WELCOME

Discrete Mathematics package consists of material from two booklets:

- Lecture Notes (this booklet), and
- Tutorials and Practice Classes: Questions.

Beside Moodle, this unit uses our server turing as the learning management system, so the whole enchilada can be found at http://turing.une.edu.au/~amth140/

Each of the 25 sections in these notes is approximately equivalent to one lecture and you should aim to cover, on average, between 2 and 3 sections (if you need some extra time for revision) per week. The Tutorials present new material and so require additional time to master. The Practice classes consist of exercises related to material in the Lecture Notes.

In the following, there are a few passages marked on the margin by the “▼” and “▲” pair (just like the pair on the right hand side of this paragraph), and a few paragraphs marked by the “■” symbol. These marked parts are optional for this unit; they are presented for those who wish to pursue further understanding of the topics concerned.

There are a number of places where subjects are explained in great details (which can at times even be irritating to some) and often in less rigorous informal terms. Such coverage is to help bridge the more intuitive understanding. To differentiate such text from the normal ones, we will always place such informal detailed exposition between a pair of “▽” and “△” symbols, just like the pair on the right hand side of this paragraph.

There are two optional reference books if you feel the need for extra reading and insights:


Information on the link between Grimaldi’s book and the topics in this unit can be found a few pages on in this booklet. These reference books are optional since this unit is entirely covered by the booklets: Lecture Notes and Unit Information (Tutorial and Practice Classes).

I wish to acknowledge the work of Dr Chris Radford, Dr Zhuhan Jiang, Dr Tim Dalby, Dr Yhong Du, Dr Shusen Yan and all others who contributed to this unit.

Best wishes with your studies.

June 2013  

Dr Ioan Despi
## Contents

### Preliminary Mathematics

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1 Absolute Value</td>
<td>11</td>
</tr>
<tr>
<td>0.2 Triangle Inequality</td>
<td>12</td>
</tr>
<tr>
<td>0.3 Generalised Triangle Inequality</td>
<td>12</td>
</tr>
<tr>
<td>0.4 Sigma, Pi Notations</td>
<td>12</td>
</tr>
<tr>
<td>0.5 Powers and Logarithms</td>
<td>13</td>
</tr>
<tr>
<td>0.6 Trigonometric Properties</td>
<td>13</td>
</tr>
<tr>
<td>0.7 Roots of Quadratic Equations</td>
<td>13</td>
</tr>
<tr>
<td>0.8 Factorising Quadratic Polynomials</td>
<td>14</td>
</tr>
<tr>
<td>0.9 Root and Factor Equivalence</td>
<td>14</td>
</tr>
<tr>
<td>0.10 Elimination Method for Linear Equations</td>
<td>15</td>
</tr>
<tr>
<td>0.11 Binomial Expansions</td>
<td>16</td>
</tr>
<tr>
<td>0.12 Matrix Multiplications</td>
<td>17</td>
</tr>
</tbody>
</table>

### Please Read Me

19

### Sets

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Basic Set Operations</td>
<td>20</td>
</tr>
<tr>
<td>1.2 Set Identities</td>
<td>23</td>
</tr>
<tr>
<td>1.3 Russell’s Paradox</td>
<td>24</td>
</tr>
</tbody>
</table>

### Mathematical Induction

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Principle of Mathematical Induction</td>
<td>27</td>
</tr>
<tr>
<td>2.2 Strong Principle of Mathematical Induction</td>
<td>32</td>
</tr>
</tbody>
</table>

### Efficiency, Big $O$

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Nested Evaluation of a Polynomial</td>
<td>35</td>
</tr>
<tr>
<td>3.2 Big $O$ Notation</td>
<td>37</td>
</tr>
<tr>
<td>3.3 Simple Features of $O$</td>
<td>39</td>
</tr>
<tr>
<td>Chapter</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>10</td>
<td><strong>Eulerian and Hamiltonian Circuits</strong></td>
</tr>
<tr>
<td>10.1</td>
<td>Two Practical Problems</td>
</tr>
<tr>
<td>10.2</td>
<td>Eulerian and Hamiltonian Paths and Circuits</td>
</tr>
<tr>
<td>10.3</td>
<td>Fleury's Algorithm for Finding an Eulerian Path or Circuit</td>
</tr>
<tr>
<td>11</td>
<td><strong>Graph Isomorphism and Matrix Representations</strong></td>
</tr>
<tr>
<td>11.1</td>
<td>Graph Isomorphism and Isomorphic Invariants</td>
</tr>
<tr>
<td>11.2</td>
<td>Matrices</td>
</tr>
<tr>
<td>11.3</td>
<td>The Adjacency Matrix</td>
</tr>
<tr>
<td>12</td>
<td><strong>Trees</strong></td>
</tr>
<tr>
<td>12.1</td>
<td>Trees, Rooted Trees and Binary Trees</td>
</tr>
<tr>
<td>12.2</td>
<td>Traversal of Binary Trees</td>
</tr>
<tr>
<td>13</td>
<td><strong>Spanning Trees</strong></td>
</tr>
<tr>
<td>13.1</td>
<td>The Concept of Spanning Trees</td>
</tr>
<tr>
<td>13.2</td>
<td>Spanning Tree Algorithm</td>
</tr>
<tr>
<td>13.3</td>
<td>Kruskal's Algorithm</td>
</tr>
<tr>
<td>14</td>
<td><strong>Number Bases</strong></td>
</tr>
<tr>
<td>14.1</td>
<td>Frequently Used Number Systems</td>
</tr>
<tr>
<td>14.2</td>
<td>Conversion to Numbers of Different Bases</td>
</tr>
<tr>
<td>15</td>
<td><strong>Relations</strong></td>
</tr>
<tr>
<td>15.1</td>
<td>Binary Relations</td>
</tr>
<tr>
<td>15.2</td>
<td>Reflexivity, Symmetry and Transitivity</td>
</tr>
<tr>
<td>16</td>
<td><strong>Equivalence Relations</strong></td>
</tr>
<tr>
<td>16.1</td>
<td>Equivalence Classes and Partitions</td>
</tr>
<tr>
<td>16.2</td>
<td>Property Closure</td>
</tr>
<tr>
<td>17</td>
<td><strong>Partial Order Relations</strong></td>
</tr>
</tbody>
</table>
17.1 Partial Orderings ................................................................. 131
17.2 Hasse Diagrams ............................................................... 133
17.3 Topological Sorting ........................................................... 138

18 Switching Circuits and Boolean Algebra 143
18.1 Switching Circuits ............................................................ 143
18.2 Boolean Algebra .............................................................. 145
18.3 Algebraic Equivalence ......................................................... 149
18.4 Sets connection with Boolean Algebra ................................. 150

19 Boolean Functions 152
19.1 Convert Boolean Function to Boolean Expression ..................... 152
19.2 Canonical Form .............................................................. 153
19.3 Minimal Representation .................................................... 156
19.4 Logic Gates ................................................................. 158

20 Karnaugh Maps 162
20.1 Grid Layout for Karnaugh Maps .......................................... 162
20.2 Simplification Procedure .................................................... 165

21 Recurrence Relations 169
21.1 Recurrence Relations ....................................................... 169
21.2 Basic Concepts Related to Recurrence Relations ....................... 171
21.3 Simplest Case of General Solutions .................................... 173

22 Solution of Linear Homogeneous Recurrence Relations 178
22.1 General Solutions for Homogeneous Problems ....................... 178
22.2 Ideas Behind the Theorem .................................................. 181

23 Basics of Linear Nonhomogeneous Recurrence Relations 183
23.1 Connection between Homogeneous and Nonhomogeneous Problems .... 183
Reading Guide

Students are only required to read and understand the material covered by the two booklets

- Lecture Notes and
- Tutorial and Practice Classes,

with the help of completing the prescribed assignments.

Anything else is optional and will therefore not be contained in the final examination.

Our School’s server, turing.une.edu.au contains all the information needed for this unit. Book this address: http://turing.une.edu.au/~amth140/

General

- It is quite likely that students can work through this unit without having to read any reference books. Hence the rest of this reading guide is mainly for those who wish to, or have already done so, get hold of a copy of the reference book cited below. However we note that not all subjects in the reference book are covered in this unit, and those covered, are not necessarily in the same order. This is only natural because the choice of the subjects depends on the School’s unit syllabus.

- Supplementary Exercises given at the end of each section are in addition to those exercises in the Practice Classes. Some are relatively challenging. The Supplementary Exercises are presented for those who wish to practice more. Some of the hints and solutions to the supplementary exercises are appended to the end of these Notes.

- Should you encounter an unfamiliar terminology or symbol at any stage, always look up the index first to see if you could quickly locate the needed information or explanations.

- All textbooks do not contain the same topics and do not present the subjects in the same order. For this unit, Grimaldi’s book seems to cover most of the unit’s syllabus.

Reference Books


Topics for Lectures, Tutorials and Practice Classes

“§” refers to a section in the above Grimaldi’s book.

1. Sets (§3.1, 3.2)

2. Mathematical induction (§4.1)
   ( \( \mathbb{R}, \mathbb{Z}, \mathbb{N} \), partition, cartesian product )
   ( sequence, \( \sum, \prod \), induction principle, binomial formula )
T-1: The bubble sort, see also subsection A. (tutorial-1)
P-1: Sets and mathematical induction (practice class-1)

Note. Internal students will start the 1st tutorial and practice class all in the 2nd week.

3. Efficiency, Big \( O \) (§5.7))

4. \( \Theta \) notation (§5.7 exercises)

5. Analysis of algorithms (§5.8)
   ( Horner’s algorithm, big \( O \) )
   ( \( \Theta \) notation, connection with big \( O \) )
   ( sequential/binary search, flooring, \( B(n) = O(\log n) \) )
T-2: Divide and conquer, see also subsection A.4
P-2: Horner’s algorithm, \( O \) and \( \Theta \) notation

6. Symbolic logic (§2.1, 2.2)

7. Propositional logic (§2.3)
   ( tautology, logical equivalence, \( p \rightarrow q \) )
   ( valid argument, inference rules, \( p \iff q \) )
T-3: Insertion sort and selection sort, see also subsections A.1 and A.3
P-3: Symbolic logic and logical equivalence

8. Predicate calculus (§2.4, 2.5)
   ( quantifiers, \( \forall, \exists \), \( \Rightarrow, \Leftrightarrow \), iff, contrapositive )
   ( directed/sub/simple/complete graphs, \( K_n, K_{m,n} \) )
   ( walks, Kuratowski theorem )
T-4: Quicksort, see also subsection A.5
P-4: Propositions and predicates

9. Graphs and their basic types (§11.1–11.3)

10. Euler and Hamiltonian circuits (§11.3, 11.5)
    ( Euler’s theorem, Fleury’s algorithm)
    ( 7 bridge problem, travelling salesman problem )
11. Graph isomorphism and matrix representation (§11.2)
   (one-to-one, onto, isomorphic invariant, adjacency matrix)
   T-5: Features of some graphs
   P-5: Isomorphism and adjacency matrix

12. Trees (§12.1, 12.2)

13. Spanning trees (§12.1, 13.2)
   (cycle, leaf, rooted tree, binary tree, traversal, postfix notation)
   (weighted graph, Kruskal’s algorithm)
   T-6: Kruskal’s algorithm
   P-6: Binary trees for representation and sorting

14. Number bases (§4.3)

15. Introduction to relations (§5.1, 7.1)
   (remainder theorem, division algorithm, blockwise conversion, octal/hex)
   (reflexive, symmetric, transitive, \( m \equiv n \pmod{d} \))
   T: This is a phantom tutorial - catch up on your work
   P-7: Number bases

16. Equivalence relations (§7.4)

17. Partial order relations (§7.3)
   (equivalence class, property closure)
   (antisymmetric, comparable, maximal element, Hasse diagram)
   T: This is a phantom tutorial - catch up on your work
   P-8: Equivalence relations and partial order relations

18. Switching circuits and Boolean algebra (§2.2, 15.4)

19. Boolean functions (§15.1)
   (mapping, binary/unary ops, equivalence of Boolean/switching/symbolic...)
   (Boolean function to expression/switches, minimal representation)
   T-7: Boolean algebra and voting machines
   P-9: Switching circuits and their design

20. Karnaugh maps (§15.2)

21. Recurrence relations (§4.2, 10.1)
   (2×4, 4×4 blocks)
   (iteration, recursion, characteristic eqn, solution for homogeneous eqn)
   T-8: Logic gates and digital adders
   P: This is a phantom practice class - catch up on your work
22. Solution of linear homogeneous recurrence relations (§10.2)

23. Basics of linear nonhomogeneous recurrence relations (R2: §10.3)
   ( repeated roots, proof hints )
   ( particular solutions for “non-resonant cases” )
   T-9: More on Karnaugh maps
   P-10: Solutions of homogeneous recurrence relations

24. Solution of linear nonhomogeneous recurrence relations (R2: §10.3)
   ( particular solutions in general cases )
   T: Start your revision or see P-12 below

Going from semesters (13 weeks) to trimesters (11-12 weeks) in 2012 definitively will have an impact on the above plan for tutorials and practicals.

Other References

There are other references to be mentioned in the Notes. You may consult them if you have time, but you don’t have to. In particular, the easy-to-read book by Susanna S Epp, *Discrete Mathematics with Applications* does not cover as many topics as the reference book by Ralph Grimaldi.

Additional Information

Please be aware that additional reading guide or instructions may be available in the form of handouts, which will provide more accurate and updated information. Latest online information is available via the web at

http://turing.une.edu.au/~amth140/

There will be lecture notes, announcements, corrections, postings, etc. appearing on the web as the unit proceeds. So you should pay close attention to the amth140 web pages on turing.

Preliminary Mathematics

This part serves as a simple reminder to the things you need to be aware of. It is basically presented through a number of examples. Most of this part can be helpful at some stage of
studying this unit. However, you may choose to skip it initially and opt to get referred back later in the unit.

0.1 Absolute Value

For any real number \( a \), its **absolute value** (or **modulus**) is the numerical value of \( a \) without regard to its sign and is denoted by \( |a| \), that is:

\[
|a| = \begin{cases} 
  a & \text{if } a \geq 0 \\
  -a & \text{if } a < 0 
\end{cases}
\]

It follows immediately that \( |-a| = |a| \), hence

\[ |23.6| = 23.6, \quad |-23.6| = 23.6, \quad |-7086| = 7086 \quad |0| = 0 \]

Because the square-root of a positive number represents the positive square root (without a sign), it follows that

\[ |a| = \sqrt{a^2} \]

can sometimes be used as a definition of absolute value. In a similar way, for any complex number \( z = x + iy \), where \( x \) and \( y \) are real numbers and \( i = \sqrt{-1} \), the **absolute value** or **modulus** of \( z \) is denoted \( |z| \) and is defined as

\[ |z| = \sqrt{x^2 + y^2} \]

Given the numbers \( a, b, c \), the absolute value has the following five fundamental properties:

1. \( |a| \geq 0 \)
2. \( |a| = 0 \iff a = 0 \) \quad or \quad \( |a - b| = 0 \iff a = b \)
3. \( |ab| = |a||b| \)
4. \( \frac{|a|}{|b|} = \frac{|a|}{|b|} \)
5. \( |a| \leq c \iff -c \leq a \leq c \)

as well as a sixth one:
0.2 Triangle Inequality

For any two numbers $a$ and $b$, one has always $|a + b| \leq |a| + |b|$, $|a + b| \geq |a| - |b|$
or, equivalently 6. $|a| - |b| \leq |a + b| \leq |a| + |b|$  

For example, if we take $a = -3.5$ and $b = -0.2$, it is easy to verify that  

$$ | -3.5 - 0.2| \leq | -3.5| + |-0.2|, \quad | -3.5 - 0.2| \geq | -3.5| - |-0.2| $$

0.3 Generalised Triangle Inequality

Let $x_1, x_2, \ldots, x_n$ be $n$ numbers. Then  

$$ |x_1 + x_2 + \cdots + x_n| \leq |x_1| + |x_2| + \cdots + |x_n|, \quad |x_1 + x_2 + \cdots + x_n| \geq |x_1| - |x_2| - \cdots - |x_n| $$

Hence it is easy to verify, for example, that  

$$ | -3.5 + 2.1 - 0.2| \leq | -3.5| + |2.1| + |-0.2|, \quad | -3.5 + 2.1 - 0.2| \geq | -3.5| - |2.1| - |-0.2| $$

where we have taken $x_1 = -3.5$, $x_2 = 2.1$ and $x_3 = -0.2$

0.4 Sigma, Pi Notations

The **sigma notation** uses the Greek letter $\Sigma$ (Sigma) to denote the sum of the terms of a sequence in a condensed form, by specifying the start and the end values of an index, that is  

$$ a_1 + a_2 + \cdots + a_m = \sum_{j=1}^{m} a_j $$

The generalised triangle inequality could be written as:

$$ \left| \sum_{k=1}^{n} x_k \right| \leq \sum_{k=1}^{n} |x_k| $$

In a similar way, **big-pi notation** uses the Greek letter $\Pi$ (Pi) to denote the product of the terms of a sequence in condensed form, by specifying the start and the end values of an index, that is  

$$ a_1 \times a_2 \times \cdots \times a_m = \prod_{j=1}^{m} a_j $$
0.5 Powers and Logarithms

Let $a$, $b$ and $c$ be positive, and $x$ and $y$ be arbitrary real values. Then

$$a^x = y \quad \text{if and only if} \quad x = \log_a y$$

Note. if and only if is shortcut in mathematics as iff and $P$ iff $Q$ means that if $P$ then $Q$ AND if $Q$ then $P$.

