@unsw.edu.au
Consults:	Thu 3:00pm-4:00pm Fri 1:00pm-2:00pm
	Room: K17 401J/K K17 401J
Research:	Artificial Intelligence, Robotics, General Game Playing
Pastimes:	Fiction, Films, Food, Football

Course Aims

The course aims to increase your level of mathematical maturity to assist with the fundamental problem of **finding, formulating, and proving** properties of programs.
The actual content is taken from a list of subjects that constitute the basis of the tool box of every serious practitioner of computing:

numbers, sets, words	week 1
logic	week 2–3
function and relation theory	week 4–5

---

graph theory	week 8
induction and recursion	week 9
program analysis	week 10
combinatorics, probability, expectation	week 11–13

Course Material

All course information is placed on the course website

www.cse.unsw.edu.au/˜cs9020/

Slides and Problem Sets are publicly readable.

Textbook:

- KA Ross and CR Wright: Discrete Mathematics

Supplementary textbook:

- E Lehman, FT Leighton, A Meyer: Mathematics for Computer Science

Assessment Summary

100% exams:

- mid-session test (1 hour in week 6) worth up to 30 marks
- final exam (2 hours) worth up to 100 marks

Your final mark for this course will be

$$\textbf{maximum}\ (\ f\ ;\ 0.7 \cdot f + m)$$

- m — mid-session test mark
- f — final exam mark

⇒ *If you do better in the final exam, your mid-session test result will be ignored*
⇒ *The mid-session test can only improve your overall mark*

Notation for Numbers

Definition

Integers $\mathbb{Z} = \{\ldots - 2, -1, 0, 1, 2, \ldots\}$

Reals $\mathbb{R}$

$\lfloor . \rfloor : \mathbb{R} \longrightarrow \mathbb{Z}$ — **floor** of x, the greatest integer $\leq x$

$\lceil . \rceil : \mathbb{R} \longrightarrow \mathbb{Z}$ — **ceiling** of x, the least integer $\geq x$

Example

$\lfloor \pi \rfloor = 3 = \lceil e \rceil \qquad \pi, e \in \mathbb{R};\ \lfloor \pi \rfloor, \lceil e \rceil \in \mathbb{Z}$

Simple properties

- $\lfloor -x \rfloor = -\lceil x \rceil$, hence $\lceil x \rceil = -\lfloor -x \rfloor$
- $\lfloor x + t \rfloor = \lfloor x \rfloor + t$ and $\lceil x + t \rceil = \lceil x \rceil + t$, for all $t \in \mathbb{Z}$

Fact

Let $k, m, n \in \mathbb{Z}$ such that $k > 0$ and $m \geq n$. The number of multiples of k in the interval $[n \mathinner{..} m]$ is

$$\left\lfloor \frac{m}{k} \right\rfloor - \left\lfloor \frac{n-1}{k} \right\rfloor$$

Exercise

Examples

1.1.4

(b) $2\lfloor 0.6\rfloor - \lfloor 1.2\rfloor = -1$

$2\lceil 0.6\rceil - \lceil 1.2\rceil = 0$

(d) $\lceil\sqrt{3}\rceil - \lfloor\sqrt{3}\rfloor = 1$; the same for every non-integer

1.1.19(a)

Give x, y s.t. $\lfloor x\rfloor + \lfloor y\rfloor < \lfloor x+y\rfloor$

$$\lfloor 3\pi\rfloor + \lfloor e\rfloor = 9 + 2 = 11 < 12 = \lfloor 9.42\ldots + 2.71\ldots\rfloor = \lfloor 3\pi + e\rfloor$$

Exercise

Examples

$\boxed{1.1.4}$

(b) $2\lfloor 0.6 \rfloor - \lfloor 1.2 \rfloor = -1$

$2\lceil 0.6 \rceil - \lceil 1.2 \rceil = 0$

(d) $\left\lceil \sqrt{3} \right\rceil - \left\lfloor \sqrt{3} \right\rfloor = 1$; the same for every non-integer

$\boxed{1.1.19(a)}$

Give x, y s.t. $\lfloor x \rfloor + \lfloor y \rfloor < \lfloor x + y \rfloor$

$$\lfloor 3\pi \rfloor + \lfloor e \rfloor = 9 + 2 = 11 < 12 = \lfloor 9.42\ldots + 2.71\ldots \rfloor = \lfloor 3\pi + e \rfloor$$

Divisibility

Let $m, n \in \mathbb{Z}$.
'$m|n$' — m is a **divisor** of n, defined by $n = k \cdot m$ for some $k \in \mathbb{Z}$
Also stated as: 'n is divisible by m', 'm is a divisor of n'
$m \nmid n$ — negation of $m|n$

Notion of divisibility applies to all integers — positive, negative and zero.
$1|m$, $-1|m$, $m|m$, $m|-m$, for every m
$n|0$ for every n; $0 \nmid n$ except $n = 0$

Numbers > 1 divisible only by 1 and itself are called **prime**.

Greatest common divisor $\gcd(m, n)$

Numbers m and n s.t. $\gcd(m, n) = 1$ are said to be **relatively prime**.

Least common multiple $\operatorname{lcm}(m, n)$

NB

$\gcd(m, n)$ *and* $\operatorname{lcm}(m, n)$ *are always taken as positive, even if* m *or* n *is negative.*

$$\begin{aligned} \gcd(-4, 6) = \gcd(4, -6) = \gcd(-4, -6) = \gcd(4, 6) &= 2 \\ \operatorname{lcm}(-5, -5) = \dots &= 5 \end{aligned}$$

Absolute Value

$$|x| = \begin{cases} x & , \text{if } x \geq 0 \\ -x & , \text{if } x < 0 \end{cases}$$

NB

$\gcd(m, n) \cdot \operatorname{lcm}(m, n) = |m| \cdot |n|$

Examples

$\boxed{1.2.2}$ *True* or *False*. Explain briefly.
(a) $n|1$
(b) $n|n$
(c) $n|n^2$

$\boxed{1.2.7(b)}$ $\gcd(0, n) \overset{?}{=}$

$\boxed{1.2.12}$ Can two even integers be relatively prime?

$\boxed{1.2.9}$ Let m, n be positive integers.
(a) What can you say about m and n if $\operatorname{lcm}(m, n) = m \cdot n$?
(b) What if $\operatorname{lcm}(m, n) = n$?

Examples

$\boxed{\text{1.2.2}}$ *True* or *False*. Explain briefly.
(a) $n|1$ — only if $n = 1$ (for $n \in \mathbb{Z}$ also $n = -1$)
(b) $n|n$ — always
(c) $n|n^2$ — always

$\boxed{\text{1.2.7(b)}}$ $\gcd(0, n) = |n|$

$\boxed{\text{1.2.12}}$ Can two even integers be relatively prime? No. (why?)

$\boxed{\text{1.2.9}}$ Let m, n be positive integers.
(a) What can you say about m and n if $\mathrm{lcm}(m, n) = m \cdot n$?
They must be relatively prime since always $\mathrm{lcm}(m, n) = \frac{mn}{\gcd(m,n)}$
(b) What if $\mathrm{lcm}(m, n) = n$?
m must be a divisor of n

Euclid's gcd Algorithm

$$f(m,n) = \begin{cases} m & \text{if } m = n \\ f(m-n,n) & \text{if } m > n \\ f(m,n-m) & \text{if } m < n \end{cases}$$

Fact

For $m > 0, n > 0$ the algorithm always terminates. (Proof?)

Fact

For $m, n \in \mathbb{Z}$, if $m > n$ then $\gcd(m,n) = \gcd(m-n,n)$

Proof.

For all $d \in \mathbb{Z}$, ($d|m$ and $d|n$) if, and only if, ($d|m-n$ and $d|n$):

"$\Rightarrow$": if $d|m$ and $d|n$ then $m = a \cdot d$ and $n = b \cdot d$, for some a, b

then $m - n = (a-b) \cdot d$, hence $d|m-n$

"$\Leftarrow$": if $d|m-n$ and $d|n$ then $\ldots$ $d|m$ (why?)

Sets

A set is defined by the collection of its elements.
Sets are typically described by:
(a) Explicit enumeration of their elements

$$\begin{aligned} S_1 &= \{a, b, c\} = \{a, a, b, b, b, c\} \\ &= \{b, c, a\} = \ldots \quad \text{three elements} \\ S_2 &= \{a, \{a\}\} \quad \text{two elements} \\ S_3 &= \{a, b, \{a, b\}\} \quad \text{three elements} \\ S_4 &= \{\} \quad \text{zero elements} \\ S_5 &= \{\{\{\}\}\} \quad \text{one element} \\ S_6 &= \{\,\{\}, \{\{\}\}\,\} \quad \text{two elements} \end{aligned}$$

(b) Specifying the properties their elements must satisfy; the elements are taken from some 'universal' domain. A typical description involves a **logical** property $P(x)$

$$S = \{\, x : x \in X \wedge P(x) \,\} = \{\, x \in X : P(x) \,\}$$

We distinguish between an element and the set comprising this single element. Thus always $a \neq \{a\}$.
Set $\{\}$ is empty (no elements);
set $\{\{\}\}$ is nonempty — it has one element.
There is only one empty set; only one set consisting of a single a; only one set of all natural numbers.

(c) Constructions from other sets (already defined)

- Union, intersection, set difference, symmetric difference, complement
- **Power set** $\mathrm{Pow}(X) = \{\, A : A \subseteq X \,\}$
- Cartesian product (below)
- Empty set $\emptyset$
  $\emptyset \subseteq X$ for all sets X.

$S \subseteq T$ — S is a **subset** of T; includes the case of $T \subseteq T$
$S \subset T$ — a **proper** subset: $S \subseteq T$ and $S \neq T$

NB

An element of a set and a subset of that set are two different concepts

$$a \in \{a, b\}, \quad a \not\subseteq \{a, b\}; \qquad \{a\} \subseteq \{a, b\}, \quad \{a\} \notin \{a, b\}$$

Cardinality

Number of elements in a set X (various notations):

$$|X| = \#(X) = \mathrm{card}(X)$$

Fact

Always $|Pow(X)| = 2^{|X|}$

$|\emptyset| = 0 \qquad \mathrm{Pow}(\emptyset) = \{\emptyset\} \qquad |\mathrm{Pow}(\emptyset)| = 1$
$\mathrm{Pow}(\mathrm{Pow}(\emptyset)) = \{\emptyset, \{\emptyset\}\} \qquad |\mathrm{Pow}(\mathrm{Pow}(\emptyset))| = 2 \quad \ldots$

$|\{a\}| = 1 \qquad \mathrm{Pow}(\{a\}) = \{\emptyset, \{a\}\} \qquad |\mathrm{Pow}(\{a\})| = 2 \quad \ldots$

$[m\,..\,n]$ — interval of integers; it is empty if $n < m$
$|[m\,..\,n]| = n - m + 1$, for $n \geq m$

Examples

$\boxed{1.3.2}$ Find the cardinalities of sets

1. $|\{\ \frac{1}{n} : n \in [1\mathinner{.\,.}4]\ \}| \stackrel{?}{=}$
2. $|\{\ n^2 - n : n \in [0\mathinner{.\,.}4]\ \}| \stackrel{?}{=}$
3. $|\{\ \frac{1}{n^2} : n \in \mathbb{P} \wedge 2|n \wedge n < 11\ \}| \stackrel{?}{=}$
4. $|\{\ 2 + (-1)^n : n \in \mathbb{N}\ \}| \stackrel{?}{=}$

Examples

$\boxed{1.3.2}$ Find the cardinalities of sets

1. $|\{ \frac{1}{n} : n \in [1\,..\,4] \}| = 4$ — four ‘indices’, no repetitions of values
2. $|\{ n^2 - n : n \in [0\,..\,4] \}| = 4$ — one ‘repetition’ of value
3. $|\{ \frac{1}{n^2} : n \in \mathbb{P} \wedge 2|n \wedge n < 11 \}| = 5$
4. $|\{ 2 + (-1)^n : n \in \mathbb{N} \}| = 2$ — what are the two elements?

Sets of Numbers

Natural numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$
Positive integers $\mathbb{P} = \{1, 2, \ldots\}$
Common notation $\mathbb{N}_{>0} = \mathbb{Z}_{>0} = \mathbb{N} \setminus \{0\}$

Integers $\mathbb{Z} = \{\ldots, -n, -(n-1), \ldots, -1, 0, 1, 2, \ldots\}$
Rational numbers (fractions) $\mathbb{Q} = \left\{ \frac{m}{n} : m, n \in \mathbb{Z}, n \neq 0 \right\}$
Real numbers (decimal or binary expansions) $\mathbb{R}$
$r = a_1 a_2 \ldots a_k \,.\, b_1 b_2 \ldots$

In $\mathbb{P} \subset \mathbb{N} \subset \mathbb{Z}$ different symbols denote different numbers.
In $\mathbb{Q}$ and $\mathbb{R}$ the standard representation is not necessarily unique.

NB

Proper ways to introduce reals include Dedekind cuts and Cauchy sequences, neither of which will be discussed here. Natural numbers etc. are either axiomatised or constructed from sets $(0 \overset{def}{=} \{\}, n+1 \overset{def}{=} n \cup \{n\})$

NB

If we need to emphasise that an object (expression, formula) is defined through an equality we use the symbol $\overset{def}{=}$*. It denotes that the object on the left is defined by the formula/expression given on the right.*

Number sets and their containments

$$\mathbb{P} \subset \mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R}$$

Derived sets of positive numbers

$$\mathbb{P} = \mathbb{N}_{>0} = \mathbb{Z}_{>0} = \{n : n \geq 1\} \subset \mathbb{Q}_{>0} = \{r : r = \tfrac{k}{l} > 0\} \subset \mathbb{R}_{>0}$$

Derived sets of integers

$$\begin{aligned} 2\mathbb{Z} &= \{\, 2x : x \in \mathbb{Z} \,\} & &\text{the even numbers} \\ 3\mathbb{Z} + 1 &= \{\, 3x + 1 : x \in \mathbb{Z} \,\} \end{aligned}$$

Intervals of numbers (applies to any type)

$$[a, b] = \{x | a \le x \le b\}; \quad (a, b) = \{x | a < x < b\}$$

$$[a, b] \supseteq [a, b), (a, b] \supseteq (a, b)$$

NB

$(a, a) = (a, a] = [a, a) = \emptyset$; *however* $[a, a] = \{a\}$.

Intervals of $\mathbb{P}, \mathbb{N}, \mathbb{Z}$ are finite: if $m \le n$

$$[m \mathinner{..} n] = \{m, m+1, \ldots, n\} \qquad |[m \mathinner{..} n]| = n - m + 1$$

Examples

$\boxed{1.3.10}$ Number of elements in the sets

1. $\{-1, 1\}$
2. $[-1, 1]$
3. $(-1, 1)$
4. $\{\, n \in \mathbb{Z} : -1 \leq n \leq 1 \,\}$

Examples

$\boxed{1.3.10}$ Number of elements in the sets

1. $\{-1, 1\}$ — 2
2. $[-1, 1]$ — 3 (if over $\mathbb{Z}$); ∞ (if over $\mathbb{Q}$ or $\mathbb{R}$)
3. $(-1, 1)$ — 1 (if over $\mathbb{Z}$); ∞ (if over $\mathbb{Q}$ or $\mathbb{R}$)
4. $\{\, n \in \mathbb{Z} : -1 \leq n \leq 1 \,\}$ — 3

Set Operations

Union $A \cup B$; Intersection $A \cap B$

Note that there is a correspondence between set operations and logical operators (to be discussed in Week 3):
One can match set A with that subset of the universal domain, where the property a holds, then match B with the subset where b holds. Then
$A \cup B \Leftrightarrow a$ or b; $A \cap B \Leftrightarrow a$ and b

We say that A, B are **disjoint** if $A \cap B = \emptyset$

NB

$A \cup B = B \Leftrightarrow A \subseteq B \qquad A \cap B = B \Leftrightarrow A \supseteq B$

Other set operations

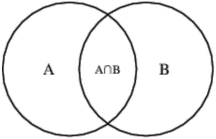
- $A \setminus B$ — **difference**, set difference, relative complement
  It corresponds (logically) to a but not b
- $A \oplus B$ — **symmetric difference**

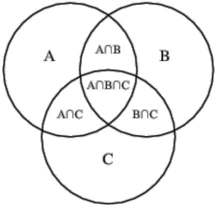
  $$A \oplus B \overset{\text{def}}{=} (A \setminus B) \cup (B \setminus A)$$

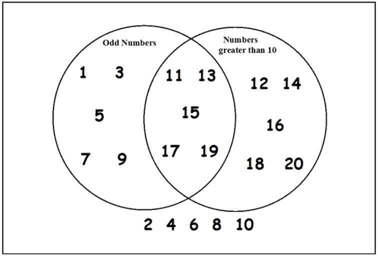
  It corresponds to a and not b or b and not a; also known as **xor (exclusive or)**
- A^c — set **complement** w.r.t. the 'universe'
  It corresponds to 'not a'

Venn Diagrams

p23–26: are a simple graphical tool to reason about the algebraic properties of set operations.







Laws of Set Operations

Commutativity	$A \cup B = B \cup A$ $A \cap B = B \cap A$
Associativity	$(A \cup B) \cup C = A \cup (B \cup C)$ $(A \cap B) \cap C = A \cap (B \cap C)$
Distribution	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
Idempotence	$A \cup A = A$ $A \cap A = A$
Identity	$A \cup \emptyset = A$ $A \cap \emptyset = \emptyset$
Double Complementation	$(A^c)^c = A$
De Morgan laws	$(A \cup B)^c = A^c \cap B^c$ $(A \cap B)^c = A^c \cup B^c$

Examples

$\boxed{1.4.4}$ $\Sigma = \{a, b\}$

(d) All subsets of Σ: ?

(e) $|\mathrm{Pow}(\Sigma)| \overset{?}{=}$

$\boxed{1.4.7}$ $A \oplus A \overset{?}{=}, \quad A \oplus \emptyset \overset{?}{=}$

$\boxed{1.4.8}$ Relate the cardinalities $|A \cup B|$, $|A \cap B|$, $|A \setminus B|$, $|A \oplus B|$, $|A|$, $|B|$

Examples

$\boxed{1.4.4}$ $\Sigma = \{a, b\}$

(d) All subsets of Σ : $\emptyset, \{a\}, \{b\}, \{a, b\}$

(e) $|\mathrm{Pow}(\Sigma)| = 4$

$\boxed{1.4.7}$ $A \oplus A \overset{?}{=} \emptyset, \quad A \oplus \emptyset \overset{?}{=} A$ for all A

$\boxed{1.4.8}$ Relate the cardinalities

$|A \cup B| = |A| + |B| - |A \cap B|$

hence $|A \cup B| + |A \cap B| = |A| + |B|$

$|A \setminus B| = |A| - |A \cap B|$

$|A \oplus B| = |A| + |B| - 2|A \cap B|$

Cartesian Product

$S \times T \overset{\text{def}}{=} \{\, (s,t) : s \in S,\ t \in T \,\}$ where (s,t) is an **ordered** pair

$\bigtimes_{i=1}^{n} S_i \overset{\text{def}}{=} \{\, (s_1, \ldots, s_n) : s_k \in S_k, \text{ for } 1 \le k \le n \,\}$

$S^2 = S \times S, \quad S^3 = S \times S \times S, \ldots, \quad S^n = \times_1^n S, \ldots$

$\emptyset \times S = \emptyset$, for every S

$|S \times T| = |S| \cdot |T|, \quad | \times_{i=1}^{n} S_i| = \prod_{i=1}^{n} |S_i|$

Formal Languages

Σ — **alphabet**, a finite, nonempty set

Examples (of various alphabets and their intended uses)

$\Sigma = \{a, b, \ldots, z\}$ for single words (in lower case)
$\Sigma = \{\sqcup, -, a, b, \ldots, z\}$ for composite terms
$\Sigma = \{0, 1\}$ for binary integers
$\Sigma = \{0, 1, \ldots, 9\}$ for decimal integers

The above cases all have a natural ordering; this is not required in general, thus the set of all Chinese characters forms a (formal) alphabet.

Definition

word — any finite string of symbols from Σ

empty word — λ

Example

$\omega = aba$, $\omega = 01101\ldots1$, etc.

length(ω) — $\#$ of symbols in ω

length$(aaa) = 3$, length$(\lambda) = 0$

The only operation on words (discussed here) is **concatenation**, written as juxtaposition $\nu\omega, \omega\nu\omega, ab\omega, \omega b\nu, \ldots$

NB

$\lambda\omega = \omega = \omega\lambda$

length$(\nu\omega)$ = length(ν) + length(ω)

Notation: Σ^k — set of all words of length k
We often identify $\Sigma^0 = \{\lambda\}$, $\Sigma^1 = \Sigma$
Σ^* — set of all words (of all lengths)
Σ^+ — set of all nonempty words (of any positive length)

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots; \quad \Sigma^{\leq n} = \bigcup_{i=0}^{n} \Sigma^i$$

$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \ldots = \Sigma^* \setminus \{\lambda\}$$

A **language** is a subset of Σ^*. Typically, only the subsets that can be formed (or described) according to certain rules are of interest. Such a collection of 'descriptive/formative' rules is called a **grammar**.

Examples: Programming languages, Database query languages

Examples

$\boxed{1.3.10}$ Number of elements in the sets (cont'd)

(e) Σ^* where $\Sigma = \{a, b, c\}$ — $|\Sigma^*| = \infty$

(f) $\{\ \omega \in \Sigma^* : \text{length}(\omega) \leq 4\ \}$ where $\Sigma = \{a, b, c\}$
$|\Sigma^{\leq 4}| = 3^0 + 3^1 + \ldots + 3^4 = \frac{3^5-1}{3-1} = \frac{243-1}{2} = 121$

Functions

We deal with functions as a set-theoretic concept, it being a special kind of correspondence (between two sets)
$f : S \longrightarrow T$ describes pairing of the sets: it means that f assigns to every element $s \in S$ a unique element $t \in T$. To emphasise that a specific element is sent, we can write $f : x \mapsto y$, which means the same as $f(x) = y$

S — **domain** of f, symbol: $\mathrm{Dom}(f)$
T — **codomain** of f, symbol: $\mathrm{Codom}(f)$
$\{ f(x) : x \in \mathrm{Dom}(f) \}$ — **image** of f, symbol: $\mathrm{Im}(f)$

$$\mathrm{Im}(f) \subseteq \mathrm{Codom}(f)$$

We observe that every function maps its domain **into** its codomain, but only **onto** its image.

Examples

$\boxed{1.5.3}$ Regarding length : $\{a, b\}^* \longrightarrow \mathbb{N}$

(c) length$(\lambda) \overset{?}{=}$

(d) Im(length) $\overset{?}{=}$

$\boxed{1.5.4}$ Σ^* as above and $g(n) \overset{\text{def}}{=} \{\, \omega \in \Sigma^* : \text{length}(\omega) \leq n \,\}$, $n \in \mathbb{N}$

Here $g(n)$ is a function that has a complex object as its value for any given argument — it maps $\mathbb{N}$ into Pow(Σ^*)

(a) $g(0) \overset{?}{=}$

(b) $g(1) \overset{?}{=}$

(c) $g(2) \overset{?}{=}$

(d) Are all $g(n)$ finite?

Examples

$\boxed{1.5.3}$ Regarding length : $\{a, b\}^* \longrightarrow \mathbb{N}$

(c) $\text{length}(\lambda) = 0$

(d) $\text{Im}(\text{length}) = \mathbb{N}$

$\boxed{1.5.4}$ Σ^* as above and $g(n) \overset{\text{def}}{=} \{\, \omega \in \Sigma^* : \text{length}(\omega) \leq n \,\}$, $n \in \mathbb{N}$

Here $g(n)$ is a function that has a complex object as its value for any given argument — it maps $\mathbb{N}$ into $\text{Pow}(\Sigma^*)$

(a) $g(0) = \{\lambda\}$

(b) $g(1) = \{\lambda, a, b\}$

(c) $g(2) = \{\lambda, a, b, aa, ab, ba, bb\}$

In general $g(n) = \bigcup_{i=0}^{n} \Sigma^i = \Sigma^{\leq n}$

(d) Are all $g(n)$ finite?

Yes; $|g(n)| = 2^0 + 2^1 + \ldots + 2^n = 2^{n+1} - 1$

Examples (cont'd)

(e) Give an example of a set in $\mathrm{Pow}(\Sigma^*)$ that is not in $\mathrm{Im}(g)$

$\boxed{1.5.6}$ Regarding $\mathrm{gcd} : \mathbb{P} \times \mathbb{P} \longrightarrow \mathbb{P}$

(c) $\mathrm{Im}(\mathrm{gcd}) \overset{?}{=}$

$\boxed{1.5.7}$

$$f(x) = \begin{cases} x^3 & x \geq 1 \\ x & 0 \leq x < 1 \\ -x^3 & x < 0 \end{cases}$$

(c) $\mathrm{Im}(f) \overset{?}{=}$

Examples (cont'd)

(e) Give an example of a set in $\mathrm{Pow}(\Sigma^*)$ that is not in $\mathrm{Im}(g)$

- any infinite subset of Σ^* (infinite language)
- any finite language that excludes some intermediate length words, e.g. $\{\lambda, a\}, \{a, b\}, \{\lambda, a, aa\}, \ldots$

$\boxed{1.5.6}$ Regarding $\mathrm{gcd} : \mathbb{P} \times \mathbb{P} \longrightarrow \mathbb{P}$

(c) $\mathrm{Im}(\mathrm{gcd}) = \mathbb{P}$ as $\mathrm{gcd}(n, n) = n$

$\boxed{1.5.7}$

$$f(x) = \begin{cases} x^3 & x \geq 1 \\ x & 0 \leq x < 1 \\ -x^3 & x < 0 \end{cases}$$

(c) $\mathrm{Im}(f) = \mathbb{R}_{\geq 0}$

Composition of Functions

Auxiliary notation

$$f : x \mapsto y, \quad f : A \mapsto B$$

The former means that x is mapped to y; the latter means that B is the image of A under f.

NB

Observe the difference between $\longrightarrow$ *and* $\mapsto$

Composition of functions is described as

$$g \circ f : x \mapsto g(f(x)), \quad \text{requiring } \mathrm{Im}(f) \subseteq \mathrm{Dom}(g)$$

If a function maps a set into itself, i.e. when $\mathrm{Dom}(f) = \mathrm{Codom}(f)$ (and thus $\mathrm{Im}(f) \subseteq \mathrm{Dom}(f)$), the function can be composed with itself — **iterated**

$$f \circ f, f \circ f \circ f, \ldots, \quad \text{also written } f^2, f^3, \ldots$$

Composition is associative

$$h \circ (g \circ f) = (h \circ g) \circ f, \quad \text{can write } h \circ g \circ f$$

Identity function on S

$$\mathrm{Id}_S(x) = x, x \in S; \mathrm{Dom}(i) = \mathrm{Codom}(i) = \mathrm{Im}(i) = S$$

For $g : S \longrightarrow T \quad g \circ \mathrm{Id}_S = g,\ \mathrm{Id}_T \circ g = g$

gcd **Example**

Reconsider gcd as a **higher-order function**, defined by

$$\mathrm{gcd}(f)(m,n) = \begin{cases} m & \text{if } m = n \\ f(m-n,n) & \text{if } m > n \\ f(m,n-m) & \text{if } m < n \end{cases}$$

Its type is now $\mathrm{gcd} : (\mathbb{P}^2 \nrightarrow \mathbb{P}) \longrightarrow (\mathbb{P}^2 \nrightarrow \mathbb{P})$
that is, it maps each partial function (from pairs of positive integers to a positive integer) to a (partial) function of the same type. The worst such function is the "nowhere defined" function

$$f_\perp(m,n) = \perp \ .$$

NB

A **partial function** *$f : S \nrightarrow T$ is a function $f : S' \longrightarrow T$ for $S' \subseteq S$*

gcd **Example cont'd**

Consider the sequence

$$f_\perp, \mathrm{gcd}(f_\perp), \mathrm{gcd}(\mathrm{gcd}(f_\perp)), \ldots, \mathrm{gcd}(\mathrm{gcd}(\ldots(f_\perp)\ldots)), \ldots$$

and observe that the i'th element of this sequence is an approximation of the gcd function that works as long as the depth of the recursion is less than $i - 1$. Since we proved that the original gcd function terminates, we can deduce that the limit of this sequence exists, and is the original gcd. It also is the **least fixpoint** of gcd i.e. the "simplest" solution f to the equation $f = \mathrm{gcd}(f)$. This, in a nutshell, explains how the semantics of recursive procedures is defined in CS. How all this works is somewhat beyond the scope of COMP9020 but still serves the purpose of motivating why we discuss functions and their composition, iteration.

Supplementary Exercises

$\boxed{\text{1.8.2(b)}}$ When is $(A \setminus B) \setminus C = A \setminus (B \setminus C)$?

$\boxed{\text{1.8.9}}$ How many third powers are $\leq 1,000,000$ and end in 9? (Solve without calculator!)

Supplementary Exercises

$\boxed{1.8.2(b)}$ When is $(A \setminus B) \setminus C = A \setminus (B \setminus C)$?
From Venn diagram
$(A \setminus B) \setminus C = A \cap B^c \cap C^c$; $A \setminus (B \setminus C) = (A \cap B^c) \cup (A \cap C)$.
Equality would require that $A \cap C \subseteq A \cap B^c \cap C^c$; however, these two sets are disjoint, thus $A \cap C = \emptyset$ is a necessary condition for the equality.
One verifies that $A \cap C = \emptyset$ is also a sufficient condition and that, in this case, both set expressions simplify to $A \setminus B$.
$\boxed{1.8.9}$ How many third powers are $\leq 1,000,000$ and end in 9?
(Solve without calculator!)
$n^3 = 9 \pmod{10}$ only when $n = 9 \pmod{10}$, and $n^3 \leq 1,000,000$ when $n \leq 100$. Hence all such n are $9, 19, \ldots, 99$.
Try the same question for n^4.

Summary

- Notation for numbers
  $\lfloor m \rfloor$, $\lceil m \rceil$, $m|n$, $|a|$, $[a, b]$, (a, b), gcd, lcm
- Sets and set operations
  $|A|$, $\in$, $\cup$, $\cap$, $\setminus$, $\oplus$, A^c, $\mathrm{Pow}(A)$, $\subseteq$, $\subset$, $\times$
- Formal languages: alphabets and words
  λ, Σ^*, Σ^+, Σ^1, $\Sigma^2, \ldots$
- Functions
  (co-)domain, image, composition $f \circ g$

COMP9020 Lecture 2–3
Session 1, 2017
Logic

- Textbook (R & W) – Ch. 2, Sec. 2.1-2.5;
  Ch. 10, Sec. 10.1-10.3
- Problem sets 2 and 3
- Supplementary Exercises Ch. 2 and 10 (R & W)
- *Guidelines for good mathematical writing*

Overview

- what's a proof?
- from English to propositional logic
- truth tables, validity, satisfiability and entailment
- *applications:* program logic, constraint satisfaction problems, reasoning about specifications, digital circuits
- proof methods
- generalisation: Boolean algebras

Proofs

A **mathematical proof** of a proposition p is a chain of logical deductions leading to p from a base set of axioms.

Example

Proposition: Every group of 6 people includes a group of 3 who each have met each other or a group of 3 who have not met a single other person in that group.
Proof: by case analysis.

But what are propositions, logical deductions, and axioms? And what is a sound case analysis?

The Real World vs Symbols

$$\begin{tikzcd}
\text{symbols} \arrow[r, "\text{symbolic manipulation}"] \arrow[d, "\text{relationship}"] & \text{symbols} \arrow[d, "\text{relationship}"] \\
\text{world} \arrow[r, "\text{physical operation}"] & \text{world}
\end{tikzcd}$$

NB

"Essentially, all models are wrong. But some are useful." (*G. Box*)

The main relationship between symbols and the world of concern in logic is that of a *sentence of a language* being *true* in the world. A sentence of a natural language (like English, Cantonese, Warlpiri) is *declarative*, or a *proposition*, if it can be meaningfully be said to be either true or false.

Examples

- Richard Nixon was president of Ecuador.
- A square root of 16 is 4.
- Euclid's program gets stuck in an infinite loop if you input 0.
- Whatever list of numbers you give as input to this program, it outputs the same list but in increasing order.
- $x^n + y^n = z^n$ has no nontrivial integer solutions for $n > 2$.

The following are *not* declarative sentences of English:

- Gubble gimble goo
- For Pete's sake, take out the garbage!
- Did you watch MediaWatch last week?
- Please waive the prerequisites for this subject for me.

Declarative sentences in natural languages can be *compound* sentences, built out of other sentences.
Propositional Logic is a formal representation of some constructions for which the truth value of the compound sentence can be determined from the truth value of its components.

- Chef is a bit of a Romeo *and* Kenny is always getting killed.
- Either Bill is a liar *or* Hillary is innocent of Whitewater.
- *It is not the case that* this program always halts.

Not all constructions of natural language are truth-functional:

- *Obama suspects that* Iran is developing nukes.
- *Chef said* they killed Kenny.
- This program always halts *because* it contains no loops.
- The disk crashed *after* I saved my file.

The Three Basic Connectives of Propositional Logic

symbol	**text**
$\wedge$	“and”, “but”, “;”, “:”
$\vee$	“or”, “either . . . or . . . ”
$\neg$	“not”, “it is not the case that”

Truth tables:

A	B	A $\wedge$ B
F	F	F
F	T	F
T	F	F
T	T	T

A	B	A $\vee$ B
F	F	F
F	T	T
T	F	T
T	T	T

A	$\neg$ A
F	T
T	F

Applications I: Program Logic

Example

```
if x > 0 or (x <= 0 and y > 100):
```

Let $p \overset{\text{def}}{=}$ (`x > 0`) and $q \overset{\text{def}}{=}$ (`y > 100`)

$p \lor (\neg p \land q)$

p	q	$p \lor (\neg p \land q)$
F	F	F
F	T	T
T	F	T
T	T	T

This is equivalent to $p \lor q$. Hence the code can be simplified to

```
if x > 0 or y > 100:
```

Somewhat more controversially, consider the following constructions:

- if A then B
- A only if B
- B if A
- A implies B
- it follows from A that B
- whenever A, B
- A is a sufficient condition for B
- B is a necessary condition for A

Each has the property that if true, and A is true, then B is true.