Moreover,

$$a^{xy} = (a^x)^y \quad \quad a^{x+y} = a^x \cdot a^y \quad \quad (a \cdot b)^x = (a^x) \cdot (b^x) \quad \quad a^0 = 1$$

$$\log_a b^x = x \cdot \log_a b \quad \quad \log_a(b \cdot c) = \log_a b + \log_a c \quad \quad \log_a 1 = 0$$

Hence we have for example

$$2^5 \times 2^7 = 2^{5+7} = 2^{12}, \quad (2^5)^7 = 2^{5 \times 7} = 2^{35}, \quad \log_2 2 = 1, \quad \log_2 2^9 = 9$$

0.6 Trigonometric Properties

Let $x$ be any real number. Then

$$\sin^2 x + \cos^2 x = 1, \quad |\sin x| \leq 1, \quad |\cos x| \leq 1$$

We note that the two inequalities can be derived easily from the first identity. Recall also that $\sin(k\pi) = 0$ and $\cos(k\pi) = (-1)^k$ hold for any integer $k$.

0.7 Roots of Quadratic Equations

A second order polynomial equation

$$ax^2 + bx + c = 0, \quad a, b, c \in \mathbb{R}, \ a \neq 0$$

has two roots $x_1$ and $x_2$, given precisely by

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

Example

1. The roots of $3x^2 + x - 2 = 0$ are given by

$$x_{1,2} = \frac{-1 \pm \sqrt{(1)^2 - 4 \times 3 \times (-2)}}{2 \times 3} = \frac{-1 \pm 5}{6}.$$

Hence $x_1 = (-1 + 5)/6 = 2/3$ and $x_2 = (-1 - 5)/6 = -1$. 

13
0.8 Factorising Quadratic Polynomials

A second order polynomial equation

\[ ax^2 + bx + c = 0, \quad a \neq 0 \]

has two roots \( x_1 \) and \( x_2 \) if and only if the polynomial can be factorised into the following form

\[ ax^2 + bx + c \equiv a(x - x_1)(x - x_2). \]

Example

2. We have shown in Example 1 that equation \( 3x^2 + x - 2 = 0 \) has two roots \( x_1 = 2/3 \) and \( x_2 = -1 \). Thus we have

\[ 3x^2 + x - 2 \equiv 3 \left( x - \frac{2}{3} \right) \left( x - (-1) \right) \equiv (3x - 2)(x + 1) \]

which can easily be verified directly.

0.9 Root and Factor Equivalence

For an \( n \)-th order polynomial

\[ P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0, \quad a_n \neq 0, \]

\( x = x_0 \) is a root of the equation

\[ P_n(x) = 0 \]

if and only if \( P_n(x) \) has \( (x - x_0) \) as a factor, i.e., \( P_n(x) \) can be written as

\[ P_n(x) = (x - x_0)Q(x) \]

for certain polynomial \( Q(x) \).

Examples

3. Let a third order polynomial \( P(x) \) be given by

\[ P(x) \equiv 6x^3 - x^2 - 5x + 2. \]

Since it is easy to verify that \( x_0 = 1/2 \) is a root of \( P(x) = 0 \) because

\[ P(x_0) = P(1/2) = 6 \left( \frac{1}{2} \right)^3 - \left( \frac{1}{2} \right)^2 - 5 \left( \frac{1}{2} \right) + 2 = 0, \]

we expect that we will be able to write \( P(x) \) as \( P(x) = (x - 1/2)Q(x) \) for a polynomial \( Q(x) \). In fact this \( Q(x) \) can be shown to be \( Q(x) = 6x^2 + 2x - 4 \). Hence we have

\[ P(x) \equiv 6x^3 - x^2 - 5x + 2 \equiv \left( x - \frac{1}{2} \right) \left( 6x^2 + 2x - 4 \right). \]
4. From examples 2 and 3 we have \(6x^2 + 2x - 4 = 2(3x^2 + x - 2) = 6(x - 2/3)(x + 1)\) and hence

\[
P(x) \equiv 6x^3 - x^2 - 5x + 2 \equiv 6 \left(x - \frac{1}{2}\right) \left(x - \frac{2}{3}\right) (x + 1).
\]

Therefore the 3 factors \((x - 1/2), (x - 2/3)\) and \((x + 1)\) will give 3 roots of \(P(x)\). They are \(x_1 = 1/2, x_2 = 2/3\) and \(x_3 = -1\).

5. How did we find the factor \(Q(x)\) in example 3? We used the so-called \textit{long division}. It is basically a generalisation of the normal division process. So if we divide \(P(x) = 6x^3 - x^2 - 5x + 2\) by the factor \((x - 1/2)\) through the following long division

\[
\begin{array}{c|cc|}
   & 6x^2 & +2x & -4 \\
\hline
   x - \frac{1}{2} & 6x^3 & -x^2 & -5x & +2 \\
   & 6x^3 & -3x^2 & \\
   \hline
   & 2x^2 & -5x & \\
   & 2x^2 & -x & \\
   \hline
   & -4x & +2 \\
   & -4x & +2 \\
   \hline
   & 0 \\
\end{array}
\]

we obtain the factorisation

\[
6x^3 - x^2 - 5x + 2 = \left(x - \frac{1}{2}\right) (6x^2 + 2x - 4).
\]

0.10 Elimination Method for Linear Equations

Examples

6. Solve the linear equations

\[
x + 3y = 5, \quad 2x - y = -4.
\]

\textbf{Solution.} Rewrite the first equation \(x + 3y = 5\) as \(x = 5 - 3y\) and substitute it into the second equation \(2x - y = -4\), that is \(2(5 - 3y) - y = -4\).

Then the variable \(x\) in the 2nd equation is \textit{eliminated}, reducing the second equation to \(10 - 7y = -4\).

The solution of this reduced equation gives \(y = 2\) which, when substituted back to the first equation via \(x = 5 - 3y = 5 - 3 \times 2 = -1\), gives the solution for \(x\).

Hence the final solution is \(x = -1\) and \(y = 2\).
7. Solve the linear equations

\[
\begin{align*}
3x - 2y - 3z &= 10 \\
2x + 3y - 5z &= 1 \\
7x - y + 2z &= 7
\end{align*}
\]

**Solution.** First we need to choose to use one of the above three equations to eliminate one variable from the other two equations. Since the third equation looks “simpler”, we choose to use the third equation \(7x - y + 2z = 7\) to eliminate variable \(y\) in the other two equations. The choice \(y\) here is actually arbitrary, though we did so here because the coefficient in front of \(y\) there is the simplest. The third equation is thus converted to

\[
y = 7x + 2z - 7.
\]

Substitute the above into the other two equations, i.e.,

\[
\begin{align*}
3x - 2y - 3z &= 10, \\
2x + 3y - 5z &= 1
\end{align*}
\]

we see that they are reduced to

\[
\begin{align*}
3x - 2(7x + 2z - 7) - 3z &= 10, \\
2x + 3(7x + 2z - 7) - 5z &= 1
\end{align*}
\]

which can be further simplified to

\[
\begin{align*}
11x + 7z &= 4, \\
23x + z &= 22
\end{align*}
\]

We thus use the elimination procedure once again, this time similar to Example 6, and use \(z = -23x + 22\) to eliminate the variable \(z\) in \(11x + 7z = 4\) which is then reduced to \(11x + 7(-23x + 22) = 4\), i.e., \(-150x + 150 = 0\). Hence we obtain \(x = 1\). Substitute \(x = 1\) back to \(23x + z = 22\) we obtain \(z = -1\). Substitute \(x = 1\) and \(z = -1\) back to \(7x - y + 2z = 7\) we obtain \(y = -2\). It is perhaps worth noting that the solutions for each individual variables are in the opposite order of elimination. We first eliminated \(y\), then \(z\). So we first solve \(x\), then \(z\) and then \(y\). The final solution for the original problem is thus simply \(x = 1\), \(y = -2\) and \(z = -1\).

### 0.11 Binomial Expansions

The expansions for \((a + b)^n\) for the first few \(n\)’s are

\[
\begin{align*}
(a + b)^1 &= a + b \\
(a + b)^2 &= a^2 + 2ab + b^2 \\
(a + b)^3 &= a^3 + 3a^2b + 3ab^2 + b^3 \\
(a + b)^4 &= a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4
\end{align*}
\]
The first argument’s powers are decreasing from \( n \) to 0 (\( a^0 = 1 \)) while the second argument’s powers are increasing from 0 to \( n \). These formulae can be obtained “on the fly” by observing the following construction pyramid for the coefficients (a.k.a. Pascal’s triangle):

\[
\begin{array}{ccccccc}
(a+b)^0 & & & & & & 1 \\
(a+b)^1 & & & & 1 & 1 \\
(a+b)^2 & & 1 & 2 & 1 \\
(a+b)^3 & 1 & 3 & 3 & 1 \\
(a+b)^4 & 1 & 4 & 6 & 4 & 1 \\
(a+b)^5 & 1 & 5 & 10 & 10 & 5 & 1 \\
\vdots & & & & & & \vdots \\
\end{array}
\]

Notice that each entry in the pyramid is no more than the sum of the 2 elements above it, see for example \( 10 = 4 + 6 \).

## 0.12 Matrix Multiplications

We multiply matrices “row by column”:

### Examples

8. Let 2 \times 2 (i.e. 2 by 2) matrices \( A \) and \( B \) be given respectively by

\[
A = \begin{bmatrix} 2 & -1 \\ 5 & 3 \end{bmatrix}, \quad B = \begin{bmatrix} 7 & 4 \\ 1 & -2 \end{bmatrix}
\]

Find \( AB \) and \( A^2 \).

**Solution.**

\[
AB = \begin{bmatrix} 2 & -1 \\ 5 & 3 \end{bmatrix} \times \begin{bmatrix} 7 & 4 \\ 1 & -2 \end{bmatrix} = \begin{bmatrix} 2 \times 7 + (-1) \times 1 & 2 \times 4 + (-1) \times (-2) \\ 5 \times 7 + 3 \times 1 & 5 \times 4 + 3 \times (-2) \end{bmatrix} = \begin{bmatrix} 13 & 10 \\ 38 & 14 \end{bmatrix}
\]

\[
A^2 = \begin{bmatrix} 2 & -1 \\ 5 & 3 \end{bmatrix} \times \begin{bmatrix} 2 & -1 \\ 5 & 3 \end{bmatrix} = \begin{bmatrix} 2 \times 2 + (-1) \times 5 & 2 \times (-1) + (-1) \times 3 \\ 5 \times 2 + 3 \times 5 & 5 \times (-1) + 3 \times 3 \end{bmatrix} = \begin{bmatrix} -1 & -5 \\ 25 & 4 \end{bmatrix}
\]

9. Let 3 \times 3 matrices \( A \) and \( B \) be given by

\[
A = \begin{bmatrix} 2 & -1 & 13 \\ 5 & 3 & -6 \\ 11 & 0 & 10 \end{bmatrix}, \quad B = \begin{bmatrix} 7 & 4 & -8 \\ 1 & -2 & 0 \\ 9 & -4 & -3 \end{bmatrix}
\]
Then the matrix product $AB$ is

$$AB = \begin{bmatrix} 2 & -1 & 13 \\ 5 & 3 & -6 \\ 11 & 0 & 10 \end{bmatrix} \times \begin{bmatrix} 7 & 4 & -8 \\ 1 & -2 & 0 \\ 9 & -4 & -3 \end{bmatrix}$$

$$= \begin{bmatrix} 2 \times 7 + (-1) \times 1 + 13 \times 9 & 2 \times 4 + (-1) \times (-2) + 13 \times (-4) & 2 \times (-8) + (-1) \times 0 + 13 \times (-3) \\ 5 \times 7 + 3 \times 1 + (-6) \times 9 & 5 \times 4 + 3 \times (-2) + (-6) \times (-4) & 5 \times (-8) + 3 \times 0 + (-6) \times (-3) \\ 11 \times 7 + 0 \times 1 + 10 \times 9 & 11 \times 4 + 0 \times (-2) + 10 \times (-4) & 11 \times (-8) + 0 \times 0 + 10 \times (-3) \end{bmatrix}$$

$$= \begin{bmatrix} 130 & -42 & -55 \\ -16 & 38 & -22 \\ 167 & 4 & -118 \end{bmatrix}$$

**Please Read Me**

Discrete Mathematics is a solid unit for a first year student. Although the subject is not necessarily difficult, it does require students to spend a good deal of time in order to master every aspect of the material covered in this unit. There will of course be highs and lows from time to time when studying this unit. This is thus to remind you that some frustrations may indeed be just an anticipated integral part of studying new concepts and methods. A simple motto is that when confused or frustrated, take a rest and read the same subject once again in the next few days. New concepts take time to sink in; hastily looking for additional reference books (which are often more time-consuming or difficult to read) is not necessarily the best strategy. In enough time, all will fall into their rightful places.

Mathematics is usually presented in quite a dense format; much new information is packed into a small space. In addition there is new terminology to master plus some everyday words take on new meanings in this context. So reading mathematics is not like reading most other subjects. Thus if you don’t understand something the first time through, the problem is most likely not your fault. Mathematics usually needs to be read several times to gain understanding.

My advice is to briefly read through each section or subsection to get a feel for the topic then read through again, slowly and try to gain understanding in small sections. A third or fourth go may be necessary.

Remember that learning mathematics requires your active participation - to do is to learn. A technique that I have found useful, is to always read mathematics with a pen in hand and paper nearby. This is so that you can make calculations and try to fill in gaps between steps in a proof say. That is, read not only with your eyes but with your hands. This technique could come in handy if you decide to create your own glossary of terms - there is a lot of new terminology coming up.
1 Sets

A set is a collection of distinct elements. Given a set $X$ and an element $x$, the element can belong or not to the set. The fact that a set $X$ has an element $x$ is denoted by $x \in X$. If a set $X$ doesn’t have $x$ as one of its elements, then we denote this by $x \notin X$. Normally, we use capital letters to denote sets and we include the elements of a set in curly braces. A set can be defined

- by its properties, in the form of
  
  \[
  X = \{ \text{element_token} \mid \text{the properties the element_token has to satisfy} \},
  \]
  where ‘|’, or alternatively ‘:’, is a separator, e.g., $B = \{ n \mid n \text{ is an odd integer} \}$ or $B = \{ 2k + 1 : k \in \mathbb{Z} \}$

- by the enumeration of its elements, e.g., $B = \{ ..., -5, -3, -1, 1, 3, 5, 9, ... \}$

If $A$ and $B$ are sets, then $A$ is a subset of $B$ (denoted by $A \subseteq B$ or by $B \supseteq A$) means that all elements of $A$ are also elements of $B$. In other words,

\[
A \subseteq B = \{ \forall x \mid \text{if } x \in A \text{ then } x \in B \}
\]

One should read the symbol $\forall$ as for all and $\subseteq$ is called the set inclusion operator. If $A$ is not a subset of $B$, we write $A \nsubseteq B$ or $B \nsubseteq A$. Note that a subset $A$ of $S$ is often defined in the form $A = \{ x \in S \mid P(x) \}$, (or $A = \{ x \in S : P(x) \}$ if you so prefer), in which predicate $P(x)$ determines which $x$ in $S$ is to be accepted in $A$.

If $A$ is a subset of $B$, but $A$ is not equal to $B$, then $A$ is called a proper (or strict) subset of $B$. Some textbooks denote this by $A \subset B$ or $B \supset A$ and call set $B$ a superset of $A$. To represent sets and their relations, one can use Venn diagrams, e.g., $A \subset B$ can be pictured as:

![Figure 1: Sets inclusion](image)

As a notational convention, use double bar letters to represent some numerical sets:

- $\mathbb{C}$ is the set of all complex numbers,
- $\mathbb{R}$ is the set of all real numbers,
- $\mathbb{Q}$ is the set of all rational numbers,
- $\mathbb{Z}$ is the set of all integers,
- $\mathbb{N} = \{ n \in \mathbb{Z} \mid n \geq 0 \}$ is the set of all non-negative integers.
1.1 Basic Set Operations

Two sets $A$ and $B$ are said to be equal iff they have the same elements. In other words, all elements of $A$ are elements of $B$ and all elements of $B$ are elements of $A$,

$$A = B \iff A \subseteq B \text{ and } B \subseteq A$$

Let $A$ and $B$ be sets. Then

- the **union** of $A$ and $B$, denoted by $A \cup B$, is the set $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$.

- the **intersection** of $A$ and $B$, denoted by $A \cap B$, is the set $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$.

- the **difference** of $A$ and $B$ (or the **relative complement** of $B$ in $A$), denoted by $A \setminus B$ or $A \setminus B$, is the set $A \setminus B = \{x \mid x \in A \text{ and } x \notin B\}$.

The union, intersection and difference of sets $A$ and $B$ are shown respectively in Figure 2 below.

![Set Operations Diagram](image)

**Figure 2: Set operations**

**Examples**

These are particularly important in terms of understanding the notation used with sets. Make sure that you understand each part of them.

1. \{ rose, daffodil, carnation \} is a set of flower names.