We can *approximate* the English meaning of these by "not (A and not B)", written $A \Rightarrow B$, which has the following truth table:

A	B	A $\Rightarrow$ B
F	F	T
F	T	T
T	F	F
T	T	T

While only an approximation to the English, 100+ years of experience have shown this to be adequate for capturing *mathematical reasoning*.

(Moral: mathematical reasoning does not need all the features of English.)

Examples

LLM: Problem 3.2

$$\begin{aligned} p &= \text{“you get an HD on your final exam”} \\ q &= \text{“you do every exercise in the book”} \\ r &= \text{“you get an HD in the course”} \end{aligned}$$

Translate into logical notation:

(a) You get an HD in the course although you do not do every exercise in the book.

(c) To get an HD in the course, you must get an HD on the exam.

(d) You get an HD on your exam, but you don't do every exercise in this book; nevertheless, you get an HD in this course.

Examples

LLM: Problem 3.2

$$\begin{aligned} p &= \text{“you get an HD on your final exam”} \\ q &= \text{“you do every exercise in the book”} \\ r &= \text{“you get an HD in the course”} \end{aligned}$$

Translate into logical notation:

(a) You get an HD in the course although you do not do every exercise in the book. $r \wedge \neg q$

(c) To get an HD in the course, you must get an HD on the exam. $r \Rightarrow p$

(d) You get an HD on your exam, but you don't do every exercise in this book; nevertheless, you get an HD in this course. $p \wedge \neg q \wedge r$

Unless

A unless B can be approximated as $\neg B \Rightarrow A$

E.g.
I go swimming unless it rains = If it is not raining I go swimming.
Correctness of the translation is perhaps easier to see in:
I don't go swimming unless the sun shines = If the sun does not shine then I don't go swimming.

Note that "I go swimming unless it rains, but sometimes I swim even though it is raining" makes sense, so the translation of "A unless B" should not imply $B \Rightarrow \neg A$.

Just in case

A just in case B usually means *A if, and only if, B*; written $A \Leftrightarrow B$

The program terminates just in case the input is a positive number.
= The program terminates if, and only if, the input is positive.

I will have an entree just in case I won't have desert.
= If I have desert I will not have an entree and vice versa.

It has the following truth table:

A	B	A ⇔ B
F	F	T
F	T	F
T	F	F
T	T	T

Same as $(A \Rightarrow B) \wedge (B \Rightarrow A)$

Propositional Logic as a Formal Language

Let $Prop = \{p, q, r, \ldots\}$ be a set of basic propositional letters. Consider the *alphabet*

$$\Sigma = Prop \cup \{\top, \bot, \neg, \wedge, \vee, \Rightarrow, \Leftrightarrow, (,)\}$$

The set of **formulae of propositional logic** is the smallest set of words over Σ such that

- $\top$, $\bot$ and all elements of $Prop$ are formulae
- If ϕ is a formula, then so is $\neg\phi$
- If ϕ and ψ are formulae, then so are $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \Rightarrow \psi)$, and $(\phi \Leftrightarrow \psi)$.

Convention: we often drop parentheses when there is no ambiguity. $\neg$ binds more tightly than $\wedge$ and $\vee$, which in turn bind more tightly than $\Rightarrow$ and $\Leftrightarrow$.

Logical Equivalence

Two formulas ϕ, ψ are **logically equivalent**, denoted $\phi \equiv \psi$ if they have the same truth value for all values of their basic propositions.

Application: If ϕ and ψ are two formulae such that $\phi \equiv \psi$, then the digital circuits corresponding to ϕ and ψ compute the same function. Thus, proving equivalence of formulas can be used to *optimise* circuits.

Some Well-Known Equivalences

Excluded Middle	$p \vee \neg p \equiv \top$
Contradiction	$p \wedge \neg p \equiv \bot$
Identity	$p \vee \bot \equiv p$
	$p \wedge \top \equiv p$
	$p \vee \top \equiv \top$
	$p \wedge \bot \equiv \bot$
Idempotence	$p \vee p \equiv p$
	$p \wedge p \equiv p$
Double Negation	$\neg\neg p \equiv p$
Commutativity	$p \vee q \equiv q \vee p$
	$p \wedge q \equiv q \wedge p$

Associativity	$(p \vee q) \vee r \equiv p \vee (q \vee r)$
	$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$
Distribution	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
De Morgan's laws	$\neg(p \wedge q) \equiv \neg p \vee \neg q$
	$\neg(p \vee q) \equiv \neg p \wedge \neg q$
Implication	$p \Rightarrow q \equiv \neg p \vee q$
	$p \Leftrightarrow q \equiv (p \Rightarrow q) \wedge (q \Rightarrow p)$

Example

$$
\begin{array}{lcl}
\multicolumn{3}{l}{((r \wedge \neg p) \vee (r \wedge q)) \vee ((\neg r \wedge \neg p) \vee (\neg r \wedge q))} \\
\equiv & (r \wedge (\neg p \vee q)) \vee (\neg r \wedge (\neg p \vee q)) & \text{Distrib.} \\
\equiv & (r \vee \neg r) \wedge (\neg p \vee q) & \text{Distrib.} \\
\equiv & \top \wedge (\neg p \vee q) & \text{Excl. Mid.} \\
\equiv & \neg p \vee q & \text{Ident.}
\end{array}
$$

Examples

$\boxed{2.2.18}$ Prove or disprove:

(a) $p \Rightarrow (q \Rightarrow r) \ \equiv \ (p \Rightarrow q) \Rightarrow (p \Rightarrow r)$

(c) $(p \Rightarrow q) \Rightarrow r \ \equiv \ p \Rightarrow (q \Rightarrow r)$

Examples

$\boxed{2.2.18}$ Prove or disprove:

(a) $(p \Rightarrow q) \Rightarrow (p \Rightarrow r)$

$$\begin{aligned}
&\equiv \neg(p \Rightarrow q) \vee (\neg p \vee r)\\
&\equiv (p \wedge \neg q) \vee \neg p \vee r\\
&\equiv (p \vee \neg p \vee r) \wedge (\neg q \vee \neg p \vee r)\\
&\equiv \top \wedge (\neg p \vee \neg q \vee r)\\
&\equiv p \Rightarrow (\neg q \vee r)\\
&\equiv p \Rightarrow (q \Rightarrow r)
\end{aligned}$$

(c) $(p \Rightarrow q) \Rightarrow r \;\equiv\; p \Rightarrow (q \Rightarrow r)$

Counterexample:

p	q	r	$(p \Rightarrow q) \Rightarrow r$	$p \Rightarrow (q \Rightarrow r)$
F	T	F	F	T

Satisfiability of Formulas

A formula is **satisfiable**, if it evaluates to T for *some* assignment of truth values to its basic propositions.

Example

A	B	$\neg(A \Rightarrow B)$
F	F	F
F	T	F
T	F	T
T	T	F

Applications II: Constraint Satisfaction Problems

These are problems such as timetabling, activity planning, etc. Many can be understood as showing that a formula is satisfiable.

Example

You are planning a party, but your friends are a bit touchy about who will be there.

1. If John comes, he will get very hostile if Sarah is there.
2. Sarah will only come if Kim will be there also.
3. Kim says she will not come unless John does.

Who can you invite without making someone unhappy?

Translation to logic: let J, S, K represent "John (Sarah, Kim) comes to the party". Then the constraints are:

1. $J \Rightarrow \neg S$
2. $S \Rightarrow K$
3. $K \Rightarrow J$

Thus, for a successful party to be possible, we want the formula $\phi = (J \Rightarrow \neg S) \wedge (S \Rightarrow K) \wedge (K \Rightarrow J)$ to be satisfiable.

Truth values for J, S, K making this true are called *satisfying assignments*, or *models*.

We figure out where the conjuncts are false, below. (so blank = T)

J	K	S	$J \Rightarrow \neg S$	$S \Rightarrow K$	$K \Rightarrow J$	ϕ
F	F	F				
F	F	T		F		F
F	T	F			F	F
F	T	T			F	F
T	F	F				
T	F	T	F	F		F
T	T	F				
T	T	T	F			F

Conclusion: a party satisfying the constraints can be held. Invite nobody, or invite John only, or invite Kim and John.

Exercise

2.7.14 (supp)

Which of the following formulae are *always* true?

(a) $(p \wedge (p \Rightarrow q)) \ \Rightarrow \ q$ — always true

(b) $((p \vee q) \wedge \neg p) \ \Rightarrow \ \neg q$ — not always true

(e) $((p \Rightarrow q) \vee (q \Rightarrow r)) \ \Rightarrow \ (p \Rightarrow r)$ — not always true

(f) $(p \wedge q) \Rightarrow q$ — always true

Exercise

2.7.14 (supp)

Which of the following formulae are *always* true?

(a) $(p \wedge (p \Rightarrow q)) \ \Rightarrow \ q$ — always true

(b) $((p \vee q) \wedge \neg p) \ \Rightarrow \ \neg q$ — not always true

(e) $((p \Rightarrow q) \vee (q \Rightarrow r)) \ \Rightarrow \ (p \Rightarrow r)$ — not always true

(f) $(p \wedge q) \Rightarrow q$ — always true

Validity, Entailment, Arguments

An *argument* consists of a set of declarative sentences called *premises* and a declarative sentence called the *conclusion*.

Example

Premises: Frank took the Ford or the Toyota.
If Frank took the Ford he will be late.
Frank is not late.

Conclusion: Frank took the Toyota

An argument is *valid* if the conclusions are true *whenever* all the premises are true. Thus: if we believe the premises, we should also believe the conclusion.
(Note: we don't care what happens when one of the premises is false.)
Other ways of saying the same thing:

- The conclusion *logically follows* from the premises.
- The conclusion is a *logical consequence* of the premises.
- The premises **entail** the conclusion.

The argument above is valid. The following is invalid:

Example

Premises:	Frank took the Ford or the Toyota.
	If Frank took the Ford he will be late.
	Frank is late.
Conclusion:	Frank took the Ford.

For arguments in propositional logic, we can capture validity as follows:
Let $\phi_1, \ldots, \phi_n$ and ϕ be formulae of propositional logic. Draw a truth table with columns for each of $\phi_1, \ldots, \phi_n$ and ϕ.
The argument with premises $\phi_1, \ldots, \phi_n$ and conclusion ϕ is valid, denoted

$$\phi_1, \ldots, \phi_n \models \phi$$

if in every row of the truth table where $\phi_1, \ldots, \phi_n$ are all true, ϕ is true also.

We mark only true locations (blank = F)

Frd	$Tyta$	$Late$	$Frd \lor Tyta$	$Frd \Rightarrow Late$	$\neg Late$	$Tyta$
F	F	F		T	T	
F	F	T		T		
F	T	F	T	T	T	T
F	T	T	T	T		T
T	F	F	T		T	
T	F	T	T	T		
T	T	F	T		T	T
T	T	T	T	T		T

This shows $Frd \lor Tyta$, $Frd \Rightarrow Late$, $\neg Late \models Tyta$

The following row shows $Frd \vee Tyta$, $Frd \Rightarrow Late$, $Late \not\models Frd$

Frd	$Tyta$	$Late$	$Frd \vee Tyta$	$Frd \Rightarrow Late$	$Late$	Frd
F	T	T	T	T	T	F

Applications III: Reasoning About Requirements/Specifications

Suppose a set of English language requirements R for a software/hardware system can be formalised by a set of formulae $\{\phi_1, \ldots \phi_n\}$.
Suppose C is a statement formalised by a formula ψ. Then

1. The requirements cannot be implemented if $\phi_1 \wedge \ldots \wedge \phi_n$ is not satisfiable.
2. If $\phi_1, \ldots \phi_n \models \psi$ then every correct implementation of the requirements R will be such that C is always true in the resulting system.
3. If $\phi_1, \ldots \phi_{n-1} \models \phi_n$, then the condition ϕ_n of the specification is redundant and need not be stated in the specification.

Example

Requirements R: A burglar alarm system for a house is to operate as follows. The alarm should not sound unless the system has been armed or there is a fire. If the system has been armed and a door is disturbed, the alarm should ring. Irrespective of whether the system has been armed, the alarm should go off when there is a fire.

Conclusion C: If the alarm is ringing and there is no fire, then the system must have been armed.

Questions

1. Will every system correctly implementing requirements R satisfy C?
2. Is the final sentence of the requirements redundant?

Expressing the requirements as formulas of propositional logic, with

- S = the alarm sounds = the alarm rings
- A = the system is armed
- D = a door is disturbed
- F = there is a fire

we get

Requirements:

1. $S \Rightarrow (A \vee F)$
2. $(A \wedge D) \Rightarrow S$
3. $F \Rightarrow S$

Conclusion: $(S \wedge \neg F) \Rightarrow A$

Our two questions then correspond to

1. Does $S \Rightarrow (A \vee F), (A \wedge D) \Rightarrow S, F \Rightarrow S \models (S \wedge \neg F) \Rightarrow A$?
2. Does $S \Rightarrow (A \vee F), (A \wedge D) \Rightarrow S \models F \Rightarrow S$?

Answers: problem set 2, exercise 2

Validity of Formulas

A formula ϕ is **valid**, or a **tautology**, denoted $\models \phi$, if it evaluates to T for *all* assignments of truth values to its basic propositions.

Example

A	B	$(A \Rightarrow B) \Rightarrow (\neg B \Rightarrow \neg A)$
F	F	T
F	T	T
T	F	T
T	T	T

Validity, Equivalence and Entailment

Theorem

The following are equivalent:

- $\phi_1, \ldots \phi_n \models \psi$
- $\models (\phi_1 \wedge \ldots \wedge \phi_n) \Rightarrow \psi$
- $\models \phi_1 \Rightarrow (\phi_2 \Rightarrow \ldots (\phi_n \Rightarrow \psi) \ldots)$

Theorem

$\phi \equiv \psi$ *if and only if* $\models \phi \Leftrightarrow \psi$

Quantifiers

We've made quite a few statements of the kind

> *"If there exists a satisfying assignment . . . "*

or

> *"Every natural number greater than 2 . . . "*

without formally capturing these quantitative aspects.

Notation: $\forall$ means "for all" and $\exists$ means "there exist(s)"

Example

Goldbach's conjecture

$$\forall n \in 2\mathbb{N}\,(n > 2 \Rightarrow \exists p, q \in \mathbb{N}\,(p, q \in \textsc{Primes} \land n = p + q))$$

Proof Rules and Methods: Proof of the Contrapositive

We want to prove $A \Rightarrow B$.
To prove it, we show $\neg B \Rightarrow \neg A$ and invoke the equivalence $(A \Rightarrow B) \equiv (\neg B \Rightarrow \neg A)$.

Example

$\forall m, n \in \mathbb{N}\,(m + n \geq 73 \;\Rightarrow\; m \geq 37 \vee n \geq 37)$

Proof Rules and Methods: Proof by Contradiction

We want to prove A.
To prove it, we assume $\neg A$, and derive both B and $\neg B$ for some proposition B.
(Hard part: working out what B should be.)

Examples

- $\sqrt{2}$ is irrational
- There exist an infinite number of primes

Proof Rules and Methods: Proof by Cases

We want to prove that A. To prove it, we find a set of cases $B_1, B_2, \ldots, B_n$ such that

1. $B_1 \vee \ldots \vee B_n$, and
2. $B_i \Rightarrow A$ for each $i = 1..n$.

(Hard Part: working out what the B_i should be.)

(Comment: often $n = 2$ and $B_2 = \neg B_1$, so $B_1 \vee B_2 = B_1 \vee \neg B_1$ holds trivially.)

Example

$|x + y| \leq |x| + |y|$ for all $x, y \in \mathbb{R}$.

Recall:

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$$

Substitution

Substitution is the process of replacing every occurrence of some symbol by an expression.

Examples

The result of substituting 3 for x in

$$x^2 + 7y = 2xz$$

is

$$3^2 + 7y = 2 \cdot 3 \cdot z$$

The result of substituting $2k + 3$ for x in

$$x^2 + 7y = 2xz$$

is

$$(2k + 3)^2 + 7y = 2 \cdot (2k + 3) \cdot z$$

We can substitute logical expressions for logical variables:

Example

The result of substituting $P \wedge Q$ for A in

$$(A \wedge B) \Rightarrow A$$

is

$$((P \wedge Q) \wedge B) \Rightarrow (P \wedge Q)$$

Substitution Rules

(a) If we substitute an expression for *all* occurrences of a logical variable in a tautology then the result is still a tautology.

If $\models \phi(P)$ then $\models \phi(\alpha)$.

Examples

$\models P \Rightarrow (P \vee Q)$, so

$$\models (A \vee B) \Rightarrow ((A \vee B) \vee Q)$$

$\boxed{2.5.7}$

$\models \neg Q \Rightarrow (Q \Rightarrow P)$, so

$$\models \neg(P \Rightarrow Q) \Rightarrow ((P \Rightarrow Q) \Rightarrow P)$$

(b) If a logical formula ϕ contains a formula α, and we replace (an occurrence of) α by a logically equivalent formula β, then the result is logically equivalent to ϕ.

If $\alpha \equiv \beta$ then $\phi(\alpha) \equiv \phi(\beta)$.

Example

$P \Rightarrow Q \equiv \neg P \vee Q$, so

$$Q \Rightarrow (P \Rightarrow Q) \equiv Q \Rightarrow (\neg P \vee Q)$$

Boolean Functions

Formulae can be viewed as **Boolean functions** mapping valuations of their propositional letters to truth values.

A Boolean function of one variable is also called **unary**.
A function of two variables is called **binary**.
A function of n input variables is called **n-ary**.

Question

How many unary Boolean functions are there?
How many binary functions? n-ary?

Question

What connectives do we need to express all of them?

Boolean Arithmetic

Consider truth values with operations $\wedge, \vee, \neg$ as an **algebraic structure**:

- $\mathbb{B} = \{0, 1\}$ with 'Boolean' arithmetic

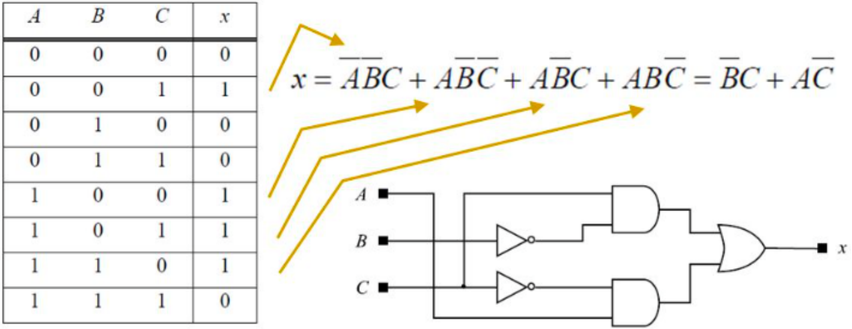
$$a \cdot b, \;\; a + b, \;\; \bar{a} = 1 - a$$

NB

We often write pq for $p \cdot q$.
In electrical and computer engineering, the notation $\overline{p}$ is more common than p', which is often used in mathematics.
Observe that using $\overline{(\cdot)}$ obviates the need for some parentheses.

Applications IV: Digital Circuits

A formula can be viewed as defining a digital circuit, which computes a Boolean function of the input propositions. The function is given by the truth table of the formula.



A	B	C	x
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

$$x = \overline{A}\,\overline{B}C + A\overline{B}\,\overline{C} + A\overline{B}C + AB\overline{C} = \overline{B}C + A\overline{C}$$

Definition: Boolean Algebra

Every structure consisting of a set T with operations *join*: $a, b \mapsto a + b$, *meet*: $a, b \mapsto a \cdot b$ and *complementation*: $a \mapsto \bar{a}$, and distinct elements 0 and 1, is called a **Boolean algebra** if it satisfies the following laws, for all $x, y, z \in T$:

commutative:
- $x + y = y + x$
- $x \cdot y = y \cdot x$

associative:
- $(x + y) + z = x + (y + z)$
- $(x \cdot y) \cdot z = x \cdot (y \cdot z)$

distributive:
- $x + (y \cdot z) = (x + y) \cdot (x + z)$
- $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$

identity: $x + 0 = x, \quad x \cdot 1 = x$

complementation: $x + \overline{x} = 1, \quad x \cdot \overline{x} = 0$

Boolean Expressions

Boolean algebra (BA) notation for propositional formulae:

	PL	**BA**
propositional atoms	$p, q, \ldots$	$p, q, \ldots$
conjunction	$p \wedge q$	$p \cdot q$ or pq
disjunction	$p \vee q$	$p + q$
negation	$\neg p$	$\overline{p}$

Example

$$(p \vee q) \wedge (\neg(p \vee \neg q) \vee \neg(\neg(r \wedge (p \vee \neg q))))$$

$$\equiv (p + q) \cdot \big(\overline{p + \overline{q}} + \overline{\overline{r \cdot (p + \overline{q})}}\big)$$

$$\equiv (p + q)\big(\overline{p + \overline{q}} + \overline{\overline{r(p + \overline{q})}})\big)$$

Terminology and Rules

- A **literal** is an expression p or $\overline{p}$, where p is a propositional atom.
- An expression is in CNF (conjunctive normal form) if it has the form

$$\prod_i C_i$$

  where each **clause** C_i is a disjunction of literals e.g. $p + q + \overline{r}$.
- An expression is in DNF (disjunctive normal form) if it has the form

$$\sum_i C_i$$

  where each clause C_i is a conjunction of literals e.g. $pq\overline{r}$.

- CNF and DNF are named after their top level operators; no deeper nesting of $\cdot$ or $+$ is permitted.
- We can assume in every clause (disjunct for the CNF, conjunct for the DNF) any given variable (literal) appears only once; preferably, no literal and its negation together.
  - $x + x = x, \quad xx = x$
  - $x\overline{x} = 0, \quad x + \overline{x} = 1$
  - $x \cdot 0 = 0$, $x \cdot 1 = x$, $x + 0 = x$, $x + 1 = 1$
- A preferred form for an expression is DNF, with as few terms as possible. In deriving such minimal simplifications the two basic rules are
  - $x + xy \Leftrightarrow x$ absorption
  - $xy + x\overline{y} \Leftrightarrow x$ combining the opposites

Theorem

For every Boolean expression ϕ, there exists an equivalent expression in conjunctive normal form and an equivalent expression in disjunctive normal form.

Proof.

We show how to apply the equivalences already introduced to convert any given formula to an equivalent one in CNF, DNF is similar. □

Step 1: Push Negations Down

Using **De Morgan's** laws and the **double negation** rule

$$\overline{x+y} = \overline{x} \cdot \overline{y}$$
$$\overline{x \cdot y} = \overline{x} + \overline{y}$$
$$\overline{\overline{x}} = x$$

we push negations down towards the atoms until we obtain a formula that is formed from literals using only $\cdot$ and $+$.

Step 2: Use Distribution to Convert to CNF

Using the distribution rules

$$x + (y_1 \cdot \ldots \cdot y_n) = (x + y_1) \cdot \ldots \cdot (x + y_n)$$
$$(y_1 \cdot \ldots \cdot y_n) + x = (y_1 + x) \cdot \ldots \cdot (y_n + x)$$

we obtain a CNF formula.

CNF/DNF in Propositional Logic

Using the equivalence

$$A \Rightarrow B \quad \equiv \quad \neg A \vee B$$

we first eliminate all occurrences of $\Rightarrow$

Example

$$\neg(\neg p \wedge ((r \wedge s) \Rightarrow q)) \equiv \neg(\neg p \wedge (\neg(r \wedge s) \vee q))$$

Step 1:

Example

$$
\begin{aligned}
\overline{\overline{p}(\overline{rs}+q)} &= \overline{\overline{p}} + \overline{\overline{rs}+q} \\
&= p + \overline{\overline{rs}} \cdot \overline{q} \\
&= p + rs\overline{q}
\end{aligned}
$$

Step 2:

Example

$$
\begin{aligned}
p + rs\overline{q} &= (p+r)(p+s\overline{q}) \\
&= (p+r)(p+s)(p+\overline{q}) \qquad \text{CNF}
\end{aligned}
$$

Canonical Form DNF

Given a Boolean expression E, we can construct an equivalent DNF E^{dnf} from the lines of the truth table where E is true:
Given an assignment π of $0, 1$ to variables $x_1 \ldots x_i$, define the literal

$$\ell_i = \begin{cases} x_i & \text{if } \pi(x_i) = 1 \\ \overline{x_i} & \text{if } \pi(x_i) = 0 \end{cases}$$

and a product $t_\pi = \ell_1 \cdot \ell_2 \cdot \ldots \cdot \ell_n$.

Example

If $\pi(x_1) = 1$ and $\pi(x_2) = 0$ then $t_\pi = x_1 \cdot \overline{x_2}$

The **canonical DNF** of E is

$$E^{dnf} = \sum_{E(\pi)=1} t_\pi$$

Example

If E is defined by

x	y	E
0	0	1
0	1	0
1	0	1
1	1	1

then $E^{dnf} = \overline{x}\overline{y} + x\overline{y} + xy$

Note that this can be simplified to either

$$\overline{y} + xy$$

or

$$\overline{x}\overline{y} \; + x$$

Exercise

$\boxed{10.2.3}$ Find the canonical DNF form of each of the following expressions in variables x, y, z

- xy
- $\overline{z}$
- $xy + \overline{z}$
- 1

Exercise

$\boxed{10.2.3}$ Find the canonical DNF form of the following expressions Remember that these are meant as expressions in three variables x, y, z.

$$\begin{aligned} xy &= xy \cdot 1 = xy \cdot (z + \overline{z}) = xyz + xy\overline{z} \\ \overline{z} &= xy\overline{z} + x\overline{y}\overline{z} + \overline{x}y\overline{z} + \overline{x}\overline{y}\overline{z} \\ xy + \overline{z} &= \text{combine the 6 so-called } \textit{min-terms} \text{ above} \\ 1 &= \text{sum of all 8 possible min-terms: } xyz + \overline{x}yz + \ldots + \overline{x}\overline{y}\overline{z} \end{aligned}$$

NB

Obviously, preferred in practice are the expressions with as few terms as possible.
However, the existence of a uniform representation as the sum of (quite a few) min-terms is important for proving the properties of Boolean expressions.

Boolean Algebras in Computer Science

Several data structures have natural operations following essentially the same rules as logical $\wedge$, $\vee$ and $\neg$.

- n-tuples of 0's and 1's with Boolean operations, e.g.

$$(1,0,0,1) \vee (1,1,0,0) = (1,1,0,1)$$

- $\mathrm{Pow}(S)$ — subsets of S

$$A \cap B, A \cup B, A^c = S \setminus A$$

Example

$\boxed{10.1.1}$ Define a Boolean algebra for the power set $\mathrm{Pow}(S)$ of $S = \{a, b, c\}$

join: $X, Y \mapsto X \cup Y$

meet: $X, Y \mapsto X \cap Y$

complementation: $X \mapsto \{a, b, c\} \setminus X$

$0 \overset{\text{def}}{=} \emptyset$

$1 \overset{\text{def}}{=} \{a, b, c\}$

Exercise:

Verify that all Boolean algebra laws (cf. slide 53) hold for $X, Y, Z \in \mathrm{Pow}(\{a, b, c\})$

Example

$\boxed{10.1.1}$ Define a Boolean algebra for the power set $\mathrm{Pow}(S)$ of
$S = \{a, b, c\}$
join: $X, Y \mapsto X \cup Y$
meet: $X, Y \mapsto X \cap Y$
complementation: $X \mapsto \{a, b, c\} \setminus X$
$0 \overset{\text{def}}{=} \emptyset$
$1 \overset{\text{def}}{=} \{a, b, c\}$

Exercise:
Verify that all Boolean algebra laws (cf. slide 53) hold for
$X, Y, Z \in \mathrm{Pow}(\{a, b, c\})$

More Examples of Boolean Algebras in CS

- Functions from any set S to $\mathbb{B}$; their set is denoted $\mathrm{Map}(S, \mathbb{B})$

  If $f, g : S \longrightarrow \mathbb{B}$ then
  - $f + g : S \longrightarrow \mathbb{B}$ is defined by $s \mapsto f(s) + g(s)$
  - $f \cdot g : S \longrightarrow \mathbb{B}$ is defined by $s \mapsto f(s) \cdot g(s)$

  There are 2^n such functions for $|S| = n$

- All Boolean functions of n variables, e.g.

$$(p_1, p_2, p_3) \mapsto (p_1 + \overline{p_2}) \cdot (p_1 + p_3) \cdot \overline{p_2 + p_3}$$

  There are 2^{2^n} of them; their collection is denoted $\textsc{Bool}(n)$

Every finite Boolean algebra satisfies: $|T| = 2^k$ for some k.
All algebras with the same number of elements are **isomorphic**, i.e. "structurally similar", written $\simeq$. Therefore, studying one such algebra describes properties of all.
A cartesian product of Boolean algebras is again a Boolean algebra. We write

$$\mathbb{B}^k = \mathbb{B} \times \ldots \times \mathbb{B}$$

The algebras mentioned above are all of this form

- n-tuples $\simeq \mathbb{B}^n$
- $\mathrm{Pow}(S) \simeq \mathbb{B}^{|S|}$
- $\mathrm{Map}(S, \mathbb{B}) \simeq \mathbb{B}^{|S|}$
- $\textsc{bool}(n) \simeq \mathbb{B}^{2^n}$

NB

Boolean algebra as the calculus of two values is fundamental to computer circuits and computer programming.
Example: Encoding subsets as bit vectors.

Summary

- Logic: syntax, truth tables; $\land$, $\lor$, $\neg$, $\Rightarrow$, $\Leftrightarrow$, $\top$, $\bot$
- Valid formulae (tautologies), satisfiable formulae
- Entailment $\models$, equivalence $\equiv$
  some well-known equivalences (slides 19 and 20)
- Proof methods: contrapositive, by contradiction, by cases
- Boolean algebra
- CNF, DNF, canonical form

Supplementary reading [LLM]

- Ch. 1, Sec. 1.5-1.9 (more about good proofs)
- Ch. 3, Sec. 3.3 (more about proving equivalences of formulae)

COMP9020 Lecture 4-5
Session 1, 2017
Functions and Relations

- Textbook - Ch. 3, Sec. 3.1, 3.3–3.4; Ch. 11, Sec. 11.1–11.2
- Problem sets 4 and 5
- Supplementary Exercises Ch. 3 and 11 (R & W)

NB

Mid-session test: Friday, 7 April, 2:30pm (1hr)

Properties of Functions

Recall:
$f : S \longrightarrow T$
S — **domain** of f, symbol: $\mathrm{Dom}(f)$
T — **codomain** of f, symbol: $\mathrm{Codom}(f)$
$\{ f(x) : x \in \mathrm{Dom}(f) \}$ — **image** of f, symbol: $\mathrm{Im}(f)$

Function is called **onto** (or **surjective**) if every element of the codomain is mapped to by at least on x in the domain, i.e.

$$\mathrm{Im}(f) = T$$

Examples (of functions that are not onto)

- $f : \mathbb{N} \longrightarrow \mathbb{N}$ with $f(x) \mapsto x^2$
- $f : \{a, \ldots, z\}^* \longrightarrow \{a, \ldots, z\}^*$ with $f(\omega) \mapsto a\omega e$

1-1 Functions

Function is called **1–1** (**one-to-one**) or **injective** if different x implies different $f(x)$, i.e.

$$f(x) = f(y) \Rightarrow x = y$$

Examples (of functions that are not 1–1)

- absolute value
- floor, ceiling
- length of a word

Inverse Functions

Inverse function — $f^{-1} : T \longrightarrow S$;
for a given $f : S \longrightarrow T$ exists exactly
when f is both 1–1 and onto.
Image of a subdomain A under a function

$$f(A) = \{ f(s) : s \in A \} = \{ t \in T : t = f(s) \text{ for some } s \in A \}$$

Inverse image — $f^{\leftarrow}(B) = \{ s \in S : f(s) \in B \} \subseteq S$;
it is defined for every f
If f^{-1} exists then $f^{\leftarrow}(B) = f^{-1}(B)$

$f(\emptyset) = \emptyset, f^{\leftarrow}(\emptyset) = \emptyset$

Examples

$\boxed{1.7.5}$ f and g are 'shift' functions $\mathbb{N} \longrightarrow \mathbb{N}$ defined by $f(n) = n + 1$, and $g(n) = \max(0, n - 1)$

(c) Is f 1–1? onto?
(d) Is g 1-1? onto?
(e) Do f and g commute, i.e. $\forall n((f \circ g)(n) = (g \circ f)(n))$?

Examples

$\boxed{1.7.5}$ f and g are 'shift' functions $\mathbb{N} \longrightarrow \mathbb{N}$ defined by $f(n) = n + 1$, and $g(n) = \max(0, n - 1)$

(c) f is 1–1, not onto: $f(\mathbb{N}) = \mathbb{N} \setminus \{0\} = \mathbb{P}$

(d) g is onto, not 1–1: $g(0) = g(1)$

(e) f and g do not commute:
$g \circ f : n \mapsto (n + 1) - 1 = n$, thus $g \circ f = \mathrm{Id}_{\mathbb{N}}$
$f \circ g : 0 \mapsto 1$, hence $f \circ g \neq \mathrm{Id}_{\mathbb{N}}$

NB

$f \circ g$ is the identity when restricted to $\mathbb{P}$

NB

For a **finite** *set S and $f : S \longrightarrow S$ the properties*

1. *onto, and*
2. *1–1*

are equivalent. (Proof suggestion?)

Examples

$\boxed{1.7.6}$ $\Sigma = \{a, b, c\}$
(c) Is $\mathrm{length} : \Sigma^* \longrightarrow \mathbb{N}$ onto?
(d) $\mathrm{length}^{\leftarrow}(2) \overset{?}{=}$

Examples

$\boxed{1.7.12}$ Verify that $f : \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R} \times \mathbb{R}$ defined by $f(x, y) = (x + y, x - y)$ is invertible.

Examples

$\boxed{1.7.6}$ $\Sigma = \{a, b, c\}$
(c) Is $\text{length} : \Sigma^* \longrightarrow \mathbb{N}$ onto?
Yes: $\text{length}^{\leftarrow}(\{n\}) = \Sigma^n \neq \emptyset$
(d) $\text{length}^{\leftarrow}(2) = \{aa, ab, ac, bb, \ldots, cc\}$

Examples

$\boxed{1.7.12}$ Verify that $f : \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R} \times \mathbb{R}$ defined by
$f(x, y) = (x + y, x - y)$ is invertible.
The inverse is $f^{-1}(a, b) = (\frac{a+b}{2}, \frac{a-b}{2})$; substituting shows that
$f \circ f^{-1} = \text{Id}_{\mathbb{R}\times\mathbb{R}}$

Examples

$\boxed{1.7.6}$ $\Sigma = \{a, b, c\}$
(c) Is $\text{length} : \Sigma^* \longrightarrow \mathbb{N}$ onto?
Yes: $\text{length}^{\leftarrow}(\{n\}) = \Sigma^n \neq \emptyset$
(d) $\text{length}^{\leftarrow}(2) = \{aa, ab, ac, bb, \ldots, cc\}$

Examples

$\boxed{1.7.12}$ Verify that $f : \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R} \times \mathbb{R}$ defined by
$f(x, y) = (x + y, x - y)$ is invertible.
The inverse is $f^{-1}(a, b) = (\frac{a+b}{2}, \frac{a-b}{2})$; substituting shows that
$f \circ f^{-1} = \text{Id}_{\mathbb{R}\times\mathbb{R}}$

Supplementary Exercises [cont'd]

$\boxed{1.8.16}$ $\Sigma = \{a, b\}$; relate it to Σ^*
(a) Is there an onto $\Sigma \longrightarrow \Sigma^*$?
(b) Is there an onto $\Sigma^* \longrightarrow \Sigma$?

Supplementary Exercises [cont'd]

$\boxed{1.8.16}$ $\Sigma = \{a, b\}$; relate it to Σ^*
(a) Is there an onto $\Sigma \longrightarrow \Sigma^*$? No: $|\Sigma| = 2, |\Sigma^*| = \infty$.
(b) Is there an onto $\Sigma^* \longrightarrow \Sigma$? Yes, eg $f(\omega) = a$ when $\text{length}(\omega)$ is odd, $f(\omega) = b$ when $\text{length}(\omega)$ is even.
The following is **not** completely correct $f : \omega \mapsto \langle\text{first letter of } \omega\rangle$
Reason: $f(\lambda)$ is not defined.

Matrices

An **m × n matrix** is a rectangular array with m horizontal rows and n vertical columns.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

NB

Matrices are important objects in Computer Science, e.g. for

- *optimisation*
- *graphics and computer vision*
- *cryptography*
- *information retrieval and web search*
- *machine learning*

Basic Matrix Operations

The **transpose** $\mathbf{A}^{\mathbf{T}}$ of an $m \times n$ matrix $\mathbf{A} = [a_{ij}]$ is the $n \times m$ matrix whose entry in the ith row and jth column is a_{ji}.

Example

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 & 4 \\ 3 & 2 & -1 & 2 \\ 4 & 0 & 1 & 3 \end{bmatrix} \qquad \mathbf{A}^{\mathbf{T}} = \begin{bmatrix} 2 & 3 & 4 \\ -1 & 2 & 0 \\ 0 & -1 & 1 \\ 4 & 2 & 3 \end{bmatrix}$$

NB

A matrix $\mathbf{M}$ *is called symmetric if* $\mathbf{M}^{\mathbf{T}} = \mathbf{M}$

The **sum** of two $m \times n$ matrices $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$ is the $m \times n$ matrix whose entry in the ith row and jth column is $a_{ij} + b_{ij}$.

Example

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 & 4 \\ 3 & 2 & -1 & 2 \\ 4 & 0 & 1 & 3 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 1 & 0 & 5 & 3 \\ 2 & 3 & -2 & 1 \\ 4 & -2 & 0 & 2 \end{bmatrix}$$

$$\mathbf{A} + \mathbf{B} = \begin{bmatrix} 3 & -1 & 5 & 7 \\ 5 & 5 & -3 & 3 \\ 8 & -2 & 1 & 5 \end{bmatrix}$$

Fact

$\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$ *and* $(\mathbf{A} + \mathbf{B}) + \mathbf{C} = \mathbf{A} + (\mathbf{B} + \mathbf{C})$

Given $m \times n$ matrix $\mathbf{A} = [a_{ij}]$ and $c \in \mathbb{R}$, the **scalar product** $c\mathbf{A}$ is the $m \times n$ matrix whose entry in the ith row and jth column is $c \cdot a_{ij}$.

Example

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 & 4 \\ 3 & 2 & -1 & 2 \\ 4 & 0 & 1 & 3 \end{bmatrix} \qquad 2\mathbf{A} = \begin{bmatrix} 4 & -2 & 0 & 8 \\ 6 & 4 & -2 & 4 \\ 8 & 0 & 2 & 6 \end{bmatrix}$$

The **product** of an $m \times n$ matrix $\mathbf{A} = [a_{ij}]$ and an $n \times p$ matrix $\mathbf{B} = [b_{jk}]$ is the $m \times p$ matrix $\mathbf{C} = [c_{ik}]$ defined by

$$c_{ik} = \sum_{j=1}^{n} a_{ij} b_{jk} \qquad \text{for } 1 \le i \le m \text{ and } 1 \le k \le p$$

Example

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

NB

The rows of **A** *must have the same number of entries as the columns of* **B***.*
The product of a $1 \times n$ *matrix and an* $n \times 1$ *matrix is usually called the* **inner product** *of two* **n-dimensional vectors.**

Example

Consider

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 2 & -1 \\ -6 & 3 \end{bmatrix}$$

Calculate **AB**, **BA**

$$\mathbf{AB} = \begin{bmatrix} -10 & 5 \\ -20 & 10 \end{bmatrix} \qquad \mathbf{BA} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

NB

In general, $\mathbf{A} \cdot \mathbf{B} \neq \mathbf{B} \cdot \mathbf{A}$

Example

Consider

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 2 & -1 \\ -6 & 3 \end{bmatrix}$$

Calculate **AB**, **BA**

$$\mathbf{AB} = \begin{bmatrix} -10 & 5 \\ -20 & 10 \end{bmatrix} \qquad \mathbf{BA} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

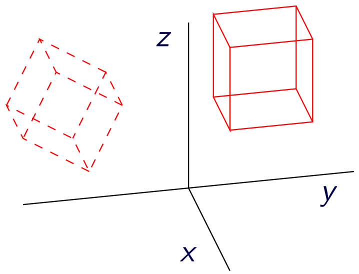
NB

In general, $\mathbf{A} \cdot \mathbf{B} \neq \mathbf{B} \cdot \mathbf{A}$

Example: Computer Graphics

Rotating an object w.r.t. the x axis by degree α:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} 5 & 5 & 7 & 7 & 5 & 7 & 5 & 7 \\ 1 & 1 & 1 & 1 & 3 & 3 & 3 & 3 \\ 9 & 7 & 7 & 9 & 7 & 7 & 9 & 9 \end{bmatrix}$$



Relations and their Representation

Relations are an abstraction used to capture the idea that the objects from certain domains (often the same domain for several objects) are *related*. These objects may

- influence one another (each other for binary relations; self(?) for unary)
- share some common properties
- correspond to each other precisely when some constraints are satisfied

In general, relations formalise the concept of interaction among objects from various domains; however, there must be a specified domain for each type of objects.

An **n-ary relation** is a subset of the cartesian product of n sets.

$$R \subseteq S_1 \times S_2 \times \ldots \times S_n$$

$$x \in R \Rightarrow x = (x_1, x_2, \ldots x_n) \;\text{ where each } x_i \in S_i$$

If $n = 2$ we have a **binary** relation $\mathcal{R} \subseteq S \times T$.
(mostly we consider binary relations)
equivalent notations: $(x_1, x_2, \ldots x_n) \in R \iff R(x_1, x_2, \ldots x_n)$
for binary relations: $(x, y) \in R \iff R(x, y) \iff xRy$.

Database Examples

Example (course enrolments)

S = set of CSE students
(S can be a subset of the set of all students)
C = set of CSE courses
(likewise)
E = enrolments = $\{\, (s, c) : s \text{ takes } c \,\}$

$$E \subseteq S \times C$$

In practice, almost always there are various ‘onto’ (nonemptiness) and 1–1 (uniqueness) constraints on database relations.

Example (class schedule)

C = CSE courses
T = starting time (hour & day)
R = lecture rooms
S = schedule =

$$\{\, (c, t, r) : c \text{ is at } t \text{ in } r \,\} \subseteq C \times T \times R$$

Example (sport stats)

$$R \subseteq \text{competitions} \times \text{results} \times \text{years} \times \text{athletes}$$

Applications

Relations are ubiquitous in Computer Science

- Databases are collections of relations
- Common data structures (e.g. graphs) are relations
- Any ordering is a relation
- Functions/procedures/programs compute relations between their input and output

Relations are therefore used in most problem specifications and to describe formal properties of programs.
For this reason, studying relations and their properties helps with formalisation, implementation and verification of programs.

n-ary Relations

Relations can be defined linking $k \geq 1$ domains $D_1, \ldots, D_k$ simultaneously.
In database situations one also allows for *unary* $(n = 1)$ relations.
Most common are **binary** relations

$$\mathcal{R} \subseteq S \times T; \quad \mathcal{R} = \{(s,t) |\ \text{“some property that links } s,t\text{”}\}$$

For related s, t we can write $(s,t) \in \mathcal{R}$ or $s\mathcal{R}t$; for unrelated items either $(s,t) \notin \mathcal{R}$ or $s \not\mathcal{R} t$.
$\mathcal{R}$ can be defined by

- explicit enumeration of interrelated k-tuples (ordered pairs in case of binary relations);
- properties that identify relevant tuples within the entire $D_1 \times D_2 \times \ldots \times D_k$;
- construction from other relations.

Functions as Relations

Any function $f : S \longrightarrow T$ can be viewed as a binary relation

$$\{\ (s, f(s)) : s \in S\ \} \subseteq S \times T$$

If a subset of $S \times T$ corresponds to a function, it must satisfy certain conditions w.r.t. S and T (which?)

Binary Relations

A binary relation, say $\mathcal{R} \subseteq S \times T$, can be presented as a matrix with rows enumerated by (the elements of) S and the columns by T; eg. for $S = \{s_1, s_2, s_3\}$ and $T = \{t_1, t_2, t_3, t_4\}$ we may have

$$\begin{bmatrix} \bullet & \circ & \bullet & \bullet \\ \circ & \bullet & \bullet & \bullet \\ \bullet & \bullet & \circ & \circ \end{bmatrix}$$

Example

$\boxed{\text{3.1.2(e)}}$ Write the following relation on $A = \{0, 1, 2\}$ as a matrix.

$(m, n) \in R$ if $m \cdot n = m$

$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & \bullet & \bullet & \bullet \\ 1 & \circ & \bullet & \circ \\ 2 & \circ & \bullet & \circ \end{array}$$

Example

$\boxed{\text{3.1.2(e)}}$ Write the following relation on $A = \{0, 1, 2\}$ as a matrix.

$(m, n) \in R$ if $m \cdot n = m$

$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & \bullet & \bullet & \bullet \\ 1 & \circ & \bullet & \circ \\ 2 & \circ & \bullet & \circ \end{array}$$

Relations on a Single Domain

Particularly important are binary relationships between the elements of the same set. We say that '$\mathcal{R}$ is a relation on S' if

$$\mathcal{R} \subseteq S \times S$$

Special (Trivial) Relations

(all w.r.t. set S)

Identity (diagonal, equality) $E = \{ (x, x) : x \in S \}$

Empty $\emptyset$

Universal $U = S \times S$

Important Properties of Binary Relations $\mathcal{R} \subseteq S \times S$

(R)	reflexive	$(x,x) \in \mathcal{R}$	$\forall x \in S$
(AR)	antireflexive	$(x,x) \notin \mathcal{R}$	$\forall x \in S$
(S)	symmetric	$(x,y) \in \mathcal{R} \Rightarrow (y,x) \in \mathcal{R}$	$\forall x,y \in S$
(AS)	antisymmetric	$(x,y),\ (y,x) \in \mathcal{R} \Rightarrow x = y$	$\forall x,y \in S$
(T)	transitive	$(x,y),\ (y,z) \in \mathcal{R} \Rightarrow (x,z) \in \mathcal{R}$	$\forall x,y,z \in S$

NB

An object, notion etc. is considered to satisfy a property if none of its instances violates any defining statement of that property.

Examples

(R) reflexive $(x,x) \in \mathcal{R}$ for all $x \in S$ $\begin{bmatrix} \bullet & \bullet & \circ \\ \circ & \bullet & \circ \\ \bullet & \circ & \bullet \end{bmatrix}$

(AR) antireflexive $(x,x) \notin \mathcal{R}$ $\begin{bmatrix} \circ & \bullet & \bullet \\ \circ & \circ & \circ \\ \bullet & \circ & \circ \end{bmatrix}$

(S) symmetric $(x,y) \in \mathcal{R} \Rightarrow (y,x) \in \mathcal{R}$ $\begin{bmatrix} \bullet & \circ & \bullet \\ \circ & \circ & \bullet \\ \bullet & \bullet & \circ \end{bmatrix}$

(AS) antisymmetric $(x,y),\ (y,x) \in \mathcal{R} \Rightarrow x = y$
$\begin{bmatrix} \bullet & \bullet & \circ \\ \circ & \circ & \bullet \\ \bullet & \circ & \circ \end{bmatrix}$

(T) transitive $(x,y),\ (y,z) \in \mathcal{R} \Rightarrow (x,z) \in \mathcal{R}$
$\begin{bmatrix} \circ & \circ & \bullet \\ \bullet & \bullet & \bullet \\ \circ & \circ & \circ \end{bmatrix}$

Example

$\boxed{3.1.1}$ The following relations are on $S = \{1,2,3\}$. Which of the properties (R), (AR), (S), (AS), (T) does each satisfy?

(a) $(m,n) \in R$ if $m + n = 3$
(AR) and (S)

(e) $(m,n) \in R$ if $\max\{m,n\} = 3$
(S)

$\boxed{3.1.2(b)}$ $(m,n) \in R$ if $m < n$
(AR), (AS), (T)

Example

$\boxed{3.1.1}$ The following relations are on $S = \{1,2,3\}$. Which of the properties (R), (AR), (S), (AS), (T) does each satisfy?

(a) $(m,n) \in R$ if $m + n = 3$
(AR) and (S)

(e) $(m,n) \in R$ if $\max\{m,n\} = 3$
(S)

$\boxed{3.1.2(b)}$ $(m,n) \in R$ if $m < n$
(AR), (AS), (T)

Interaction of Properties

A relation *can* be both symmetric and antisymmetric. Namely, when $\mathcal{R}$ consists only of some pairs $(x,x), x \in S$.
A relation *cannot* be simultaneously reflexive and antireflexive (unless $S = \emptyset$).

NB

$$\left.\begin{array}{r}\textit{nonreflexive}\\ \textit{nonsymmetric}\end{array}\right\} \textit{ is not the same as } \left\{\begin{array}{l}\textit{antireflexive/irreflexive}\\ \textit{antisymmetric}\end{array}\right.$$

Most important kinds of relations on S

- total order $\begin{bmatrix} \bullet & \bullet & \bullet \\ \circ & \bullet & \bullet \\ \circ & \circ & \bullet \end{bmatrix}$
- partial order $\begin{bmatrix} \bullet & \bullet & \bullet \\ \circ & \bullet & \circ \\ \circ & \circ & \bullet \end{bmatrix}$, $\begin{bmatrix} \bullet & \bullet & \circ \\ \circ & \bullet & \circ \\ \circ & \circ & \bullet \end{bmatrix}$
- equivalence $\begin{bmatrix} \bullet & \bullet & \circ \\ \bullet & \bullet & \circ \\ \circ & \circ & \bullet \end{bmatrix}$
- identity $\begin{bmatrix} \bullet & \circ & \circ \\ \circ & \bullet & \circ \\ \circ & \circ & \bullet \end{bmatrix}$

NB

Some of those are special cases of the others, eg. 'total order' of a 'partial order', 'identity' of an 'equivalence'.

Relation $\mathcal{R}$ as Correspondence From S to T

$\mathcal{R}(A) \overset{\text{def}}{=} \{t \in T | (s,t) \in \mathcal{R} \text{ for some } s \in A \subseteq S\}$
$\mathcal{R}^{\leftarrow}(B) \overset{\text{def}}{=} \{s \in S | (s,t) \in \mathcal{R} \text{ for some } t \in B \subseteq T\}$
Converse relation $\mathcal{R}^{\leftarrow}$

$$\mathcal{R}^{\leftarrow} = \{(t,s) \in T \times S | (s,t) \in \mathcal{R}\}$$

Note that $\mathcal{R}^{\leftarrow} \subseteq T \times S$.
Observe that $(\mathcal{R}^{\leftarrow})^{\leftarrow} = \mathcal{R}$.

NB

Viewed this way $\mathcal{R}$ becomes a function from $Pow(S)$ to $Pow(T)$. However, not every $g : Pow(S) \longrightarrow Pow(T)$ can be matched to a relation.

(Using a small domain like $S = \{a, b\}$ provide an example of a function $g : \text{Pow}(S) \longrightarrow \text{Pow}(S)$ which does not correspond to any relation on S. Can you do it with $S' = \{a\}$?)

NB

The order of axes – S and T – is important. For $\mathcal{R} \subseteq S \times S$, its converse $\mathcal{R}^{\leftarrow}$ is usually quite different from $\mathcal{R}$.

Example: divisibility relation on $\mathbb{P}$

$$\begin{aligned} D &\overset{\text{def}}{=} \{\ (p,q) : p|q\ \} = \{(1,1),(1,2),\ldots,(2,2),(2,4),\ldots\} \\ D^{\leftarrow} &= \{\ (p,q) : p \in q\mathbb{P}\ \} \\ &= \{(1,1),(2,1),(2,2),(3,1),(3,3),(4,1),(4,2),\ldots\} \end{aligned}$$

For every $n \in \mathbb{P}$, $D(\{n\})$ is infinite, $D^{\leftarrow}(\{n\})$ is finite.

Question

$f^{\leftarrow}$ is a relation; when is it a function?

Question

$f^{\leftarrow}$ is a relation; when is it a function?

Answer

When f is 1-1 and onto.

Example

$\boxed{3.1.9}$ Find the properties of the *empty relation* $\emptyset \subset S \times S$ and the *universal relation* $U = S \times S$. Assume that S is a nonempty domain.

(a) $\emptyset$ is (AR), (S), (AS), (T); if $S = \emptyset$ itself then $\emptyset$ is also (R).
(b) U is (R), (S), (T); if $|S| \leq 1$ then also (AS)

Example

$\boxed{3.1.9}$ Find the properties of the *empty relation* $\emptyset \subset S \times S$ and the *universal relation* $U = S \times S$. Assume that S is a nonempty domain.

(a) $\emptyset$ is (AR), (S), (AS), (T); if $S = \emptyset$ itself then $\emptyset$ is also (R).
(b) U is (R), (S), (T); if $|S| \leq 1$ then also (AS)

Example

$\boxed{\text{3.1.10(a)}}$ Give examples of relations with specified properties.
(AS), (T), ¬(R).

Examples over $\mathbb{N}$, Pow($\mathbb{N}$)

- strict order of numbers $x < y$
- simple (weak) order, but with some pairs (x, x) removed from $\mathcal{R}$
- being a prime divisor
  $(p, n) \in \mathcal{R}$ iff p is prime and $p|n$
    - not reflexive: $(1, 1) \notin \mathcal{R}, (4, 4) \notin \mathcal{R}, (6, 6) \notin \mathcal{R}$
    - transitivity is meaningful only for the pairs $(p, p), (p, n),\ p|n$ for p prime

Example

$\boxed{\text{3.1.10(a)}}$ Give examples of relations with specified properties.
(AS), (T), ¬(R).

Examples over $\mathbb{N}$, Pow($\mathbb{N}$)

- strict order of numbers $x < y$
- simple (weak) order, but with some pairs (x, x) removed from $\mathcal{R}$
- being a prime divisor
  $(p, n) \in \mathcal{R}$ iff p is prime and $p|n$
    - not reflexive: $(1, 1) \notin \mathcal{R}, (4, 4) \notin \mathcal{R}, (6, 6) \notin \mathcal{R}$
    - transitivity is meaningful only for the pairs $(p, p), (p, n),\ p|n$ for p prime

Example

$\boxed{\text{3.1.10(b)}}$ Give examples of relations with specified properties.
(S), ¬(R), ¬(T).

Easiest examples - inequality

- $\mathcal{R} = \{(x, y)|x \neq y,\ x, y \in \mathbb{N}\}$
- $\mathcal{R} = \{(A, B)|A \neq B,\ A, B \subseteq S\}$

Example

$\boxed{\text{3.1.10(b)}}$ Give examples of relations with specified properties.
(S), ¬(R), ¬(T).

Easiest examples - inequality

- $\mathcal{R} = \{(x, y)|x \neq y,\ x, y \in \mathbb{N}\}$
- $\mathcal{R} = \{(A, B)|A \neq B,\ A, B \subseteq S\}$

Example

$\boxed{3.1.14}$ Which properties carry from individual relations to their union?

(a) $\mathcal{R}_1, \mathcal{R}_2 \in (\mathrm{R}) \Rightarrow \mathcal{R}_1 \cup \mathcal{R}_2 \in (\mathrm{R})$

(b) $\mathcal{R}_1, \mathcal{R}_2 \in (\mathrm{S}) \Rightarrow \mathcal{R}_1 \cup \mathcal{R}_2 \in (\mathrm{S})$

(c) $\mathcal{R}_1, \mathcal{R}_2 \in (\mathrm{T}) \not\Rightarrow \mathcal{R}_1 \cup \mathcal{R}_2 \in (\mathrm{T})$

Eg. $S = \{a, b, c\}, a\mathcal{R}_1 b, b\mathcal{R}_2 c$

and no other relationships

Example

$\boxed{3.1.14}$ Which properties carry from individual relations to their union?

(a) $\mathcal{R}_1, \mathcal{R}_2 \in (\mathrm{R}) \Rightarrow \mathcal{R}_1 \cup \mathcal{R}_2 \in (\mathrm{R})$

(b) $\mathcal{R}_1, \mathcal{R}_2 \in (\mathrm{S}) \Rightarrow \mathcal{R}_1 \cup \mathcal{R}_2 \in (\mathrm{S})$

(c) $\mathcal{R}_1, \mathcal{R}_2 \in (\mathrm{T}) \not\Rightarrow \mathcal{R}_1 \cup \mathcal{R}_2 \in (\mathrm{T})$

Eg. $S = \{a, b, c\}, a\mathcal{R}_1 b, b\mathcal{R}_2 c$

and no other relationships

Equivalence Relations and Partitions

Relation $\mathcal{R}$ is called an *equivalence* relation if it satisfies (R), (S), (T). Every equivalence $\mathcal{R}$ defines *equivalence classes* on its domain S.

The equivalence class $[s]$ (w.r.t. $\mathcal{R}$) of an element $s \in S$ is

$$[s] = \{\ t \in S : t\mathcal{R}s\ \}$$

This notion is well defined only for $\mathcal{R}$ which is an equivalence relation. Collection of all equivalence classes $[S]_{\mathcal{R}} = \{\ [s] : s \in S\ \}$ is a partition of S

$$S = \bigcup_{s \in S} [s]$$

Thus the equivalence classes are disjoint and jointly cover the entire domain. It means that every element belongs to one (and only one) equivalence class.

We call $s_1, s_2, \ldots$ *representatives* of (different) equivalence classes

For $s, t \in S$ either $[s] = [t]$, when $s\mathcal{R}t$, or $[s] \cap [t] = \emptyset$, when $s \not\mathcal{R} t$.

We commonly write $s \sim_{\mathcal{R}} t$ when s, t are in the same equivalence class.

In the opposite direction, a partition of a set defines the equivalence relation on that set. If $S = S_1 \dot\cup \ldots \dot\cup S_k$, then we specify $s \sim t$ exactly when s and t belong to the same S_i.

If the relation $\sim$ is an equivalence on S and $[S]$ the corresponding partition, then

$$\nu : S \longrightarrow [S], \quad \nu : s \mapsto [s] = \{\, x \in S : x \sim s \,\}$$

is called the *natural* map. It is always onto.

Question

When is ν also 1–1 ?

If the relation $\sim$ is an equivalence on S and $[S]$ the corresponding partition, then

$$\nu : S \longrightarrow [S], \quad \nu : s \mapsto [s] = \{\, x \in S : x \sim s \,\}$$

is called the *natural* map. It is always onto.

Question

When is ν also 1–1 ?

Answer

When $\sim$ is the identity on S.

A function $f : S \longrightarrow T$ defines an equivalence relation on S by

$$s_1 \sim s_2 \quad \text{iff} \quad f(s_1) = f(s_2)$$

These sets $f^{\leftarrow}(t)$, $t \in T$ that are nonempty form the corresponding partition

$$S = \bigcup_{t \in T} f^{\leftarrow}(t)$$

Question

When are all $f^{\leftarrow}(t) \neq \emptyset$?

A function $f : S \longrightarrow T$ defines an equivalence relation on S by

$$s_1 \sim s_2 \quad \text{iff} \quad f(s_1) = f(s_2)$$

These sets $f^{\leftarrow}(t)$, $t \in T$ that are nonempty form the corresponding partition

$$S = \bigcup_{t \in T} f^{\leftarrow}(t)$$

Question

When are all $f^{\leftarrow}(t) \neq \emptyset$?

Answer

When f is onto.

Example

Partition of $\mathbb{Z}$ into classes of numbers with the same remainder $(\mathrm{mod}\ p)$; it is particularly important for p prime

$$\mathbb{Z}(p) = \mathbb{Z}_p = \{0, 1, \ldots, p-1\}$$

One can define all four arithmetic operations (with the usual properties) on $\mathbb{Z}_p$ for a prime p; division has to be restricted when p is not prime.

Standard notation:
$m = n \pmod{p}$ stands for: $m \bmod p = n \bmod p$

Example

$\boxed{3.6.6}$ Show that $m \sim n$ when $m^2 = n^2 \pmod{5}$ is an equivalence on $S = \{1, \ldots, 7\}$. Find all the equivalence classes.

(a) It just means that $m = n \pmod{5}$ or $m = -n \pmod{5}$, e.g. $1 = -4 \pmod{5}$. This satisfies (R), (S), (T).

(b) We have
$[0] = \{0, 5\}$
$[1] = \{1, 4, 6\}$
$[2] = \{2, 3, 7\}$

Supplementary Exercises

$\boxed{3.6.10}$
$\mathcal{R}$ is a relation on $\mathbb{N} \times \mathbb{N}$, i.e. it is a subset of $\mathbb{N}^4$
$(m, n) \sim (p, q)$ if $m = p \pmod{3}$ or $n = q \pmod{5}$.
(a) $\mathcal{R} \in$ (R)?
Yes: $(m, n) \sim (m, n)$ iff $m = m \pmod{3}$ or $n = n \pmod{5}$ iff true or true.
(b) $\mathcal{R} \in$ (S)?
Yes: by symmetry of $. = . \pmod{n}$.
(c) $\mathcal{R} \in$ (T)?
No — for arbitrary two pairs (m_1, n_1) and (m_2, n_2) one can create a chain $(m_1, n_1)\mathcal{R}(m_2, n_1)$ and $(m_2, n_1)\mathcal{R}(m_2, n_2)$, but not all pairs are related.

Supplementary Exercises

$\boxed{3.6.10}$
$\mathcal{R}$ is a relation on $\mathbb{N} \times \mathbb{N}$, i.e. it is a subset of $\mathbb{N}^4$
$(m, n) \sim (p, q)$ if $m = p \pmod{3}$ or $n = q \pmod{5}$.
(a) $\mathcal{R} \in$ (R)?
Yes: $(m, n) \sim (m, n)$ iff $m = m \pmod{3}$ or $n = n \pmod{5}$ iff true or true.
(b) $\mathcal{R} \in$ (S)?
Yes: by symmetry of $. = . \pmod{n}$.
(c) $\mathcal{R} \in$ (T)?
No — for arbitrary two pairs (m_1, n_1) and (m_2, n_2) one can create a chain $(m_1, n_1)\mathcal{R}(m_2, n_1)$ and $(m_2, n_1)\mathcal{R}(m_2, n_2)$, but not all pairs are related.

Supplementary Exercises

$\boxed{3.6.10}$

$\mathcal{R}$ is a relation on $\mathbb{N} \times \mathbb{N}$, i.e. it is a subset of $\mathbb{N}^4$
$(m, n) \sim (p, q)$ if $m = p \pmod 3$ or $n = q \pmod 5$.
(a) $\mathcal{R} \in$ (R)?
Yes: $(m, n) \sim (m, n)$ iff $m = m \pmod 3$ or $n = n \pmod 5$ iff true or true.
(b) $\mathcal{R} \in$ (S)?
Yes: by symmetry of $. = . \pmod n$.
(c) $\mathcal{R} \in$ (T)?
No — for arbitrary two pairs (m_1, n_1) and (m_2, n_2) one can create a chain $(m_1, n_1)\mathcal{R}(m_2, n_1)$ and $(m_2, n_1)\mathcal{R}(m_2, n_2)$, but not all pairs are related.

Supplementary Exercises

$\boxed{3.6.10}$

$\mathcal{R}$ is a relation on $\mathbb{N} \times \mathbb{N}$, i.e. it is a subset of $\mathbb{N}^4$
$(m, n) \sim (p, q)$ if $m = p \pmod 3$ or $n = q \pmod 5$.
(a) $\mathcal{R} \in$ (R)?
Yes: $(m, n) \sim (m, n)$ iff $m = m \pmod 3$ or $n = n \pmod 5$ iff true or true.
(b) $\mathcal{R} \in$ (S)?
Yes: by symmetry of $. = . \pmod n$.
(c) $\mathcal{R} \in$ (T)?
No — for arbitrary two pairs (m_1, n_1) and (m_2, n_2) one can create a chain $(m_1, n_1)\mathcal{R}(m_2, n_1)$ and $(m_2, n_1)\mathcal{R}(m_2, n_2)$, but not all pairs are related.

Order Relations

Total order $\leq$ on S
(R) $x \leq x$ for all $x \in S$
(AS) $x \leq y, y \leq x \Rightarrow x = y$
(T) $x \leq y, y \leq z \Rightarrow x \leq z$
(L) *Linearity* — any two elements are comparable:
for all x, y either $x \leq y$ or $y \leq x$ (and both if $x = y$)

On a finite set all total orders are isomorphic

$$x_1 \leq x_2 \leq \cdots \leq x_n$$

On an infinite set there is quite a variety of possibilities.

Examples

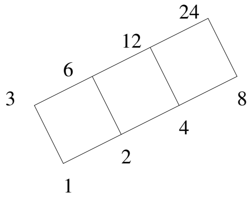
- discrete with a least element, e.g. $\mathbb{N} = \{0, 1, 2, \ldots\}$
- discrete without a least element, e.g. $\mathbb{Z} = \{\ldots, 0, 1, 2, \ldots\}$
- various dense/locally dense orders
  - rational numbers $\mathbb{Q}$: $\quad p < q \Rightarrow \exists_r p < r < q$
  - $S = [a, b]$ — both least and greatest elements
  - $S = (a, b]$ — no least element
  - $S = [a, b)$ — no greatest element
  - other $[0, 1] \cup [2, 3] \cup [4, 5] \cup \ldots$

Partial Order

A **partial order** $\preceq$ on S satisfies (R), (AS), (T); need not be (L)
We call $(S, \preceq)$ a **poset** — partially ordered set

Finite posets can be represented as so-called **Hasse diagrams**

11.1.1(a) Hasse diagram for positive divisors of 24



Ordering Concepts

- *Minimal* and *maximal* elements (they always exist in every finite poset)
- *Minimum* and *maximum* — unique minimal and maximal element
- *lub* (least upper bound) and *glb* (greatest lower bound) of a subset $A \subseteq S$ of elements
  $\text{lub}(A)$ — smallest element $x \in S$ s.t. $x \succeq a$ for all $a \in A$
  $\text{glb}(A)$ — greatest element $x \in S$ s.t. $x \preceq a$ for all $a \in A$
- *Lattice* — a poset where lub and glb exist for every pair of elements
  (by induction, they then exist for every *finite* subset of elements)

Examples

- $\text{Pow}(\{a, b, c\})$ with the order $\subseteq$
  $\emptyset$ is minimum; $\{a, b, c\}$ is maximum
- 11.1.4
  $\text{Pow}(\{a, b, c\}) \setminus \{\{a, b, c\}\}$ (proper subsets of $\{a, b, c\}$)
  Each two-element subset $\{a, b\}, \{a, c\}, \{b, c\}$ is maximal.
  - But there is no maximum
- $\{1, 2, 3, 4, 6, 8, 12, 24\}$ partially ordered by divisibility is a lattice
  - e.g. $\text{lub}(\{4, 6\}) = 12$; $\text{glb}(\{4, 6\}) = 2$
- $\{1, 2, 3\}$ partially ordered by divisibility is not a lattice
  - $\{2, 3\}$ has no lub
- $\{2, 3, 6\}$ partially ordered by divisibility is not a lattice
  - $\{2, 3\}$ has no glb
- $\{1, 2, 3, 12, 18, 36\}$ partially ordered by divisibility is not a lattice
  - $\{2, 3\}$ has no lub ($12, 18$ are minimal upper bounds)

NB

An infinite lattice need not have a lub (or no glb) for an arbitrary infinite subset of its elements, in particular no such bound may exist for ***all*** *its elements.*

Examples

- $\mathbb{Z}$ — neither lub nor glb;
- $\mathbb{F}(\mathbb{N})$ — all finite subsets, has no *arbitrary* lub property; glb exists, it is the intersection, hence always finite;
- $\mathbb{I}(\mathbb{N})$ — all infinite subsets, may not have an arbitrary glb; lub exists, it is the union, which is always infinite.