2. Let $A = \{1, 2, 3, \{4\}, \{1, 2\}\}$ and $B = \{1, 2, \{4\}, 5\}$ be two sets. Then

   - (a) $2 \in A$ 2 is an element of $A$
   - (b) $4 \notin A$ 4 is not an element of $A$
   - (c) $\{4\} \in A$ $\{4\}$ is an element of $A$
   - (d) $\{4\} \notin A$ $\{4\}$ is not a subset of $A$ because the element 4 in $\{4\}$ is not an element in $A$
   - (e) $\{1, 2\} \subseteq A$ 1,2 are both elements of $A$
   - (f) $\{1, 2\} \in A$ $\{1, 2\}$ is an element of $A$
   - (g) $B \notin A$ 5 $\in B$ but 5 $\notin A$
   - (h) $A \cup B = \{1, 2, 3, \{4\}, 5, \{1, 2\}\}$ union of $A$ and $B$
   - (i) $A \cap B = \{1, 2, \{4\}\}$ intersection of $A$ and $B$
   - (j) $A \setminus B = \{3, \{1, 2\}\}$ set $A$ minus set $B$
   - (k) $B \setminus A = \{5\}$ set $B$ minus set $A
If all the sets we are interested in are subsets of an universal set \( U \), then we can define for each set \( A \) its complement 

\[
A' = \{ x \in U \mid x \notin A \}
\]

We note that some texts use \( \overline{A} \) or \( \complement A \) instead of \( A' \) to denote the complement of the set \( A \).

Hence 

\[
\complement A = \overline{A} = A' = U - A.
\]

Figure 3: Set complement

There exists an unique empty set, denoted by \( \emptyset \), which is a set with no elements. Obviously \( \emptyset \) is a subset of any set, \( \forall A \in U, \emptyset \subseteq A \).

If \( S \) is a set, then the power set of \( S \), denoted by \( \mathcal{P}(S) \), is the set of all subsets of \( S \), i.e.,

\[
\mathcal{P}(S) = \{ X \mid X \subseteq S \}
\]

If a set \( S \) has \( n \) elements then \( \mathcal{P}(S) \) has \( 2^n \) elements (do you know why?)

Two sets \( A \) and \( B \) are disjoint iff \( A \cap B = \emptyset \).

A collection of nonempty sets \( \{ A_1, \cdots, A_n \} \) is a partition of a set \( A \) iff

(i) \( A = A_1 \cup A_2 \cup \cdots \cup A_n \),

(ii) \( A_1, \cdots, A_n \) are mutually disjoint, \( A_i \cap A_j = \emptyset, \forall i \neq j \).

Figure 4: Set partition

Examples

3. The power set of \( A = \{ 1, 2, 3 \} \) is \( \mathcal{P}(A) = \{ \emptyset, \{ 1 \}, \{ 2 \}, \{ 3 \}, \{ 1, 2 \}, \{ 1, 3 \}, \{ 2, 3 \}, \{ 1, 2, 3 \} \} \).

4. Let \( A_1 = \{ 1, 2 \} \) and \( A_2 = \{ 3 \} \), then \( \{ A_1, A_2 \} \) is a partition of \( A = \{ 1, 2, 3 \} \) because \( A_1 \) and \( A_2 \) are disjoint and \( A_1 \cup A_2 = A \).

Let \( B_1 = \{ x \in \mathbb{R} \mid x < 0 \} \), \( B_2 = \{ x \in \mathbb{R} \mid x \geq 0 \} \) and \( B_3 = \{ x \in \mathbb{R} \mid x \leq 0 \} \). Then \( \{ B_1, B_2 \} \) is a partition of \( \mathbb{R} \) but \( \{ B_2, B_3 \} \) is not a partition because \( B_2 \cap B_3 \neq \emptyset \).

5. Let \( \mathbb{R} \) be taken as the universal set and let \( \mathbb{Q} \) be the set of rational numbers. Then the complement of \( \mathbb{Q} \), i.e. \( \mathbb{Q}' = \{ x \in \mathbb{R} \mid x \notin \mathbb{Q} \} \) is the set of irrational numbers.
For any given sets $A_1, \ldots, A_n$, the **Cartesian product** of $A_1, \ldots, A_n$, denoted by $A_1 \times A_2 \times \cdots \times A_n$, is the set of all ordered $n$-tuples $(a_1, \ldots, a_n)$ where $a_1 \in A_1, \ldots, a_n \in A_n$. In other words,

$$A_1 \times A_2 \times \cdots \times A_n = \{(a_1, \ldots, a_n) \mid a_i \in A_i, \ i = 1, 2, \ldots, n\}.$$ 

In the special case when all the $A_i$’s are equal to each other, then we can use the shorthand

$$A^n = A \times A \times \cdots \times A$$

**A function** (map) uniquely associates elements of one set with elements of another set. Formally, if $f$ is a function $f$ from $A$ to $B$, $f : A \to B$, then $\forall x \in A \ \exists y \in B$ such that $y = f(x)$ (uniquely associates every $x \in A$ with an element $f(x) \in B$). The symbol $f$ refers to the function itself, the set $A$ is called its domain and the set $B$ is called its range (codomain).

A function $f : A \to B$ is **onto** (surjective) if and only if $\forall y \in B, \exists x \in A$ such that $f(x) = y$ and $f$ is **one-to-one** (injective) if and only if whenever $f(x) = f(y)$, $x = y$ (equivalently: $\forall x, y \in A, x \neq y \Rightarrow f(x) \neq f(y)$). A **bijection** is both an onto and an one-to-one function (an injection and a surjection).

**Examples**

6. $\mathbb{R} \times \mathbb{R} = \mathbb{R}^2 = \{(x, y) \mid x, y \in \mathbb{R}\}$ represents a plane.

7. $\{1, 2\} \times \{3, 4, 5\} = \{(1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5)\}.$

8. $\{1, 2\} = \{2, 1\}$ but $(1, 2) \neq (2, 1)$. In general, $A \times B \neq B \times A$ because their elements are ordered pairs with the first component from the first set and the second element from the second set.

9. Suppose sets $A$ and $B$ are given respectively by $A = \{a, \ b, c\}$, $B = \{c\}$ . Give the Cartesian product $A \times B$ and the power set $\mathcal{P}(A)$ explicitly in terms of their elements.

**Solution.** The Cartesian product is $A \times B = \{(a, c), \ (b, c), \ (c, c)\}$ and the power set is $\mathcal{P}(A) = \{\emptyset, \ \{a\}, \ \{b, c\}, \ \{a, b, c\}\}$ . We note that $\{b, c\}$ is an element of the set $A$ although this element $\{b, c\}$ is itself a set.

Given a set $A$, the number of elements in $A$ is called the **cardinality** of $A$ and is denoted by $|A|$. That is, if $A$ is finite (has $n$ elements), then $|A| = n$.

Hence $|\emptyset| = 0$ and if, for instance, $A = \{3, 6, 9, x, y\}$, then $|A| = 5$.

We note that the symbol “$\mid$” here should not be confused with the normal symbol for taking an absolute value. Hence $|a|$ means the absolute value of $a$ if $a$ is a value, and $|A|$ means the cardinality of $A$ if $A$ is a set. An immediate and simple counting formula is that

$$|A \cup B| = |A| + |B| - |A \cap B|$$
1.2 Set Identities

holds for any finite sets \( A \) and \( B \).
For example, if \( A = \{3, 6, 9, x, y\} \) and \( B = \{-1, 6, y, 10, 11, 12\} \), then \(|A| = 5\), \(|B| = 6\), \(|A \cap B| = |\{6, y\}| = 2\), and \(|A \cup B| = |\{-1, 3, 6, 9, x, y, 10, 11, 12\}| = 9\).
Hence \(|A \cup B| = 9 = |A| + |B| - |A \cap B| = 5 + 6 - 2\).

1.2 Set Identities

**Theorem 1.** (1) Let \( A \), \( B \) and \( C \) be subsets of an universal set \( U \). Then

\[
\begin{align*}
S1. & \quad A \cup B = B \cup A \quad \text{commutativity} \\
& \quad A \cap B = B \cap A \\
S2. & \quad (A \cup B) \cup C = A \cup (B \cup C) \quad \text{associativity} \\
& \quad (A \cap B) \cap C = A \cap (B \cap C) \\
S3. & \quad A \cup (B \cap C) = (A \cup B) \cap (A \cup C) \quad \text{distributivity} \\
& \quad A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \\
S4. & \quad A \cup \emptyset = A \quad \text{“identity”} \\
& \quad A \cap U = A \\
S5. & \quad A \cup A' = U \quad \text{complementation} \\
& \quad A \cap A' = \emptyset \\
S6. & \quad (A \cup B)' = A' \cap B' \quad \text{De Morgan’s laws} \\
& \quad (A \cap B)' = A' \cup B' \\
S7. & \quad (A')' = A \quad \text{double complement} \\
S8. & \quad A \cup (A \cap B) = A \quad \text{absorption} \\
& \quad A \cap (A \cup B) = A \\
S9. & \quad A \cap A = A \quad \text{idempotent laws} \\
& \quad A \cup A = A \\
S10. & \quad A - B = A \cap B' \quad \text{set difference representation}
\end{align*}
\]

**Proof.** It’s often very useful to first draw an interpretation diagram whenever possible as it often provides good insights. Here we are only going to exemplify the proofs of the second of S1 and the first of S6, i.e.

\[
\begin{align*}
A \cap B & = B \cap A \quad (a) \\
(A \cup B)' & = A' \cap B' \quad (b)
\end{align*}
\]

and we leave the rest of them as an exercise for the astute reader.

(a) We need to show \( A \cap B \subseteq B \cap A \), and \( B \cap A \subseteq A \cap B \).

(i) \( \forall x \in A \cap B \), we have \( x \in A \) and \( x \in B \), and hence \( x \in B \cap A \). Thus \( A \cap B \subseteq B \cap A \).

(ii) \( \forall x \in B \cap A \), then \( x \in B \) and \( x \in A \). Hence \( x \in A \cap B \) and thus \( B \cap A \subseteq A \cap B \).
From (i) and (ii) we see (a) is valid.

(b) (i) We show first \((A \cup B)' \subseteq A' \cap B'\).

\[
\forall x \in (A \cup B)', \text{ we have } x \in U \text{ and } x \notin A \cup B. \text{ Hence } x \notin A \text{ and } x \notin B, \text{ which means by definition } x \in A' \text{ and } x \in B'. \text{ Thus } x \in A' \cap B' \text{ implying } (A \cup B)' \subseteq A' \cap B'.
\]

(ii) We need to show furthermore \(A' \cap B' \subseteq (A \cup B)\).

\[
\forall x \in A' \cap B', \text{ we have } x \in A' \text{ and } x \in B'. \text{ Hence } x \notin U-A \text{ and } x \notin U-B, \text{ implying } x \notin A \text{ and } x \notin B. \text{ We thus have } x \notin A \cup B. \text{ But since } x \in U, \text{ we conclude } x \in U - (A \cup B) = (A \cup B)' \text{. Therefore } A' \cap B' \subseteq (A \cup B)'\.
\]

From (i) and (ii) we see (b) is true.

**Example**

10. Let \(A, B\) and \(C\) be 3 sets. Prove the following set identities.

(a) \((A - B) \cap (C - B) = (A \cap C) - B\).

(b) \((A - B) \cup (B - A) = (A \cup B) - (A \cap B)\).

**Solution.** The proofs are the easiest if we make use of the set identities S1–S10 given in the above theorem, although direct proofs are equally acceptable.

(a)

\[
(A - B) \cap (C - B) = (A \cap B') \cap (C \cap B') \quad \text{S10}
\]

\[
= A \cap C \cap B' \cap B' \quad \text{S1 & S2}
\]

\[
= (A \cap C) \cap B' \quad \text{S9}
\]

\[
= (A \cap C) - B \quad \text{S10}
\]

(b)

\[
(A - B) \cup (B - A) = (A \cap B') \cup (B \cap A') \quad \text{S10}
\]

\[
= (A \cup (B \cap A')) \cap (B' \cup (B \cap A')) \quad \text{S3}
\]

\[
= ((A \cup B) \cap (A \cup A')) \cap ((B' \cup B) \cap (B' \cup A')) \quad \text{S5}
\]

\[
= (A \cup B) \cap (B \cap A') \quad \text{S5 & S6}
\]

\[
= (A \cup B) - (A \cap B) \quad .
\]

### 1.3 Russell’s Paradox

The precise definition of a set is actually quite involved: it has to be determined by a number of axioms. One of them, the axiom of comprehension (or separability), states that every formula defines a set. That is
Axiom 2 (comprehension). If $X$ is a set, $P$ is a property, then there exists a set $Y$ whose elements are precisely those of $X$ having property $P$.

Russell’s Paradox: A logical contradiction in set theory discovered by Bertrand Russell. If $R$ is the set of all sets which don’t contain themselves, does $R$ contain itself? If it does then it doesn’t and vice versa. For more details, see the book by Elizabeth J Billinton et al, Discrete Mathematics, Logic and Structures, Longman, 1990.

Proof.

We now show that $R = \{ x | x \notin x \}$ is not a set. If otherwise, then $S = \{ x \in R | x \notin x \}$ must also be a set according to the above axiom of comprehension. This is not possible for the following reasons.

(i) If $S \in R$ then $S \notin S$ by the definition of $R$. This then implies from the definition of $S$ that $S \in S$ because $S \in R$ and $S \notin S$, leading thus to a contradiction.

(ii) If $S \notin R$ then $S \in S$ by the definition of $R$. This then implies $S \in R$ from the definition of $S$, leading again to a contradiction.

Hence $S$ can’t be a set.

To conclude, Russel’s paradox “$R$ is neither an element of $R$ nor not an element of $R$” would be true only if $R$ where considered a valid set. But we have just shown that $R$ can not be a (valid) set, hence the paradox does not arise under the axiom of comprehension. ▲

Exercises

1. Let $A = \{a, 5, 6\}$ and $B = \{x, \emptyset\}$ be 2 sets. Write down $A \times B$ and $\mathcal{P}(B)$ explicitly in terms of their elements.

2. Let $A$, $B$ and $C$ be 3 sets. Draw a Venn diagram for $A \cap B \cap C$. From the Venn diagram, under what conditions can one draw the conclusion

$$(A - B) \cup (B - C) \cup (C - A) = A \cup B \cup C ?$$

3. Prove that the identity $(A \cap B)' = A' \cup B'$ holds for any two sets $A$ and $B$.

4. The number of ways of dividing $n$ objects into $k$ non-empty disjoint subsets is called a Stirling Number of the Second Kind and is denoted by $S_2(n, k)$. It is obvious that $S_2(n, k) = 0$, if $n < k$, and say that $S_2(0, 0) = 1$. Also, if $n > 0$, then $S_2(n, 0) = 0$, $S_2(n, 1) = S_2(n, n) = 1$. It can be proved that, for $n > k > 1$

$$S_2(n, k) = S_2(n - 1, k - 1) + kS_2(n - 1, k)$$

Find $S_2(5, 3)$ and $S_2(n, 2)$. 25
2 Mathematical Induction

A sequence

\[ a_m, a_{m+1}, a_{m+2}, \ldots \]

is an ordered set of elements, denoted by \( \{a_i\}_{i \in I} \) in which \( I \) is the set of indices, that is a set whose members index (label) members of another set. In other words, a sequence is a surjective function defined on an index set \( I \) and with values in any set \( A \).

For example, sequence \( a_0, a_1, a_2, \ldots \) may be written as \( \{a_i\}_{i \in \mathbb{N}} \) or \( \{a_i\}_{i \geq 0} \) or \( \{a_i; i \geq 0\} \). In the world of sequences, the \textbf{sigma notation} \( \sum \) provides a convenient way of writing sums of a partial or complete sequence while the \textbf{big-pi notation} \( \prod \) will denote the product of corresponding sequence. For example

\[
\begin{align*}
\sum_{j=2}^{4} (j^3 + j) &= (2^3 + 2) + (3^3 + 3) + (4^3 + 4) \\
\prod_{j=2}^{4} (j^3 + j) &= (2^3 + 2) \cdot (3^3 + 3) \cdot (4^3 + 4)
\end{align*}
\]

\[
\sum_{i=m}^{n} a_i = a_m + a_{m+1} + \cdots + a_n
\]

\[
\prod_{i=m}^{n} a_i = a_m \cdot a_{m+1} \cdots a_n
\]

\[
\sum_{k \in \mathbb{N}, k \geq 1} \frac{1}{k^2} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \cdots
\]

\[
\prod_{i=1}^{n} i = 1 \times 2 \times 3 \times \cdots \times n = n!
\]

\[
\sum_{n=1}^{\infty} \prod_{i=1}^{n} i = 1 + 2! + 3! + 4! + 5! + \cdots
\]

It is obvious that for \( m, n \in \mathbb{Z} \) with \( n \geq m \), one has

\[
\sum_{i=m}^{n} (a_i + b_i) = \sum_{i=m}^{n} a_i + \sum_{i=m}^{n} b_i
\]

\[
\sum_{i=m}^{n} c \cdot a_i = c \sum_{i=m}^{n} a_i
\]

\[
\prod_{i=m}^{n} (a_i \cdot b_i) = \left( \prod_{i=m}^{n} a_i \right) \cdot \left( \prod_{i=m}^{n} b_i \right)
\]

where \( c \) is any constant, i.e., \( c \) is independent of the dummy variable \( i \).
2.1 Principle of Mathematical Induction

Mathematical induction is the most powerful technique for verifying assertions in all of mathematics. It is based on the fact that any non-empty subset of the natural numbers set has a least element. This can be generalised as follows.

If $m$ is an integer, and $W$ is the set $W = \{m, m + 1, m + 2, \ldots\}$, then every nonempty subset of $W$ has a least element.

The following theorem is the basis for the mathematical induction proof technique.