Example

$\boxed{11.1.5}$ Consider poset $(\mathbb{R}, \leq)$

(a) Is this a lattice?

(b) Give an example of a non-empty subset of $\mathbb{R}$ that has no upper bound.

(c) Find $\mathrm{lub}(\{ x \in \mathbb{R} : x < 73 \})$

(d) Find $\mathrm{lub}(\{ x \in \mathbb{R} : x \leq 73 \})$

(e) Find $\mathrm{lub}(\{ x : x^2 < 73 \})$

(f) Find $\mathrm{glb}(\{ x : x^2 < 73 \})$

Example

$\boxed{11.1.5}$ Consider poset $(\mathbb{R}, \leq)$

(a) It is a lattice.

(b) subset with no upper bound: $\mathbb{R}_{>0} = \{ r \in \mathbb{R} : r > 0 \}$

(c) and (d) $\mathrm{lub}(\{ x : x < 73 \}) = \mathrm{lub}(\{ x : x \leq 73 \}) = 73$

(e) $\mathrm{lub}(\{ x : x^2 < 73 \}) = \sqrt{73}$

(f) $\mathrm{glb}(\{ x : x^2 < 73 \}) = -\sqrt{73}$

Example

$\boxed{11.1.13}$ $\mathbb{F}(\mathbb{N})$ — collection of all *finite* subsets of $\mathbb{N}$

(a) Does it have a maximal element?

(b) Does it have a minimal element?

(c) Given $A, B \in \mathbb{F}(\mathbb{N})$, does $\{A, B\}$ have a lub in $\mathbb{F}(\mathbb{N})$?

(d) Given $A, B \in \mathbb{F}(\mathbb{N})$, does $\{A, B\}$ have a glb in $\mathbb{F}(\mathbb{N})$?

(e) Is $\mathbb{F}(\mathbb{N})$ a lattice?

Example

$\boxed{11.1.13}$ $\mathbb{F}(\mathbb{N})$ — collection of all *finite* subsets of $\mathbb{N}$

(a) No maximal elements

(b) $\emptyset$ is the minimum

(c) $\mathrm{lub}(A, B) = A \cup B$

(d) $\mathrm{glb}(A, B) = A \cap B$

(e) $\mathbb{F}(\mathbb{N})$ is a lattice — is has *finite* union and intersection properties.

Example

$\boxed{11.1.14}$ $\mathbb{I}(\mathbb{N}) = \text{Pow}(\mathbb{N}) \setminus \mathbb{F}(\mathbb{N})$ — collection of all *infinite* subsets of $\mathbb{N}$

(a) Does it have a maximal element?

(b) Does it have a minimal element?

(c) Given $A, B \in \mathbb{I}(\mathbb{N})$, does $\{A, B\}$ have a lub in $\mathbb{I}(\mathbb{N})$?

(d) Given $A, B \in \mathbb{I}(\mathbb{N})$, does $\{A, B\}$ have a glb in $\mathbb{I}(\mathbb{N})$?

(e) Is $\mathbb{I}(\mathbb{N})$ a lattice?

Example

$\boxed{11.1.14}$ $\mathbb{I}(\mathbb{N}) = \text{Pow}(\mathbb{N}) \setminus \mathbb{F}(\mathbb{N})$ — collection of all *infinite* subsets of $\mathbb{N}$

(a) $\mathbb{N}$ is the maximum

(b) No minimum element ($\emptyset$ is not in $\mathbb{I}(\mathbb{N})$)

(c) $\text{lub}(A, B) = A \cup B$

(d) $\text{glb}(A, B) = A \cap B$ *if it exists*; it does not exist when $A \cap B$ is finite, eg. when empty.

(e) $\mathbb{I}(\mathbb{N})$ is not a lattice — it has finite union but not finite intersection property; eg. sets $2\mathbb{N}$ and $2\mathbb{N} + 1$ have the empty intersection.

Well-Ordered Sets

Well-ordered set: every subset has a least element.

NB

The greatest element is not required.

Examples

- $\mathbb{N} = \{0, 1, \ldots\}$
- $\mathbb{N}_1 \dot\cup \mathbb{N}_2 \dot\cup \mathbb{N}_3 \dot\cup \ldots$, where each $\mathbb{N}_i \sim \mathbb{N}$ and $\mathbb{N}_1 < \mathbb{N}_2 < \mathbb{N}_3 \cdots$

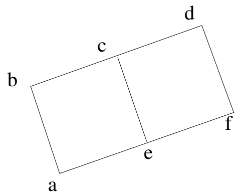
NB

Well-order sets are an important mathematical tool to prove termination of programs.

Ordering of a Poset — Topological Sort

For a poset $(S, \preceq)$ any linear order $\leq$ that is consistent with $\preceq$ is called **topological sort**. Consistency means that $a \preceq b \Rightarrow a \leq b$.

Consider



Various possible topological sortings

The following all are topological sorts:

$a \leq b \leq e \leq c \leq f \leq d$

$a \leq e \leq b \leq f \leq c \leq d$

......

$a \leq e \leq f \leq b \leq c \leq d$

Combining Orders

Product order — can combine any partial orders. In general, it is only a *partial order*, even if combining total orders.
For $s, s' \in S$ and $t, t' \in T$ define

$$(s,t) \preceq (s',t') \quad \text{if } s \preceq s' \text{ and } t \preceq t'$$

Practical Orderings

They are, effectively, *total* orders on the *product* of ordered sets.

- **Lexicographic order** — defined on all of Σ^*. It extends a total order already assumed to exist on Σ.
- **Lenlex** — the order on (potentially) the entire Σ^*, where the elements are ordered first by length.
  $\Sigma^{(1)} \prec \Sigma^{(2)} \prec \Sigma^{(3)} \prec \cdots$, then lexicographically within each $\Sigma^{(k)}$. In practice it is applied only to the finite subsets of Σ^*.
- **Filing order** — lexicographic order confined to the strings of the same length.
  It defines total orders on Σ^i, separately for each i.

Example

$\boxed{11.2.5}$ Let $\mathbb{B} = \{0,1\}$ with the usual order $0 < 1$. List the elements $101, 010, 11, 000, 10, 0010, 1000$ of $\mathbb{B}^*$ in the
(a) Lexicographic order
$000, 0010, 010, 10, 1000, 101, 11$
(b) Lenlex order
$10, 11, 000, 010, 101, 0010, 1000$

$\boxed{11.2.8}$ When are the lexicographic order and *lenlex* on Σ^* the same?
Only when $|\Sigma| = 1$.

Example

$\boxed{11.2.5}$ Let $\mathbb{B} = \{0,1\}$ with the usual order $0 < 1$. List the elements $101, 010, 11, 000, 10, 0010, 1000$ of $\mathbb{B}^*$ in the
(a) Lexicographic order
$000, 0010, 010, 10, 1000, 101, 11$
(b) Lenlex order
$10, 11, 000, 010, 101, 0010, 1000$

$\boxed{11.2.8}$ When are the lexicographic order and *lenlex* on Σ^* the same?
Only when $|\Sigma| = 1$.

Supplementary Exercises

$\boxed{11.6.6}$ True or false?
(a) If a set Σ is totally ordered, then the corresponding lexicographic partial order on Σ^* also must be totally ordered.
(b) If a set Σ is totally ordered, then the corresponding lenlex order on Σ^* also must be totally ordered.
(c) Every finite partially ordered set has a Hasse diagram.
(d) Every finite partially ordered set has a topological sorting.
(e) Every finite partially ordered set has a smallest element.
(f) Every finite totally ordered set has a largest element.
(g) An infinite partially ordered set cannot have a largest element.

Supplementary Exercises

$\boxed{11.6.6}$
(a) and (b) – True; this is the idea behind various lex-sorts
(c) Yes.
(d) Yes.
(e) False – consider a two-element set with the identity as p.o.
(f) True – due to the finiteness
(g) False, eg. $\mathbb{Z}_{<0}$

Summary

- Properties of functions: onto, 1-1; f^{-1}, $f^{\leftarrow}$
- Properties of binary relations: (R), (AR); (S), (AS); (T)
- Matrix operations: transposition, sum, scalar product, product
- Equivalence relations $\sim$, equivalence classes $[S]$, example $\mathbb{Z}_p$
- Ordering concepts: total, partial, lub, glb, lattice, topological sort
- Orderings: product, lexicographic, lenlex, filing

COMP9020 Lecture 6
Session 1, 2017
Graphs and Trees

- Textbook (R & W) - Ch. 3, Sec. 3.2; Ch. 6, Sec. 6.1–6.5
- Problem set 6
- Supplementary Exercises Ch. 6 (R & W)
- A. Aho & J. Ullman. Foundations of Computer Science in C, p. 522–526 (Ch. 9, Sec. 9.10)

Graphs

Binary relations on finite sets correspond to directed graphs.
Symmetric relations correspond to undirected graphs.

Terminology (the most common; there are many variants):

(Undirected) Graph — pair (V, E) where
V – set of vertices
E – set of edges
Every edge $e \in E$ corresponds uniquely to the set (an unordered pair) $\{x_e, y_e\}$ of vertices $x_e, y_e \in V$.

A *directed* edge is called an *arc*; it corresponds to the ordered pair (x_a, y_a). A **directed graph** consist of vertices and arcs.

NB

Edges $\{x, y\}$ and arcs (x, y) with $x = y$ are called loops.
We will only consider graphs without loops.

Graphs in Computer Science

Examples

1. The WWW can be considered a massive graph where the nodes are web pages and arcs are hyperlinks.
2. The possible states of a program form a directed graph.
3. The map of the earth can be represented as an undirected graph where edges delineate countries.

NB

Applications of graphs in Computer Science are abundant, e.g.

- *route planning in navigation systems, robotics*
- *optimisation, e.g. timetables, utilisation of network structures*
- *compilers using "graph colouring" to assign registers to program variables*

Vertex Degrees

- **Degree** of a vertex

$$\deg(v) = |\{ w \in V : (v, w) \in E \}|$$

  i.e., the number of edges attached to the vertex
- **Regular graph** — all degrees are equal
- *Degree sequence* $D_0, D_1, D_2, \ldots, D_k$ of graph $G = (V, E)$, where D_i = no. of vertices of degree i

Question

What is $D_0 + D_1 + \ldots + D_k$?

- $\sum_{v \in V} \deg(v) = 2 \cdot e(G)$; thus the sum of vertex degrees is always even.
- There is an even number of vertices of odd degree ($\boxed{6.1.8}$)

Paths

- A **path** in a graph (V, E) is a sequence of edges that link up

$$v_0 \xrightarrow{\{v_0,v_1\}} v_1 \xrightarrow{\{v_1,v_2\}} \dots \xrightarrow{\{v_{n-1},v_n\}} v_n$$

  where $e_i = \{v_{i-1}, v_i\} \in E$
- **length** of the path is the number of edges: n
  neither the vertices nor the edges have to be all different
- Subpath of length r: $(e_m, e_{m+1}, \dots, e_{m+r-1})$
- Path of length 0: single vertex v_0
- **Connected graph** — each pair of vertices joined by a path

Exercises

$\boxed{\text{6.1.13(a)}}$ Draw a connected, regular graph on four vertices, each of degree 2

$\boxed{\text{6.1.13(b)}}$ Draw a connected, regular graph on four vertices, each of degree 3

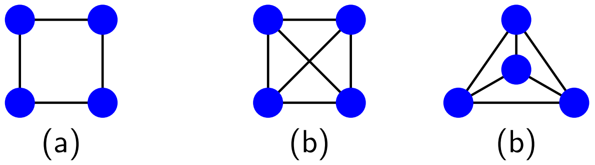
$\boxed{\text{6.1.13(c)}}$ Draw a connected, regular graph on five vertices, each of degree 3

$\boxed{\text{6.1.14(a)}}$ Graph with 3 vertices and 3 edges

$\boxed{\text{6.1.14(b)}}$ Two graphs each with 4 vertices and 4 edges

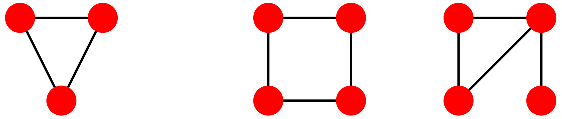
Exercises

$\boxed{\text{6.1.13}}$ Connected, regular graphs on four vertices



(a) (b) (b) none (c)

$\boxed{\text{6.1.14}}$ Graphs with 3 vertices and 3 edges must have a *cycle*



(a) the only one (b) (b)

Exercises

NB

We use the notation
$v(G) = |V|$ *for the no. of vertices of graph* $G = (V, E)$
$e(G) = |E|$ *for the no. of edges of graph* $G = (V, E)$

$\boxed{\text{6.1.20(a)}}$ Graph with $e(G) = 21$ edges has a degree sequence $D_0 = 0, D_1 = 7, D_2 = 3, D_3 = 7, D_4 = ?$
Find $v(G)$!

$\boxed{\text{6.1.20(b)}}$ How would your answer change, if at all, when $D_0 = 6$?

Exercises

$\boxed{\text{6.1.20(a)}}$ Graph with $e(G) = 21$ edges has a degree sequence
$D_0 = 0, D_1 = 7, D_2 = 3, D_3 = 7, D_4 = ?$
Find $v(G)$

$\sum_v \deg(v) = 2|E|$; here
$7 \cdot 1 + 3 \cdot 2 + 7 \cdot 3 + x \cdot 4 = 2 \cdot 21$ giving $x = 2$, thus
$v(G) = \sum D_i = 19$.

$\boxed{\text{6.1.20(b)}}$ How would your answer change, if at all, when $D_0 = 6$?
No change to D_4; $v(G) = 25$.

Cycles

Recall paths $v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \ \ldots \ \xrightarrow{e_n} v_n$

- *simple path* — $e_i \neq e_j$ for all edges of the path $(i \neq j)$
- *closed path* — $v_0 = v_n$
- **cycle** — closed path, all other v_i pairwise distinct and $\neq v_0$
- *acyclic path* — $v_i \neq v_j$ for *all* vertices in the path $(i \neq j)$

NB

1. *$C = (e_1, \ldots, e_n)$ is a cycle iff removing any single edge leaves an acyclic path. (Show that the 'any' condition is needed!)*
2. *C is a cycle if it has the same number of edges and vertices and no subpath has this property.*
   (Show that the 'subpath' condition is needed, i.e., there are graphs G that are **not** *cycles and $|E_G| = |V_G|$; every such G must contain a cycle!)*

Trees

- **Acyclic graph** — graph that doesn't contain any cycle
- **Tree** — connected acyclic graph
- A graph is acyclic *iff* it is a *forest* (collection of disjoint trees)

NB

Graph G is a tree iff

$\Leftrightarrow$ *it is acyclic and $|V_G| = |E_G| + 1$.*
(Show how this implies that the graph is connected!)

$\Leftrightarrow$ *there is exactly one simple path between any two vertices.*

$\Leftrightarrow$ *G is connected, but becomes disconnected if any single edge is removed.*

$\Leftrightarrow$ *G is acyclic, but has a cycle if any single edge on already existing vertices is added.*

Exercise (Supplementary)

$\boxed{\text{6.7.3}}$ (Supp) Tree with n vertices, $n \geq 3$.
Always true, false or could be either?
(a) $e(T) \overset{?}{=} n$
(b) at least one vertex of deg 2
(c) at least two v_1, v_2 s.t. $\deg(v_1) = \deg(v_2)$
(d) exactly one path from v_1 to v_2

Exercise (Supplementary)

$\boxed{6.7.3}$ (Supp) Tree with n vertices, $n \geq 3$.
Always true, false or could be either?

(a) $e(T) \overset{?}{=} n$ — False

(b) at least one vertex of deg 2 — Could be either

(c) at least two v_1, v_2 s.t. $\deg(v_1)=\deg(v_2)$ — True

(d) exactly one path from v_1 to v_2 — True (characterises a tree)

NB

A tree with one vertex designated as its root is called a rooted tree*. It imposes an ordering on the edges: 'away' from the root — from parent nodes to children. This defines a* level number *(or:* depth*) of a node as its distance from the root.*
Another very common notion in Computer Science is that of a DAG *— a* directed, acyclic graph*.*

Graph Isomorphisms

$\phi : G \longrightarrow H$ is a *graph isomorphism* if

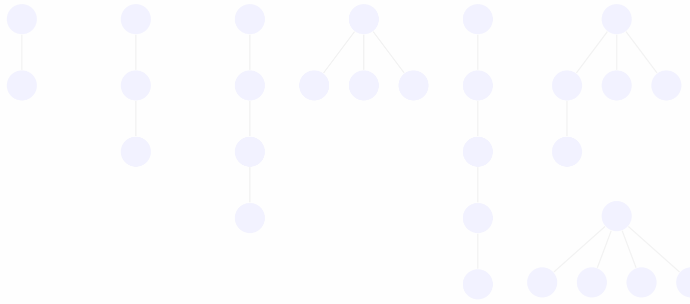
(i) $\phi : V_G \longrightarrow V_H$ is 1–1 and onto (a so-called *bijection*)

(ii) $(x, y) \in E_G$ iff $(\phi(x), \phi(y)) \in E_H$

Two graphs are called *isomorphic* if there exists (at least one) isomorphism between them.

Example

All nonisomorphic trees on 2, 3, 4 and 5 vertices.



Graph Isomorphisms

$\phi : G \longrightarrow H$ is a *graph isomorphism* if

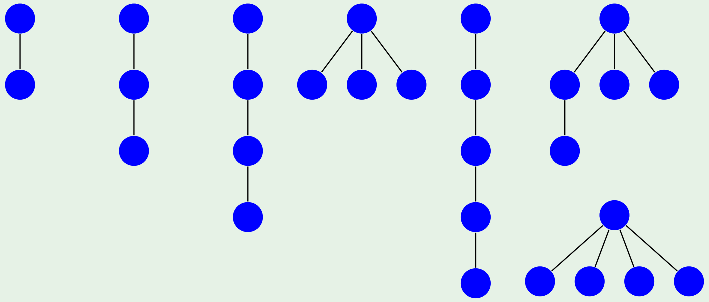
(i) $\phi : V_G \longrightarrow V_H$ is 1–1 and onto (a so-called *bijection*)

(ii) $(x, y) \in E_G$ iff $(\phi(x), \phi(y)) \in E_H$

Two graphs are called *isomorphic* if there exists (at least one) isomorphism between them.

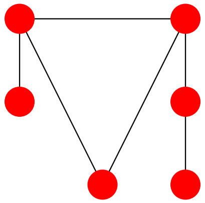
Example

All nonisomorphic trees on $2, 3, 4$ and 5 vertices.



Automorphisms and Asymmetric Graphs

An isomorphism from a graph to itself is called *automorphism*.
Every graph has at least the trivial automorphism;
(trivial meaning $\phi(v) = v$ for all $v \in V_G$)
Graphs with no non-trivial automorphisms are called *asymmetric*.
The smallest non-trivial asymmetric graphs have 6 vertices.



(Can you find another one with 6 nodes? There are seven more.)

Edge Traversal

Definition

- **Euler path** — path containing every edge exactly once
- **Euler circuit** — closed Euler path

Characterisations

- G (connected) has an Euler circuit iff $\deg(v)$ is even for all $v \in V$.
- G (connected) has an Euler path iff either it has an Euler circuit (above) or it has exactly two vertices of odd degree.

NB

- *These characterisations apply to graphs with loops as well*
- *For directed graphs the condition for existence of an Euler circuit is $indeg(v) = outdeg(v)$ for all $v \in V$*

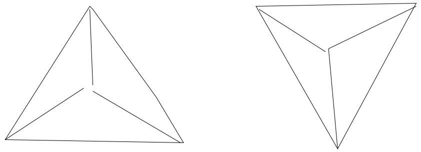
Exercises

$\boxed{6.2.11}$ Construct a graph with vertex set $\{0,1\} \times \{0,1\} \times \{0,1\}$ and with an edge between vertices if they differ in exactly two coordinates.
(a) How many components does this graph have?
(b) How many vertices of each degree?
(c) Euler circuit?

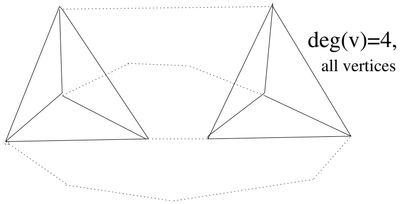
$\boxed{6.2.12}$ As Ex. 6.2.11 but with an edge between vertices if they differ in two or three coordinates.

Exercises

$\boxed{6.2.11}$ This graph consists of all the *face diagonals* of a cube. It has two disjoint components.
No Euler circuit



$\boxed{6.2.12}$ (Refer to Ex. 6.2.11 and connect the vertices from different components in pairs)



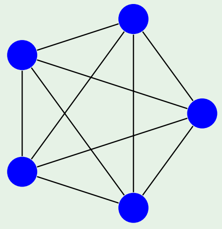
Must have an Euler circuit (why?)

Special Graphs

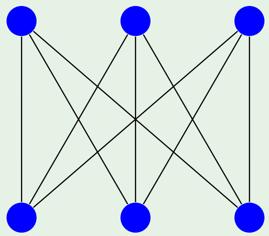
- **Complete graph** K_n
  n vertices, all pairwise connected, $\frac{n(n-1)}{2}$ edges.
- **Complete bipartite graph** $K_{m,n}$
  Has $m + n$ vertices, partitioned into two (disjoint) sets, one of n, the other of m vertices.
  All vertices from different parts are connected; vertices from the same part are disconnected. No. of edges is $m \cdot n$.
- **Complete k-partite graph** $K_{m_1,\dots,m_k}$
  Has $m_1 + \ldots + m_k$ vertices, partitioned into k disjoint sets, respectively of $m_1, m_2, \ldots$ vertices.
  No. of edges is $\sum_{i<j} m_i m_j = \frac{1}{2} \sum_{i \neq j} m_i m_j$
  - These graphs generalise the complete graphs $K_n = K_{\underbrace{1,\ldots,1}_{n}}$

Example

K_5 :



$K_{3,3}$:

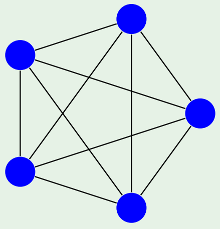


$\boxed{6.2.14}$ Which complete graphs K_n have an Euler circuit?
When do bipartite, 3-partite complete graphs have an Euler circuit?

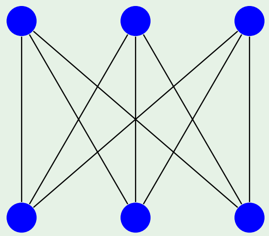
K_n has an Euler circuit for n odd
$K_{m,n}$ — when both m and n are even
$K_{p,q,r}$ — when $p+q, p+r, q+r$ are all even, ie. when p, q, r are all even or all odd

Example

K_5 :



$K_{3,3}$:



$\boxed{6.2.14}$ Which complete graphs K_n have an Euler circuit?
When do bipartite, 3-partite complete graphs have an Euler circuit?

K_n has an Euler circuit for n odd
$K_{m,n}$ — when both m and n are even
$K_{p,q,r}$ — when $p+q, p+r, q+r$ are all even, ie. when p, q, r are all even or all odd

Vertex Traversal

Definition

- **Hamiltonian path** visits every vertex of graph exactly once
- **Hamiltonian circuit** visits every vertex exactly once except the last one, which duplicates the first

NB

Finding such a circuit, or proving it does not exist, is a difficult problem — the worst case is NP-complete.

Examples (when the circuit exists)

- All five regular polyhedra (verify!)
- n-cube; Hamiltonian circuit $=$ *Gray code*
- K_m for all m; $K_{m,n}$ iff $m = n$; $K_{a,b,c}$ iff a, b, c satisfy the triangle inequalities: $a + b \geq c$, $a + c \geq b$, $b + c \geq a$
- Knight's tour on a chessboard (incl. rectangular boards)

Examples when a Hamiltonian circuit does not exist are much harder to construct.
Also, given such a graph it is nontrivial to verify that indeed there is no such a circuit: there is nothing obvious to specify that could assure us about this property.
In contrast, if a circuit is given, it is immediate to verify that it is a Hamiltonian circuit.
These situations demonstrate the often enormous discrepancy in difficulty of 'proving' versus (simply) 'checking'.

Exercises

$\boxed{6.5.5(a)}$ How many Hamiltonian circuits does $K_{n,n}$ have?

Let $V = V_1 \dot\cup V_2$

- start at any vertex in V_1
- go to any vertex in V_2
- go to any *new* vertex in V_1
- ......

There are $n!$ ways to order each part and two ways to choose the 'first' part, implying $c = 2(n!)^2$ circuits.

Exercises

$\boxed{6.5.5(a)}$ How many Hamiltonian circuits does $K_{n,n}$ have?

Let $V = V_1 \dot\cup V_2$

- start at any vertex in V_1
- go to any vertex in V_2
- go to any *new* vertex in V_1
- ......

There are $n!$ ways to order each part and two ways to choose the 'first' part, implying $c = 2(n!)^2$ circuits.

Colouring

Informally: assigning a "colour" to each vertex (e.g. a node in an electric or transportation network) so that the vertices connected by an edge have different colours.
Formally: A mapping $c : V \longrightarrow [1 \,..\, n]$ such that for every $e = (v, w) \in E$

$$c(v) \neq c(w)$$

The minimum n sufficient to effect such a mapping is called the **chromatic number** of a graph $G = (E, V)$ and is denoted $\chi(G)$.

NB

This notion is extremely important in operations research, esp. in scheduling.
There is a dual notion of 'edge colouring' — two edges that share a vertex need to have different colours. Curiously enough, it is much less useful in practice.

Properties of the Chromatic Number

- $\chi(K_n) = n$
- If G has n vertices and $\chi(G) = n$ then $G = K_n$

Proof.

Suppose that G is 'missing' the edge (v, w), as compared with K_n. Colour all vertices, except w, using $n - 1$ colours. Then assign to w the same colour as that of v. □

- If $\chi(G) = 1$ then G is totally disconnected: it has 0 edges.
- If $\chi(G) = 2$ then G is bipartite.
- For any tree $\chi(T) = 2$.
- For any cycle C_n its chromatic number depends on the parity of n — for n even $\chi(C_n) = 2$, while for n odd $\chi(C_n) = 3$.

Cliques

Graph (V', E') *subgraph* of (V, E) — $V' \subseteq V$ and $E' \subseteq E$.

Definition

A **clique** in G is a *complete* subgraph of G. A clique of k nodes is called *k-clique*.
The size of the largest clique is called the *clique number* of the graph and denoted $\kappa(G)$.

Theorem

$\chi(G) \geq \kappa(G)$.

Proof.

Every vertex of a clique requires a different colour, hence there must be at least $\kappa(G)$ colours. □

However, this is the only restriction. For any given k there are graphs with $\kappa(G) = k$, while $\chi(G)$ can be arbitrarily large.

NB

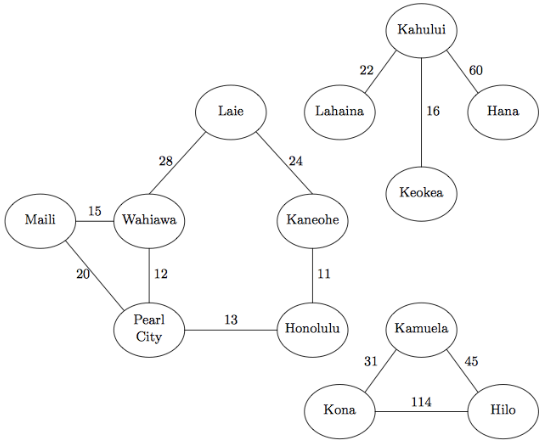
This fact (and such graphs) are important in the analysis of parallel computation algorithms.

- $\kappa(K_n) = n,\ \kappa(K_{m,n}) = 2,\ \kappa(K_{m_1,\ldots,m_r}) = r$.
- If $\kappa(G) = 1$ then G is totally disconnected.
- For a tree $\kappa(T) = 2$.
- For a cycle C_n
  $\kappa(C_3) = 3, \quad \kappa(C_4) = \kappa(C_5) = \ldots = 2$

The difference between $\kappa(G)$ and $\chi(G)$ is apparent with just $\kappa(G) = 2$ — this does not imply that G is bipartite. For example, the cycle C_n for any odd n has $\chi(C_n) = 3$.

Exercise

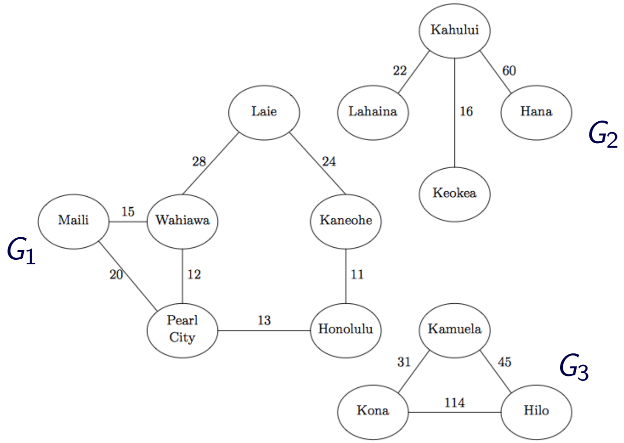
$\boxed{9.10.1}$ (Ullmann)



$\chi(G)$? $\kappa(G)$? A largest clique?

Exercise

$\boxed{9.10.1}$ (Ullmann)



$\chi(G_1) = \kappa(G_1) = 3; \quad \chi(G_2) = \kappa(G_2) = 2; \quad \chi(G_3) = \kappa(G_3) = 3$

Exercise

$\boxed{9.10.3}$ (Ullmann) Let $G = (V, E)$ be an undirected graph. What inequalities must hold between

- the maximal $deg(v)$ for $v \in V$
- $\chi(G)$
- $\kappa(G)$

$max_{v \in V} deg(v) + 1 \geq \chi(G) \geq \kappa(G)$

Exercise

$\boxed{9.10.3}$ (Ullmann) Let $G = (V, E)$ be an undirected graph. What inequalities must hold between

- the maximal $deg(v)$ for $v \in V$
- $\chi(G)$
- $\kappa(G)$

$max_{v \in V} deg(v) + 1 \geq \chi(G) \geq \kappa(G)$

Planar Graphs

Definition

A graph is **planar** if it can be embedded in a plane without its edges intersecting.

Theorem

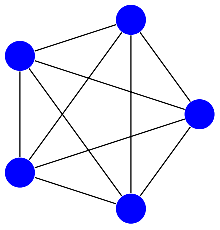
If the graph is planar it can be embedded (without self-intersections) in a plane so that all its edges are straight lines.

NB

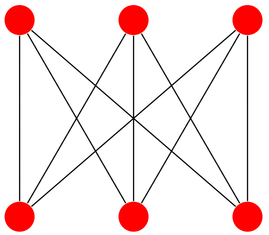
This notion and its related algorithms are extremely important to VLSI and visualizing data.

Two minimal nonplanar graphs

K_5 :

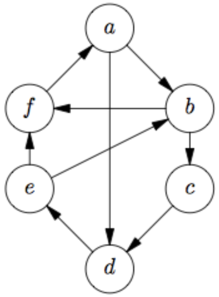


$K_{3,3}$:



Exercise

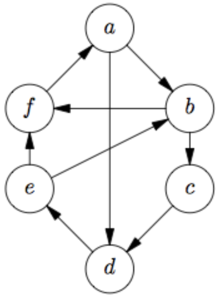
$\boxed{9.10.2}$ (Ullmann)



Is (the undirected version of) this graph planar? Yes

Exercise

$\boxed{9.10.2}$ (Ullmann)

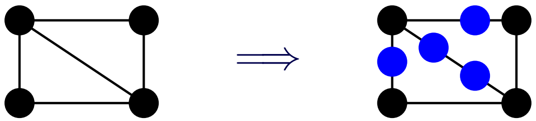


Is (the undirected version of) this graph planar? Yes

Theorem

If graph G contains, as a subgraph, a nonplanar graph, then G itself is nonplanar.

For a graph, *edge subdivision* means to introduce some new vertices, all of degree 2, by placing them on existing edges.



We call such a derived graph a *subdivision* of the original one.

Theorem

If a graph is nonplanar then it must contain a subdivision of K_5 or $K_{3,3}$.

Theorem

K_n for $n \geq 5$ is nonplanar.

Proof.

It contains K_5: choose any five vertices in K_n and consider the subgraph they define. □

Theorem

$K_{m,n}$ is nonplanar when $m \geq 3$ and $n \geq 3$.

Proof.

They contain $K_{3,3}$ — choose any three vertices in each of two vertex parts and consider the subgraph they define. □

Question

Are all $K_{m,1}$ planar?

Answer

Yes, they are trees of two levels — the root and m leaves.

Question

Are all $K_{m,1}$ planar?

Answer

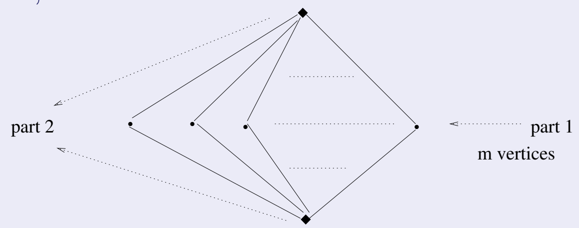
Yes, they are trees of two levels — the root and m leaves.

Question

Are all $K_{m,2}$ planar?

Answer

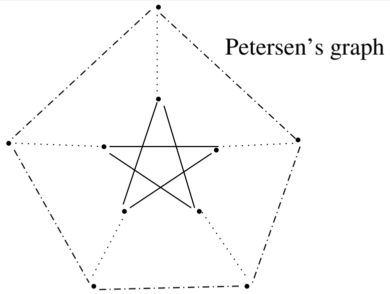
Yes; they can be represented by "glueing" together two such trees at the leaves.
Sketching $K_{m,2}$



Also, among the k-partite graphs, planar are $K_{2,2,2}$ and $K_{1,1,m}$. The latter can be depicted by drawing one extra edge in $K_{2,m}$, connecting the top and bottom vertices.

NB

Finding a 'basic' nonplanar obstruction is not always simple



It contains a subdivision of both $K_{3,3}$ and K_5 while it does not directly contain either of them.

Summary

- Graphs, trees, vertex degree, connected graphs, paths, cycles
- Graph isomorphisms, automorphisms
- Special graphs: complete, complete bi-, k-partite
- Traversals
  - Euler paths and circuits (edge traversal)
  - Hamiltonian paths and circuits (vertex traversal)
- Graph properties: chromatic number, clique number, planarity

COMP9020 Lecture 7
Session 1, 2017
Induction and Recursion

- Textbook (R & W) - Ch. 4, Sec. 4.2, 4.4, 4.6
- Problem set 7
- Supplementary Exercises Ch. 4 (R & W)

Inductive Reasoning

Suppose we would like to reach a conclusion of the form

P(x) *for all* x (of some type)

Inductive reasoning (as understood in philosophy) proceeds from examples.
E.g. From "This swan is white, that swan is white, in fact every swan I have seen so far is white"
Conclude: "Every Swan is white"

NB

This may be a good way to discover hypotheses.
But it is not a valid principle of reasoning!

Mathematical induction *is a variant that is valid.*

Inductive Reasoning

Suppose we would like to reach a conclusion of the form

P(x) *for all* x (of some type)

Inductive reasoning (as understood in philosophy) proceeds from examples.
E.g. From "This swan is white, that swan is white, in fact every swan I have seen so far is white"
Conclude: "Every Swan is white"

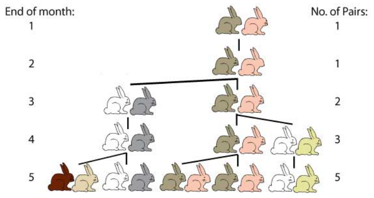
NB

This may be a good way to discover hypotheses.
But it is not a valid principle of reasoning!

Mathematical induction *is a variant that is valid.*

Example

Fibonacci Numbers:

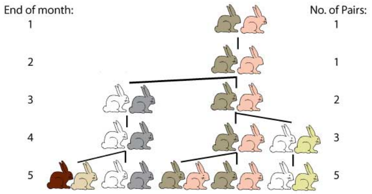


$\mathrm{FIB}(1) = 1$
$\mathrm{FIB}(2) = 1$
$\mathrm{FIB}(n) = \mathrm{FIB}(n-1) + \mathrm{FIB}(n-2)$ for all $n > 2$

$\mathrm{FIB}(1)$	1	$\mathrm{FIB}(5)$	5	$\mathrm{FIB}(9)$	34
$\mathrm{FIB}(2)$	1	$\mathrm{FIB}(6)$	8	$\mathrm{FIB}(10)$	55
$\mathrm{FIB}(3)$	2	$\mathrm{FIB}(7)$	13	$\mathrm{FIB}(11)$	89
$\mathrm{FIB}(4)$	3	$\mathrm{FIB}(8)$	21	$\mathrm{FIB}(12)$	144

Example

Fibonacci Numbers:



$\textsc{fib}(1) = 1$
$\textsc{fib}(2) = 1$
$\textsc{fib}(n) = \textsc{fib}(n-1) + \textsc{fib}(n-2)$ for all $n > 2$

$\textsc{fib}(1)$	1
$\textsc{fib}(2)$	1
$\textsc{fib}(3)$	2
$\textsc{fib}(4)$	3

$\textsc{fib}(5)$	5
$\textsc{fib}(6)$	8
$\textsc{fib}(7)$	13
$\textsc{fib}(8)$	21

$\textsc{fib}(9)$	34
$\textsc{fib}(10)$	55
$\textsc{fib}(11)$	89
$\textsc{fib}(12)$	144

Example

$\textsc{fib}(1) = 1$
$\textsc{fib}(2) = 1$
$\textsc{fib}(n) = \textsc{fib}(n-1) + \textsc{fib}(n-2)$ for all $n > 2$

$\textsc{fib}(1)$	1
$\textsc{fib}(2)$	1
$\textsc{fib}(3)$	2
$\textsc{fib}(4)$	3

$\textsc{fib}(5)$	5
$\textsc{fib}(6)$	8
$\textsc{fib}(7)$	13
$\textsc{fib}(8)$	21

$\textsc{fib}(9)$	34
$\textsc{fib}(10)$	55
$\textsc{fib}(11)$	89
$\textsc{fib}(12)$	144

Claim: Every 4th Fibonacci number is divisible by 3
How can we prove this?

Mathematical Induction

Mathematical Induction is based not just on a set of examples, but also a rule for deriving new cases of $P(x)$ from cases for which P is known to hold.
General structure of reasoning by mathematical induction:

Base Case [B]: $P(a_1), P(a_2), \ldots, P(a_n)$ for some small set of examples $a_1 \ldots a_n$ (often $n = 1$)
Inductive Step [I]: A general rule showing that if $P(x)$ holds for some cases $x = x_1, \ldots, x_k$ then $P(y)$ holds for some new case y, constructed in some way from $x_1, \ldots, x_k$.

Conclusion: Starting with $a_1 \ldots a_n$ and repeatedly applying the construction of y from existing values, we can eventually construct all values in the domain of interest.

Example

Suppose we start with $x = 0$ and repeatedly apply the construction $x \mapsto x + 1$.
Then we construct values
$0, 0+1 = 1, 1+1 = 2, 2+1 = 3, 3+1 = 4, \ldots$
In the limit, this is all of $\mathbb{N}$
The corresponding principle of Mathematical Induction on $\mathbb{N}$:

Base Case [B]: $P(0)$
Inductive Step [I]: $\forall k \geq 0 \, (P(k) \Rightarrow P(k+1))$

Conclusion: $\forall n \in \mathbb{N} \; P(n)$

Example

Suppose we start with $x = 0$ and repeatedly apply the construction $x \mapsto x + 1$.
Then we construct values
$0, 0 + 1 = 1, 1 + 1 = 2, 2 + 1 = 3, 3 + 1 = 4, \ldots$
In the limit, this is all of $\mathbb{N}$
The corresponding principle of Mathematical Induction on $\mathbb{N}$:

Base Case [B]: $P(0)$
Inductive Step [I]: $\forall k \geq 0 \, (P(k) \Rightarrow P(k+1))$

Conclusion: $\forall n \in \mathbb{N} \; P(n)$

Example

Suppose we start with $x = 0$ and repeatedly apply the construction $x \mapsto x + 1$.
Then we construct values
$0, 0 + 1 = 1, 1 + 1 = 2, 2 + 1 = 3, 3 + 1 = 4, \ldots$
In the limit, this is all of $\mathbb{N}$
The corresponding principle of Mathematical Induction on $\mathbb{N}$:

Base Case [B]: $P(0)$
Inductive Step [I]: $\forall k \geq 0 \, (P(k) \Rightarrow P(k+1))$

Conclusion: $\forall n \in \mathbb{N} \; P(n)$

Inductive Hypothesis

To prove the Inductive Step, $P(k) \Rightarrow P(k+1)$ for $k \geq 0$, we typically proceed as follows:

Assume $P(k)$, for an arbitrary $k \geq 0$

$\vdots$ (steps of reasoning, often using the assumption that $P(k)$)

Conclude $P(k+1)$.

Here $P(k)$ is called the *Inductive Hypothesis*

Example

Theorem. For all $n \in \mathbb{N}$, we have

$$P(n): \quad \sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$

Proof.

[B] $P(0)$, i.e.

$$\sum_{i=0}^{0} i = \frac{0(0+1)}{2}$$

[I] $\forall k \geq 0 \, (P(k) \Rightarrow P(k+1))$, i.e.

$$\sum_{i=0}^{k} i = \frac{k(k+1)}{2} \Rightarrow \sum_{i=0}^{k+1} i = \frac{k(k+1)}{2} + (k+1) = \frac{(k+1)(k+2)}{2}$$

□

Example

Theorem. For all $n \in \mathbb{N}$, we have

$$P(n): \quad \sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$

Proof.

[B] $P(0)$, i.e.

$$\sum_{i=0}^{0} i = \frac{0(0+1)}{2}$$

[I] $\forall k \geq 0\,(P(k) \Rightarrow P(k+1))$, i.e.

$$\sum_{i=0}^{k} i = \frac{k(k+1)}{2} \Rightarrow \sum_{i=0}^{k+1} i = \frac{k(k+1)}{2} + (k+1) = \frac{(k+1)(k+2)}{2}$$

□

Example

Theorem. For all $n \in \mathbb{N}$, we have

$$P(n): \quad \sum_{i=0}^{n} i = \frac{n(n+1)}{2}$$

Proof.

[B] $P(0)$, i.e.

$$\sum_{i=0}^{0} i = \frac{0(0+1)}{2}$$

[I] $\forall k \geq 0\,(P(k) \Rightarrow P(k+1))$, i.e.

$$\sum_{i=0}^{k} i = \frac{k(k+1)}{2} \Rightarrow \sum_{i=0}^{k+1} i = \frac{k(k+1)}{2} + (k+1) = \frac{(k+1)(k+2)}{2}$$

□

Variations

1. Induction from m upwards
2. Induction steps >1
3. Strong induction
4. Backward induction
5. Forward-backward induction
6. Structural induction

Induction From m Upwards

If

[B] $P(m)$

[I] $\forall k \geq m\,(P(k) \Rightarrow P(k+1))$

then

[C] $\forall n \geq m\,(P(n))$

Example

Theorem. For all $n \geq 1$, the number $8^n - 2^n$ is divisible by 6.

[B] $8^1 - 2^1$ is divisible by 6
[I] if $8^k - 2^k$ is divisible by 6, then so is $8^{k+1} - 2^{k+1}$, for all $k \geq 1$

Prove [I] using the "trick" to rewrite 8^{k+1} as $8 \cdot (8^k - 2^k + 2^k)$ which allows you to apply the Ind. Hyp. on $8^k - 2^k$

Exercise

Consider an **increasing** function $f : \mathbb{N} \longrightarrow \mathbb{N}$
i.e., $\forall m, n \, (m \leq n \;\Rightarrow\; f(m) \leq f(n))$
and a function $g : \mathbb{N} \longrightarrow \mathbb{N}$ such that

- $f(0) < g(0)$
- $f(1) = g(1)$
- if $f(k) \geq g(k)$ then $f(k+1) \geq g(k+1)$, for all $k \in \mathbb{N}$

Always true, false or could be either?
(a) $f(n) > g(n)$ for all $n \geq 1$ — false
(b) $f(n) > g(n)$ for some $n \geq 1$ — could be either
(c) $f(n) \geq g(n)$ for all $n \geq 1$ — true
(d) g is decreasing $(m \leq n \;\Rightarrow\; g(m) \geq g(n))$ — could be either

Exercise

Consider an **increasing** function $f : \mathbb{N} \longrightarrow \mathbb{N}$
i.e., $\forall m, n \, (m \leq n \;\Rightarrow\; f(m) \leq f(n))$
and a function $g : \mathbb{N} \longrightarrow \mathbb{N}$ such that

- $f(0) < g(0)$
- $f(1) = g(1)$
- if $f(k) \geq g(k)$ then $f(k+1) \geq g(k+1)$, for all $k \in \mathbb{N}$

Always true, false or could be either?
(a) $f(n) > g(n)$ for all $n \geq 1$ — false
(b) $f(n) > g(n)$ for some $n \geq 1$ — could be either
(c) $f(n) \geq g(n)$ for all $n \geq 1$ — true
(d) g is decreasing $(m \leq n \;\Rightarrow\; g(m) \geq g(n))$ — could be either

Induction Steps $\ell > 1$

If
[B] $P(m)$
[I] $P(k) \Rightarrow P(k + \ell)$ for all $k \geq m$
then
[C] $P(n)$ for every ℓ'th $n \geq m$

Example

$$\begin{aligned}
&\textsc{fib}(1) = 1\\
&\textsc{fib}(2) = 1\\
&\textsc{fib}(n) = \textsc{fib}(n-1) + \textsc{fib}(n-2)
\end{aligned}$$

Every 4th Fibonacci number is divisible by 3.

[B] $\textsc{fib}(4) = 3$ is divisible by 3
[I] if $3 \mid \textsc{fib}(k)$, then $3 \mid \textsc{fib}(k+4)$, for all $k \geq 4$

Prove [I] by rewriting $\textsc{fib}(k+4)$ in such a way that you can apply the Ind. Hyp. on $\textsc{fib}(k)$

Strong Induction

This is a version in which the inductive hypothesis is stronger. Rather than using the fact that $P(k)$ holds for a single value, we use *all* values up to k.

If
[B] $P(m)$
[I] $[P(m) \wedge P(m+1) \wedge \ldots \wedge P(k)] \Rightarrow P(k+1)$ for all $k \geq m$
then
[C] $P(n)$, for all $n \geq m$

Example

Claim: All integers ≥ 2 can be written as a product of primes.

[B] 2 is a product of primes
[I] If all x with $2 \leq x \leq k$ can be written as a product of primes, then $k+1$ can be written as a product of primes, for all $k \geq 2$

Proof for [I]?

Negative Integers, Backward Induction

NB

Induction can be conducted over any subset of $\mathbb{Z}$ with least element. Thus m can be negative; eg. base case $m = -10^6$.

NB

One can apply induction in the 'opposite' direction $p(m) \Rightarrow p(m-1)$. It means considering the integers with the opposite ordering where the next number after n is $n-1$. Such induction would be used to prove some $p(n)$ for all $n \leq m$.

NB

Sometimes one needs to reason about all integers $\mathbb{Z}$. This requires two separate simple induction proofs: one for $\mathbb{N}$, another for $-\mathbb{N}$. They both would start form some initial values, which could be the same, e.g. zero. Then the first proof would proceed through positive integers; the second proof through negative integers.

Forward-Backward Induction

Idea

To prove $P(n)$ for all $n \geq k_0$

- *verify $P(k_0)$*
- *prove $P(k_i)$ for infinitely many $k_0 < k_1 < k_2 < k_3 < \dots$*
- *fill the gaps*
  $P(k_1) \Rightarrow P(k_1 - 1) \Rightarrow P(k_1 - 2) \Rightarrow \dots \Rightarrow P(k_0 + 1)$
  $P(k_2) \Rightarrow P(k_2 - 1) \Rightarrow P(k_2 - 2) \Rightarrow \dots \Rightarrow P(k_1 + 1)$
  . . . .

Example

Claim

Binary search in an (ordered) list of $n - 1$ elements requires no more than $\lceil \log_2 n \rceil$ comparisons.

Proof.

(i) it holds for $n = 1$

(ii) if it holds for k then it holds for $2k$,
thus true for $2,\ 4,\ 8,\ 16,\ \dots$

(iii) if it holds for 2^i then it holds for $2^i - 1,\ 2^i - 2, \dots, 2^{i-1} + 1$,
thus true for all n.

□

Forward-Backward Induction: Formalisation

[B] $P(k_0)$
[I] if $P(k)$ then $P(k')$ for *some* $k' > k$
[I] and [B] alone imply $P(k_i)$ for infinitely many $k_0 < k_1 < k_2 < \dots$
[D] $P(k) \Rightarrow P(k-1)$ for all k between k_i's and k_{i+1}'s (downward step)
[C] $\forall n \geq k_0\, (P(n))$

NB

This form of induction is extremely important for the analysis of algorithms.

Various Inductive Arguments

Induction by cases
Consider separately various subsets $S_1, S_2, \dots \subset \mathbb{N}$, eg. odd and even numbers, making sure that they jointly cover all of $\mathbb{N}$.
Complete the proof (by induction) separately for each subset.

Example

Any amount $n \in \mathbb{N}$ greater than \$1 can be paid using units ('coins') of \$2 and \$3.
To prove it we conduct **two** inductive arguments: one over the even numbers, the other over the odd numbers.
Equivalently, we can split the proof into **three** cases: one for the numbers divisible by 3, one for those with remainder 1 and one for those with remainder 2. This means more cases, but each one of them is a bit simpler.

NB

One can use the same type of argument for any two coin values \$m and \$n if $\gcd(m,n)=1$ *and amount* $> \$(mn-m-n)$*.*
The proof splits into m *cases: one for numbers divisible by* m*, then one for those having remainder 1 mod* m*, then ...*

Infinite Descent

To prove that $Q(n)$, for all $n \geq m$, show

- $\neg Q(n) \Rightarrow \neg Q(n')$ for some $n' < n$
- there cannot be arbitrarily small n s.t. $Q(n)$ is false; in particular the "base case" $Q(m)$ is *true*

This amounts to a proof by contradiction: to verify $\forall n\, Q(n)$ we assume (provisionally) its negation $\exists n \neg Q(n)$ and proceed to show that there would have to exist a smaller n' such that $\neg Q(n')$.
Usually the conditions of the problem make it clear that no such infinite decreasing chain $\ldots < n'' < n' < n$ can possibly exist.

Example

Theorem

For a planar, connected graph let F *be the number of faces (enclosures) including the exterior face,* E *the number of edges, and* V *the number of vertices.*
Euler's formula *holds:*

$$V - E + F = 2 \tag{EF}$$

Proof.

Suppose $G = (V, E)$ is a planar connected graph that **violates** (EF).

First observe that G must have an edge because it is connected and the graph with just one vertex satisfies (EF).

If G has an outside edge, that is, an edge separating the exterior face from an interior face, then removing that edge results in a smaller (planar, connected) graph, also violating (EF) because both E and F are reduced by 1.

If G has no outside edge then it has a vertex v of degree 1. Removing v reduces both V and E by 1 while F remains unchanged. It follows that we again found a smaller (planar, connected) graph violating (EF). □

Structural Induction

The induction schemes can be applied not only to natural numbers (and integers) but to any partially ordered set in general.

The basic approach is always the same — we need to verify that

- **[I]** for any given object, if the property in question holds for all its predecessors (‘smaller’ objects) then it holds for the object itself
- **[B]** the property holds for all minimal objects — objects that have no predecessors; they are usually very simple objects allowing immediate verification

Example: Induction on Rooted Trees

We write $T = \langle r;\ T_1, T_2, \ldots, T_k \rangle$ for a tree T with root r and k subtrees at the root $T_1, \ldots, T_k$

If

[B] $p(\langle v; \rangle)$ for trees with only a root

[I] $p(T_1) \wedge \ldots \wedge p(T_k) \Rightarrow p(T)$ for all trees $T = \langle r;\ T_1, T_2, \ldots, T_k \rangle$

then

[C] $p(T)$ for every tree T

Example

Theorem

In any rooted tree the number of vertices is one more than the number of edges.

Proof.

[B] If $T = \langle v; \rangle$ then $v(T) = 1$ and $e(T) = 0$

□

Example

Theorem

In any rooted tree the number of vertices is one more than the number of edges.

Proof.

[B] If $T = \langle v; \rangle$ then $v(T) = 1$ and $e(T) = 0$

[I] If $T = \langle r;\ T_1, T_2, \ldots, T_k \rangle$ then
$v(T) = 1 + \sum_{i=1}^{k} v(T_i)$ and $e(T) = k + \sum_{i=1}^{k} e(T_i)$

□

Example

Theorem

In any rooted tree the number of vertices is one more than the number of edges.

Proof.

[B] If $T = \langle v; \rangle$ then $v(T) = 1$ and $e(T) = 0$

[I] If $T = \langle r;\ T_1, T_2, \dots, T_k \rangle$ then

$$v(T) = 1 + \sum_{i=1}^{k} v(T_i) \quad \text{and} \quad e(T) = k + \sum_{i=1}^{k} e(T_i)$$

From the Ind. Hyp. on $T_1, \dots, T_k$ it follows that

$$\sum_{i=1}^{k} v(T_i) = \sum_{i=1}^{k}(e(T_i) + 1) = (\sum_{i=1}^{k} e(T_i)) + k$$

Therefore

$$v(T) = 1 + (\sum_{i=1}^{k} e(T_i)) + k = 1 + e(T)$$

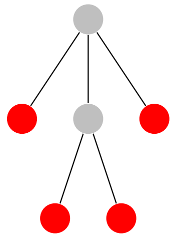
□

Example

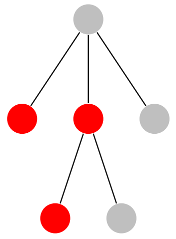
Theorem

In any rooted tree the number of leaves is one more than the number of vertices that have a right sibling.

Proof: exercise



4 leaves



3 vertices with right sibling

Recursive Definitions

They comprise basis (B) and recursive process (R).
A sequence is recursively defined when (typically)
(B) some initial terms are specified, perhaps only the first one;
(R) later terms stated as functional expressions of the earlier terms.

Examples

Factorial:

(B) $0! = 1$

(R) $(n+1)! = (n+1) \cdot n!$

Fibonacci numbers:

(B) $\textsc{fib}(1) = 1$

(B) $\textsc{fib}(2) = 1$

(R) $\textsc{fib}(n) = \textsc{fib}(n-1) + \textsc{fib}(n-2)$

NB

(R) also called **recurrence formula**

Inductive Proofs About Recursive Definitions

Proofs about recursively defined function very often proceed by a mathematical induction following the structure of the definition.

Example

$\forall n \in \mathbb{N}\left(n! \geq 2^{n-1}\right)$

Proof.

[B] $0! = 1 \geq \frac{1}{2} = 2^{0-1}$

[I] Assume $n \geq 1$.

$$\begin{aligned} (n+1)! = n! \cdot (n+1) &\geq 2^{n-1} \cdot (n+1) && \text{by Ind. Hyp.} \\ &\geq 2^{n-1} \cdot 2 && \text{by } n \geq 1 \\ &= 2^n \end{aligned}$$

□

Exercise

$\boxed{4.4.2}$ Define $s_1 = 1$ and $s_{n+1} = \frac{1}{1+s_n}$ for $n \geq 1$

Then $s_1 = 1,\ s_2 = \frac{1}{2},\ s_3 = \frac{2}{3},\ s_4 = \frac{3}{5},\ s_5 = \frac{5}{8}, \ldots$

The numbers in numerator and denominator remind one of the Fibonacci sequence.

Prove by induction that

$$s_n = \frac{\text{FIB}(n)}{\text{FIB}(n+1)}$$

Example (continued)

Furthermore,

$$\lim_{n\to\infty} s_n = \frac{2}{\sqrt{5}+1} = \frac{\sqrt{5}-1}{2} \approx 0.6$$

This is obtained by showing (using induction!) that

$$\text{FIB}(n) = \frac{1}{\sqrt{5}}(r_1^n - r_2^n)$$

where $r_1 = \frac{1+\sqrt{5}}{2}$ and $r_2 = \frac{1-\sqrt{5}}{2}$

Exercise

$\boxed{4.4.4}$ (a) Give a recursive definition for the sequence

$$(2,\ 4,\ 16,\ 256,\ \ldots)$$

To generate $a_n = 2^{2^n}$ use $a_n = (a_{n-1})^2$.
(The related "Fermat numbers" $F_n = 2^{2^n} + 1$ are used in cryptography.)

(b) Give a recursive definition for the sequence

$$(2,\ 4,\ 16,\ 65536,\ \ldots)$$

To generate a "stack" of n 2's use $b_n = 2^{b_{n-1}}$.
(These are *Ackermann's numbers*, first used in logic. The inverse function is extremely slow growing; it is important for the analysis of several data organisation algorithms.)

Exercise

4.4.4 (a) Give a recursive definition for the sequence

$$(2,\ 4,\ 16,\ 256,\ \ldots)$$

To generate $a_n = 2^{2^n}$ use $a_n = (a_{n-1})^2$.
(The related "Fermat numbers" $F_n = 2^{2^n} + 1$ are used in cryptography.)

(b) Give a recursive definition for the sequence

$$(2,\ 4,\ 16,\ 65536,\ \ldots)$$

To generate a "stack" of n 2's use $b_n = 2^{b_{n-1}}$.
(These are *Ackermann's numbers*, first used in logic. The inverse function is extremely slow growing; it is important for the analysis of several data organisation algorithms.)

Correctness of Recursive Definition

A recurrence formula is correct if the computation of any later term can be reduced to the initial values given in (B).

Example (Incorrect definition)

- Function $g(n)$ is defined recursively by

$$g(n) = g(g(n-1) - 1) + 1, \qquad g(0) = 2.$$

The definition of $g(n)$ is incomplete — the recursion may not terminate:
Attempt to compute $g(1)$ gives

$$g(1) = g(g(0)-1)+1 = g(1)+1 = \ldots = g(1)+1+1+1\ldots$$

When implemented, it leads to an overflow; most static analyses cannot detect this kind of ill-defined recursion.

Example (continued)

However, the definition could be repaired. For example, we can add the specification specify $g(1) = 2$.

Then $g(2) = g(2-1) + 1 = 3$,
$g(3) = g(g(2)-1) + 1 = g(3-1) + 1 = 4$,
$\ldots$

In fact, by induction … $g(n) = n + 1$

This illustrates a very important principle: the boundary (limiting) cases of the definition are evaluated *before* applying the recursive construction.

Example

Function $f(n)$ is defined by

$$f(n) = f(\lceil n/2 \rceil), \quad f(0) = 1$$

When evaluated for $n = 1$ it leads to

$$f(1) = f(1) = f(1) = \ldots$$

This one can also be repaired. For example, one could specify that $f(1) = 1$.
This would lead to a constant function $f(n) = 1$ for all $n \geq 0$.

Mutual Recursion

Several more sophisticated programs employ a technique of two procedures calling each other. Of course, it should be designed so that each consecutive call refers to ever smaller parameters, so that the entire process terminates. This method is often used in computer graphics, in particular for generating fractal images (basis of various imaginary landscapes, among others).

Summary

- Mathematical induction:
  base case(s), induction hypothesis $P(k)$,
  inductive step $\forall k\,(P(k) \Rightarrow P(k+1))$, conclusion
- Variations:
  strong ind., forward-backward ind., ind. by cases, structural ind.
- Recursive definitions

COMP9020 Lecture 8
Session 1, 2017
Running Time of Programs

aka “Big-Oh Notation”

- Textbook (R & W) - Ch. 4, Sec. 4.3, 4.5
- Problem set 8
- Supplementary Exercises Ch. 4 (R & W)

Motivation

We would like to be able to talk about the resources (running time, memory, energy consumption) required by a program/algorithm as a function $f(n)$ of the size n of its input.

Example

How long does a given sorting algorithm take to run on a list of n elements?

Problem 1: the exact running time may depend on

- compiler optimisations
- processor speed
- cache size

Each of these may affect the resource usage by up to a *linear* factor, making it hard to state a general claim about running times.

Problem 2: Many algorithms that arise in practice have resource usage that can be expressed only as a rather complicated function. E.g.

$$f(n) = 20n^2 + 2n\log(n) + (n-100)\log(n)^2 + \frac{1}{2^n}\log(\log(n))$$

The main contribution to the value of the function for “large” input sizes n is the term of the *highest order*:

$$20n^2$$

We would like to be able to *ignore the terms of lower order*

$$2n\log(n) + (n-100)\log(n)^2 + \frac{1}{2^n}\log(\log(n))$$

Order of Growth

Example

Consider two algorithms, one with running time $f_1(n) = \frac{1}{10}n^2$, the other with running time $f_2 = 10n \log n$ (measured in milliseconds).

Input size	$f_1(n)$	$f_2(n)$
100	0.01s	2s
1000	1s	30s
10000	1m40s	6m40s
100000	2h47m	1h23m
1000000	11d14h	16h40h
10000000	3y3m	8d2h

Order of growth provides a way to abstract away from these two problems, and focus on what is essential to the size of the function, by saying that "the (complicated) function f is of *roughly the same size* (for large input) as the (simple) function g"

"Big-Oh" Asymptotic Upper Bounds

Definition

Let $f, g : \mathbb{N} \to \mathbb{R}$. We say that g is *asymptotically less than* f (or: f **is an upper bound of** g) if there exists $n_0 \in \mathbb{N}$ and a real constant $c > 0$ such that for all $n \geq n_0$,

$$g(n) \leq c \cdot f(n)$$

Write $\mathcal{O}(f(n))$ for the class of all functions g that are asymptotically less than f.

Example

$$\frac{1}{10}n^2 \in \mathcal{O}(n^2) \qquad 10n \log n \in \mathcal{O}(n \log n)$$

$$\mathcal{O}(n \log n) \subsetneq \mathcal{O}(n^2)$$

The traditional notation has been

$$g(n) = \mathcal{O}(f(n))$$

instead of

$$g(n) \in \mathcal{O}(f(n))$$

It allows one to use $\mathcal{O}(f(n))$ or similar expressions as part of an equation; of course these 'equations' express only an approximate equality.

Thus,

$$T(n) = 2 \cdot T\left(\frac{n}{2}\right) + \mathcal{O}(n)$$

means

"There exists a function $f(n) \in \mathcal{O}(n)$ such that $T(n) = 2T(\frac{n}{2}) + f(n)$."

Examples

$$5n^2 + 3n + 2 = \mathcal{O}(n^2)$$

$$n^3 + 2^{100}n^2 + 2n + 2^{2^{100}} = \mathcal{O}(n^3)$$

Generally, for constants $a_k \ldots a_0$,

$$a_k n^k + a_{k-1}n^{k-1} + \ldots + a_0 = \mathcal{O}(n^k)$$

“Big-Theta” Notation

Definition

Two functions f, g have the *same order of growth* if they scale up in the same way:
There exists $n_0 \in \mathbb{N}$ and real constants $c > 0, d > 0$ such that for all $n \geq n_0$,

$$c \cdot f(n) \leq g(n) \leq d \cdot f(n)$$

Write $\Theta(f(n))$ for the class of all functions g that have the same order of growth as f.

If $g \in \mathcal{O}(f)$ we say that f is (gives) an *upper bound* on the order of growth of g; if $g \in \Theta(f)$ we call it a **tight bound**.

Observe that, somewhat symmetrically

$$g \in \Theta(f) \iff f \in \Theta(g)$$

We obviously have

$$\Theta(f(n)) \subseteq \mathcal{O}(f(n))$$

At the same time the ‘Big-Oh’ is *not* a symmetric relation

$$g \in \mathcal{O}(f) \not\Rightarrow f \in \mathcal{O}(g)$$

More Examples

- All logarithms $\log_b x$ have the same order, irrespective of the value of b

$$\mathcal{O}(\log_2 n) = \mathcal{O}(\log_3 n) = \ldots = \mathcal{O}(\log_{10} n) = \ldots$$

- Exponentials r^n, s^n to different bases $r < s$ have different orders, e.g. there is no $c > 0$ such that $3^n < c \cdot 2^n$ for all n

$$\mathcal{O}(r^n) \subsetneq \mathcal{O}(s^n) \subsetneq \mathcal{O}(t^n) \ \ldots \quad \text{for} \quad r < s < t \ldots$$

- Similarly for polynomials

$$\mathcal{O}(n^k) \subsetneq \mathcal{O}(n^l) \subsetneq \mathcal{O}(n^m) \ \ldots \quad \text{for} \quad k < l < m \ldots$$

Here are some of the most common functions occurring in the analysis of the performance of programs (algorithm complexity):

$1,\ \log\log n,\ \log n,\ \sqrt{n},\ \sqrt{n}(\log n)^k,\ \sqrt{n}(\log n)^2, \ldots$
$n,\ n\log\log n,\ n\log n,\ n^{1.5},\ n^2,\ n^3, \ldots$
$2^n,\ 2^n \log n,\ n2^n,\ 3^n,\ \ldots$
$n!,\ n^n,\ n^{2n}, \ldots,\ n^{n^2},\ n^{2^n}, \ldots$

Notation: $\mathcal{O}(1) \equiv$ const, although technically it could be any function that varies between two constants c and d.

Exercise

4.3.5 True or false?

(a) $2^{n+1} = \mathcal{O}(2^n)$ — true

(b) $(n+1)^2 = \mathcal{O}(n^2)$ — true

(c) $2^{2n} = \mathcal{O}(2^n)$ — false

(d) $(200n)^2 = \mathcal{O}(n^2)$ — true

4.3.6 True or false?

(b) $\log(n^{73}) = \mathcal{O}(\log n)$ — true

(c) $\log n^n = \mathcal{O}(\log n)$ — false

(d) $(\sqrt{n}+1)^4 = \mathcal{O}(n^2)$ — true

Exercise

4.3.5 True or false?

(a) $2^{n+1} = \mathcal{O}(2^n)$ — true

(b) $(n+1)^2 = \mathcal{O}(n^2)$ — true

(c) $2^{2n} = \mathcal{O}(2^n)$ — false

(d) $(200n)^2 = \mathcal{O}(n^2)$ — true

4.3.6 True or false?

(b) $\log(n^{73}) = \mathcal{O}(\log n)$ — true

(c) $\log n^n = \mathcal{O}(\log n)$ — false

(d) $(\sqrt{n}+1)^4 = \mathcal{O}(n^2)$ — true

Analysing the Complexity of Algorithms

We want to know what to expect of the running time of an algorithm as the input size goes up. To avoid vagaries of the specific computational platform we measure the performance in the number of *elementary operations* rather than clock time.
Typically we consider the four arithmetic operations, comparisons, and logical operations as elementary; they take one processor cycle (or a fixed small number of cycles).