**Theorem 3.** Let $m \in \mathbb{Z}$ and $W = \{m, m + 1, m + 2, \ldots\}$. Let $S$ be a nonempty subset of $W$ such that the following two conditions hold:

1. $m \in S$
2. whenever $k \in S$ then $k + 1 \in S$

Then $S = W$.

**Proof.** We will prove $S = W$ by contradiction, that is let us suppose $S \neq W$. Then $W - S$ is not empty and contains a least element $x$ because every nonempty subset of $W$ has a least element ($x \in W$ and $x \notin S$). The first condition (1) tells us that $m \in S$, so it must be the case that $x > m$. Thus $x - 1 \geq m$, and it follows that $x - 1 \in S$. Now we apply the second condition (2) to get that $(x - 1) + 1 \in S$, that is $x \in S$. This is a contradiction, since we cannot have both $x \in S$ and $x \notin S$ at the same time. Therefore $S = W$.

A **proposition** is a statement that is unambiguously true or false (even if we do not know which). A **sequence of propositions** is a rule (function) that associates with each integer $n$ (index) a proposition $S(n)$, also denoted by $S_n$.

**Theorem 4** (Principle of Mathematical Induction). If $m \in \mathbb{Z}$ and $\{S(n)\}_{n \geq m}$ is a sequence of propositions, then to prove that $S(n)$ is true for all integers $n \geq m$, perform the following two steps:

1. Prove that $S(m)$ is true.
2. Assume that $S(n)$ is true for arbitrary $n \geq m$. Prove that $S(n + 1)$ is true.

**Proof.** Let $W = \{n \mid n \geq m\}$ and let $S = \{n \mid n \geq m$ and $S(n)$ is true\}. Assume we have performed the two steps of Theorem 4. Then $S$ satisfies the hypothesis of Theorem 3, therefore $S = W$. So $S(n)$ is true for all $n \geq m$.

Roughly speaking, the first condition (1) in the Principle of Mathematical Induction (P.M.I.) ensures that $S_n$ is initially true, i.e., true when $n =$ the initial index $m$, while the second condition (2) ensures that a true statement $S_n$ for any $n \geq m$ will guarantee that at least the next statement $S_{n+1}$ is also true.
Examples

1. Suppose

(a) Cats can only give birth to cats, and
(b) Anubis is a cat.

Will Anubis ever have a dog as one of her future offspring?

Solution. Just about everyone will say “not possible” immediately. Why? This is because the P.M.I. is already in our natural mind without being noticed. However, let us see how the underlying logic works mathematically.

Let $S_n$ denote the statement that Anubis’ offsprings of the $n$–th generation are all cats. Then

(i) $S_0$ is true, because Anubis’ offsprings of 0–th generation consists of Anubis herself alone, who from (b) is a cat.

(ii) Assume $S_k$ is true for $k \geq 0$, i.e., the $k$–th generation of Anubis’ offsprings are all cats. Then, according to (a), all the children of this $k$–th generation can only be cats. That is the $(k+1)$–th generation of Anubis’ offsprings will all be cats, i.e., $S_{k+1}$ is true.

From the P.M.I., we conclude $S_n$ is true for all $n \geq 0$. Hence all Anubis’ offsprings will be cats. In other words, Anubis will never have a “dog” as one of her offsprings. It is perhaps surprising that just about everyone can subconsciously perform the above reasoning, with the use of the principle of mathematical induction, in a nip of time.

2. Prove inductively for all integers $n \geq 1$ that,

$$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}.$$  

Solution. Let $S_n$ denote the statement (*). Since the initial index for $n \geq 1$ is $n = 1$, we first verify that

(i) $S_1$ is true because

$$\text{l.h.s. of (*)} = 1, \quad \text{r.h.s. of (*)} = \frac{1(1+1)}{2} = 1 = \text{l.h.s.},$$

where l.h.s. (or LHS, L.H.S.) stands for left hand side and r.h.s. (or RHS, R.H.S.) stands for right hand side. Hence equality (*) holds for $n = 1$, meaning $S_1$ is true.

(ii) Assume $S_k$ is true for some $k \geq 1$, then by the definition of $S_k$, we have

$$1 + 2 + \cdots + k = \frac{k(k+1)}{2}.$$  

(**)
We now prove $S_{k+1}$ is also true. For $n = k + 1$,

\[\text{l.h.s. of } (***) = \left(\frac{1+2+\cdots+k+(k+1)}{2}\right) = \frac{k(k+1)}{2} + (k+1)\]

\[= \frac{(k+1)(k+2)}{2} = \text{r.h.s. of } (**),\]

i.e., $S_{k+1}$ is true.

We thus conclude from the P.M.I. that $S_n$ is true for all $n \geq 1$. This means (***) will hold for all integers $n \geq 1$.

3. Can you do the above example once again, but this time perhaps more informally?

**Solution.** Let us consider how to prove by induction the following statement

$$S_n : \sum_{j=1}^{n} j = \frac{1}{2} n(n+1). \quad (1)$$

First, we are happy with the first case (initial index $m = 1$) because $S_1$ is true as is readily verified via $\sum_{j=1}^{1} j = 1$ and $\frac{1}{2} 1 \cdot (1 + 1) = 1$. The main task is the step from the assumption of $S_k$ being true to the proof of $S_{k+1}$ being true. In other words, we, by assuming

$$\sum_{j=1}^{k} j = \frac{1}{2} k(k+1), \quad (2)$$

have to show

$$\sum_{j=1}^{k+1} j = \frac{1}{2} (k+1)[(k+1) + 1] \quad (3)$$

is true as well. But how can we prove (3) by perhaps the use of knowledge in the form of (2)? Well, the idea is to examine both sides of the statement in (3) to see if we can “dis-integrate” or “break” the expressions into the forms that are present in the existing knowledge (2), plus the “nasty bits”. For instance, the l.h.s. of (3)

$$\sum_{j=1}^{k+1} j \quad (4)$$

is “deceiving”: we can’t use the existing knowledge (2) on it directly. So we break the expression (4) into

$$\left[ \sum_{j=1}^{k} j \right] + (k+1) \quad (5)$$

in which the last term $(k+1)$ is the “nasty bit”, while the first expression is in the ready form for the use of (2). In general, one can not expect to prove a nontrivial statement for
2.1 Principle of Mathematical Induction AMTH140, Discrete Mathematics 2013

\[ n = k + 1 \] without using the assumption for \( n = k \). Thus we expect that we won’t be able prove the statement (3) without replacing part of its expression by using the assumption (2). Hence the purpose in establishing (5) for (4) and thus for the l.h.s. of (3) is that we can now use the knowledge (2). Since now we are able to break the l.h.s. of (3) into the form of (5), and the \( \sum_{j=1}^{k} j \) part in (5) can be replaced by the known (2), (thus successfully making use of the assumption 2: almost a must!), we can now hope that the outcome will be equal to the r.h.s. of (3) because we have used the assumed knowledge (2). (If this were not the case, then further this type of “breaking” would be needed.)

We are now ready to proceed. The l.h.s. of (3) is

\[
\sum_{j=1}^{k+1} j = \left[ \sum_{j=1}^{k} j \right] + \left[ \sum_{j=k+1}^{k+1} j \right] = \left[ \sum_{j=1}^{k} j \right] + (k + 1) .
\]

(6)

After replacing \( \sum_{j=1}^{n} j \) by \( \frac{1}{2}k(k + 1) \) due to (2), it reads

\[
\left[ \frac{1}{2}k(k + 1) \right] + (k + 1) = (k + 1)\left[ \frac{1}{2}k + 1 \right] = (k + 1)\left[ \frac{k + 2}{2} \right] = \frac{1}{2}(k + 1)(k + 2)
\]

This completes the proof that assumption (2) for \( n = k \) implies the result (3) for \( n = k + 1 \). Combining with the earlier validation of \( S_1 \), the statement \( S_n \) is proved by mathematical induction for any integer \( n \geq 1 \).

4. What would happen, if I were unlucky and looked at the expression (3) in example 3 on the right hand side?

**Solution.** If we are “unlucky”, no worries, at most just a bit extra work. Let us assume that we looked the other way and saw the r.h.s. of (3) first and we still want to prove it by making use of the induction assumption (2). The first step is to “break” that expression into containing forms that are identifiable with some of those in (2). But for expression like

\[
\frac{1}{2}(k + 1)\left[ (k + 1) + 1 \right] ?
\]

(7)

Yes, a bit tricky, but can still be done. Let us first examine the expression in (7). It is a second order polynomial in \( k \) with a leading coefficient \( \frac{1}{2} \). Let us now examine the expression in r.h.s. of (2) – it is also a second order polynomial in \( k \). So we can imagine

\[
\frac{1}{2}(k + 1)\left[ (k + 1) + 1 \right] = \frac{1}{2}k(k + 1) + “nasty~bits”,
\]

(8)

in which the “nasty bits” can be calculated as

\[
“nasty~bits” = \frac{1}{2}(k + 1)\left[ (k + 1) + 1 \right] - \frac{1}{2}k(k + 1)
\]

(9)

which is just (after simplification)

\[
(k + 1)
\]

(10)
Thus the r.h.s. of (3) is equal to
\[ \frac{1}{2}k(k + 1) + \text{“nasty bits”} \] (11)
in which the first part is \( \sum_{j=1}^{k} j \) due to (3) and second part is \((k + 1)\) due to (8) to (10). We thus finally have that the r.h.s. of (3) is equal to
\[
\left[ \sum_{j=1}^{k} j \right] + (k + 1) = \text{l.h.s. of (3)} \] (12)
and that (3) is proved through the use of (2) △

5. Prove through the use of the P.M.I. that the following identity
\[ 1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{n(n + 1)(2n + 1)}{6} \] (†)
holds for all positive integers \( n \).

**Solution.** Let \( S_n \) denote the statement (†). Then

(i) \( S_1 \) is true because

\[
\text{l.h.s. of (†)} = 1^2 = 1, \quad \text{r.h.s. of (†)} = \frac{1 \times (1 + 1)(2 \times 1 + 1)}{6} = 1
\]

implies l.h.s. = r.h.s., i.e. (†) is valid for \( n = 1 \).

(ii) Assume \( S_k \) is true for some \( k \geq 1 \), then

\[
1^2 + 2^2 + 3^2 + \cdots + k^2 = \frac{k(k + 1)(2k + 1)}{6} \] (‡)

Hence
\[
1^2 + 2^2 + \cdots + k^2 + (k + 1)^2 = \frac{(1^2 + 2^2 + \cdots + k^2) + (k + 1)^2}{k(k + 1)(2k + 1)} + (k + 1)^2
\]
from (‡)
\[
= \frac{(k + 1)}{6} \left[ (2k^2 + k) + (6k + 6) \right]
= \frac{(k + 1)(2k^2 + 7k + 6)}{6}
= \frac{(k + 1)(k + 2)(2k + 3)}{6}
= \frac{(k + 1)((k + 1) + 1)(2(k + 1) + 1)}{6}
\]
i.e. \( S_{k+1} \) is true.

Hence the principle of mathematical induction implies \( S_n \) is true for all integers \( n \geq 1 \).
2.2 Strong Principle of Mathematical Induction

The Principle of Mathematical Induction can also be equivalently represented in other forms. One of them is the following Strong Principle of M. I. (S.P.M.I.).

**Theorem 5** (Strong Principle of Mathematical Induction). Suppose \( m, m' \in \mathbb{Z} \) and for any integer \( n \geq m \), \( S_n \) is a statement.

(i) Prove that \( S_m, \ldots, S_m' \) with \( m' \geq m \) are all true.

(ii) Assume that \( S_m, \ldots, S_n \) are all true and assume \( n \geq m' \) and prove that \( S_{n+1} \) is true.

Then \( S_n \) is true for all \( n \geq m \).

**Proof.** S.P.M.I. implies P.M.I. is obvious. As for the converse one only needs to define \( T_n \) to be the statement that \( S_k \) for \( m \leq k \leq n \) are all true. Then the P.M.I. on \( T_n \) will imply the S.P.M.I. on \( S_n \). We shall however skip the simple details here.

**Examples**

6. Let \( n, r \in \mathbb{N} \) with \( r \leq n \). We define \( \binom{n}{r} \), called “\( n \) choose \( r \)” (binomial coefficients), to be the total number of \( r \)-element subsets that can be chosen from a set of \( n \) elements. Or, more simply, the number of ways of choosing \( r \) items from a list of \( n \) items. We thus have

(a) \( \binom{n}{0} = 1 \), \( \exists \) only 1 empty set

(b) \( \binom{n}{n} = 1 \), the only subset with all elements in it is the set itself

(c) \( \binom{n}{r} = \binom{n-1}{r} + \binom{n-1}{r-1} \). if \( r \geq 1 \), \( n > 1 \)

Let us briefly explain (c). Suppose \( S \) be a set of \( n \) elements and \( a \in S \) is any fixed element, then \( S = T \cup \{a\} \) if \( T = S \setminus \{a\} \).

To choose \( r \) elements from \( S \) [there are \( \binom{n}{r} \) different ways to do so], we can either

– choose \( r \) elements from \( T \) entirely [there are \( \binom{n-1}{r} \) different ways to do so], or

– choose \( a \) as one element and another \( r - 1 \) elements from \( T \) [there are \( \binom{n-1}{r-1} \) different ways to do so].

Hence (c) is valid.

Let us prove

\[
\binom{n}{r} = \frac{n!}{r!(n-r)!}, \quad n \geq r \geq 0.
\]
2.2 Strong Principle of Mathematical Induction

**Solution.** Recall the convention $0! = 1$. Let $S_n$ denote the following statement

$$\binom{n}{r} = \frac{n!}{(n-r)!r!}$$ is true for all $0 \leq r \leq n$.

(i) $S_0$ is true from either (a) or (b), because $\binom{0}{0} = \frac{0!}{0! \cdot 0!} = 1$.

(ii) Assume for $n \geq 0$, $S_k$ is true for all $0 \leq k \leq n$. Then from (c) for $1 \leq r \leq n$ we have

$$\binom{n+1}{r} = \binom{n}{r} + \binom{n}{r-1} = \frac{n!}{r!(n-r)!} + \frac{n!}{(r-1)!(n-r+1)!}$$

(from induction assumption)

$$= \frac{n!}{(r-1)!(n-r)!} \left[ \frac{1}{r} + \frac{1}{n-r+1} \right]$$

$$= \frac{(n+1)!}{r!(n+1-r)!}$$

As for $r = 0$ and $r = n + 1$, we see from (a) and (b)

$$\binom{n+1}{0} = \binom{n+1}{n+1} = 1 = \frac{(n+1)!}{(n+1)!0!}.$$ 

Hence $S_{n+1}$ is true. From (i) and (ii) and the S.P.M.I. (with $m = m' = 0$), we conclude $S_n$ is true for all $n \geq 0$.

7. For integer $n \geq 1$ show the following statement (often called binomial theorem or binomial expansion) is true

$$S_n: \quad (x+y)^n = \left(\begin{array}{c}n \\ 0\end{array}\right)x^n + \left(\begin{array}{c}n \\ 1\end{array}\right)x^{n-1}y + \left(\begin{array}{c}n \\ 2\end{array}\right)x^{n-2}y^2 + \cdots + \left(\begin{array}{c}n \\ n-1\end{array}\right)xy^{n-1} + \left(\begin{array}{c}n \\ n\end{array}\right)y^n$$

**Solution.**

(i) $S_1$ is true because

$$(x+y)^1 = x + y = \left(\begin{array}{c}1 \\ 0\end{array}\right)x + \left(\begin{array}{c}1 \\ 1\end{array}\right)y.$$ 

(ii) Assume $S_k$ is true for some $k \geq 1$, then

$$(x+y)^{k+1} = (x+y)\left((x+y)^k\right)$$

induction assumption

$$= (x+y)\left[\binom{k}{0}x^k + \binom{k}{1}x^{k-1}y + \cdots + \binom{k}{k}y^k\right]$$

$$= \binom{k}{0}x^k + \binom{k}{1}x^{k-1}y + \left[\binom{k}{1} + \binom{k}{2}\right]x^{k-2}y^2$$

$$+ \cdots + \left[\binom{k}{k-1} + \binom{k}{k}\right]xy^k + \binom{k}{k}y^{k+1}$$

from (a), (b), (c) of example 5

$$= \binom{k+1}{0}x^{k+1} + \binom{k+1}{1}x^ky + \cdots + \binom{k+1}{k+1}y^{k+1}$$

i.e., $S_{k+1}$ is true. Hence from the P.M.I. we conclude $S_n$ is true for all $n \geq 1$. 

33
8. Prove \(2n + 1 < 2^n, \quad n \geq 3\).

**Solution.** Let \(S_n\) be the proposition: \(2n + 1 < 2^n, \quad n \geq 3\).

For \(n = 3\), \(S_3\) is true because \(LHS = 2 \times 3 + 1 = 7, \quad RHS = 2^3 = 8, \quad 7 < 8\).

Assume \(S_n\) is true for some \(n \geq 3\), i.e., \(2n + 1 < 2^n, \quad n \geq 3\).

Then \(2(n + 1) + 1 = 2n + 1 + 2 < 2^n + 2 < 2^n + 2^n = 2^{n+1}\), hence \(S_{n+1}\) is true. Therefore \(S_n\) is true for all integers \(n \geq 3\).

9. Let \(x \in \mathbb{R}, x \neq 1\) and \(n \in \mathbb{N}\). Then \(1 + x + x^2 + \cdots + x^n = \frac{x^{n+1} - 1}{x - 1}\).

**Solution.** We will prove it by induction. If \(n = 0\), then both sides are 1. Assume the above formula is true for \(n\) and prove that it is true for \(n + 1\). Starting with the LHS, we have:

\[
1 + x + x^2 + \cdots + x^n + x^{n+1} = (1 + x + x^2 + \cdots + x^n) + x^{n+1}
\]

\[
= \frac{x^{n+1} - 1}{x - 1} + x^{n+1}
\]

\[
= \frac{x^{n+1} - 1 + (x - 1)x^{n+1}}{x - 1}
\]

\[
= \frac{x^{n+1} - 1}{x - 1}
\]

so the formula is true for \(n + 1\), and therefore it is true for all natural numbers \(n\).

**Exercises**

1. Prove by mathematical induction that

\[2^0 + 2^1 + \cdots + 2^n = 2^{n+1} - 1\]

holds for all integers \(n \geq 1\).

2. Prove by the use of P.M.I. that \(3^n > 4n\) holds for all integers \(n \geq 2\).

3. Suppose a sequence \(a_n\) is defined by the recurrence relation

\[a_0 = 1, \quad a_{n+1} = 2a_n + 1\]

for \(n = 0, 1, 2, 3, \ldots\). Prove by induction that \(a_n = 2^{n+1} - 1\) holds for all integers \(n \geq 0\).

4. Let \(F_{n+2} = F_{n+1} + F_n\), for \(n \geq 0\), \(F_0 = 0\), \(F_1 = 1\). Prove by induction that

\[
\begin{bmatrix}
1 & 1 \\
1 & 0
\end{bmatrix}^n = 
\begin{bmatrix}
F_{n+1} & F_n \\
F_n & F_{n-1}
\end{bmatrix}
\]
3 Efficiency, Big $O$

3.1 Nested Evaluation of a Polynomial

A monomial is
- a constant, or
- a mathematical expression involving positive integer powers in one or more variables and multiplied by a non-zero constant.

For example, $53$, $x$, $3x^2y$, $-2uv^2t^6$ are monomials, but $4 + x$, $5/y$, $43^x$, $7pq^{-2}$ are not. The variables part is called a term and the constant is called the coefficient. The degree of a monomial is the sum of the exponents of all included variables. Constants have the monomial degree of 0.

An algebraic sum of monomials is called a polynomial. The degree (order) of the polynomial is the greatest degree of its terms. For example, $3x^5 - x + 1$ is a polynomial of degree 5, $-y^7 + 5y^6 + y^2$ is a polynomial of degree 7, and $5x^2y^4 + 4xy^2 + x$ is a polynomial of degree 6.

A polynomial of degree $n$ in variable $x$ is an expression of the form

$$a_nx^n + a_{n-1}x^{n-1} + \cdots + a_1x + a_0,$$

where $a_0, a_1, \ldots, a_n$ are (real) constants.

The sum of two polynomials is obtained by adding together the coefficients sharing the same powers of variables (i.e., the same terms) so, for example,

$$(ax^2 + bx + c) + (dx + e) = ax^2 + (b + d)x + (c + e)$$

and has order less than or equal to the maximum order of the original two polynomials.

The product of two polynomials is obtained by multiplying term by term and combining the results, for example

$$(ax^2 + bx + c)(dx + e) = ax^2(dx + e) + bx(dx + e) + c(dx + e)$$

$$= adx^3 + (ae + bd)x^2 + (be + cd)x + ce$$

and has order equal to the sum of the orders of the two original polynomials.

A polynomial quotient

$$R(x) = \frac{P(x)}{Q(x)}$$

of two polynomials $P(x)$ and $Q(x)$ is known as a rational function. The process of performing such a division is called long division.
In mathematics, evaluation means to simplify an expression down to a single numerical value. There are often different ways to perform an evaluation, some may thus be better or more efficient than the others. For example, the direct term by term evaluation of polynomial

\[ f(x) = 2x^3 + 9x^2 + 5x - 1 \]

requires a total of 9 operations (6 multiplications + 3 additions/subtractions), while the evaluation of the same polynomial in the following form

\[ f(x) = x[(2x + 9) + 5] - 1 \]

requires only 6 operations (3 multiplications plus 3 additions/subtractions)

- first evaluate: \( 2x + 9 \), 2 operations
- then evaluate: \( x(2x + 9) + 5 \), 2 (more) operations
- then evaluate: \( x(x(2x + 9) + 5) - 1 \), 2 (more) operations

This method of evaluation is essentially Horner’s algorithm, also called the nested evaluation.

Incidentally, the reason that we often don’t differentiate additions and subtractions while counting multiplications separately is because multiplications are more expensive (in terms of computer time) than additions and subtractions and the latter two will take approximately the same amount of computer time.

In general for an \( n \)-th order polynomial

\[ P(x) = \sum_{i=0}^{n} a_i x^i = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 , \]

we can rewrite \( P(x) \) as

\[
\begin{align*}
P(x) &= x(a_n x^{n-1} + a_{n-1} x^{n-2} + \cdots + a_1) + a_0 \\
&= x(x(a_n x^{n-2} + \cdots + a_2) + a_1) + a_0 \\
&= \cdots \\
&= x \left[ x \cdots x(x(a_n x + a_{n-1}) + a_{n-2}) + \cdots \right] + a_1 + a_0
\end{align*}
\]

The above nested form or telescoping form, requires no more than \( n \) multiplications and \( n \) additions/subtractions.

For instance, to evaluate \( 3x^4 + 5x^3 + 2x + 6 \) by using Horner’s scheme, one needs only four multiplications, because \( 3x^4 + 5x^3 + 2x + 6 = x(x(x(3x + 5)) + 2) + 6 \).

It is obviously more efficient than the direct term by term evaluation (how many operations are needed in term by term evaluation?). In other words, the Horner’s algorithm is more efficient for polynomial evaluations.
3.2 Big $O$ Notation

When solving real life problems, we have to devise (to find) algorithms and then to implement them in a programming language to be run on computers. We often need to choose one out of a set of 'similar' ones. **Analysis of algorithms** is a branch of computer science that studies the performance of algorithms. Especially, it refers to the process of finding estimates for the time and space needed to execute the algorithm. Usually, the time is expressed as the number of steps and the space is expressed as the number of memory cells needed to execute the algorithm. That is, analysis of algorithms deals with functions that define the quantity of some resource consumed by a particular algorithm. We call such a function the **complexity of the algorithm** (sometimes the cost function of the algorithm). Quite often, the magnitude rather than the precise value is sufficient or is significant for these functions and we are interested in comparing our functions with well-known ones (polynomial, exponential, logarithmic, etc.) to find their upper or lower bounds. The big $O$ notation will serve well in this regard.

**Definition 6.** Let $D \subseteq \mathbb{R}$ and $f : D \to \mathbb{R}$ and $g : D \to \mathbb{R}$ be two (real-valued) functions. We say $f(x)$ is $O(g(x))$ ("big oh"), and we write $f(x) = O(g(x))$, if $\exists M \in \mathbb{R}$ and $\exists C > 0$ such that $\forall x \in D$,

$$|f(x)| \leq C|g(x)|, \quad \text{whenever } x \geq M.$$  

Notice that this is just a notation, that is

- the equal sign in the expression does not really denote mathematical equality, and
- $O(\cdot)$ does not really mean that $O$ is a mathematical function.

The diagram below (see Figure 5) depicts an intuitive interpretation of the above inequality: the curve $y = |f(x)|$ will always be below the curve $y = C|g(x)|$ after $x$ has passed the mark $x = M$. The values $M$ and $C > 0$ themselves are not important: the significance lies in the existence of such $M$ and $C$, rather than the magnitude of these values.

We note that if a property is to be established from **first principles**, then the property has to be derived or proved from the basic **definitions**. Hence if we are to show, for instance, $F(n) = O(G(n))$ from first principles for some given functions $F(n)$ and $G(n)$, then we have to derive it from the very definition of big $O$. In other words we need to find a $C > 0$ and an $M$ such that $|F(n)| \leq C|G(n)|$ holds whenever $n \geq M$. In establishing big $O$ properties, one typically has to make use of the following **triangle inequalities**

$$|a + b| \leq |a| + |b|, \quad |a + b| \geq |a| - |b|,$$

$$|x_1 + x_2 + \cdots + x_n| \leq |x_1| + |x_2| + \cdots + |x_n|, \quad |x_1 + x_2 + \cdots + x_n| \geq |x_1| - |x_2| - \cdots - |x_n|.$$  

Hence we have for example $|x - 2y| \leq |x| + |2y|$, and $|x + y - 13| \geq |x| - |y| - |13|$.

**Note.** Some texts use $f \in O(g)$ to denote that $f(x)$ is $O(g(x))$. The convenience of our notation however lies in the fact that we can use $O(g(x))$ to denote any quantity that is $O(g(x))$. However, both notations are acceptable as long as they are used consistently.
3.2 Big O Notation

**Examples**

1. Let \( f : \mathbb{N} \to \mathbb{R} \) be given by \( f(n) = 2n - 3 \). Show \( f(n) = O(n) \).

   **Solution.** Observe \( |f(n)| = |2n - 3| \leq |2n| + | - 3| \leq 2|n| + 3 \) if \( n \geq 3 \)

   \[ 2n + n = 3n \]

   i.e., \( |f(n)| \leq 3|n|, \quad \forall n \geq 3 \),

   where we have made use of the triangle inequality

   \[ |2n - 3| = |(2n) + (-3)| \leq |(2n)| + |(-3)| = |2n| + |3| \]

   By taking \( C = 3 \) and \( M = 3 \) (and \( D = \mathbb{N} \)), we see \( f(n) \) is \( O(n) \). We note that there are many different yet all valid choices of \( C \) and \( M \). For example,

   \[ |f(n)| \leq 2n + 3 |n| \leq 6 |n| \]

   i.e., \( |f(n)| \leq 5 |n|, \quad \forall n \geq 1 \)

   implies we can choose \( C = 5 \) and \( M = 1 \) in the definition of \( f(n) = O(n) \).

2. Show \( f(x) = \frac{3 \sqrt{x}(2x + 5)}{|x| + 1} \) is \( O(\sqrt{x}) \) for \( x \in \mathbb{R}^+ \). The set of positive real numbers is defined as \( \mathbb{R}^+ = \{ x \in \mathbb{R} | x > 0 \} \).

   **Solution.** The inequality

   \[ |f(x)| = \frac{3 \sqrt{x}(2x + 5)}{|x| + 1} \]

   \[ \leq \frac{3 \sqrt{x}(2x + 5)}{x} = 6 \sqrt{x} + \frac{15}{\sqrt{x}} \]

   assume \( x \geq 1 \)

   \[ \leq 6 \sqrt{x} + 15 \leq 6 \sqrt{x} + 15 \sqrt{x} = 21 \sqrt{x}, \]

   i.e., \( |f(x)| \leq 21 \sqrt{x} \) for \( x \geq 1 \), gives immediately \( f(x) = O(\sqrt{x}) \) (by choosing \( C = 21 \) and \( M = 1 \) in the definition).
Let $f(n) = \frac{1}{2}n(n + 5)$. Prove $f(n) = O(n^2)$.

Solution. We need to find constant $M$ and positive constant $C$ such that $|f(n)| \leq Cn^2$ for all $n \geq M$ ($|n^2| = n^2$). There are different ways to achieve this, and we give below the 2 most obvious ones.

(a) $|f(n)| = \frac{1}{2}n(n + 5)$

\[
= \frac{1}{2}n(n + 5) \leq \frac{1}{2}n(n + 5n) \quad \text{if assume } n \geq 1
\]

i.e. $|f(n)| \leq 3n^2$

We take in this case $M = 1$ and $C = 3$.

(b) $|f(n)| = \frac{1}{2}n(n + 5)$

\[
= \frac{1}{2}n^2 + \frac{5}{2}n
\]

\[
\leq \frac{1}{2}n^2 + \frac{1}{2}n^2 \quad \text{if assume } n \geq 5
\]

i.e. $|f(n)| \leq n^2$

We take in this case $M = 5$ and $C = 1$.

From (a) or (b) we see that $f(n) = O(n^2)$.

4. For $m \geq n \geq 0$, $x^n = O(x^m)$.

Solution. Try it yourself!.

3.3 Simple Features of $O$

Let $f$ and $g$ be mappings from $D \subseteq \mathbb{R}$ to $\mathbb{R}$, that is $f$, $g : D \to \mathbb{R}$. Then

- $f(x) = O(f(x))$, $|f(x)| = O(f(x))$, $c \cdot f(x) = O(f(x))$ (c is any constant).
  
For example, $n^2 = O(n^2)$.

- If $g(x) = O(f(x))$, then $f(x) + g(x) = O(f(x))$.

This is because $g(x) = O(f(x))$ implies the existence of constants $C > 0$ and $M$ such that $|g(x)| \leq C|f(x)|$, $\forall x \geq M$. Hence

\[
|f(x) + g(x)| \leq |f(x)| + |g(x)| \leq (C + 1)|f(x)|, \quad x \geq M
\]

which means $f(x) + g(x) = O(f(x))$. For example, $n^3 + 4n = O(n^3)$. 


If \( g(x) = O(f(x)) \), then \( f(x) + O(g(x)) = O(f(x)) \).

Since \( f(x) + O(g(x)) \) represents a quantity \( f(x) + h(x) \) such that \( h(x) = O(g(x)) \), we see from \( f(x) + h(x) = O(f(x)) \) that \( f(x) + O(g(x)) = O(f(x)) \).

For example, \( n^3 + 2n^2 + 5n = n^3 + (2n^2 + 5n) = n^3 + O(n^3) = O(n^3) \).

In order to make the full use of the notation \( O(f(x)) \), we need to assign an exact meaning to an identity in which \( O \) notations are used. Suppose \( F \) and \( G \) are two general expressions, with \( F \) containing \( O(f_1(x)), \cdots, O(f_m(x)) \) each exactly once and \( G \) containing \( O(g_1(x)), \cdots, O(g_n(x)) \) each exactly once, then the identity

\[
F(O(f_1(x)), \cdots, O(f_m(x)), \cdots) = G(O(g_1(x)), \cdots, O(g_n(x)), \cdots)
\]

means that

\[
\forall p_1(x), \cdots, \forall p_m(x) \text{ such that } p_i(x) = O(f_i(x)) \text{ for all } i = 1, \cdots, m, \\
\exists q_1(x), \cdots, \exists q_n(x) \text{ such that } q_j(x) = O(g_j(x)) \text{ for all } j = 1, \cdots, n,
\]

such that the identity

\[
F(p_1(x), \cdots p_m(x), \cdots) = G(q_1(x), \cdots q_n(x), \cdots)
\]

is held in the conventional sense. Obviously this extension is naturally consistent with the basic definition of \( f(x) = O(g(x)) \).

We hasten to remark that with above such definition, \( F(O(f_1), \cdots) = G(O(g_1), \cdots) \) is in general not same as \( G(O(g_1), \cdots) = F(O(f_1), \cdots) \). In other words the equality “=” is not the conventional one, but with the extended meaning.

**Example**

5. \( n = O(n) \) is obviously true. But the reversed identity \( O(n) = n \) is not true, because \( 2n \) is \( O(n) \) but is not equal to the r.h.s. \( n \).

6. If \( f_1(n) = O(g_1(n)) \) and \( g_2(n) \) is a function whose value is non-negative for integers \( n \geq 0 \), then \( f_1(n) \times g_2(n) = O(g_1(n) \times g_2(n)) \)

**Solution.** By definition, there are integers \( M_0 \) and \( C_0 \) such that \( |f_1(n)| \leq C_0 g_1(n) \) for \( n \geq M_0 \). Since \( g_2(n) \) is never negative,

\[
|f_1(n) \times g_2(n)| \leq |f_1(n)| \times |g_2(n)| \leq C_0 |g_1(n)| \times g_2(n) \leq C_0 |g_1(n)| \times |g_2(n)| \leq C_0 |g_1(n)| \times g_2(n)
\]

therefore \( f_1(n) \times g_2(n) = O(g_1(n) \times g_2(n)) \).
Exercises

1. How many multiplications have to be performed if one is to evaluate $2x^3 + 5x^2 + 3x + 1$ using Horner’s method?

2. Show $2n^3 - 9n + 1 = O(n^3)$.

3. Show $\frac{n + 1}{\sqrt{|n - 1|} + 1} = O(\sqrt{n})$.

4. Let $f(x)$ and $g(x)$ be 2 functions on $\mathbb{R}$ with $f(x) = O(g(x))$. Show that there exists a constant $M$ such that $|f(x)| \leq |x \cdot g(x)|$ holds for all $x \geq M$.

5. Prove that if $f(x) = O(g(x))$ and $g(x) = O(h(x))$, then $f(x) = O(h(x))$. 
4 Θ Notation

We recall that big $O$ is essentially a measurement for magnitude of functions at “infinity”. We say $f(x) = O(g(x))$ if the magnitude of $f(x)$ will be eventually bounded from above by a constant multiple of the magnitude of $g(x)$, i.e., there exist constants $C > 0$ and $M$ such that $|f(x)| \leq C|g(x)|$ for all $x \geq M$. Obviously it is the existence of such $C$ and $M$, rather than their actual values, which is significant.

It often turns out that upper bound by big $O$ is not precise enough. For example, to say $3n = O(n^2)$, while true, is not describing “$3n$” more precisely than one could have. On the other hand, $O(n)$ describes much better the magnitude of function $3n$ than say $O(n^2)$. For better distinction of such cases, we introduce below the so-called Θ notation.