A typical approach to determining the **complexity** of an algorithm, i.e. an asymptotic estimate of its running time, is to write down a recurrence for the number of operations as a function of the size of the input.
We then *solve the recurrence up to an order of size.*

Example: Insertion Sort

Consider the following recursive algorithm for sorting a list. We take the cost to be the number of list element comparison operations.
Let $T(n)$ denote the total cost of running InsSort(L)

```
InsSort(L):
  Input  list $L[0..n-1]$ containing $n$ elements

  if $n \le 1$ then return $L$                                  cost $= 0$
  let $L_1 = $ InsSort$(L[0..n-2])$                             cost $= T(n-1)$
  let $L_2 = $ result of inserting element $L[n-1]$ into $L_1$ (sorted!)
              in the appropriate place                          cost $\le n-1$
  return $L_2$
```

$$T(n) = T(n-1) + n - 1 \qquad T(1) = 0$$

Example: Insertion Sort

Consider the following recursive algorithm for sorting a list. We take the cost to be the number of list element comparison operations.
Let $T(n)$ denote the total cost of running InsSort(L)

```
InsSort($L$):
  Input  list $L[0..n-1]$ containing $n$ elements

  if $n \leq 1$ then return $L$                                   cost $= 0$
  let $L_1 = $ InsSort($L[0..n-2]$)                               cost $= T(n-1)$
  let $L_2 = $ result of inserting element $L[n-1]$ into $L_1$ (sorted!)
                in the appropriate place                          cost $\leq n-1$
  return $L_2$
```

$$T(n) = T(n-1) + n - 1 \qquad T(1) = 0$$

Solving the Recurrence

Unwinding $T(n) = T(n-1) + (n-1)$, $T(1) = 0$

$$\begin{aligned} T(n) &= T(n-1) + (n-1) \\ &= T(n-2) + (n-2) + (n-1) \\ &= T(n-3) + (n-3) + (n-2) + (n-1) \\ &\vdots \\ &= T(1) + 1 + \ldots + (n-1) \\ &= 0 + 1 + \ldots + (n-1) \\ &= \frac{n(n-1)}{2} \\ &= \mathcal{O}(n^2) \end{aligned}$$

Hence, Insertion Sort is in $\mathcal{O}(n^2)$
We also say: "The complexity of Insertion Sort is quadratic."

Exercise

Linear recurrence

$$T(n) = T(n-1) + g(n), \quad T(0) = a$$

has the precise solution (cf. last week's homework, Exercise 4)

$$T(n) = a + \sum_{j=1}^{n} g(j)$$

Give a tight big-Oh upper bound on the solution if $g(n) = n^2$

$$T(n) = a + \sum_{j=1}^{n} j^2 = a + \frac{n(n+1)(2n+1)}{6} = \mathcal{O}(n^3)$$

Exercise

Linear recurrence

$$T(n) = T(n-1) + g(n), \quad T(0) = a$$

has the precise solution (cf. last week's homework, Exercise 4)

$$T(n) = a + \sum_{j=1}^{n} g(j)$$

Give a tight big-Oh upper bound on the solution if $g(n) = n^2$

$$T(n) = a + \sum_{j=1}^{n} j^2 = a + \frac{n(n+1)(2n+1)}{6} = \mathcal{O}(n^3)$$

A General Result

Recurrences for algorithm complexity often involve a linear reduction in subproblem size.

Theorem

- *(case 1)* $T(n) = T(n-1) + bn^k$
  solution $T(n) = \mathcal{O}(n^{k+1})$
- *(case 2)* $T(n) = cT(n-1) + bn^k, \quad c > 1:$
  solution $T(n) = \mathcal{O}(c^n)$

This contrasts with *divide-and-conquer algorithms*, where we solve a problem of size n by recurrence to subproblems of size $\frac{n}{c}$ for some c (often $c = 2$).

A General Result

Recurrences for algorithm complexity often involve a linear reduction in subproblem size.

Theorem

- *(case 1)* $T(n) = T(n-1) + bn^k$
  solution $T(n) = \mathcal{O}(n^{k+1})$
- *(case 2)* $T(n) = cT(n-1) + bn^k, \quad c > 1:$
  solution $T(n) = \mathcal{O}(c^n)$

This contrasts with *divide-and-conquer algorithms*, where we solve a problem of size n by recurrence to subproblems of size $\frac{n}{c}$ for some c (often $c = 2$).

A General Result

Recurrences for algorithm complexity often involve a linear reduction in subproblem size.

Theorem

- *(case 1)* $T(n) = T(n-1) + bn^k$
  solution $T(n) = \mathcal{O}(n^{k+1})$
- *(case 2)* $T(n) = cT(n-1) + bn^k, \quad c > 1:$
  solution $T(n) = \mathcal{O}(c^n)$

This contrasts with *divide-and-conquer algorithms*, where we solve a problem of size n by recurrence to subproblems of size $\frac{n}{c}$ for some c (often $c = 2$).

A Divide-and-Conquer Algorithm: Merge Sort

```
MergeSort($L$):
  Input  list $L$ of $n$ elements

  if $n \leq 1$ then return $L$                                            cost $= 0$
  let $L_1 = $ MergeSort($L[0 \,..\, \lceil \frac{n}{2} \rceil - 1]$)      cost $= T(\frac{n}{2})$
  let $L_2 = $ MergeSort($L[\lceil \frac{n}{2} \rceil \,..\, n-1]$)        cost $= T(\frac{n}{2})$
  merge $L_1$ and $L_2$ into a sorted list $L_3$                           cost $\leq n-1$
    by repeatedly extracting the least element from $L_1$ or $L_2$
        (both are sorted!) and placing in $L_3$
  return $L_3$
```

Let $T(n)$ be the number of comparison operations required by MergeSort(L) on a list L of length n

$$T(n) = 2T\left(\frac{n}{2}\right) + (n-1) \qquad T(1) = 0$$

A Divide-and-Conquer Algorithm: Merge Sort

```
MergeSort($L$):
  Input  list $L$ of $n$ elements

  if $n \leq 1$ then return $L$                                   cost $= 0$
  let $L_1 =$ MergeSort($L[0 \mathinner{..} \lceil \frac{n}{2} \rceil - 1]$)          cost $= T(\frac{n}{2})$
  let $L_2 =$ MergeSort($L[\lceil \frac{n}{2} \rceil \mathinner{..} n - 1]$)          cost $= T(\frac{n}{2})$
  merge $L_1$ and $L_2$ into a sorted list $L_3$                      cost $\leq n - 1$
    by repeatedly extracting the least element from $L_1$ or $L_2$
        (both are sorted!) and placing in $L_3$
  return $L_3$
```

Let $T(n)$ be the number of comparison operations required by MergeSort(L) on a list L of length n

$$T(n) = 2T\left(\frac{n}{2}\right) + (n-1) \qquad T(1) = 0$$

Solving the Recurrence

$$T(n) = 2T\left(\frac{n}{2}\right) + (n-1), \quad T(1) = 0$$

$$\begin{aligned}
T(1) &= 0 \\
T(2) &= 2T(1) + (2-1) & & = 0 + 1 \\
T(4) &= 2T(2) + (4-1) & = 2(0+1) + (4-1) & = 4 + 1 \\
T(8) &= 2T(4) + (8-1) & = 2(4+1) + (8-1) & = 16 + 1 \\
T(16) &= 2T(8) + (16-1) & = 2(16+1) + (16-1) & = 48 + 1 \\
T(32) &= 2T(16) + (32-1) & = 2(48+1) + (32-1) & = 128 + 1
\end{aligned}$$

Value of n	4	8	16	32
$T(n)$	5	17	49	129
Ratio	1	2	3	4

Conjecture: $T(n) = n(\log_2 n - 1) + 1$ for $n = 2^k$ (Proof?)

Hence, Merge Sort is in $\mathcal{O}(n \log n)$

Solving the Recurrence

$$T(n) = 2T\left(\frac{n}{2}\right) + (n-1), \quad T(1) = 0$$

$$\begin{aligned}
T(1) &= 0 \\
T(2) &= 2T(1) + (2-1) & & = 0 + 1 \\
T(4) &= 2T(2) + (4-1) & = 2(0+1) + (4-1) & = 4 + 1 \\
T(8) &= 2T(4) + (8-1) & = 2(4+1) + (8-1) & = 16 + 1 \\
T(16) &= 2T(8) + (16-1) & = 2(16+1) + (16-1) & = 48 + 1 \\
T(32) &= 2T(16) + (32-1) & = 2(48+1) + (32-1) & = 128 + 1
\end{aligned}$$

Value of n	4	8	16	32
$T(n)$	5	17	49	129
Ratio	1	2	3	4

Conjecture: $T(n) = n(\log_2 n - 1) + 1$ for $n = 2^k$ (Proof?)

Hence, Merge Sort is in $\mathcal{O}(n \log n)$

Solving the Recurrence

$$T(n) = 2T\left(\frac{n}{2}\right) + (n-1), \quad T(1) = 0$$

$$\begin{aligned}
T(1) &= 0 \\
T(2) &= 2T(1) + (2-1) & & = 0 + 1 \\
T(4) &= 2T(2) + (4-1) & = 2(0+1) + (4-1) & = 4 + 1 \\
T(8) &= 2T(4) + (8-1) & = 2(4+1) + (8-1) & = 16 + 1 \\
T(16) &= 2T(8) + (16-1) & = 2(16+1) + (16-1) & = 48 + 1 \\
T(32) &= 2T(16) + (32-1) & = 2(48+1) + (32-1) & = 128 + 1
\end{aligned}$$

Value of n	4	8	16	32
$T(n)$	5	17	49	129
Ratio	1	2	3	4

Conjecture: $T(n) = n(\log_2 n - 1) + 1$ for $n = 2^k$ (Proof?)

Hence, Merge Sort is in $\mathcal{O}(n \log n)$

Exercise

Give a tight big-Oh upper bound on the solution to the divide-and-conquer recurrence

$$T(n) = T\left(\frac{n}{2}\right) + g(n), \quad T(1) = a$$

for the case $g(n) = n^2$

$$T(n) = n^2+(\frac{n}{2})^2+(\frac{n}{4})^2+\ldots = n^2(1+\frac{1}{4}+\frac{1}{16}+\ldots) = \mathcal{O}(\frac{4}{3}n^2) = \mathcal{O}(n^2)$$

Exercise

Give a tight big-Oh upper bound on the solution to the divide-and-conquer recurrence

$$T(n) = T\left(\frac{n}{2}\right) + g(n), \quad T(1) = a$$

for the case $g(n) = n^2$

$$T(n) = n^2+(\frac{n}{2})^2+(\frac{n}{4})^2+\ldots = n^2(1+\frac{1}{4}+\frac{1}{16}+\ldots) = \mathcal{O}(\frac{4}{3}n^2) = \mathcal{O}(n^2)$$

Master Theorem

Theorem

The following cases cover many divide-and-conquer recurrences that arise in practice:

$$T(n) = d^\alpha \cdot T\left(\frac{n}{d}\right) + \mathcal{O}(n^\beta)$$

- *(case 1)* $\alpha > \beta$
  solution $T(n) = \mathcal{O}(n^\alpha)$
- *(case 2)* $\alpha = \beta$
  solution $T(n) = \mathcal{O}(n^\alpha \log n)$
- *(case 3)* $\alpha < \beta$
  solution $T(n) = \mathcal{O}(n^\beta)$

The situations arise when we reduce a problem of size n *to several subproblems of size* n/d*. If the number of such subproblems is* d^α*, while the cost of combining these smaller solutions is* n^β*, then the overall cost depends on the relative magnitude of* α *and* β*.*

Master Theorem: Examples

Example

$$T(n) = T\left(\frac{n}{2}\right) + n^2, \quad T(1) = a$$

Here $d = 2$, $\alpha = 0$, $\beta = 2$, so we have case 3 and the solution is

$$T(n) = \mathcal{O}(n^\beta) = n^2$$

Example

Mergesort has

$$T(n) = 2T\left(\frac{n}{2}\right) + (n-1)$$

recurrence for the number of comparisons.
Here $d = 2$, $\alpha = 1 = \beta$, so we have case 2, and the solution is

$$T(n) = \mathcal{O}(n^\alpha \log(n)) = \mathcal{O}(n \log(n))$$

Exercise

Solve $T(n) = 3^n T(\frac{n}{2})$ with $T(1) = 1$

Let $n \geq 2$ be a power of 2 then

$$T(n) = 3^n \cdot 3^{\frac{n}{2}} \cdot 3^{\frac{n}{4}} \cdot 3^{\frac{n}{8}} \cdot \ldots = 3^{n(1+\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\ldots)} = \mathcal{O}(3^{2n})$$

Exercise

Solve $T(n) = 3^n T(\frac{n}{2})$ with $T(1) = 1$

Let $n \geq 2$ be a power of 2 then

$$T(n) = 3^n \cdot 3^{\frac{n}{2}} \cdot 3^{\frac{n}{4}} \cdot 3^{\frac{n}{8}} \cdot \ldots = 3^{n(1+\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\ldots)} = \mathcal{O}(3^{2n})$$

Exercise

$\boxed{4.3.22}$ The following algorithm raises a number a to a power n.

```
$p = 1$
$i = n$
while $i > 0$ do
    $p = p * a$
    $i = i - 1$
end while
return $p$
```

Determine the complexity (no. of comparisons and arithmetic ops).

Solution

$\boxed{4.3.22}$ Number of comparisons and arithmetic operations:

$\text{cost}(n = 1) = 4$ (why?)

$\text{cost}(n > 1) = 3 + \text{cost}(n - 1)$

This can be described by the recurrence
$T(n) = 3 + T(n - 1)$ with $T(1) = 4$

Solution: $T(n) = \mathcal{O}(n)$

Exercise

$\boxed{4.3.21}$ The following algorithm gives a fast method for raising a number a to a power n.

```
$p = 1$
$q = a$
$i = n$
while $i > 0$ do
    if $i$ is odd then
        $p = p * q$
    $q = q * q$
    $i = \lfloor \frac{i}{2} \rfloor$
end while
return $p$
```

Determine the complexity (no. of comparisons and arithmetic ops).

Solution

$\boxed{4.3.21}$ Number of comparisons and arithmetic operations:

$\text{cost}(n = 1) = 6$ (why?)

$\text{cost}(n > 1) = 4 + \text{cost}(\lfloor \frac{n}{2} \rfloor)$ if n even
$\text{cost}(n > 1) = 5 + \text{cost}(\lfloor \frac{n}{2} \rfloor)$ if n odd

This can be described by the recurrence
$T(n) = 5 + T(\frac{n}{2})$ with $T(1) = 6$

Solution: $T(n) = \mathcal{O}(\log n)$

Application: Efficient Matrix Multiplication

The running time of a straightforward algorithm for the multiplication of two $n \times n$ matrices is $\mathcal{O}(n^3)$. (Why?)

Matrix mutliplication can also be carried out blockwise:

$$\begin{bmatrix} [\mathbf{A}] & [\mathbf{B}] \\ [\mathbf{C}] & [\mathbf{D}] \end{bmatrix} \cdot \begin{bmatrix} [\mathbf{E}] & [\mathbf{F}] \\ [\mathbf{G}] & [\mathbf{H}] \end{bmatrix} = \begin{bmatrix} [\mathbf{AE} + \mathbf{BG}] & [\mathbf{AF} + \mathbf{BH}] \\ [\mathbf{CE} + \mathbf{DG}] & [\mathbf{CF} + \mathbf{DH}] \end{bmatrix}$$

This can be implemented by a divide-and-conquer algorithm, recursively computing eight size-$\frac{n}{2}$ matrix products plus a few $\mathcal{O}(n^2)$-time matrix additions.
Determine a recurrence to describe the total running time!

$$T(n) = 8 \cdot T\left(\frac{n}{2}\right) + \mathcal{O}(n^2)$$

Solution (Master Theorem)?

Application: Efficient Matrix Multiplication

The running time of a straightforward algorithm for the multiplication of two $n \times n$ matrices is $\mathcal{O}(n^3)$. (Why?)

Matrix mutliplication can also be carried out blockwise:

$$\begin{bmatrix} [\mathbf{A}] & [\mathbf{B}] \\ [\mathbf{C}] & [\mathbf{D}] \end{bmatrix} \cdot \begin{bmatrix} [\mathbf{E}] & [\mathbf{F}] \\ [\mathbf{G}] & [\mathbf{H}] \end{bmatrix} = \begin{bmatrix} [\mathbf{AE} + \mathbf{BG}] & [\mathbf{AF} + \mathbf{BH}] \\ [\mathbf{CE} + \mathbf{DG}] & [\mathbf{CF} + \mathbf{DH}] \end{bmatrix}$$

This can be implemented by a divide-and-conquer algorithm, recursively computing eight size-$\frac{n}{2}$ matrix products plus a few $\mathcal{O}(n^2)$-time matrix additions.
Determine a recurrence to describe the total running time!

$$T(n) = 8 \cdot T\left(\frac{n}{2}\right) + \mathcal{O}(n^2)$$

Solution (Master Theorem)? $\mathcal{O}(n^3)$

Application: Efficient Matrix Multiplication

The running time of a straightforward algorithm for the multiplication of two $n \times n$ matrices is $\mathcal{O}(n^3)$. (Why?)

Matrix mutliplication can also be carried out blockwise:

$$\begin{bmatrix} [\mathbf{A}] & [\mathbf{B}] \\ [\mathbf{C}] & [\mathbf{D}] \end{bmatrix} \cdot \begin{bmatrix} [\mathbf{E}] & [\mathbf{F}] \\ [\mathbf{G}] & [\mathbf{H}] \end{bmatrix} = \begin{bmatrix} [\mathbf{AE}+\mathbf{BG}] & [\mathbf{AF}+\mathbf{BH}] \\ [\mathbf{CE}+\mathbf{DG}] & [\mathbf{CF}+\mathbf{DH}] \end{bmatrix}$$

This can be implemented by a divide-and-conquer algorithm, recursively computing eight size-$\frac{n}{2}$ matrix products plus a few $\mathcal{O}(n^2)$-time matrix additions.
Determine a recurrence to describe the total running time!

$$T(n) = 8 \cdot T\left(\frac{n}{2}\right) + \mathcal{O}(n^2)$$

Solution (Master Theorem)? $\mathcal{O}(n^3)$

Application: Efficient Matrix Multiplication

Strassen's algorithm improves the efficiency by some clever algebra:

$$\mathbf{X} = \begin{bmatrix} [\mathbf{A}] & [\mathbf{B}] \\ [\mathbf{C}] & [\mathbf{D}] \end{bmatrix} \qquad \mathbf{Y} = \begin{bmatrix} [\mathbf{E}] & [\mathbf{F}] \\ [\mathbf{G}] & [\mathbf{H}] \end{bmatrix}$$

$$\mathbf{X} \cdot \mathbf{Y} = \begin{bmatrix} [\mathbf{P}_5 + \mathbf{P}_4 - \mathbf{P}_2 + \mathbf{P}_6] & [\mathbf{P}_1 + \mathbf{P}_2] \\ [\mathbf{P}_3 + \mathbf{P}_4] & [\mathbf{P}_1 + \mathbf{P}_5 - \mathbf{P}_3 - \mathbf{P}_7] \end{bmatrix}$$

where

$$\begin{array}{lll} \mathbf{P}_1 = \mathbf{A}(\mathbf{F}-\mathbf{H}) & \mathbf{P}_3 = (\mathbf{C}+\mathbf{D})\mathbf{E} & \mathbf{P}_5 = (\mathbf{A}+\mathbf{D})(\mathbf{E}+\mathbf{H}) \\ \mathbf{P}_2 = (\mathbf{A}+\mathbf{B})\mathbf{H} & \mathbf{P}_4 = \mathbf{D}(\mathbf{G}-\mathbf{E}) & \mathbf{P}_6 = (\mathbf{B}-\mathbf{D})(\mathbf{G}+\mathbf{H}) \\ & & \mathbf{P}_7 = (\mathbf{A}-\mathbf{C})(\mathbf{E}+\mathbf{F}) \end{array}$$

Its total running time is described by the recurrence

$$T(n) = 7 \cdot T\left(\frac{n}{2}\right) + \mathcal{O}(n^2) \qquad (= \mathcal{O}(n^{\log_2 7}) \simeq \mathcal{O}(n^{2.807}))$$

Summary

- "Big-Oh" notation $\mathcal{O}(f(n))$ for the class of functions for which $f(n)$ is an upper bound; $\Theta(f(n))$
- Analysing the complexity of algorithms using recurrences
- Solving recurrences
- General results for recurrences with linear reductions (slide 23) and exponential reductions ("Master Theorem")

COMP9020 Lectures 9-11
Session 1, 2017
Counting, Probability and Expectation

- Textbook (R & W) - Ch. 5, Sec. 5.1–5.3; Ch. 9
- Problem sets 9–11
- Supplementary Exercises Ch. 5, 9 (R & W)

Announcements

Final Exam ...

- **Wednesday, 21 June, 8:45am**
- **Scientia Building (G19), Leighton Hall**

Of course, assessment isn't a "one-way street" ...

- I get to assess you in the final exam
- you get to assess me in UNSW's **MyExperience** Evaluation

Please fill it out ...

- give me some feedback on how you might like the course to run in the future
- even if that is "Exactly the same. It was perfect this time."

Overview

1. Counting techniques
2. Basic and conditional probability
3. Expectation
4. Probability distributions

NB

Combinatorics and probability arise in many areas of Computer Science, e.g.

- *Complexity of algorithms, data management*
- *Reliability, quality assurance*
- *Computer security*
- *Data mining, machine learning, robotics*

Counting Techniques

General idea: find methods, algorithms or precise formulae to count the number of elements in various sets or collections derived, in a structured way, from some basic sets.

Examples

Single base set $S = \{s_1, \ldots, s_n\}$, $|S| = n$; find the number of

- all subsets of S
- ordered selections of r different elements of S
- unordered selections of r different elements of S
- selections of r elements from S s.t. $\ldots$
- functions $S \longrightarrow S$ (onto, 1-1)
- partitions of S into k equivalence classes
- graphs/trees with elements of S as labelled vertices/leaves

Basic Counting Rules (1)

Union rule — S and T *disjoint*

$$|S \cup T| = |S| + |T|$$

$S_1, S_2, \ldots, S_n$ pairwise disjoint ($S_i \cap S_j = \emptyset$ for $i \neq j$)

$$|S_1 \cup \ldots \cup S_n| = \sum |S_i|$$

Example

How many numbers in $A = [1, 2, \ldots, 999]$ are divisible by 31 or 41?

$\lfloor 999/31 \rfloor = 32$ divisible by 31
$\lfloor 999/41 \rfloor = 24$ divisible by 41
No number in A divisible by both
Hence, $32 + 24 = 56$ divisible by 31 or 41

Basic Counting Rules (1)

Union rule — S and T *disjoint*

$$|S \cup T| = |S| + |T|$$

$S_1, S_2, \ldots, S_n$ pairwise disjoint ($S_i \cap S_j = \emptyset$ for $i \neq j$)

$$|S_1 \cup \ldots \cup S_n| = \sum |S_i|$$

Example

How many numbers in $A = [1, 2, \ldots, 999]$ are divisible by 31 or 41?

$\lfloor 999/31 \rfloor = 32$ divisible by 31
$\lfloor 999/41 \rfloor = 24$ divisible by 41
No number in A divisible by both
Hence, $32 + 24 = 56$ divisible by 31 or 41

Basic Counting Rules (2)

Product rule

$$|S_1 \times \ldots \times S_k| = |S_1| \cdot |S_2| \cdots |S_k| = \prod_{i=1}^{k} |S_i|$$

If all $S_i = S$ (the same set) and $|S| = m$ then $|S^k| = m^k$

Example

Let $\Sigma = \{a, b, c, d, e, f\}$.
How many 5-letter words? How many with no letter repeated?

$$|\Sigma^5| = |\Sigma|^5 = 7^5 = 16,807$$

$$\prod_{i=0}^{4} (|\Sigma| - i) = 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 = 2,520$$

Exercises

S, T finite. How many functions $S \longrightarrow T$ are there?

$$|T|^{|S|}$$

$\boxed{5.1.19}$ Consider a *complete* graph on n vertices.

(a) No. of paths of length 3
Take any vertex to start, then every next vertex different from the preceding one. Hence $n \cdot (n-1)^3$

(b) paths of length 3 with all vertices distinct
$n(n-1)(n-2)(n-3)$

(c) paths of length 3 with all edges distinct
$n(n-1)(n-2)^2$

Exercises

S, T finite. How many functions $S \longrightarrow T$ are there?

$$|T|^{|S|}$$

$\boxed{5.1.19}$ Consider a *complete* graph on n vertices.

(a) No. of paths of length 3
Take any vertex to start, then every next vertex different from the preceding one. Hence $n \cdot (n-1)^3$

(b) paths of length 3 with all vertices distinct
$n(n-1)(n-2)(n-3)$

(c) paths of length 3 with all edges distinct
$n(n-1)(n-2)^2$

Basic Inferences

For arbitrary sets $S, T, \ldots$

$$\begin{aligned} |S \cup T| &= |S| + |T| - |S \cap T| \\ |T \setminus S| &= |T| - |S \cap T| \\ |S_1 \cup S_2 \cup S_3| &= |S_1| + |S_2| + |S_3| \\ &\quad - |S_1 \cap S_2| - |S_1 \cap S_3| - |S_2 \cap S_3| \\ &\quad + |S_1 \cap S_2 \cap S_3| \end{aligned}$$

Exercise

$\boxed{5.3.1}$ 200 people. 150 swim or jog, 85 swim and 60 do both.
How many jog?

S – (set of) people who swim, J – people who jog
$|S \cup J| = |S| + |J| - |S \cap J|$; thus $150 = 85 + |J| - 60$ hence
$|J| = 125$; answer *does not* depend on the number of people overall

$\boxed{5.6.38}$ (Supp) There are 100 problems, 75 of which are ‘easy’ and 40 ‘important’.
What’s the smallest number of easy *and* important problems?

$|E \cap I| = |E| + |I| - |E \cup I| = 75 + 40 - |E \cup I| \geq 75 + 40 - 100 = 15$

Exercise

$\boxed{5.3.1}$ 200 people. 150 swim or jog, 85 swim and 60 do both.
How many jog?

S – (set of) people who swim, J – people who jog
$|S \cup J| = |S| + |J| - |S \cap J|$; thus $150 = 85 + |J| - 60$ hence
$|J| = 125$; answer *does not* depend on the number of people overall

$\boxed{5.6.38}$ (Supp) There are 100 problems, 75 of which are ‘easy’ and 40 ‘important’.
What’s the smallest number of easy *and* important problems?

$|E \cap I| = |E| + |I| - |E \cup I| = 75 + 40 - |E \cup I| \geq 75 + 40 - 100 = 15$

Exercise

$\boxed{5.3.2}$ $S = [100 \dots 999]$, thus $|S| = 900$.

(a) How many numbers have at least one digit that is a 3 or 7?
$A_3 = \{\text{at least one '3'}\}$
$A_7 = \{\text{at least one '7'}\}$

$(A_3 \cup A_7)^c = \{\ n \in [100, 999] : n \text{ digits } \in \{0, 1, 2, 4, 5, 6, 8, 9\}\ \}$

7 choices for the first digit and 8 choices for the later digits

$$|(A_3 \cup A_7)^c| = |\{1, 2, 4, 5, 6, 8, 9\}| \cdot |\{0, 1, 2, 4, 5, 6, 8, 9\}|^2$$

Therefore $|A_3 \cup A_7| = 900 - 448 = 452$

(b) How many numbers have a 3 *and* a 7?
$|A_3 \cap A_7| = |A_3| + |A_7| - |A_3 \cup A_7| =$
$(900 - 8 \cdot 9 \cdot 9) + (900 - 8 \cdot 9 \cdot 9) - 452 = 2 \cdot 252 - 452 = 52$

Exercise

$\boxed{5.3.2}$ $S = [100 \dots 999]$, thus $|S| = 900$.

(a) How many numbers have at least one digit that is a 3 or 7?
$A_3 = \{\text{at least one '3'}\}$
$A_7 = \{\text{at least one '7'}\}$

$(A_3 \cup A_7)^c = \{\ n \in [100, 999] : n \text{ digits } \in \{0, 1, 2, 4, 5, 6, 8, 9\}\ \}$

7 choices for the first digit and 8 choices for the later digits

$$|(A_3 \cup A_7)^c| = |\{1, 2, 4, 5, 6, 8, 9\}| \cdot |\{0, 1, 2, 4, 5, 6, 8, 9\}|^2$$

Therefore $|A_3 \cup A_7| = 900 - 448 = 452$

(b) How many numbers have a 3 *and* a 7?
$|A_3 \cap A_7| = |A_3| + |A_7| - |A_3 \cup A_7| =$
$(900 - 8 \cdot 9 \cdot 9) + (900 - 8 \cdot 9 \cdot 9) - 452 = 2 \cdot 252 - 452 = 52$

Corollaries

- If $|S \cup T| = |S| + |T|$ then S and T are disjoint
- If $|\bigcup_{i=1}^{n} S_i| = \sum_{i=1}^{n} |S_i|$ then S_i are pairwise disjoint
- If $|T \setminus S| = |T| - |S|$ then $S \subseteq T$

These properties can serve to identify cases when sets are disjoint (resp. one is contained in the other).

Proof.

$|S| + |T| = |S \cup T|$ means $|S \cap T| = |S| + |T| - |S \cup T| = 0$

$|T \setminus S| = |T| - |S|$ means $|S \cap T| = |S|$ means $S \subseteq T$ □

Combinatorial Objects: How Many?

permutations
Ordering of all objects from a set S; equivalently: Selecting all objects while *recognising* the order of selection.
The number of permutations of n elements is

$$n! = n \cdot (n-1) \cdots 1, \quad 0! = 1! = 1$$

***r*-permutations**
Selecting any r objects from a set S of size n without repetition while *recognising* the order of selection.
Their number is

$$\Pi(n, r) = n \cdot (n-1) \cdots (n-r+1) = \frac{n!}{(n-r)!}$$

r-**selections** (or: *r*-**combinations**)
Collecting any r distinct objects without repetition;
equivalently: selecting r objects from a set S of size n and *not* recognising the order of selection.
Their number is

$$\binom{n}{r} = \frac{n!}{(n-r)!r!} = \frac{n\cdot(n-1)\cdots(n-r+1)}{1\cdot 2\cdots r}$$

NB

These numbers are usually called binomial coefficients due to

$$(a+b)^n = a^n + \binom{n}{1}a^{n-1}b + \binom{n}{2}a^{n-2}b^2 + \ldots + b^n = \sum_{i=0}^{n}\binom{n}{i}a^{n-i}b^i$$

Also defined for any $\alpha \in \mathbb{R}$ *as* $\binom{\alpha}{r} = \dfrac{\alpha(\alpha-1)\cdots(\alpha-r+1)}{r!}$

Simple Counting Problems

Example

$\boxed{5.1.2}$ Give an example of a counting problem whose answer is

(a) $\Pi(26, 10)$

(b) $\binom{26}{10}$

Draw 10 cards from a half deck (eg. black cards only)
(a) the cards are recorded in the order of appearance
(b) only the complete draw is recorded

Examples

- Number of edges in a complete graph K_n
- Number of diagonals in a convex polygon
- Number of poker hands
- Decisions in games, lotteries etc.

Simple Counting Problems

Example

$\boxed{5.1.2}$ Give an example of a counting problem whose answer is

(a) $\Pi(26, 10)$

(b) $\binom{26}{10}$

Draw 10 cards from a half deck (eg. black cards only)
(a) the cards are recorded in the order of appearance
(b) only the complete draw is recorded

Examples

- Number of edges in a complete graph K_n
- Number of diagonals in a convex polygon
- Number of poker hands
- Decisions in games, lotteries etc.

Exercise

$\boxed{5.1.6}$ From a group of 12 men and 16 women, how many committees can be chosen consisting of

(a) 7 members? $\binom{12+16}{7}$

(b) 3 men and 4 women? $\binom{12}{3}\binom{16}{4}$

(c) 7 women or 7 men? $\binom{12}{7} + \binom{16}{7}$

$\boxed{5.1.7}$ As above, but any 4 people (male or female) out of 9 and two, Alice and Bob, unwilling to serve on the same committee.

{all committees} − {committees with both A and B}
$= \binom{9}{4} - \binom{7}{2} = 126 - 21 = 105$

equivalently, {A in, B out} + {A out, B in} + {none in}
$= \binom{7}{3} + \binom{7}{3} + \binom{7}{4} = 35 + 35 + 35 = 105$

Exercise

$\boxed{5.1.6}$ From a group of 12 men and 16 women, how many committees can be chosen consisting of

(a) 7 members? $\binom{12+16}{7}$

(b) 3 men and 4 women? $\binom{12}{3}\binom{16}{4}$

(c) 7 women or 7 men? $\binom{12}{7}+\binom{16}{7}$

$\boxed{5.1.7}$ As above, but any 4 people (male or female) out of 9 and two, Alice and Bob, unwilling to serve on the same committee.