4.1 Definition of the Θ Notation

**Definition 7.** Let $f$ and $g$ be real-valued functions over $S \subseteq \mathbb{R}$. We say $f(x) = \Theta(g(x))$ if and only if there exist constants $C > 0$, $D > 0$ and $M$ such that $orall x \in S$

$$D|g(x)| \leq |f(x)| \leq C|g(x)|, \quad \text{whenever } x \geq M.$$ 

We note that the inequality on the r.h.s., $|f(x)| \leq C|g(x)|$, gives an upper bound for the magnitude of function $f(x)$ while the inequality on the l.h.s., $D|g(x)| \leq |f(x)|$, gives a lower bound. Sometimes, we say that $f = \Omega(g(x))$ if $D|g(x)| \leq |f(x)|$. In other words, $f(x) = \Theta(g(x))$ iff $|f(x)|$ is “sandwiched” between two nonzero multiples of $|g(x)|$, that is $f$ is Big Oh and Big Omega of $g$.

**Examples**

1. Let $f: \mathbb{N} \to \mathbb{R}$ be given by $f(n) = 2n + 1$. Then
   (a) $f(n) = O(n^2)$
   (b) $f(n) = \Theta(n)$

**Solution.**

(a) $|f(n)| = |2n + 1| = 2n + 1 \leq 2n^2 + n^2 = 3n^2$ for $n \geq 1$ implies $f(n) = O(n^2)$ by the first principle, i.e., by the definition.

(b) We need to find the upper bound and the lower bound for the function $f(n)$. From

$$|f(n)| = |2n + 1| = 2n + 1 \quad \text{if } n \geq 1 \leq 3n, \quad |f(n)| = |2n + 1| = 2n + 1 \geq 2n \quad \text{i.e.,}$$

$$2n \leq |f(n)| \leq 3n \quad \text{for } n \geq 1, \quad \text{or } 2|n| \leq |f(n)| \leq 3|n| \quad \text{for } n \geq 1,$$

we easily conclude $f(n) = \Theta(n)$ with $D = 2$, $C = 3$, $M = 1$. 

2. Let \( f : \mathbb{N} \to \mathbb{R} \) be given by \( f(n) = \frac{n^2 - 1}{2n^2 + n + 1} \). Show

(a) \( f(n) = O(1) \)

(b) \( f(n) = \Theta(1) \)

Solution.

(a) 

\[
|f(n)| = \left| \frac{n^2 - 1}{2n^2 + n + 1} \right| \leq \frac{n^2}{2n^2} = \frac{1}{2} \leq \frac{1}{2} \cdot |1|,
\]

i.e., \(|f(n)| \leq \left( \frac{1}{2} \right) \cdot |1| \) for any \( n \geq 1 \). Hence \( f(n) = O(1) \) (with \( C = \frac{1}{2} \) and \( M = 1 \)).

(b) We already found an upper bound for \( f(n) \) in (a); we just need to find a lower bound in order to show \( f(n) = \Theta(1) \). Since \( n \geq 2 \) implies \( n - 1 \geq n/2 \) (prove it!), we have

\[
|f(n)| = \left| \frac{n^2 - 1}{2n^2 + n + 1} \right| \geq \frac{(n - 1)(n - 1)}{4n^2} = \frac{1}{8} \cdot \frac{n}{2} \cdot \frac{n}{2} = \frac{1}{8} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot |1|,
\]

i.e., \( \frac{1}{8} \cdot |1| \leq |f(n)| \) for \( n \geq 2 \). Hence, together with the upper bound found in (a), we have

\[
\frac{1}{8} \times |1| \leq |f(n)| \leq \frac{1}{2} \times |1|, \quad n \geq 2,
\]

or \( \frac{1}{8} |g(n)| \leq |f(n)| \leq \frac{1}{2} |g(n)| \) for \( g(n) \equiv 1 \), with \( M = 2, C = \frac{1}{2} \) and \( D = \frac{1}{8} \).

According to the definition, we thus have \( f(n) = \Theta(g(n)) \), i.e. \( f(n) = \Theta(1) \).

Though the above 2 examples have very short proofs, it’s not uncommon for someone new to the subject to feel completely lost. So in what follows we’ll explain several examples in painstaking details.

### 4.2 Examples in Exaggerated Details

#### Examples

3. What on earth is the big “\( O \)”?

Solution. Big \( O \) is essentially a measurement for the magnitude of functions at “infinity”, and tells roughly who is “bigger”. We say function \( f(n) \) is of order of big \( O \) of function \( g(n) \), i.e. \( f(n) = O(g) \), if the magnitude of \( f(n) \) will be eventually bounded from above by a constant multiple of the magnitude of \( g(n) \). Magnitude \(|f(n)|\) bounded by a constant multiple of that of \( g \) means

\[
|f(n)| \leq C \cdot |g(n)| \quad (\ast)
\]

for a constant \( C > 0 \), while “eventually” means \((\ast)\) needs only to be valid for the “later” \( n \)’s. In other words, \((\ast)\) needs only to be true for \( n \geq M \) for a constant \( M \). We note that
what the actual values the positive constant $C$ and the constant $M$ take is not important, what is important is that once a $C$ is chosen, it has to remain unchanged in (*) for all $n \geq M$. This is like comparing the potential weight (magnitude) of a cat and a dog: we have to wait until they have fully grown up ($n \geq M$), but exactly when (i.e. the value of $M$) is not really important.

4. Let $f : \mathbb{N} \to \mathbb{R}$ be given by $f(n) = \frac{n^3 + 2n^2 - 1}{n + 1}$. Show in very rudimentary details how we can prove $f(n) = \Theta(n^2)$.

**Solution.** From the definition of $\Theta$ notation, we have to establish an upper bound and lower bound, corresponding to the second and the first inequality relation in the formula below

$$D \cdot |n^2| \leq |f(n)| \leq C \cdot |n^2|$$

To establish an upper bound is to find some positive constants $C$ and $M$ such that

$$\left| \frac{n^3 + 2n^2 - 1}{n + 1} \right| \leq C \cdot n^2 \quad (13)$$

is true at least for $n \geq M$. But how can we do this? And in what general direction should we proceed? We observe that on the r.h.s. of (13) it is a just a single term, while the l.h.s. of (13) is a polynomial fraction.

We will now try to “move” from the expression

$$\frac{n^3 + 2n^2 - 1}{n + 1} \quad (14)$$

to the expression

$$C \cdot n^2 \quad (15)$$

gradually, in terms of the “$\leq$” sign. In such “moves”, we will keep some terms intact, dump some terms and replace some terms. The general rule is that one should always keep the dominant (leading) terms, while throwing away or manipulating other non-dominant terms in an expression. Back in expression (14), we see that the dominant term in the numerator $n^3 + 2n^2 - 1$ is $n^3$ while the the dominant term in the denominator $n + 1$ is $n$. So we are happy to throw away (legitimately) the term $-1$ in the numerator to arrive at a simpler form (which is closer to the final target in (15). We thus get the first “move”

$$\frac{n^3 + 2n^2 - 1}{n + 1} \leq \frac{n^3 + 2n^2}{n + 1} \quad (16)$$

Because the numerator and denominator of the result

$$\frac{n^3 + 2n^2}{n + 1} \quad (17)$$
of the first “move” are both non-negative, we throw away the “1” in the denominator of (17), because the less the denominator, the larger the total fraction. Hence our 2nd “move” is

\[
\frac{n^3 + 2n^2}{n + 1} \leq \frac{n^3 + 2n^2}{n} = \frac{n(n^2 + 2n)}{n} = n^2 + 2n
\]

(18)

The newly obtained result after the 2nd move is thus

\[
n^2 + 2n
\]

(19)

which is even closer to (15). How do we make the last “move”? We observe that (19) contains inhomogeneous terms (i.e., different powers) while (15) contains only an homogeneous term. So our next step is to “promote” legitimately other orders to the same leading order \(n^2\). In doing so, we notice that

\[
2n \leq 2n^2
\]

when \(n \geq 0\), hence our third “move” gives

\[
n^2 + 2n \leq n^2 + 2n^2 = 3n^2.
\]

Putting all the above “moves” together, we obtain for \(n \geq 0\)

\[
\frac{n^3 + 2n^2 - 1}{n + 1} \leq \frac{n^3 + 2n^2}{n + 1} \leq \frac{n^3 + 2n^2}{n} = \frac{n(n^2 + 2n)}{n} = n^2 + 2n \leq n^2 + 2n^2 = 3n^2
\]

(20)

Since value of (14) is non-negative if \(n \geq 1\), we see that (20) implies

\[
\frac{n^3 + 2n^2 - 1}{n + 1} \leq \frac{n^3 + 2n^2}{n + 1} \leq \frac{n^3 + 2n^2}{n} = \frac{n(n^2 + 2n)}{n} = n^2 + 2n \leq n^2 + 2n^2 = 3n^2
\]

(21)

is true for all \(n \geq 1\). In other words, (13) is true if we choose \(C = 3\) and \(M = 1\) \((n^2 = |n^2|)\).

Now we proceed to establish a lower bound for \(f(n)\). We need to find positive constant \(D\) and constant \(L\) such that

\[
\frac{n^3 + 2n^2}{n + 1} \geq D \cdot n^2
\]

(22)

for all \(n \geq L\). If we can establish (22), then along with the already established (13), we see that

\[
D \cdot n^2 \leq \frac{n^3 + 2n^2 - 1}{n + 1} \leq C \cdot n^2
\]

(23)

is true for \(n \geq \tilde{M} = \max(M, L)\).

To derive (22), we proceed similarly. First let us throw away some non-significant terms \textit{legitimately}, by keeping the dominant ones and assuming \(n \geq 1\) (thus (14) is non-negative). Our first “move” is to throw away the term \(2n^2\) in the numerator of (14). Notice that our new “moves” are for the “\(\geq\)” sign, not the “\(\leq\)” as in the previous case.
Thus we can’t throw way the “$-1$” in the numerator and “$1$” in the denominator as they would break the sought “$\geq$” sign. Hence after the first “move”, we obtain

$$\frac{n^3 + 2n^2 - 1}{n + 1} \geq \frac{n^3 - 1}{n + 1}$$

(24)

Next, let us promote the (non-leading) “$1$” in the denominator to arrive at an homogeneous expression, i.e., we need to promote the “$1$” into something like “constant $\cdot n^3$”. This turns out easy as $1 \leq n$ thus $\frac{1}{n+1} \geq \frac{1}{n+n}$ for $n \geq 1$. (If $a < b$ then $\frac{1}{a} > \frac{1}{b}$).

Hence our second “move” gives

$$\frac{n^3 - 1}{n + 1} \geq \frac{n^3 - 1}{n + n} = \frac{n^3 - 1}{2n}$$

(25)

Observe the r.h.s. of (25), we see that we need to promote the “$-1$” in the numerator to the form of “constant $\cdot n^3$” to ensure homogeneity. In other words, we need to establish something like

$$-1 \geq \text{constant} \cdot n^3$$

(26)

It would be obviously true if we choose the constant in (26) to be just say $-2$ and ask for $n \geq 1$. But this would inflict out-of-proportion damages on our leading term $n^3$. So we have to choose the constant carefully. Since $n^3$ is eventually much larger than the lower order term $-1$, we can take out a portion of $n^3$, say $\frac{1}{4}n^3$ without affecting the magnitude order, and use it to compensate the “$-1$”. In other words,

$$n^3 - 1 = \frac{3}{4}n^3 + \left[\frac{1}{4}n^3 - 1\right]$$

(27)

If we choose $n$ large enough, then the terms in the square brackets in (27) will be non-negative, and then we can legitimately throw that away because we are establishing a “$\geq$” relation. To ensure

$$\left[\frac{1}{4}n^3 - 1\right] \geq 0$$

(28)

we need $n^3 \geq 4$ or $n \geq 4^{\frac{1}{3}}$. Hence if we choose $n \geq 2$, it will be enough to guarantee the non-negativeness in (28). Hence our third “move” is

$$n^3 - 1 = \frac{3}{4}n^3 + \left[\frac{1}{4}n^3 - 1\right] \geq \frac{3}{4}n^3, \quad \text{for } n \geq 2$$

or

$$\frac{n^3 - 1}{n} \geq \frac{3}{4}n^2, \quad \text{for } n \geq 2$$

(29)

Putting “moves” 1–3 together, we obtain

$$\left|\frac{n^3 + 2n^2 - 1}{n + 1}\right| \geq \frac{n^3 - 1}{n + 1} \geq \frac{n^3 - 1}{n + n} = \frac{n^3 - 1}{2n} \geq \frac{1}{2} \times \frac{3}{4}n^2, \quad \text{for } n \geq 2$$

(30)
4.2 Examples in Exaggerated Details

Notice that (30) and (21) can be put together as
\[
\frac{3}{8} n^2 \leq \left| \frac{n^3 + 2n^2 - 1}{n + 1} \right| \leq 3n^2, \quad \text{for } n \geq 2 ,
\]
and \( n^2 = |n^2| \), which implies \( f(n) = \Theta(n^2) \) with \( D = \frac{3}{8}, C = 3, M = 2 \).

5. Redo example 4 in such a (recommended) way that the proof will be both concise and mathematically acceptable.

Solution. To find an upper bound we first observe
\[
\left| \frac{n^3 + 2n^2 - 1}{n + 1} \right| = \frac{n^3 + (2n^2 - 1)}{n + 1} \quad \text{assume } n \geq 1 \text{ (thus } 2n^2 - 1 \geq 0 \text{ and } f(n) \geq 0) \leq \frac{n^3 + 2n^2}{n + 1} \quad \text{increase the numerator} \leq \frac{n^3 + n^3}{n + 1} \quad \text{assume furthermore } n \geq 2 \text{ (thus } 2n^2 \leq n^3) \leq 2n^2 .
\]
The inequality above was due to the (legitimate!) increase of the numerator and the decrease of the denominator, it could be done so because we were looking for an upper bound. Obviously we won’t allow in this case a decrease of the numerator or an increase of the denominator. To find a lower bound, incidentally, the opposite is true. We have thus so far shown
\[
|f(n)| \leq 2n^2, \quad \text{whenever } n \geq 2 .
\]

For a lower bound, we observe
\[
\left| \frac{n^3 + 2n^2 - 1}{n + 1} \right| = \left| \frac{n^3 + (2n^2 - 1)}{n + 1} \right| \leq \frac{n^3 + (2n^2 - 1)}{n + 1} \quad \text{if } n \geq 1 \geq \frac{n^3}{n + 1} \quad \text{because } 2n^2 - 1 \geq 0 \geq \frac{n^3}{n + n} \quad \text{legitimate increase of denominator} = \frac{1}{2} n^2 ,
\]
which means
\[
|f(n)| \geq \frac{1}{2} n^2, \quad \text{whenever } n \geq 1 .
\]

Hence, by putting together the upper bound and the lower bound, we have
\[
\frac{1}{2} n^2 \leq |f(n)| \leq 2n^2, \quad \text{whenever } n \geq 2 .
\]
Once again, don’t forget that $0 < n^2 = |n^2|$. This in the definition of $\Theta$ notation corresponds to $C = 2$, $D = \frac{1}{2}$ and $M = \max(1, 2) = 2$ where, for any values $a$ and $b$, $\max(a, b)$ denotes their maximum value. Hence we finally conclude $f(n) = \Theta(n^2)$.

6. Show that $\log_2 n = O(n)$ for positive integer $n$.

**Solution.** We first prove by induction the statement

$$S_n : \quad \log_2 n < n$$

for $n \geq 1$. Obviously $S_1$ is true because l.h.s. = $\log_2 1 = 0 < 1 = $ r.h.s. Before we embark on the inductive step, we observe that $\log_2 n$ is an increasing function in the sense that $\log_2 m < \log_2 n$ whenever $m < n$ (and $m, n > 0$). This is true for any logarithm to base $> 1$. Assume $S_k$ is true for an integer $k \geq 1$, i.e., $\log_2 k < k$, then

$$\log_2(k + 1) \leq \log_2(k + 1) = \log_2(2k) = \log_2 2 + \log_2 k = 1 + \log_2 k < 1 + k,$$

i.e., $\log_2(k + 1) < k + 1$. Hence $S_{k+1}$ is also true. From the P.M.I. we have thus shown that $S_n$ is true for all integers $n \geq 1$. Since $\log_2 n \geq 0$ for $n \geq 1$ implies $|\log_2 n| = \log_2 n \leq n = |n| = 1 \cdot |n|$ for $n \geq 1$, we finally conclude $\log_2 n = O(n)$, with $C = 1$ and $M = 1$.

7. Show $n^2 \neq O(n)$.

**Solution.** If otherwise, i.e., $n^2 = O(n)$, then according to the definition of big $O$ there would exist a positive constant $C$ and another constant $M$, such that $|n^2| \leq n|n|$ whenever $n \geq M$. But this is not possible because, if we choose an $n_0$ such that $|n_0| > \max(C, M)$, we would have $|n_0|^2 > C|n_0|$ which contradicts the requirement $|n^2| \leq C|n|$ whenever $n \geq M$. Hence the earlier assumption $n^2 = O(n)$ that leads to this requirement must be incorrect. Thus $n^2 \neq O(n)$. This is a proof by contradiction.

**Note.** From the above examples we see easily that the choice of constants $C$, $D$ and $M$ may vary and their values often depend on the different approaches we may choose.