{all committees} − {committees with both A and B}
$= \binom{9}{4} - \binom{7}{2} = 126 - 21 = 105$

equivalently, {A in, B out} + {A out, B in} + {none in}
$= \binom{7}{3} + \binom{7}{3} + \binom{7}{4} = 35 + 35 + 35 = 105$

Counting Poker Hands

$\boxed{5.1.15}$ A poker hand consists of 5 cards drawn without replacement from a standard deck of 52 cards

$$\{A, 2\text{-}10, J, Q, K\} \times \{\text{club, spade, heart, diamond}\}$$

(a) Number of "4 of a kind" hands (e.g. 4 Jacks)
|rank of the 4-of-a-kind| · |any other card| $= 13 \cdot (52-4)$

(b) Number of non-straight flushes, i.e. all cards of same suit but *not* consecutive (e.g. 8,9,10,J,K)
|all flush| − |straight flush|
= |suit| · |5-hand in a given suit| −
|suit| · |rank of a straight flush in a given suit|
$= 4 \cdot \binom{13}{5} - 4 \cdot 10$

Counting Poker Hands

$\boxed{5.1.15}$ A poker hand consists of 5 cards drawn without replacement from a standard deck of 52 cards

$$\{A, 2\text{-}10, J, Q, K\} \times \{\text{club, spade, heart, diamond}\}$$

(a) Number of "4 of a kind" hands (e.g. 4 Jacks)
|rank of the 4-of-a-kind| · |any other card| $= 13 \cdot (52-4)$

(b) Number of non-straight flushes, i.e. all cards of same suit but *not* consecutive (e.g. 8,9,10,J,K)
|all flush| − |straight flush|
= |suit| · |5-hand in a given suit| −
|suit| · |rank of a straight flush in a given suit|
$= 4 \cdot \binom{13}{5} - 4 \cdot 10$

Difficult Counting Problems

Example (Ramsay numbers)

An example of a *Ramsay number* is $R(3,3) = 6$, meaning that

> *"K_6 is the smallest complete graph s.t. if all edges are painted using two colours, then there must be at least one monochromatic triangle"*

This serves as the basis of a game called S-I-M (invented by Simmons), where two adversaries connect six dots, respectively using blue and red lines. The objective is to *avoid* closing a triangle of one's own colour. The second player has a winning strategy, but the full analysis requires a computer program.

Using Programs to Count

Two dice, a red die and a black die, are rolled.
(Note: one *die*, two or more *dice*)

Write a program to list all the pairs $\{(R, B) : R > B\}$

Similarly, for three dice, list all triples $R > B > G$

Generally, for n dice, all of which are m-sided ($n \leq m$), list all *decreasing* n-tuples

NB

In order to just find the number of such n-tuples, it is not necessary to list them all. One can write a recurrence relation for these numbers and compute (or try to solve) it.

Approximate Counting

NB

A Count may be a precise value or an **estimate**.

The latter should be asymptotically correct or at least give a good asymptotic bound, whether upper or lower. If S is the base set, $|S| = n$ its size, and we denote by $c(S)$ some collection of objects from S we are interested in, then we seek constants a, b s.t.

$$a \leq \lim_{n \to \infty} \frac{est(|c(S)|)}{|c(S)|} \leq b$$

Probability

Elementary Probability

Sample space:

$$\Omega = \{\omega_1, \ldots, \omega_n\}$$

Each point represents an outcome, each outcome ω_i equally likely:

$$P(\omega_1) = P(\omega_2) = \ldots = P(\omega_n) = \frac{1}{n}$$

This a called a **uniform probability distribution** over Ω

Examples

Tossing a coin: $\Omega = \{H, T\}$

$$P(H) = P(T) = 0.5$$

Rolling a die: $\Omega = \{1, 2, 3, 4, 5, 6\}$

$$P(1) = P(2) = P(3) = P(4) = P(5) = P(6) = \frac{1}{6}$$

Non-uniform Probability

Slight modification is needed to define an arbitrary (in general non-uniform) probability distribution:

$$\Omega = \{\omega_1, \dots, \omega_n\}$$

Let

$$P(\omega_1) = p_1, P(\omega_2) = p_2, \dots, P(\omega_n) = p_n$$

Then

$$\sum_{i=1}^{n} p_i = 1$$

Events

Definition

Event — a collection of outcomes = subset of Ω

Probability of an event:

$$P(E) = \sum_{\omega \in E} P(\omega)$$

Fact

$$P(\emptyset) = 0, \quad P(\Omega) = 1, \quad P(E^c) = 1 - P(E)$$

Exercises

$\boxed{5.2.7}$ Suppose an experiment leads to events A, B with probabilities $P(A) = 0.5, P(B) = 0.8, P(A \cap B) = 0.4$.
Find

- $P(B^c) = 1 - P(B) = 0.2$
- $P(A \cup B) = P(A) + P(B) - P(A \cap B) = 0.9$
- $P(A^c \cup B^c) = 1 - P((A^c \cup B^c)^c) = 1 - P(A \cap B) = 0.6$

$\boxed{5.2.8}$ Given $P(A) = 0.6,\ P(B) = 0.7$, show $P(A \cap B) \geq 0.3$

$$\begin{aligned} P(A \cap B) &= P(A) + P(B) - P(A \cup B) \\ &= 0.6 + 0.7 - P(A \cup B) \\ &\geq 0.6 + 0.7 - 1 = 0.3 \end{aligned}$$

Exercises

$\boxed{5.2.7}$ Suppose an experiment leads to events A, B with probabilities $P(A) = 0.5, P(B) = 0.8, P(A \cap B) = 0.4$.
Find

- $P(B^c) = 1 - P(B) = 0.2$
- $P(A \cup B) = P(A) + P(B) - P(A \cap B) = 0.9$
- $P(A^c \cup B^c) = 1 - P((A^c \cup B^c)^c) = 1 - P(A \cap B) = 0.6$

$\boxed{5.2.8}$ Given $P(A) = 0.6,\ P(B) = 0.7$, show $P(A \cap B) \geq 0.3$

$$\begin{aligned} P(A \cap B) &= P(A) + P(B) - P(A \cup B) \\ &= 0.6 + 0.7 - P(A \cup B) \\ &\geq 0.6 + 0.7 - 1 = 0.3 \end{aligned}$$

Computing Probabilities by Counting

Computing probabilities with respect to a *uniform* distribution comes down to counting the size of the event.
If $E = \{e_1, \ldots, e_k\}$ then

$$P(E) = \sum_{i=1}^{k} P(e_i) = \sum_{i=1}^{k} \frac{1}{|\Omega|} = \frac{|E|}{|\Omega|}$$

Most of the counting rules carry over to probabilities wrt. a uniform distribution.

NB

The expression "selected at random", when not further qualified, means:
"subject to / according to / . . . a uniform *distribution."*

Examples

$\boxed{5.6.38}$ (Supp) Of 100 problems, 75 are 'easy' and 40 'important'.
(b) n problems chosen randomly. What is the probability that all n are important?

$$p = \frac{\binom{40}{n}}{\binom{100}{n}} = \frac{40 \cdot 39 \cdots (41-n)}{100 \cdot 99 \cdots (101-n)}$$

$\boxed{5.2.3}$ A 4-letter word is selected at random from Σ^4, where $\Sigma = \{a, b, c, d, e\}$. What is the probability that
(a) the letters in the word are all distinct?
(b) there are no vowels ("a", "e") in the word?
(c) the word begins with a vowel?

(a) $|E| = \Pi(5,4)$, $P(E) = \frac{5 \cdot 4 \cdot 3 \cdot 2}{5^4} = \frac{120}{625} \approx 19\%$
(b) $|E| = 3^4$, $P(E) = \frac{81}{625} \approx 13\%$
(c) $|E| = 2 \cdot 5^3$, $P(E) = \frac{2}{5}$

Examples

$\boxed{5.6.38}$ (Supp) Of 100 problems, 75 are 'easy' and 40 'important'.
(b) n problems chosen randomly. What is the probability that all n are important?

$$p = \frac{\binom{40}{n}}{\binom{100}{n}} = \frac{40 \cdot 39 \cdots (41-n)}{100 \cdot 99 \cdots (101-n)}$$

$\boxed{5.2.3}$ A 4-letter word is selected at random from Σ^4, where $\Sigma = \{a, b, c, d, e\}$. What is the probability that
(a) the letters in the word are all distinct?
(b) there are no vowels ("a", "e") in the word?
(c) the word begins with a vowel?

(a) $|E| = \Pi(5,4)$, $P(E) = \frac{5 \cdot 4 \cdot 3 \cdot 2}{5^4} = \frac{120}{625} \approx 19\%$
(b) $|E| = 3^4$, $P(E) = \frac{81}{625} \approx 13\%$
(c) $|E| = 2 \cdot 5^3$, $P(E) = \frac{2}{5}$

Exercise

$\boxed{5.2.11}$ Two dice, a red die and a black die, are rolled.
What is the probability that
(a) the sum of the values is even?
$P(R + B \in \{2, 4, \ldots, 12\}) = \frac{18}{36} = \frac{1}{2}$

(b) the number on the red die is bigger than on the black die?
$P(R > B) = P(R < B)$; also $P(R = B) = \frac{1}{6}$
Therefore $P(R < B) = \frac{1}{2}(1 - P(R = B)) = \frac{5}{12}$

(c) the number on the black die is twice the one on the red die?
$P(R = 2 \cdot B) = P(\{(2,1), (4,2), (6,3)\}) = \frac{3}{36} = \frac{1}{12}$

$\boxed{5.2.12}$ (a) the maximum of the numbers is 4? $P(E_1) = \frac{7}{36}$
(b) their minimum is 4? $P(E_2) = \frac{5}{36}$

Check:

$P(E_1 \cup E_2) = \frac{7}{36} + \frac{5}{36} - P(E_1 \cap E_2) = \frac{7+5-1}{36} = \frac{11}{36}$

$P(\text{at least one '4'}) = 1 - P(\text{no '4'}) = 1 - \frac{5}{6} \cdot \frac{5}{6} = \frac{11}{36}$

Exercise

$\boxed{5.2.11}$ Two dice, a red die and a black die, are rolled.
What is the probability that
(a) the sum of the values is even?
$P\big(R+B \in \{2,4,\dots,12\}\big) = \frac{18}{36} = \frac{1}{2}$

(b) the number on the red die is bigger than on the black die?
$P(R>B) = P(R<B)$; also $P(R=B) = \frac{1}{6}$
Therefore $P(R<B) = \frac{1}{2}(1-P(R=B)) = \frac{5}{12}$

(c) the number on the black die is twice the one on the red die?
$P(R = 2\cdot B) = P(\{(2,1),(4,2),(6,3)\}) = \frac{3}{36} = \frac{1}{12}$

$\boxed{5.2.12}$ (a) the maximum of the numbers is 4? $P(E_1) = \frac{7}{36}$
(b) their minimum is 4? $P(E_2) = \frac{5}{36}$

Check:

$P(E_1 \cup E_2) = \frac{7}{36} + \frac{5}{36} - P(E_1 \cap E_2) = \frac{7+5-1}{36} = \frac{11}{36}$
$P(\text{at least one '4'}) = 1 - P(\text{no '4'}) = 1 - \frac{5}{6}\cdot\frac{5}{6} = \frac{11}{36}$

Exercise

$\boxed{5.2.5}$ An urn contains 3 red and 4 black balls. 3 balls are removed without replacement. What are the probabilities that
(a) all 3 are red
(b) all 3 are black
(c) one is red, two are black

All probabilities are computed using the same sample space: all possible ways to draw three balls without replacement.
The size of the sample space is $\dfrac{7\cdot6\cdot5}{3!} = 35$
(a) $E =$ All balls are red: 1 combination
(b) $E =$ All balls are black: $\binom{4}{3} = 4$ combinations
(c) $E =$ One red and two black: $\binom{3}{1}\cdot\binom{4}{2} = 18$ combinations

Exercise

$\boxed{5.2.5}$ An urn contains 3 red and 4 black balls. 3 balls are removed without replacement. What are the probabilities that
(a) all 3 are red
(b) all 3 are black
(c) one is red, two are black

All probabilities are computed using the same sample space: all possible ways to draw three balls without replacement.
The size of the sample space is $\dfrac{7\cdot6\cdot5}{3!} = 35$
(a) $E =$ All balls are red: 1 combination
(b) $E =$ All balls are black: $\binom{4}{3} = 4$ combinations
(c) $E =$ One red and two black: $\binom{3}{1}\cdot\binom{4}{2} = 18$ combinations

Asymptotic Estimate of Relative Probabilities

Example

Event $A \overset{\text{def}}{=}$ one die rolled n times and you obtain two 6's
Event $B \overset{\text{def}}{=}$ n dice rolled simultaneously and you obtain one 6

$$P(A) = \frac{\binom{n}{2}}{6^n} \qquad P(B) = \frac{\binom{n}{1}}{6^n}$$

Therefore $\frac{P(A)}{P(B)} = \frac{\binom{n}{2}}{\binom{n}{1}} = \frac{n(n-1)}{2}\cdot\frac{1}{n} = \frac{n-1}{2} \in \Theta(n)$

n	1	2	3	4	...
$P(A)$	0	$\frac{1}{36}$	$\frac{1}{108}$	$\frac{1}{216}$	...
$P(B)$	$\frac{1}{6}$	$\frac{1}{18}$	$\frac{1}{108}$	$\frac{1}{324}$	...

Inclusion-Exclusion

This is one of the most universal counting procedures. It allows you to compute the size of

$$A_1 \cup \ldots \cup A_n$$

from the sizes of all possible intersections

$$A_{i_1} \cap A_{i_2} \cap \ldots \cap A_{i_k},\ a_{i_1} < a_{i_2} < \ldots < a_{i_k}$$

Two sets $|A \cup B| = |A| + |B| - |A \cap B|$

Three sets $|A \cup B \cup C| = |A| + |B| + |C|$
$-|A \cap B| - |A \cap C| - |B \cap C|$
$+|A \cap B \cap C|$

NB

Inclusion-exclusion is often applied informally without making clear or explicit why certain quantities are subtracted or put back in.

Interpretation

Each A_i defined as the set of objects that satisfy some property P_i

$$A_i = \{\ x \in X : P_i(x)\ \}$$

Union $A_1 \cup \ldots \cup A_n$ is the set of objects that satisfy **at least one** property P_i

$$A_1 \cup \ldots \cup A_n = \{\ x \in X : P_1(x) \vee P_2(x) \vee \ldots \vee P_n(x)\ \}$$

Intersection $A_{i_1} \cap \ldots \cap A_{i_r}$ is the set of objects that satisfy **all** properties $P_{i_1}, \ldots, P_{i_r}$

$$A_{i_1} \cap \ldots \cap A_{i_r} = \{\ x \in X : P_{i_1}(x) \wedge P_{i_2}(x) \wedge \ldots \wedge P_{i_r}(x)\ \}$$

Special case $r = 1$: $A_{i_1} = \{x \in X : P_{i_1}(x)\}$

Inclusion-Exclusion is a very common method for deriving probabilities from other probabilities.

Two sets

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

Three sets

$$\begin{aligned} P(A \cup B \cup C) &= P(A \cup B) + P(C) - P((A \cup B) \cap C) \\ &= P(A) + P(B) - P(A \cap B) + P(C) \\ &\quad - P((A \cap C) \cup (B \cap C)) \\ &= P(A) + P(B) - P(A \cap B) + P(C) \\ &\quad - (P(A \cap C) + P(B \cap C) - P(A \cap C \cap B \cap C)) \\ &= P(A) + P(B) + P(C) \\ &\quad - P(A \cap C) - P(A \cap C) - P(B \cap C) \\ &\quad + P(A \cap B \cap C) \end{aligned}$$

Example

A four-digit number n is selected at random (i.e. randomly from $[1000 \ldots 9999]$). Find the probability p that n has each of 0, 1, 2 among its digits.

Let $q = 1 - p$ be the complementary probability and define

$A_i = \{n : \text{no digit } i\}, A_{ij} = \{n : \text{no digits } i, j\}, A_{ijk} = \{n : \text{no } i, j, k\}$

Then define
$T = A_0 \cup A_1 \cup A_2 = \{n :\ \text{missing at least one of } 0, 1, 2\}$
$S = (A_0 \cup A_1 \cup A_2)^c = \{n :\ \text{containing each of } 0, 1, 2\}$

Example (cont'd)

Once we find the cardinality of T, the solution is

$$q = \frac{|T|}{9000}, \; p = 1 - q$$

To find $|A_i|, |A_{ij}|, |A_{ijk}|$ we reflect on how many choices are available for the first digit, for the second etc. A special case is the leading digit, which must be $1, \ldots, 9$

Example (cont'd)

$$\begin{aligned} |A_0| &= 9^4, \quad |A_1| = |A_2| = 8 \cdot 9^3 \\ |A_{01}| &= |A_{02}| = 8^4, \quad |A_{12}| = 7 \cdot 8^3 \\ |A_{012}| &= 7^4 \end{aligned}$$

$$\begin{aligned} |T| &= |A_0 \cup A_1 \cup A_2| \\ &= |A_0| + |A_1| + |A_2| - |A_0 \cap A_1| - |A_0 \cap A_2| - |A_1 \cap A_2| \\ &\quad + |A_0 \cap A_1 \cap A_2| \\ &= 9^4 + 2 \cdot 8 \cdot 9^3 - 2 \cdot 8^4 - 7 \cdot 8^3 + 7^4 \\ &= 25 \cdot 9^3 - 23 \cdot 8^3 + 7^4 = 8850 \end{aligned}$$

$$q = \frac{8850}{9000}, \quad p = 1 - q \quad \approx 0.01667$$

Previous example generalised: Probability of an r-digit number having all of 0,1,2,3 among its digits.
We use the previous notation: A_i — set of numbers n *missing* digit i, and similarly for all $A_{ij\ldots}$
We aim to find the size of $T = A_0 \cup A_1 \cup A_2 \cup A_3$, and then to compute $|S| = 9 \cdot 10^{r-1} - |T|$.

$$\begin{aligned} |A_0 \cup A_1 \cup A_2 \cup A_3| &= \text{sum of } |A_i| \\ &- \text{sum of } |A_i \cap A_j| \\ &+ \text{sum of } |A_i \cap A_j \cap A_k| \\ &- \text{sum of } |A_i \cap A_j \cap A_k \cap A_l| \end{aligned}$$

Probability of Sequential Outcomes

Example

Team A has probability $p = 0.5$ of winning a game against B.
What is the probability P_p of A winning a best-of-seven match if
(a) A already won the first game?
(b) A already won the first two games?
(c) A already won two out of the first three games?

(a) Sample space S — 6-sequences, formed from wins (W) and losses (L)

$$|S| = 2^6 = 64$$

Favourable sequences F — those with three to six W

$$|F| = \binom{6}{3} + \binom{6}{4} + \binom{6}{5} + \binom{6}{6} = 20 + 15 + 6 + 1 = 42$$

Therefore $P_{0.5} = \frac{42}{64} \approx 66\%$

Probability of Sequential Outcomes

Example

Team A has probability $p = 0.5$ of winning a game against B.
What is the probability P_p of A winning a best-of-seven match if
(a) A already won the first game?
(b) A already won the first two games?
(c) A already won two out of the first three games?

(a) Sample space S — 6-sequences, formed from wins (W) and losses (L)

$$|S| = 2^6 = 64$$

Favourable sequences F — those with three to six W

$$|F| = \binom{6}{3} + \binom{6}{4} + \binom{6}{5} + \binom{6}{6} = 20 + 15 + 6 + 1 = 42$$

Therefore $P_{0.5} = \frac{42}{64} \approx 66\%$

Example (cont'd)

(b) Sample space S — 5-sequences of W and L

$$|S| = 2^5 = 32$$

Favourable sequences F — those with two to five W

$$|F| = \binom{5}{2} + \binom{5}{3} + \binom{5}{4} + \binom{5}{5} = 10 + 10 + 5 + 1 = 26$$

Therefore $P_{0.5} = \frac{26}{32} \approx 81\%$

(c)

$$|S| = 2^4 = 16$$

$$|F| = \binom{4}{2} + \binom{4}{3} + \binom{4}{4} = 6 + 4 + 1 = 11$$

Therefore $P_{0.5} = \frac{11}{16} \approx 69\%$

Example (cont'd)

Redo for arbitrary p
(a)

$$P_p = \binom{6}{3}p^3(1-p)^3 + \binom{6}{4}p^4(1-p)^2 + \binom{6}{5}p^5(1-p) + \binom{6}{6}p^6$$

(b)

$$P_p = \binom{5}{2}p^2(1-p)^3 + \binom{5}{3}p^3(1-p)^2 + \binom{5}{4}p^5(1-p) + \binom{5}{5}p^5$$

(c)

$$P_p = \binom{4}{2}p^2(1-p)^2 + \binom{4}{3}p^3(1-p) + \binom{4}{4}p^4$$

Use of Recursion in Probability Computations

Question

Given n tosses of a coin, what is the probability of two HEADS *in a row? Compute for $n = 5, 10, 20, \ldots$*

Approaches:

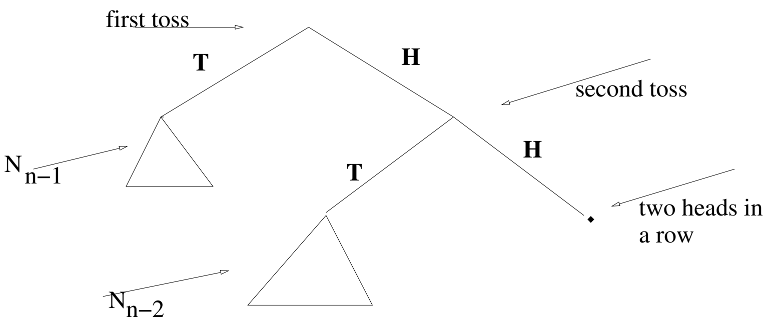
I. Write down all possibilities — 32 for $n = 5$, 1024 for $n = 10$, ...

II. Write a program; running time $\mathcal{O}(2^n)$ — why?

III. Inter-relate the numbers of relevant possibilities
$N_n \overset{\text{def}}{=}$ No. of sequences of n tosses *without* ...HH... pattern
Initial values:
$N_0 = 1,\ N_1 = 2,\ N_2 = 3$ (all except "HH")
$N_3 = 5$ (why?) $\quad N_4 = 8$ (why?)

Answer

We can summarise all possible outcomes in a **recursive tree**



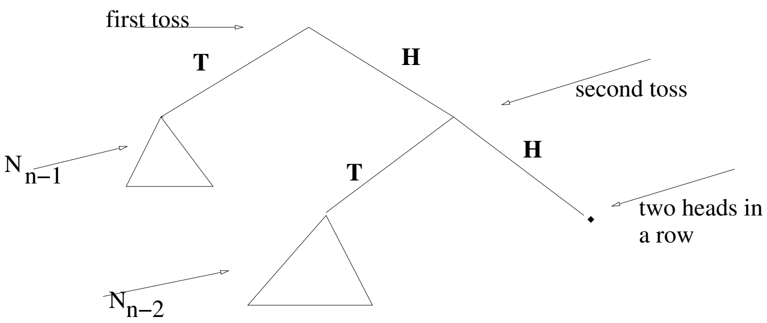
$N_n = N_{n-1} + N_{n-2}$ — Fibonacci recurrence: $N_n = \mathrm{FIB}(n+1)$

$N_n \approx \frac{1}{\sqrt{5}}\left(\frac{\sqrt{5}+1}{2}\right)^{n+1} \approx 0.72 \cdot (1.6)^n$

$p_n = \frac{2^n - \mathrm{FIB}(n+1)}{2^n} \approx 1 - 0.72 \cdot (0.8)^n$

Answer

We can summarise all possible outcomes in a **recursive tree**



$N_n = N_{n-1} + N_{n-2}$ — Fibonacci recurrence: $N_n = \mathrm{FIB}(n+1)$

$N_n \approx \frac{1}{\sqrt{5}}\left(\frac{\sqrt{5}+1}{2}\right)^{n+1} \approx 0.72 \cdot (1.6)^n$

$p_n = \frac{2^n - \mathrm{FIB}(n+1)}{2^n} \approx 1 - 0.72 \cdot (0.8)^n$

Example

Question

Given n tosses, what is the probability q_n of at least one HHH?

$q_0 = q_1 = q_2 = 0;\ q_3 = \frac{1}{8}$

Then recursive computation:

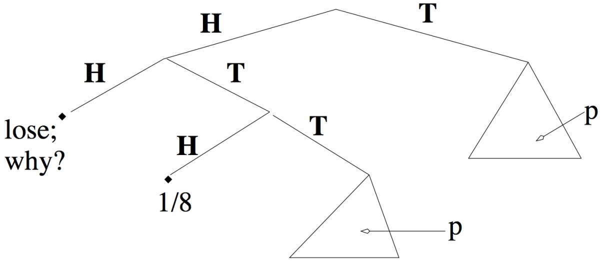
$$\begin{aligned} q_n = & \frac{1}{2} q_{n-1} && \text{(initial: T)} \\ & + \frac{1}{4} q_{n-2} && \text{(initial: HT)} \\ & + \frac{1}{8} q_{n-3} && \text{(initial: HHT)} \\ & + \frac{1}{8} && \text{(start with: HHH)} \end{aligned}$$

Example

Question

A coin is tossed 'indefinitely'. Which pattern is more likely (and by how much) to appear first, HTH or HHT?

let $p = P(HTH \text{ first})$



$p = \frac{1}{8} + \frac{1}{8}p + \frac{1}{2}p \;\Rightarrow\; \frac{3}{8}p = \frac{1}{8} \;\Rightarrow\; p = \frac{1}{3}$

NB

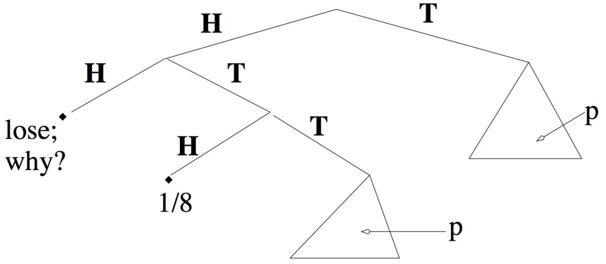
Probability that either pattern would appear at a given, prespecified point in the sequence of tosses is, obviously, the same.

Example

Question

A coin is tossed 'indefinitely'. Which pattern is more likely (and by how much) to appear first, HTH or HHT?

let $p = P(HTH \text{ first})$



$p = \frac{1}{8} + \frac{1}{8}p + \frac{1}{2}p \;\Rightarrow\; \frac{3}{8}p = \frac{1}{8} \;\Rightarrow\; p = \frac{1}{3}$

NB

Probability that either pattern would appear at a given, prespecified point in the sequence of tosses is, obviously, the same.

Example

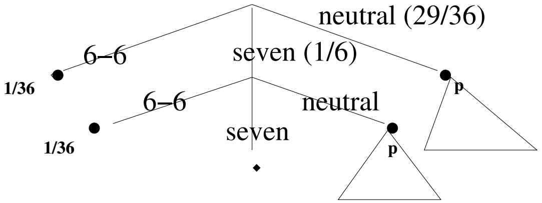
Question

Two dice are rolled repeatedly. What is the probability that '6–6' will occur before two consecutive (back-to-back) 'totals seven'?

NB

The probability of either occurring at a given roll is the same: $\frac{1}{36}$.

Let $p = P(6\text{–}6 \text{ first})$



$p = \frac{1}{36} + \frac{1}{6}\cdot\frac{1}{36} + \frac{1}{6}\cdot\frac{29}{36}p + \frac{29}{36}p \;\Rightarrow\; 216p = 7 + 203p \;\Rightarrow\; p = \frac{7}{13}$

Example

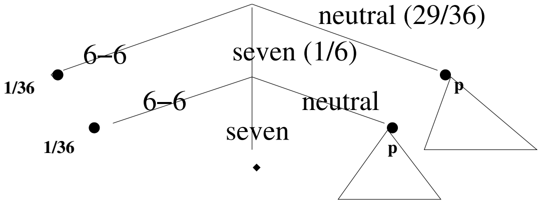
Question

Two dice are rolled repeatedly. What is the probability that '6–6' will occur before two consecutive (back-to-back) 'totals seven'?

NB

The probability of either occurring at a given roll is the same: $\frac{1}{36}$.

Let $p = P(6\text{–}6 \text{ first})$



$p = \frac{1}{36} + \frac{1}{6}\cdot\frac{1}{36} + \frac{1}{6}\cdot\frac{29}{36}p + \frac{29}{36}p \;\Rightarrow\; 216p = 7 + 203p \;\Rightarrow\; p = \frac{7}{13}$

NB

The majority of problems in probability and statistics do not have such elegant solutions. Hence the use of computers for either precise calculations or approximate simulations is mandatory. However, it is the use of recursion that simplifies such computing or, quite often, makes it possible in the first place.

Conditional Probability

Conditional Probability

Definition

Conditional probability of E **given** S:

$$P(E|S) = \frac{P(E \cap S)}{P(S)}, \quad E, S \subseteq \Omega$$

It is defined only when $P(S) \neq 0$

NB

$P(A|B)$ and $P(B|A)$ are, in general, not related — one of these values predicts, by itself, essentially nothing about the other. The only exception, applicable when $P(A), P(B) \neq 0$, is that $P(A|B) = 0$ iff $P(B|A) = 0$ iff $P(A \cap B) = 0$.

If P is the uniform distribution over a finite set Ω, then

$$P(E|S) = \frac{\frac{|E \cap S|}{|\Omega|}}{\frac{|S|}{|\Omega|}} = \frac{|E \cap S|}{|S|}$$

This observation can help in calculations...

Example

$\boxed{9.1.6}$ A coin is tossed four times. What is the probability of
(a) two consecutive HEADS
(b) two consecutive HEADS *given* that ≥ 2 tosses are HEADS

T	T	T	T	H	T	T	T
T	T	T	H	H	T	T	H
T	T	H	T	H	T	H	T
T	T	H	H	H	T	H	H
T	H	T	T	H	H	T	T
T	H	T	H	H	H	T	H
T	H	H	T	H	H	H	T
T	H	H	H	H	H	H	H

(a) $\frac{8}{16}$ (b) $\frac{8}{11}$

Some General Rules

Fact

- $A \subseteq B \Rightarrow P(A|B) \geq P(A)$
- $A \subseteq B \Rightarrow P(B|A) = 1$
- $P(A \cap B|B) = P(A|B)$
- $P(\emptyset|A) = 0$ *for* $A \neq \emptyset$
- $P(A|\Omega) = P(A)$
- $P(A^c|B) = 1 - P(A|B)$

NB

- $P(A|B)$ *and* $P(A|B^c)$ *are not related*
- $P(A|B), P(B|A), P(A^c|B^c), P(B^c|A^c)$ *are not related*

Example

Two dice are rolled and the outcomes recorded as b for the black die, r for the red die and $s = b + r$ for their total.
Define the events $B = \{b \geq 3\},\ R = \{r \geq 3\},\ S = \{s \geq 6\}$.

$P(S|B) = \frac{4+5+6+6}{24} = \frac{21}{24} = \frac{7}{8} = 87.5\%$

$P(B|S) = \frac{4+5+6+6}{26} = \frac{21}{26} = 80.8\%$

The (common) numerator $4 + 5 + 6 + 6 = 21$ represents the size of the $B \cap S$ — the common part of B and S, that is, the number of rolls where $b \geq 3$ and $s \geq 6$. It is obtained by considering the different cases: $b = 3$ and $s \geq 6$, then $b = 4$ and $s \geq 6$ etc.

The denominators are $|B| = 24$ and $|S| = 26$

Example (cont'd)

Recall: $B = \{b \geq 3\},\ R = \{r \geq 3\},\ S = \{s \geq 6\}$

$P(B) = P(R) = 2/3 = 66.7\%$
$P(S) = \frac{5+6+5+4+3+2+1}{36} = \frac{26}{36} = 72.22\%$
$P(S|B \cup R) = \frac{2+3+4+5+6+6}{32} = \frac{26}{32} = 81.25\%$

The set $B \cup R$ represents the event 'b or r'.
It comprises all the rolls except for those with *both* the red and the black die coming up either 1 or 2.

$P(S|B \cap R) = 1 = 100\%$ — because $S \supseteq B \cap R$

Exercise

$\boxed{9.1.9}$ Consider three red and eight black marbles; draw two without replacement. We write b_1 — Black on the first draw, b_2 — Black on the second draw, r_1 — Red on first draw, r_2 — Red on second draw
Find the probabilities
(a) both Red:

$$P(r_1 \wedge r_2) = P(r_1)P(r_2|r_1) = \frac{3}{11} \cdot \frac{2}{10} = \frac{3}{55}$$

Equivalently:
$|\text{two-samples}| = \binom{11}{2} = 55$; $|\text{Red two-samples}| = \binom{3}{2} = 3$
$P(\cdot) = \frac{\binom{3}{2}}{\binom{11}{2}} = \frac{3}{55}$

(b) both Black:
$$P(b_1 \wedge b_2) = P(b_1)P(b_2|b_1) = \frac{8}{11} \cdot \frac{7}{10} = \frac{28}{55} = \frac{\binom{8}{2}}{\binom{11}{2}}$$

Exercise

$\boxed{9.1.9}$ Consider three red and eight black marbles; draw two without replacement. We write b_1 — Black on the first draw, b_2 — Black on the second draw, r_1 — Red on first draw, r_2 — Red on second draw
Find the probabilities
(a) both Red:

$$P(r_1 \wedge r_2) = P(r_1)P(r_2|r_1) = \frac{3}{11} \cdot \frac{2}{10} = \frac{3}{55}$$

Equivalently:
$|\text{two-samples}| = \binom{11}{2} = 55$; $|\text{Red two-samples}| = \binom{3}{2} = 3$
$P(\cdot) = \frac{\binom{3}{2}}{\binom{11}{2}} = \frac{3}{55}$

(b) both Black:
$$P(b_1 \wedge b_2) = P(b_1)P(b_2|b_1) = \frac{8}{11} \cdot \frac{7}{10} = \frac{28}{55} = \frac{\binom{8}{2}}{\binom{11}{2}}$$

(c) one Red, one Black:

$$P(r_1 \wedge b_2) + P(b_1 \wedge r_2) = \frac{3 \cdot 8}{\binom{11}{2}} \quad \text{— why?}$$

By textbook (the 'hard way')

$$P(r_1 \wedge b_2) + P(b_1 \wedge r_2) = \frac{3}{11} \cdot \frac{8}{10} + \frac{8}{11} \cdot \frac{3}{10}$$

or

$$P(\cdot) = 1 - P(r_1 \wedge r_2) - P(b_1 \wedge b_2) = \frac{55 - 3 - 28}{55}$$

(c) one Red, one Black:

$$P(r_1 \wedge b_2) + P(b_1 \wedge r_2) = \frac{3 \cdot 8}{\binom{11}{2}} \quad \text{— why?}$$

By textbook (the 'hard way')

$$P(r_1 \wedge b_2) + P(b_1 \wedge r_2) = \frac{3}{11} \cdot \frac{8}{10} + \frac{8}{11} \cdot \frac{3}{10}$$

or

$$P(\cdot) = 1 - P(r_1 \wedge r_2) - P(b_1 \wedge b_2) = \frac{55 - 3 - 28}{55}$$

Exercise

$\boxed{9.1.12}$ What is the probability of a flush given that all five cards in a Poker hand are red?

Red cards = ♢'s + ♡'s
flush = all cards of the same suit

$$P(\text{flush} \mid \text{all five cards are Red}) = \frac{2 \cdot \binom{13}{5}}{\binom{26}{5}} = \frac{9}{230} \approx 4\%$$

Exercise

$\boxed{9.1.12}$ What is the probability of a flush given that all five cards in a Poker hand are red?

Red cards = ♢'s + ♡'s
flush = all cards of the same suit

$$P(\text{flush} \mid \text{all five cards are Red}) = \frac{2 \cdot \binom{13}{5}}{\binom{26}{5}} = \frac{9}{230} \approx 4\%$$

Exercise

$\boxed{9.1.22}$ Prove the following:
If $P(A|B) > P(A)$ ("positive correlation") then $P(B|A) > P(B)$

$P(A|B) > P(A)$

$\Rightarrow P(A \cap B) > P(A)P(B)$

$\Rightarrow \frac{P(A \cap B)}{P(A)} > P(B)$

$\Rightarrow P(B|A) > P(B)$

Exercise

$\boxed{9.1.22}$ Prove the following:
If $P(A|B) > P(A)$ ("positive correlation") then $P(B|A) > P(B)$

$P(A|B) > P(A)$

$\Rightarrow P(A \cap B) > P(A)P(B)$

$\Rightarrow \frac{P(A \cap B)}{P(A)} > P(B)$

$\Rightarrow P(B|A) > P(B)$

Stochastic Independence

Definition

A and B are **stochastically independent** (notation: $A \perp B$) if $P(A \cap B) = P(A) \cdot P(B)$

If $P(A) \neq 0$ and $P(B) \neq 0$, all of the following are *equivalent* definitions:

- $P(A \cap B) = P(A)P(B)$
- $P(A|B) = P(A)$
- $P(B|A) = P(B)$
- $P(A^c|B) = P(A^c)$ or $P(A|B^c) = P(A)$ or $P(A^c|B^c) = P(A^c)$

The last one claims that

$$A \perp B \Leftrightarrow A^c \perp B \Leftrightarrow A \perp B^c \Leftrightarrow A^c \perp B^c$$

Basic non-independent sets of events

- $A \subseteq B$
- $A \cap B = \emptyset$
- Any pair of one-point events $\{x\}, \{y\}$:
  either $x = y$ and $P(x|y) = 1$
  or $x \neq y$ and $P(x|y) = 0$

Independence of $A_1, \ldots, A_n$

$$P(A_{i_1} \cap A_{i_2} \cap \ldots \cap A_{i_k}) = P(A_{i_1}) \cdot P(A_{i_2}) \cdots P(A_{i_k})$$

for all possible collections $A_{i_1}, A_{i_2}, \ldots, A_{i_k}$.
This is often called (for emphasis) a *full* independence

Basic non-independent sets of events

- $A \subseteq B$
- $A \cap B = \emptyset$
- Any pair of one-point events $\{x\}, \{y\}$:
  either $x = y$ and $P(x|y) = 1$
  or $x \neq y$ and $P(x|y) = 0$

Independence of $A_1, \ldots, A_n$

$$P(A_{i_1} \cap A_{i_2} \cap \ldots \cap A_{i_k}) = P(A_{i_1}) \cdot P(A_{i_2}) \cdots P(A_{i_k})$$

for all possible collections $A_{i_1}, A_{i_2}, \ldots, A_{i_k}$.
This is often called (for emphasis) a *full* independence

Pairwise independence is a *weaker* concept.

Example

Toss of two coins

$$\left.\begin{array}{l} A = \langle \text{first coin } H \rangle \\ B = \langle \text{second coin } H \rangle \\ C = \langle \text{exactly one } H \rangle \end{array}\right\} \begin{array}{l} P(A) = P(B) = P(C) = \frac{1}{2} \\ P(A \cap B) = P(A \cap C) = P(B \cap C) = \frac{1}{4} \\ \text{However: } P(A \cap B \cap C) = 0 \end{array}$$

One can similarly construct a set of n events where any k of them are independent, while any $k+1$ are dependent (for $k < n$).

Independence of events, even just pairwise independence, can greatly simplify computations and reasoning in AI applications. It is common for many expert systems to make an approximating assumption of independence, even if it is not completely satisfied.

$$P(\text{sense}_t \mid \text{loc}_t, \text{sense}_{t-1}, \text{loc}_{t-1}, \ldots) = P(\text{sense}_t \mid \text{loc}_t)$$

Exercise

$\boxed{9.1.7}$ Suppose that an experiment leads to events A, B and C with $P(A) = 0.3$, $P(B) = 0.4$ and $P(A \cap B) = 0.1$

(a) $P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{1}{4}$

(b) $P(A^c) = 1 - P(A) = 0.7$

(c) Is $A \perp B$? No. $P(A) \cdot P(B) = 0.12 \neq P(A \cap B)$

(d) Is $A^c \perp B$? No, as can be seen from (c).

Note: $P(A^c \cap B) = P(B) - P(A \cap B) = 0.4 - 0.1 = 0.3$
$P(A^c) \cdot P(B) = 0.7 \cdot 0.4 = 0.28$

Exercise

$\boxed{9.1.7}$ Suppose that an experiment leads to events A, B and C with $P(A) = 0.3$, $P(B) = 0.4$ and $P(A \cap B) = 0.1$

(a) $P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{1}{4}$

(b) $P(A^c) = 1 - P(A) = 0.7$

(c) Is $A \perp B$? No. $P(A) \cdot P(B) = 0.12 \neq P(A \cap B)$

(d) Is $A^c \perp B$? No, as can be seen from (c).

Note: $P(A^c \cap B) = P(B) - P(A \cap B) = 0.4 - 0.1 = 0.3$
$P(A^c) \cdot P(B) = 0.7 \cdot 0.4 = 0.28$

Exercise

$\boxed{9.1.8}$ Given $A \perp B,\ P(A) = 0.4,\ P(B) = 0.6$

$P(A|B) = P(A) = 0.4$

$P(A \cup B) = P(A) + P(B) - P(A)P(B) = 0.76$

$P(A^c \cap B) = P(A^c)P(B) = 0.36$

Exercise

$\boxed{9.1.8}$ Given $A \perp B,\ P(A) = 0.4,\ P(B) = 0.6$

$P(A|B) = P(A) = 0.4$

$P(A \cup B) = P(A) + P(B) - P(A)P(B) = 0.76$

$P(A^c \cap B) = P(A^c)P(B) = 0.36$

Exercise

$\boxed{9.1.25}$ Does $A \perp B \perp C$ imply $(A \cap B) \perp (A \cap C)$?

No; this is almost never the case.
If somehow $(A \cap B) \perp (A \cap C)$ then it would give

$$P(A \cap B \cap C) = P(A \cap B \cap A \cap C) = P(A \cap B) \cdot P(A \cap C)$$

As A is independent of B and of C it would suggest

$$P(A \cap B \cap C) \overset{?}{=} P(A) \cdot P(B) \cdot P(A) \cdot P(C)$$

instead of the correct

$$P(A \cap B \cap C) = P(A) \cdot P(B) \cdot P(C)$$

Exercise

$\boxed{9.1.25}$ Does $A \perp B \perp C$ imply $(A \cap B) \perp (A \cap C)$?

No; this is almost never the case.
If somehow $(A \cap B) \perp (A \cap C)$ then it would give

$$P(A \cap B \cap C) = P(A \cap B \cap A \cap C) = P(A \cap B) \cdot P(A \cap C)$$

As A is independent of B and of C it would suggest

$$P(A \cap B \cap C) \overset{?}{=} P(A) \cdot P(B) \cdot P(A) \cdot P(C)$$

instead of the correct

$$P(A \cap B \cap C) = P(A) \cdot P(B) \cdot P(C)$$

Supplementary Exercise

$\boxed{9.5.5}$ (Supp) We are given two events with $P(A) = \frac{1}{4}$, $P(B) = \frac{1}{3}$. True, false or could be either?

(a) $P(A \cap B) = \frac{1}{12}$ — possible; it holds when $A \perp B$

(b) $P(A \cup B) = \frac{7}{12}$ — possible; it holds when A, B are disjoint

(c) $P(B|A) = \frac{P(B)}{P(A)}$ — false; correct is: $P(B|A) = \frac{P(B \cap A)}{P(A)}$

(d) $P(A|B) \geq P(A)$ — possible (it means that B “supports” A)

(e) $P(A^c) = \frac{3}{4}$ — true, since $P(A^c) = 1 - P(A)$

(f) $P(A) = P(B)P(A|B) + P(B^c)P(A|B^c)$ — true (also known as *total probability*)

Supplementary Exercise

$\boxed{9.5.5}$ (Supp) We are given two events with $P(A) = \frac{1}{4}$, $P(B) = \frac{1}{3}$. True, false or could be either?

(a) $P(A \cap B) = \frac{1}{12}$ — possible; it holds when $A \perp B$

(b) $P(A \cup B) = \frac{7}{12}$ — possible; it holds when A, B are disjoint

(c) $P(B|A) = \frac{P(B)}{P(A)}$ — false; correct is: $P(B|A) = \frac{P(B \cap A)}{P(A)}$

(d) $P(A|B) \geq P(A)$ — possible (it means that B “supports” A)

(e) $P(A^c) = \frac{3}{4}$ — true, since $P(A^c) = 1 - P(A)$

(f) $P(A) = P(B)P(A|B) + P(B^c)P(A|B^c)$ — true (also known as *total probability*)

Expectation

Random Variables

Definition

An (integer) **random variable** is a function from Ω to $\mathbb{Z}$.
In other words, it associates a number value with every outcome.

Random variables are often denoted by $X, Y, Z, \ldots$

Example

Random variable $X_s \overset{\text{def}}{=}$ sum of rolling two dice

$\Omega = \{(1,1), (1,2), \ldots, (6,6)\}$

$X_s((1,1)) = 2 \qquad X_s((1,2)) = 3 = X_s((2,1)) \quad \ldots$

$\boxed{9.3.3}$ Buy one lottery ticket for \$1. The only prize is \$1M.

$\Omega = \{win, lose\} \qquad X_L(win) = \$999,999 \qquad X_L(lose) = -\$1$

Expectation

Definition

The **expected value** (often called "expectation" or "average") of a random variable X is

$$E(X) = \sum_{k\in\mathbb{Z}} P(X=k)\cdot k$$

Example

The expected sum when rolling two dice is

$$E(X_s) = \frac{1}{36}\cdot 2 + \frac{2}{36}\cdot 3 + \ldots + \frac{6}{36}\cdot 7 + \ldots + \frac{1}{36}\cdot 12 = 7$$

$\boxed{9.3.3}$ Buy one lottery ticket for \$1. The only prize is \$1M. Each ticket has probability $6\cdot 10^{-7}$ of winning.

$E(X_L) = 6\cdot 10^{-7}\cdot \$999,999 + (1-6\cdot 10^{-7})\cdot -\$1 = -\$0.4$

NB

Expectation is a truly universal concept; it is the basis of all decision making, of estimating gains and losses, in all actions under risk. Historically, a rudimentary concept of expected value arose long before the notion of probability.

Theorem (linearity of expected value)

$E(X+Y) = E(X)+E(Y)$
$E(c\cdot X) = c\cdot E(X)$

Example

The expected sum when rolling two dice can be computed as

$$E(X_s) = E(X_1)+E(X_2) = 3.5+3.5 = 7$$

since $E(X_i) = \frac{1}{6}\cdot 1 + \frac{1}{6}\cdot 2 + \ldots + \frac{1}{6}\cdot 6$, for each die X_i

Example

$E(S_n)$, where $S_n \overset{\text{def}}{=} |\text{no. of HEADS in } n \text{ tosses}|$

- 'hard way'

  $E(S_n) = \sum_{k=0}^{n} P(S_n=k)\cdot k = \sum_{k=0}^{n} \frac{1}{2^n}\binom{n}{k}\cdot k$

  since there are $\binom{n}{k}$ sequences of n tosses with k HEADS, and each sequence has the probability $\frac{1}{2^n}$

  $= \frac{1}{2^n}\sum_{k=1}^{n}\frac{n}{k}\binom{n-1}{k-1}k = \frac{n}{2^n}\sum_{k=0}^{n-1}\binom{n-1}{k} = \frac{n}{2^n}\cdot 2^{n-1} = \frac{n}{2}$

  using the 'binomial identity' $\sum_{k=0}^{n}\binom{n}{k} = 2^n$

- 'easy way'

  $E(S_n) = E(S_1^1+\ldots+S_1^n) = \sum_{i=1\ldots n} E(S_1^i) = nE(S_1) = n\cdot\frac{1}{2}$

Note: $S_n \overset{\text{def}}{=} |\text{HEADS in } n \text{ tosses}|$ while each $S_1^i \overset{\text{def}}{=} |\text{HEADS in 1 toss}|$

Example

$E(S_n)$, where $S_n \overset{\text{def}}{=} |\text{no. of HEADS in } n \text{ tosses}|$

- 'hard way'

  $E(S_n) = \sum_{k=0}^{n} P(S_n=k)\cdot k = \sum_{k=0}^{n} \frac{1}{2^n}\binom{n}{k}\cdot k$

  since there are $\binom{n}{k}$ sequences of n tosses with k HEADS, and each sequence has the probability $\frac{1}{2^n}$

  $= \frac{1}{2^n}\sum_{k=1}^{n}\frac{n}{k}\binom{n-1}{k-1}k = \frac{n}{2^n}\sum_{k=0}^{n-1}\binom{n-1}{k} = \frac{n}{2^n}\cdot 2^{n-1} = \frac{n}{2}$

  using the 'binomial identity' $\sum_{k=0}^{n}\binom{n}{k} = 2^n$

- 'easy way'

  $E(S_n) = E(S_1^1+\ldots+S_1^n) = \sum_{i=1\ldots n} E(S_1^i) = nE(S_1) = n\cdot\frac{1}{2}$

Note: $S_n \overset{\text{def}}{=} |\text{HEADS in } n \text{ tosses}|$ while each $S_1^i \overset{\text{def}}{=} |\text{HEADS in 1 toss}|$

Example

$E(S_n)$, where $S_n \overset{\text{def}}{=} |\text{no. of } \textsc{heads} \text{ in } n \text{ tosses}|$

- 'hard way'

$$E(S_n) = \sum_{k=0}^{n} P(S_n = k)\cdot k = \sum_{k=0}^{n} \frac{1}{2^n}\binom{n}{k}\cdot k$$

since there are $\binom{n}{k}$ sequences of n tosses with k HEADS, and each sequence has the probability $\frac{1}{2^n}$

$$= \frac{1}{2^n}\sum_{k=1}^{n} \frac{n}{k}\binom{n-1}{k-1}k = \frac{n}{2^n}\sum_{k=0}^{n-1}\binom{n-1}{k} = \frac{n}{2^n}\cdot 2^{n-1} = \frac{n}{2}$$

using the 'binomial identity' $\sum_{k=0}^{n}\binom{n}{k} = 2^n$

- 'easy way'

$$E(S_n) = E(S_1^1 + \ldots + S_1^n) = \sum_{i=1\ldots n} E(S_1^i) = nE(S_1) = n\cdot\frac{1}{2}$$

Note: $S_n \overset{\text{def}}{=} |\textsc{heads} \text{ in } n \text{ tosses}|$ while each $S_1^i \overset{\text{def}}{=} |\textsc{heads} \text{ in } 1 \text{ toss}|$

NB

If $X_1, X_2, \ldots, X_n$ are independent, identically distributed random variables, then $E(X_1 + X_2 + \ldots + X_n)$ happens to be the same as $E(nX_1)$, but these are very different random variables.

Exercise

$\boxed{9.3.7}$

An urn has $m + n = 10$ marbles, $m \geq 0$ red and $n \geq 0$ blue.
7 marbles selected at random without replacement.
What is the expected number of red marbles drawn?

$$\frac{\binom{m}{0}\binom{n}{7}}{\binom{10}{7}}\cdot 0 + \frac{\binom{m}{1}\binom{n}{6}}{\binom{10}{7}}\cdot 1 + \frac{\binom{m}{2}\binom{n}{5}}{\binom{10}{7}}\cdot 2 + \ldots + \frac{\binom{m}{7}\binom{n}{0}}{\binom{10}{7}}\cdot 7$$

e.g.

$$\frac{\binom{5}{2}\binom{5}{5}}{\binom{10}{7}}\cdot 2 + \frac{\binom{5}{3}\binom{5}{4}}{\binom{10}{7}}\cdot 3 + \frac{\binom{5}{4}\binom{5}{3}}{\binom{10}{7}}\cdot 4 + \frac{\binom{5}{5}\binom{5}{2}}{\binom{10}{7}}\cdot 5$$

$$= \frac{10}{120}\cdot 2 + \frac{50}{120}\cdot 3 + \frac{50}{120}\cdot 4 + \frac{10}{120}\cdot 5 = \frac{420}{120} = 3.5$$

Exercise

$\boxed{9.3.7}$

An urn has $m + n = 10$ marbles, $m \geq 0$ red and $n \geq 0$ blue.
7 marbles selected at random without replacement.
What is the expected number of red marbles drawn?

$$\frac{\binom{m}{0}\binom{n}{7}}{\binom{10}{7}}\cdot 0 + \frac{\binom{m}{1}\binom{n}{6}}{\binom{10}{7}}\cdot 1 + \frac{\binom{m}{2}\binom{n}{5}}{\binom{10}{7}}\cdot 2 + \ldots + \frac{\binom{m}{7}\binom{n}{0}}{\binom{10}{7}}\cdot 7$$

e.g.

$$\frac{\binom{5}{2}\binom{5}{5}}{\binom{10}{7}}\cdot 2 + \frac{\binom{5}{3}\binom{5}{4}}{\binom{10}{7}}\cdot 3 + \frac{\binom{5}{4}\binom{5}{3}}{\binom{10}{7}}\cdot 4 + \frac{\binom{5}{5}\binom{5}{2}}{\binom{10}{7}}\cdot 5$$

$$= \frac{10}{120}\cdot 2 + \frac{50}{120}\cdot 3 + \frac{50}{120}\cdot 4 + \frac{10}{120}\cdot 5 = \frac{420}{120} = 3.5$$

Example

Find the average waiting time for the first HEAD, with no upper bound on the 'duration' (one allows for all possible sequences of tosses, regardless of how many times TAILS occur initially).

$$\begin{aligned} A = E(X_w) &= \textstyle\sum_{k=1}^{\infty} k \cdot P(X_w = k) = \sum_{k=1}^{\infty} k \frac{1}{2^k} \\ &= \tfrac{1}{2^1} + \tfrac{2}{2^2} + \tfrac{3}{2^3} + \dots \end{aligned}$$

This can be evaluated by breaking the sum into a sequence of geometric progressions

$$\frac{1}{2} + \frac{2}{2^2} + \frac{3}{2^3} + \dots$$

$$= \left(\frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \dots\right) + \left(\frac{1}{2^2} + \frac{1}{2^3} + \dots\right) + \left(\frac{1}{2^3} + \dots\right) + \dots$$

$$= 1 + \frac{1}{2} + \frac{1}{2^2} + \dots = 2$$

Example

Find the average waiting time for the first HEAD, with no upper bound on the 'duration' (one allows for all possible sequences of tosses, regardless of how many times TAILS occur initially).

$$\begin{aligned} A = E(X_w) &= \textstyle\sum_{k=1}^{\infty} k \cdot P(X_w = k) = \sum_{k=1}^{\infty} k \frac{1}{2^k} \\ &= \tfrac{1}{2^1} + \tfrac{2}{2^2} + \tfrac{3}{2^3} + \dots \end{aligned}$$

This can be evaluated by breaking the sum into a sequence of geometric progressions

$$\frac{1}{2} + \frac{2}{2^2} + \frac{3}{2^3} + \dots$$

$$= \left(\frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \dots\right) + \left(\frac{1}{2^2} + \frac{1}{2^3} + \dots\right) + \left(\frac{1}{2^3} + \dots\right) + \dots$$

$$= 1 + \frac{1}{2} + \frac{1}{2^2} + \dots = 2$$

There is also a recursive 'trick' for solving the sum

$$A = \sum_{k=1}^{\infty} \frac{k}{2^k} = \sum_{k=1}^{\infty} \frac{k-1}{2^k} + \sum_{k=1}^{\infty} \frac{1}{2^k} = \frac{1}{2} \sum_{k=1}^{\infty} \frac{k-1}{2^{k-1}} + 1 = \frac{1}{2} A + 1$$

Now $\quad A = \frac{A}{2} + 1$ and $A = 2$

NB

A much simpler but equally valid argument is that you expect 'half' a HEAD in 1 toss, so you ought to get a 'whole' HEAD in 2 tosses.

Theorem

The average number of trials needed to see an event with probability p is $\frac{1}{p}$.

Exercise

$\boxed{9.4.12}$ A die is rolled until the first 4 appears. What is the expected waiting time?

$P(\text{roll } 4) = \frac{1}{6}$ hence $E(\text{no. of rolls until first } 4) = 6$

Exercise

$\boxed{9.4.12}$ A die is rolled until the first 4 appears. What is the expected waiting time?

$P(\text{roll } 4) = \frac{1}{6}$ hence $E(\text{no. of rolls until first } 4) = 6$

Example

You face a quiz consisting of six true/false questions, and your plan is to guess the answer to each question (randomly, with probability 0.5 of being right). There are no negative marks, and answering four or more questions correctly suffices to pass.
What is the expected score and what is the probability of passing?

To pass you would need four, five or six correct guesses. Therefore,

$$p(\text{pass}) = \frac{\binom{6}{4}+\binom{6}{5}+\binom{6}{6}}{64} = \frac{15+6+1}{64}$$

The expected score from a single question is 0.5, as there is no penalty for errors. For six questions the expected value is $6 \cdot 0.5 = 3$

Example

You face a quiz consisting of six true/false questions, and your plan is to guess the answer to each question (randomly, with probability 0.5 of being right). There are no negative marks, and answering four or more questions correctly suffices to pass.
What is the expected score and what is the probability of passing?

To pass you would need four, five or six correct guesses. Therefore,

$$p(\text{pass}) = \frac{\binom{6}{4}+\binom{6}{5}+\binom{6}{6}}{64} = \frac{15+6+1}{64}$$

The expected score from a single question is 0.5, as there is no penalty for errors. For six questions the expected value is $6 \cdot 0.5 = 3$

Example

To find an object $\mathcal{X}$ in an unsorted list L of elements, one needs to search linearly through L. Let the probability of $\mathcal{X} \in L$ be p, hence there is $1-p$ likelihood of $\mathcal{X}$ being absent altogether. Find the expected number of comparison operations.

If the element is in the list, then the number of comparisons averages to $\frac{1}{n}(1+\ldots+n)$; if absent we need n comparisons. The first case has probability p, the second $1-p$. Combining these we find

$$E_n = p\frac{1+\ldots+n}{n} + (1-p)n = p\frac{n+1}{2} + (1-p)n = (1-\frac{p}{2})n + \frac{p}{2}$$

As one would expect, increasing p leads to a lower E_n.

Example

To find an object $\mathcal{X}$ in an unsorted list L of elements, one needs to search linearly through L. Let the probability of $\mathcal{X} \in L$ be p, hence there is $1 - p$ likelihood of $\mathcal{X}$ being absent altogether. Find the expected number of comparison operations.

If the element is in the list, then the number of comparisons averages to $\frac{1}{n}(1 + \ldots + n)$; if absent we need n comparisons. The first case has probability p, the second $1 - p$. Combining these we find

$$E_n = p\frac{1 + \ldots + n}{n} + (1-p)n = p\frac{n+1}{2} + (1-p)n = (1 - \frac{p}{2})n + \frac{p}{2}$$

As one would expect, increasing p leads to a lower E_n.

One may expect that this would indicate a practical rule — that high probability of success might lead to a high expected value. Unfortunately this is *not* the case in a great many practical situations.
Many lottery advertisements claim that buying more tickets leads to better expected results — and indeed, obviously you will have more potentially winning tickets. However, the expected value *decreases* when the number of tickets is increased.

As an example, let us consider a punter placing bets on a roulette (outcomes: $0, 1 \ldots 36$). Tired of losing, he decides to place \$1 on 24 ‘ordinary’ numbers $a_1 < a_2 < \ldots < a_{24}$, selected from among 1 to 36.

His probability of winning is high indeed — $\frac{24}{37} \approx 65\%$; he scores on any of his choices, and loses only on the remaining thirteen numbers.

But what about his performance?

- If one of his numbers comes up, say a_i, he wins \$35 from the bet on that number and loses \$23 from the bets on the remaining numbers, thus collecting \$12.
  This happens with probability $p = \frac{24}{37}$.
- With probability $q = \frac{13}{37}$ none of his numbers appears, leading to loss of \$24.

The expected result

$$p \cdot \$12 - q \cdot \$24 = \$12\frac{24}{37} - \$24\frac{13}{37} = -\$\frac{24}{37} \approx -65¢$$

Many so-called 'winning systems' that purports to offer a winning strategy do something akin — they provide a scheme for frequent relatively moderate wins, but at the cost of an occasional very big loss.

It turns out (it is a formal theorem) that there can be *no system* that converts an ‘unfair’ game into a 'fair' one. In the language of decision theory, ‘unfair’ denotes a game whose individual bets have negative expectation.

It can be easily checked that any individual bets on roulette, on lottery tickets or on just about any commercially offered game have negative expected value.

Standard Deviation and Variance

Definition

For random variable X with expected value (or: **mean**) $\mu = E(X)$, the **standard deviation** of X is

$$\sigma = \sqrt{E((X-\mu)^2)}$$

and the **variance** of X is

$$\sigma^2$$

Standard deviation and variance measure how spread out the values of a random variable are. The smaller σ^2 the more confident we can be that $X(\omega)$ is close to $E(X)$, for a randomly selected ω.

NB

The variance can be calculated as $E((X-\mu)^2) = E(X^2) - \mu^2$

Example

Random variable $X_d \overset{\text{def}}{=}$ value of a rolled die

$$\mu = E(X_d) = 3.5$$

$$E(X_d^2) = \frac{1}{6}\cdot 1 + \frac{1}{6}\cdot 4 + \frac{1}{6}\cdot 9 + \frac{1}{6}\cdot 16 + \frac{1}{6}\cdot 25 + \frac{1}{6}\cdot 36 = \frac{91}{6}$$

$$\text{Hence,}\quad \sigma^2 = E(X_d^2) - \mu^2 = \frac{35}{12} \quad\Rightarrow\quad \sigma \approx 1.71$$

Exercise

$\boxed{9.5.10}$ (Supp) Two independent experiments are performed.
$P(\text{1st experiment succeeds}) = 0.7$
$P(\text{2nd experiment succeeds}) = 0.2$
Random variable X counts the number of successful experiments.

(a) Expected value of X? $E(X) = 0.7 + 0.2 = 0.9$

(b) Probability of exactly one success? $0.7\cdot 0.8 + 0.3\cdot 0.2 = 0.62$

(c) Probability of at most one success? (b)$+0.3\cdot 0.8 = 0.86$

(e) Variance of X? $\sigma^2 = (0.62\cdot 1 + 0.14\cdot 4) - 0.9^2 = 0.37$

Exercise

$\boxed{9.5.10}$ (Supp) Two independent experiments are performed.
$P(\text{1st experiment succeeds}) = 0.7$
$P(\text{2nd experiment succeeds}) = 0.2$
Random variable X counts the number of successful experiments.

(a) Expected value of X? $E(X) = 0.7 + 0.2 = 0.9$

(b) Probability of exactly one success? $0.7\cdot 0.8 + 0.3\cdot 0.2 = 0.62$

(c) Probability of at most one success? (b)$+0.3\cdot 0.8 = 0.86$

(e) Variance of X? $\sigma^2 = (0.62\cdot 1 + 0.14\cdot 4) - 0.9^2 = 0.37$

Cumulative Distribution Functions

Definition

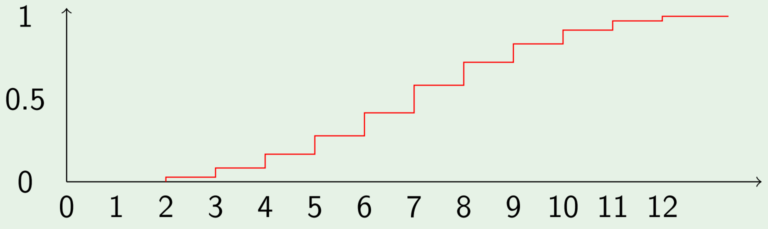
The **cumulative distribution function** $\mathrm{CDF}_X : \mathbb{Z} \longrightarrow \mathbb{R}$ of an integer random variable X is defined as

$$\mathrm{CDF}_X(y) \mapsto \sum_{k \leq y} P(X = k)$$

$\mathrm{CDF}_X(y)$ collects the probabilities $P(X)$ for all values up to y

Example

Cumulative distribution function for sum of 2 dice



Example: Binomial Distributions

Definition

Binomial random variables count the number of ‘successes’ in n independent experiments with probability p for each experiment.

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

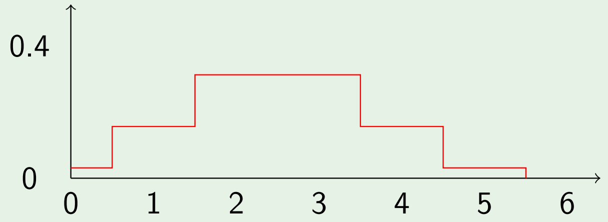
$$\mathrm{CDF}_B(y) \mapsto \sum_{k \leq y} \binom{n}{k} p^k (1-p)^{n-k}$$

Theorem

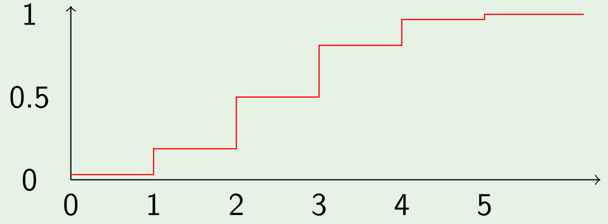
If X is a binomially distributed random variable based on n and p, then $E(X) = n \cdot p$ with variance $\sigma^2 = n \cdot p \cdot (1-p)$

Example (binomial distribution)

No. of HEADS in 5 coin tosses



CDF for no. of HEADS in 5 coin tosses



Exercise

$\boxed{\text{9.4.10}}$ An experiment is repeated 30,000 times with probability of success $\frac{1}{4}$ each time.

(a) Expected number of successes? $E(X) = 30,000 \cdot \frac{1}{4} = 7500$

(b) Standard deviation? $\sigma = \sqrt{30,000 \cdot \frac{1}{4} \cdot \frac{3}{4}} = 75$

Exercise

$\boxed{9.4.10}$ An experiment is repeated 30,000 times with probability of success $\frac{1}{4}$ each time.

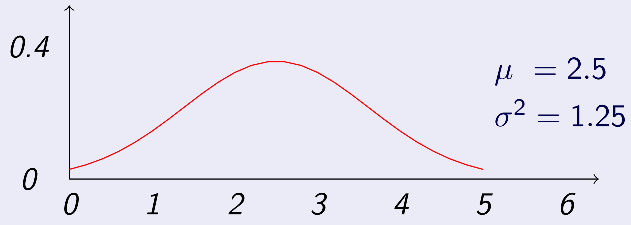
(a) Expected number of successes? $E(X) = 30,000 \cdot \frac{1}{4} = 7500$

(b) Standard deviation? $\sigma = \sqrt{30,000 \cdot \frac{1}{4} \cdot \frac{3}{4}} = 75$

Normal Distribution

Fact

For large n, binomial distributions can be approximated by **normal distributions** *(a.k.a.* **Gaussian distributions***) with mean $\mu = n \cdot p$ and variance $\sigma^2 = n \cdot p \cdot (1 - p)$*



$$\frac{1}{\sqrt{2\sigma^2\pi}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Summary

- counting
  - union rule, product rule, $n!$, $\Pi(n, r)$, $\binom{n}{r}$
- events and their probability
- counting, inclusion-exclusion, recursion for probabilities
- conditional probability $P(A|B)$, independence $A \perp B$
- random variables X, expected value $E(X)$ ($=$ mean μ)
- CDF, standard deviation σ, variance σ^2