**Exercises**

1. Show $2n^3 + 9 \log_2 n + 1 = \Theta(n^3)$.

2. Show $\frac{n + \sin n}{\sqrt{|n|} + 1} = \Theta(\sqrt{n})$.

3. Suppose $f(n)$ and $g(n)$ are two functions defined on $\mathbb{N}$ such that $f(n) = \Theta(g(n))$ and $f(n) \neq 0$ for all $n \in \mathbb{N}$. Show that $nf(n) \neq O(g(n))$. 

5 Analysis of Algorithms

5.1 Floor and Ceiling Functions

Before going further, we first introduce two useful functions defined on the set of real numbers and taking values in the set of integers that relate any real number to its integer neighbours,

\[
\lfloor \cdot \rfloor, \lceil \cdot \rceil : \mathbb{R} \to \mathbb{N}
\]

For any \( x \in \mathbb{R} \),

- **the floor of** \( x \), denoted by \( \lfloor x \rfloor \), is the greatest integer not exceeding \( x \) (the greatest integer that is less than or equal to \( x \)), e.g., \( \lfloor 3.2 \rfloor = 3 \), \( \lfloor -4.5 \rfloor = -5 \), and \( \lfloor -4 \rfloor = -4 \).

- **the ceiling of** \( x \), denoted by \( \lceil x \rceil \), is the smallest integer not less than \( x \) (the least integer that is greater than or equal to \( x \)), e.g., \( \lceil 3.2 \rceil = 4 \), \( \lceil -4.5 \rceil = -4 \), and \( \lceil -4 \rceil = -4 \).

![The floor and ceiling functions](image)

Figure 6: The floor and ceiling functions

It is obvious from the definitions that for any \( x \in \mathbb{R} - \mathbb{Z} \) and \( n \in \mathbb{N} \)

\[
\begin{align*}
\lfloor x \rfloor &\leq \lfloor x \rfloor + n \\
\lfloor x + n \rfloor &\leq \lfloor x \rfloor + n \\
\lfloor -x \rfloor &\leq \lfloor -x \rfloor + n \\
\lfloor x \rfloor + \lceil x \rceil &\leq 1 \\
\lfloor n \rfloor &\leq n \\
\lceil n \rceil &\leq n \\
\lfloor x + n \rfloor &\leq \lfloor x \rfloor + n \\
\lfloor x \rfloor + \lceil x \rceil &\leq 1 \\
\lfloor x \rfloor + \lceil -x \rceil &\leq 0
\end{align*}
\]
5.2 Searching Algorithms

An algorithm is a set of instructions that can be mechanically executed to produce an unique output from a valid input in a finite amount of time or steps. Given an algorithm, one often needs to determine such things as the cost of computer implementation and the efficiency of the algorithm.

One important algorithm is one that does a search of a list for a particular item. Suppose we are given a finite ordered list of items \( I(1), \ldots, I(n) \). Then for any item named KEY, we look through the given list to see if there is a match in the list. Report the position \( k \) at which item \( I(k) \) matches KEY. Otherwise report a phantom position “−1” which is thus used to denote that the KEY is not found.

<table>
<thead>
<tr>
<th>Input: KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
</tr>
<tr>
<td>Output: First position ( k ) in the list of which ( I(k) ) matches KEY, otherwise output “−1”</td>
</tr>
</tbody>
</table>

In the following we shall examine two search algorithms for the above problem, the binary search and the sequential search, and show that, in general, the former is a more efficient one. For simplicity, the efficiency of a searching algorithm will always be measured according to the total number of comparisons involved.

5.3 Sequential Search

The simplest (brute force) algorithm to search a list for a given key is to test the key against each element in the list. We search successively the list from the first item to the second, then to the third and so on to locate the position of the matched item. For a given item, KEY, the algorithm is as follows

---

**Algorithm 5.1 Sequential Search**

\[
\text{for } k = 1 \text{ to } n \text{ do}
\]

\[
\text{if } \text{KEY} = I(k) \text{ then}
\]

\[\text{return } k\]

\[
\text{end if}
\]

\[
\text{end for}
\]

\[\text{return } -1 \text{ ("KEY not found")}
\]
We note that the last step will not be executed if the execution of the algorithm is terminated in step 1. Usually, we output the phantom position “−1” to indicate no match has been found (in contrast to a positive number \(k\), returned when a match was found).

This works correctly regardless of the order of the elements in the list. Let \(S(n)\) be the number of comparisons needed in a worst case to complete the sequential search in a list of \(n\) items. Since the worst case is when either the searched item is not found or it is seated at the last position, we have in both cases \(S(n) = n\).

### 5.4 Binary Search

Binary search is a divide-and-conquer strategy. For an ordered list of \(n\) items, the general strategy of binary search is to compare a given KEY with the item at the middlemost position, i.e., at the \(\lfloor \frac{n+1}{2} \rfloor\)–th position. Then report this position if the items are matched. If not, we only need to search one of the two (halved) sublists broken up at the middlemost point (remember: the list is sorted!). Which sublist is to be continued for the search depends on whether the given KEY is after or before the item at the middlemost position. Continue the binary search on the correct sublist until either a match is found, or the latest sublist is reduced to an empty list. It is easy to see that binary search has a great deal in common with how we would search for a word in a dictionary.

Let us recall that if a list \(a_m, a_{m+1}, \cdots, a_n\) is indexed from \(m\) to \(n\), then the middlemost position is at \(k = \lfloor \frac{m+n}{2} \rfloor\). Suppose a given list \(I(1), \cdots, I(n)\) is sorted in the increasing order, then the binary search algorithm can be rephrased as follows. Here (….) indicate some comments.

**Algorithm 5.2 Binary Search**

\[
\begin{align*}
F &= 1 \quad \text{(first index, last index)} \\
L &= n \\
\text{while } F \leq L \text{ do} \\
\quad k &= \left\lfloor \frac{F + L}{2} \right\rfloor \quad \text{(middlemost position)} \\
\quad \text{if } KEY = I(k) \text{ then} \\
\qquad \text{return } k \quad \text{(success!)} \\
\quad \text{else if } KEY > I(k) \text{ then} \\
\qquad F &= k + 1 \quad \text{(keep the second half of the list)} \\
\quad \text{else} \\
\qquad L &= k - 1 \quad \text{(keep the first half of the list)} \\
\quad \text{end if} \\
\text{end while} \\
\text{return } -1 \quad \text{(output the phantom position)}
\end{align*}
\]
Example

1. Find 13 from the ordered list \(\{1, 3, 5, 7, 9, 13, 15, 17, 19\}\) with both the binary search and the sequential search. Give the number of comparisons needed in both cases.

**Binary search.** 3 comparisons are needed as can be seen from the diagram below

**Sequential search.** 6 comparisons are needed in this case, see

5.5 Complexity of Binary Search

Let \(B(n)\) be the maximum number of comparisons the binary search needs to complete the search for an ordered list of \(n\) items. Then for \(m, n \in \mathbb{N}\) with \(m \geq 1\),

- (i) \(B(1) = 1\)
- (ii) \(B(m) \leq B(n)\) if \(m \leq n\)
- (iii) \(B(2m) = 1 + B(m)\) see: \(* \cdots * * \cdots *\)
- (iv) \(B(2m + 1) = 1 + B(m)\) see: \(* \cdots * * \cdots *\)
Hence we have

\[ B(2^k) = B(2^{k-1}) + 1 = B(2^{k-2}) + 2 = \ldots = B(2^0) + k = k + 1, \quad \text{for } k \geq 0. \]

For any \( n \in \mathbb{N} \) with \( n \geq 1 \), \( \exists k \in \mathbb{N} \) such that \( 2^k \leq n < 2^{k+1} \) (every natural number sits between two consecutive powers of 2). Taking a \( \log_2 \) on both sides of the inequality we obtain \( k \leq \log_2 n < k + 1 \), see Preliminary Mathematics at the beginning of these notes for a quick reminder of powers and logarithms. Hence

\[ k = \lfloor \log_2 n \rfloor \iff 2^k \leq n < 2^{k+1}. \]

From (ii) we thus obtain

\[ B(2^k) \leq B(n) \leq B(2^{k+1}) \]

which gives \( k + 1 \leq B(n) \leq k + 2 \), or simply

\[ \lfloor \log_2 n \rfloor + 1 \leq B(n) \leq \lfloor \log_2 n \rfloor + 2. \]

Hence for \( n \geq 4 \), we have \( \log_2 n \geq 2 \) and \( \log_2 n \leq \lfloor \log_2 n \rfloor + 1 \) for \( n \geq 1 \) so

\[ \log_2 n \leq B(n) \leq 2 \log_2 n. \]

Thus \( B(n) = O(\log_2 n) \) and \( \log_2 n = O(B(n)) \). We note that in the literature of computer science, \( \log_2 n \) is often abbreviated to \( \log n \).

\[ \blacklozenge \]

**Note.** The inequality (ii), \( B(m) \leq B(n) \) if \( m \leq n \), is easy to understand. However a rigorous proof can be done by inductively proving that \( B(n) \leq B(n + 1) \) for any \( n \geq 1 \). The idea is sketched in the following diagrams

\begin{align*}
\text{n even:} & \quad \bullet \ldots \bullet \quad \bullet \quad \bullet \ldots \bullet & \quad B(2m) = 1 + B(m) & \quad \text{these two give:} \\
\text{n + 1:} & \quad \bullet \ldots \bullet \quad \bullet \quad \bullet \ldots \bullet & \quad B(2m + 1) = 1 + B(m) & \quad B(2m) \leq B(2m + 1) \\
\text{n odd :} & \quad \bullet \ldots \bullet \quad \bullet \quad \bullet \ldots \bullet & \quad B(2m + 1) = 1 + B(m) \\
\text{n + 1:} & \quad \bullet \ldots \bullet \quad \bullet \quad \bullet \ldots \bullet & \quad B(2m + 2) = 1 + B(m + 1) & \quad \geq 1 + B(m) & \quad \text{induction assumption}
\end{align*}

In fact, it is not difficult to derive a much neater expression \( B(n) = \lfloor \log_2 n \rfloor + 1 \) for \( n \geq 1 \) via, for instance, first an induction on \( B(n) \leq 1 + \log_2 n \).

\[ \blacklozenge \]

### 5.6 Performance of Sorting Algorithms

A number of sorting algorithms will be introduced in the tutorials over the semester. They include **bubble sort**, **insertion sort**, **selection sort**, **merge sort** and **quick sort**. The performance of these sorting algorithms, for simplicity, will be based purely on the number of comparisons involved during the sorting process.
In sorting a list of \( n \) items, selection sort, bubble sort, insertion sort and quick sort each require a total of \( O(n^2) \) comparisons in the corresponding worst (i.e., the most “laborious”) cases. Let \( M(n) \) be number of comparisons needed for the merge sort in the worst case, and \( \overline{Q}(n) \) be the average number of comparisons needed for the quick sort, then both \( M(n) \) and \( \overline{Q}(n) \) are \( O(n \log_2 n) \). More precisely, however, one can show for sorting a list of \( n \) (\( \geq 1 \)) items

\[
M(n) \leq 4n \log_2 n, \quad \overline{Q}(n) \leq (2 \ln 2)n \log_2 n.
\]

This subsection should be regarded as just the passing remarks. Full explanations of the sorting algorithms and the associated complexity and performance will be covered in a sequence of the tutorials. For a quick summary of these sorting algorithms, please also see the appendix Tutorials: Sorting Algorithms.

**Exercises**

1. Let \( 0 < x < 1 \). What is the value of \( \left\lceil \frac{3 - x}{2} \right\rceil \), and what is value of \( \left\lceil \frac{3 - x}{3} \right\rceil \)?

2. How many comparisons are needed to search “G” from the following list

\[
A, B, C, D, E, F, G, H, I, J
\]

through the use of the binary search algorithm?
6 Symbolic Logic

Symbolic logic forms the mathematical basis for much of what happens in a computer and is an important area of mathematics, in its own right. The purpose here is to deal with various forms of statements or compound statements, to symbolize and represent the underlying logic, as well as to set up the basic concepts for further analysis.

6.1 Basic Concepts

- A **proposition** (statement) is a complete declarative sentence which is either true or false. Questions, commands, exclamations and phrases are not mathematical sentences since they cannot be judged to be true or false.

- A **simple proposition** is one that does not contain any other statement as a part. In other words, it cannot be broken down without a loss in meaning. We will use the lowercase letters, \( p, q, r, \ldots \) as symbols for simple statements.

- A **compound proposition** is one with two or more simple statements as parts of what we will call components.
  
  - Components may themselves be compound propositions.
  
  - Components are “glued” together by the use of the **connectives** “and”, “or”, “not”, “implies”, “equivalent to”.

- A **tautology** is a proposition which is always true.

- A **contradiction** is a proposition which is always false.

Suppose we have two statements or propositions denoted by \( p \) and \( q \) respectively, then the above connectives give rise to the following compound propositions:

- \( p \land q \), the **conjunction** of \( p \) and \( q \), meaning “\( p \) and \( q \)”

- \( p \lor q \), the **disjunction** of \( p \) and \( q \), meaning “\( p \) or \( q \)”

- \( \neg p \), the **negation** of \( p \), meaning “not \( p \)”

- \( p \implies q \), the **implication**, meaning “\( p \) implies \( q \)”

- \( p \iff q \), the **equivalence** of \( p \) and \( q \), meaning “\( p \) and \( q \) are equivalent”

**Note.** The negation \( \neg p \) is sometimes also written as \( \sim p \). Both notations are acceptable in this unit.
Example

1. Among the following four sentences
   (a) Today is a rainy day.
   (b) David was wet this morning.
   (c) Did David get soaked in the rain?
   (d) Please read the notes.

only (a) and (b) are propositions. If we denote by \( p \) and \( q \) respectively the propositions (a) and (b), then

- \( p \land q \) represents “Today is a rainy day and David was wet this morning”.
- \( p \lor q \) represents “Either today is a rainy day, or David was wet this morning, or both”.
- \( \sim p \) represents “Today is not a rainy day”.

The truth value of a compound statement depends on the truth values of its components. For instance, when \( p \) and \( q \) are propositions, \( p \land q \) (i.e., “\( p \) and \( q \)” is true \( T \)) if and only if both \( p \) and \( q \) are true. (“if and only if” is a standard phrase in mathematics meaning “equivalent to”). The precise effect or truth values for the above connectives can be summerized in the following truth tables:

\[
\begin{array}{c|c|}
 p & q & p \land q \\
 T & T & T \\
 T & F & F \\
 F & T & F \\
 F & F & F \\
\end{array}
\]

\[
\begin{array}{c|c|}
 p & q & p \lor q \\
 T & T & T \\
 T & F & T \\
 F & T & T \\
 F & F & F \\
\end{array}
\]

\[
\begin{array}{c|}
 p & \sim p \\
 T & F \\
 F & T \\
\end{array}
\]

A truth table is a complete list of the possible truth values of a statement. We use “T” to mean “true”, and “F” to mean “false” (though it may be clearer and quicker to use ”1” and ”0”, respectively).

In setting up these truth tables it is necessary to have all possible combinations of T’s and F’s: with two propositions there are 4 possibilities, with three propositions there are 8 possibilities and with \( n \) propositions there are \( 2^n \) possibilities. It is a mathematical tradition to split the first column in two - the first half being all T’s and the second half being all F’s, then to split the second column into quarters with T’s in the first quarter, F’s in the second quarter and so on, then to split the third column, if there is one, into eights with blocks of T’s and F’s alternating, and so on.

In fact we can take these tables as the precise definitions for the corresponding connectives. Likewise one way of describing completely a compound proposition is to give explicitly its truth table. Going the other way, we note that a truth table can also be used to define a compound proposition.
6.2 Logical Equivalence

When ever a new mathematical concept or structure is introduced, it is important to know when two items are the same or are equal or are equivalent. One simple reason for this is that we don’t want to waste effort considering two things as if they were different when they were the same all along. You will meet this idea of equivalence, in various guises, throughout this unit.

Two (compound) propositions $P$ and $Q$ are said to be equivalent or logically equivalent, denoted by $P \equiv Q$ or by $P \iff Q$, if (i.e., if and only if) they have the same truth values. In other words, for all possible truth values of the component statements, the compound propositions will have the same truth values.

Examples

2. Show $(\neg p) \lor (\neg q)$ and $\neg (p \land q)$ are equivalent.

Solution. We first construct below the truth table for the two compound propositions. When first tackling problems like this, many students have trouble knowing how to start. The key is to work out what is required for the first line - the headings of the columns. Once that is done, the rest is just filling in the T’s and F’s by following the basic rules. The first line comes from asking ”What are the components of each of the propositions?”.

For example, the first proposition is made up of $\neg p$ and $\neg q$ so there needs to be a column for each.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$\neg p$</th>
<th>$\neg q$</th>
<th>$p \land q$</th>
<th>$\neg (p \land q)$</th>
<th>$\neg p \lor \neg q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Since the last two columns are the same, we conclude $(\neg p) \lor (\neg q)$ and $\neg (p \land q)$ are equivalent.

3. Show $\neg (p \land q)$ and $(\neg p) \land (\neg q)$ are not logically equivalent.

Solution. This is manifested in the following truth table.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$\neg p$</th>
<th>$\neg q$</th>
<th>$p \land q$</th>
<th>$\neg (p \land q)$</th>
<th>$(\neg p) \land (\neg q)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

Because the corresponding truth values differ (at 2 places).
4. Show \((p \lor q) \lor (\neg p)\) is a tautology and \((p \land q) \land (\neg p)\) is a contradiction.

**Solution.** From the following truth table

\[
\begin{array}{cccccccc}
 p & q & \neg p & \neg q & p \lor q & p \land q & (p \lor q) \lor (\neg p) & (p \land q) \land (\neg p) \\
 T & T & F & F & T & T & T & T \\
 T & F & F & T & T & F & T & F \\
 F & T & T & F & T & F & T & F \\
 F & F & T & T & T & F & T & F \\
\end{array}
\]

we see \((p \lor q) \lor (\neg p)\) is always true and is thus a tautology and \((p \land q) \land (\neg p)\) is always false and is thus a contradiction.

**Note.** For simplicity, one may use “0” and “1” to denote “F” and “T” respectively in the truth tables.

A number of logical equivalences are summarised in the following theorem. Proofs are left as exercises.

**Theorem 8. (Logical Equivalences)** Let \(p, q, r\) be propositions and denote by \(\top\) and \(\bot\) tautology, respectively contradiction. Then the following logical equivalences hold.

1. **Commutative laws** \(p \land q \equiv q \land p\) \(p \lor q \equiv q \lor p\)
2. **Associative laws** \((p \land q) \land r \equiv p \land (q \land r)\) \((p \lor q) \lor r \equiv p \lor (q \lor r)\)
3. **Distributive laws** \(p \land (q \lor r) \equiv (p \land q) \lor (p \land r)\) \(p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)\)
4. **Identity laws** \(p \land \top \equiv p\) \(p \lor \bot \equiv p\)
5. **Negation laws** \(p \land \neg p \equiv \bot\) \(p \lor \neg p \equiv \top\)
6. **Double Negation law** \(\neg \neg (p) \equiv p\)
7. **Idempotent laws** \(p \land p \equiv p\) \(p \lor p \equiv p\)
8. **Universal bound laws** \(p \land \bot \equiv \bot\) \(p \lor \top \equiv \top\)
9. **De Morgan’s laws** \(\neg (p \land q) \equiv (\neg p) \lor (\neg q)\) \(\neg (p \lor q) \equiv (\neg p) \land (\neg q)\)
10. **Absorption laws** \(p \land (p \lor q) \equiv p\) \(p \lor (p \land q) \equiv p\)
11. **Negations of \(\top\) and \(\bot\)** \(\neg \top \equiv \bot\) \(\neg \bot \equiv \top\)

**Examples**

5. Use the laws of Theorem 8 to verify the logical equivalence

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

**Solution.** Starting with the most complex side, working toward the other side:

\[
\begin{align*}
\neg (p \lor (\neg p \land q)) & \equiv \neg ((p \lor \neg p) \land (p \lor q)) \quad \text{Distributivity} \\
& \equiv \neg (\top \land (p \lor q)) \quad \text{Identity} \\
& \equiv \neg (p \lor q) \quad \text{De Morgan} \\
& \equiv \neg p \land \neg q
\end{align*}
\]
6.3 Conditional Statements

The compound proposition implication

\[ p \to q \]

is a conditional statement, and can be read as “if \( p \) then \( q \)” or “\( p \) implies \( q \)”. Its precise definition is given by the following truth table

<table>
<thead>
<tr>
<th>( p )</th>
<th>( q )</th>
<th>( p \to q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

Let us briefly see why the above definition via the truth table is “reasonable” and is consistent with our day to day understanding of the notion of implications. We observe that the only explicit contradiction to “if \( p \) then \( q \)” comes from the case when \( p \) is true but \( q \) is false, and this explains the only “F” entry in the \( p \to q \) column. We also note that some people would never use “\( p \) implies \( q \)” to refer to \( p \to q \); they would instead use “\( p \) implies \( q \)” to exclusively refer to \( p \Rightarrow q \), i.e. \( p \to q \) is a tautology. More details on “\( \Rightarrow \)” can be found in one of the later lectures.

Example

6. Let \( p \) denote “I buy shares” and \( q \) denote “I’ll be rich”. Then \( p \to q \) means “If I buy shares then I’ll be rich”. Let us check row by row the “reasonableness” of the truth table for \( p \to q \) given shortly before.

Row 1: “I buy shares” (\( p \) true) and “I’ll be rich” (\( q \) true) is certainly consistent with \( p \to q \) being true.

Row 2: “I buy shares” and “I won’t be rich” means “If I buy shares then I’ll be rich” (i.e. \( p \to q \)) is false.

Row 3 and 4: “I don’t buy shares” won’t contradict our statement \( p \to q \), regardless of whether I’ll be rich, as obviously there are other ways to get rich.

Representation

\[ p \to q \equiv (\sim p) \lor q \]

This can easily be proved by the use of the truth table.

Note. Obviously a string like \( p) \wedge \wedge \to qr \) is not a legitimate logical expression. In this unit, we always assume that all the concerned strings of logical expressions are well-formed formulas, or wffs, i.e. the strings are legitimate.
Exercises

1. Let \( p \) and \( q \) be propositions. Show

   (a) \((p \lor q) \land (\neg p \lor q) \equiv q\).

   (b) \(p \rightarrow q \equiv \neg q \rightarrow \neg p\).

   (c) \((p \lor q) \lor (\neg p) \land (\neg q)\) is a tautology.

   (d) \((p \lor q) \land (\neg p) \land (\neg q)\) is a contradiction.

2. Let \( p, q \) and \( r \) be propositions. Use a truth table to show that \((p \land q) \lor r\) is not logically equivalent to \( p \land (q \lor r)\).
7 Propositional Logic

7.1 Argument Forms

We have already encountered a few basic concepts related to propositions. They include truth values true \(T\) and false \(F\), tautology, contradiction, and \(\land\), or \(\lor\), not \(\neg\), and implication \(\rightarrow\). If we further define an equivalence connective \(\leftrightarrow\) for any two propositions \(p\) and \(q\) by \((p \rightarrow q) \land (q \rightarrow p)\), denoted by \(p \leftrightarrow q\), then the usual “order of precedence” is

1. connectives within parentheses, innermost parentheses first
2. \(\neg\)
3. \(\land\), \(\lor\)
4. \(\rightarrow\)
5. \(\leftrightarrow\)

You may also find that some people actually place a higher precedence for \(\land\) than for \(\lor\). To avoid possible confusion we shall always insert the parentheses at the appropriate places.

Example

1. \(p \leftrightarrow q \lor \neg r \land p\) is same as \(p \leftrightarrow ((q \lor (\neg r)) \land p)\). However \(p \leftrightarrow q \lor \neg r \land p\) would be same as \(p \leftrightarrow (q \lor ((\neg r) \land p))\) had we adopted higher precedence for \(\land\) than for \(\lor\).

An argument form, or argument for short, is a sequence of statements. All statements but the last one are called premises or hypotheses. The final statement is called the conclusion, and is often preceded by a symbol “\(\therefore\)”, pronounced as therefore. Typically an argument form will take the form

\[ p_1, p_2, \ldots, p_n, \therefore q \]

where propositions \(p_1, \ldots, p_n\) are the premises and proposition \(q\) is the conclusion. The above argument form can also be represented vertically by

\[
\begin{array}{c}
p_1 \\
p_2 \\
\vdots \\
p_n \\
\therefore q
\end{array}
\]

or by

\[
\begin{array}{c}
p_1 \\
p_2 \\
\vdots \\
p_n \\
\therefore q
\end{array}
\]
An argument is **valid** if the conclusion is true whenever all the premises are true. In other words, an argument form \((p_1, p_2, \ldots, p_n, \therefore q)\) is valid if and only if the proposition \((p_1 \land p_2 \land \cdots \land p_n) \rightarrow q\) is a tautology.

Notice that a valid argument may have false premises and a false conclusion. The validity of an argument can be tested through the use of the truth table by checking if the **critical rows**, i.e., the rows in which all premises are true, will correspond to the value “true” for the conclusion.

An **invalid** argument is an argument which is not valid. In other words, it has true premises but a false conclusion.

**Examples**

2. Show that \((p \lor q, p \rightarrow r, q \rightarrow r, \therefore r)\) is a valid argument.

**Solution.** From the table

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>r</th>
<th>(p \lor q)</th>
<th>p (\rightarrow) r</th>
<th>q (\rightarrow) r</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

we see all critical rows (in this case, those with the shaded positions all containing a T) correspond to (the circled) T (true) for r. Hence the argument is valid.

3. Show that the argument \((p \rightarrow q, \therefore \sim p \rightarrow \sim q)\) is invalid.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>(\sim p)</th>
<th>(\sim q)</th>
<th>p (\rightarrow) q</th>
<th>(\sim p \rightarrow \sim q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

We see that on the third row, a critical row, the premise \(p \rightarrow q\) is true while the conclusion \(\sim p \rightarrow \sim q\) is false. Hence the argument \((p \rightarrow q, \therefore \sim p \rightarrow \sim q)\) is invalid.
### 7.2 Converse, Inverse and Contrapositive

For any conditional statement \( p \rightarrow q \) (where \( p \) is the premise or hypothesis and \( q \) is the conclusion), we can derive other three conditional statements:

- \( q \rightarrow p \) called the **converse** of \( p \rightarrow q \);
- \( \sim p \rightarrow \sim q \) called the **inverse** of \( p \rightarrow q \);
- \( \sim q \rightarrow \sim p \) called the **contrapositive** of \( p \rightarrow q \).

**Example**

4. Let \( p \) be the statement “I’m sick” and \( q \) be the statement “I go and see a doctor”. The \( p \rightarrow q \) means “**If** I’m sick, **then** I go and see a doctor”. The converse of this statement is “**If** I go and see a doctor, **then** I’m sick”, while the inverse reads “**If** I’m not sick, **then** I won’t go and see a doctor”. However, the contrapositive takes the form “**If** I don’t go and see a doctor, **then** I’m not sick”.

We note that the result in example 3 amounts to saying the inverse of a statement is not necessarily true. It represents one of the following two typical fallacies:

- **converse error**: \( p \rightarrow q, \therefore q \rightarrow p \)
- **inverse error**: \( p \rightarrow q, \therefore \sim p \rightarrow \sim q \)

### 7.3 Rules of Inference

Rules of inference are no more than valid arguments. The simplest yet most fundamental valid arguments are

- **modus ponens**: \( p \rightarrow q, p, \therefore q \)
- **modus tollens**: \( p \rightarrow q, \sim q, \therefore \sim p \)

Latin phrases **modus ponens** and **modus tollens** carry the meaning of “method of affirming” and “method of denying”, respectively. That they are valid can be easily established. Modus tollens, for instance, can be seen or derived by the following truth table

<table>
<thead>
<tr>
<th></th>
<th>( p \rightarrow q )</th>
<th>( \sim q )</th>
<th>( \sim p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

The only critical row which shows the validity of the argument form.
Other rules of inference are:

- **Disjunctive addition:** \( p, \therefore p \lor q \).
- **Conjunctive addition:** \( p, q, \therefore p \land q \).
- **Conjunctive simplification:** \( p \land q, \therefore p \).
- **Disjunctive syllogism:** \( p \lor q, \sim q, \therefore p \).
- **Hypothetical syllogism:** \( p \rightarrow q, q \rightarrow r, \therefore p \rightarrow r \).
- **Division into cases:** \( p \lor q, p \rightarrow r, q \rightarrow r, \therefore r \).
- **Rule of contradiction:** \( \sim p \rightarrow \text{contradiction}, \therefore p \).
- **Contrapositive equivalence:** \( p \rightarrow q, \therefore \sim q \rightarrow \sim p \).

The validity of the above argument forms can all be easily verified via truth tables. In fact the case of “division into cases” has been proven in example 2. These rules may not mathematically look very familiar. But it is most likely that everyone has used them all, individually or jointly, at some stage subconsciously.

The validity of an argument form may be established through the use of the above rules of inference along with other laws of logic such as De Morgan’s laws. This type of validation is particularly useful when the argument form under question contains a relatively larger number of premises, which means a proof by a truth table is often unrealistically tedious.

**Examples**

5. Show that the following argument form

\[
p \lor q, \quad q \rightarrow r, \quad p \land s \rightarrow t, \quad \sim r, \quad \sim q \rightarrow u \land s, \quad \therefore t
\]

is valid by breaking it into a list of known elementary valid argument forms or rules. In other words, show the validity of the above argument form (*) through the basic inference rules or the laws of logic.

**Solution.** We’ll treat all the rules of inference introduced earlier in this subsection as the known elementary argument forms. The logical inference for the argument form in the question is as follows.
6. Can you explain, in additional details, how those 6 proof steps in the above example come into existence?

**Solution.** In the following we shall give the “reverse route” for the proof of the above argument form. Basically we shall start with the conclusion $t$, then go downwards towards other “required” intermediate propositions by making use of the known premises. The final proof of the argument form will be essentially in the reverse order of the “reverse route”. The main idea is to try to lead from the conclusion to eventually reach only the premises. If the final objective is $t$, the only premise that involves $t$ explicitly is $p \land s \rightarrow t$. Hence to derive “$t$” (i.e. $t$ is true) one needs “$p \land s$” to be true, obtained by conjunctive addition of $p$ and $s$. To derive “$p$” from “$p \lor q$”, one needs “$\neg q$” (disjunctive syllogism). To have “$\neg q$” from “$q \rightarrow r$” one must have “$\neg r$” (premise). $s$ can be obtained by conjunctive simplification of $u \land s$.

With the help of the above order of derivation, we have

(a) $\neg r, q \rightarrow r, \therefore \neg q$ (modus tollens)
(b) $\neg q, p \lor q, \therefore p$ (disjunctive syllogism)
(c) $\neg q, \neg q \rightarrow u \land s, \therefore u \land s$ (modus ponens)
(d) $u \land s, \therefore s$ (conjunctive simplification)
(e) $p, s, \therefore p \land s$ (conjunctive addition)
(f) $p \land s, p \land s \rightarrow t, \therefore t$ (modus ponens)

That is, the conclusion is derived from the use of the basic inference rules.
7. Can you do example 6 once again, with some differences?

**Solution.** The “dumbest” way is to try to determine for each proposition (symbol) if it is true or not. This way one could be wasting a lot of time unnecessarily; but this often ensures one gets closer and closer to the solution of the problem.

In the argument form in examples 5-6, we have 6 basic propositions $p, q, r, s, t$ and $u$. We will now proceed to determine (in the order dictated by the actual circumstances) whether each of these 6 propositions is true or not.

(i) $\sim r$ is given as a premise, so the truth value of proposition $r$ is already determined ($r$ is false).

(ii) $q \rightarrow r, \sim r, \therefore \sim q$, so the truth value of $q$ is determined ($q$ is false).

(iii) $p \lor q, \sim q, \therefore p$, so the truth value of $p$ is determined ($p$ is true).

(iv) $\sim q, \sim q \rightarrow u \land s, \therefore u \land s$.

(v) $u \land s, \therefore u$, so $u$ is determined ($u$ is true).

(vi) $u \land s, \therefore s$, so $s$ is determined ($s$ is true).

(vii) $p, s, \therefore p \land s$.

(viii) $p \land s, p \land s \rightarrow t, \therefore t$, so the truth value of $t$ is determined ($t$ is true).

We have by now established the truth value of all the concerned basic propositions $p, q, r, s, t, u$.

We can now proceed to determine if the conclusion $t$ of the original argument form is true or not. In this case, it is obvious because we have already shown $t$ is true. So the argument form (*) in example 5 is valid. Notice that step (v) is completely unnecessary. But we probably wouldn’t know it at that time.

**Note.** This “dumb” method can be very useful if you want to determine whether or not the conclusion of a lengthy argument form is true or not.

8. Let $p, q, r, s$ and $t$ be propositions.

(a) Use a truth table to show $p \rightarrow q$ is equivalent to $(\sim p) \lor q$.

(b) Use a truth table to determine whether or not the following argument form is valid

$$p \rightarrow q, \quad q \rightarrow r, \quad \therefore r.$$  

(c) Show the following argument form

$$r \rightarrow p, \quad \sim p \lor q, \quad s \rightarrow p \land r, \quad \sim p \land \sim r \rightarrow s \lor t, \quad \sim q, \quad \therefore t$$

is valid by deducing the conclusion from the premises step by step through the use of basic inference rules.
Solution.

(a) From the truth table below

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$\sim p$</th>
<th>$(\sim p) \lor q$</th>
<th>$p \rightarrow q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

we see that the last two columns, corresponding to $(\sim p) \lor q$ and $p \rightarrow q$ respectively, are exactly the same. This means $(\sim p) \lor q$ is equivalent to $p \rightarrow q$.

(b) The truth table for the argument form reads

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$r$</th>
<th>$p \rightarrow q$</th>
<th>$q \rightarrow r$</th>
<th>$r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

Among the 8 rows in the above table, the 1st, 5th, 7th and 8th rows are critical rows. Since the conclusion $r$ fails at the last critical row, the argument form is not valid.

(c) The derivation is as follows:

1. $\sim q, \sim p \lor q, \therefore \sim p$ (disjunctive syllogism)
2. $\sim p, r \rightarrow p, \therefore \sim r$ (modus tollens)
3. $\sim p, \sim r, \therefore \sim p \land \sim r$ (conjunctive addition)
4. $\sim p \land \sim r, \sim p \lor r, \therefore s \lor t, \therefore s \lor t$ (modus ponens)
5. $\sim r, \therefore \sim r \lor p$ (disjunctive addition)
6. $s \rightarrow \sim p \land r$ is the same as $s \rightarrow (\sim r \lor p)$ (De Morgan’s laws)
7. $\sim r \lor p, s \rightarrow (\sim r \lor p), \therefore \sim s$ (modus tollens)
8. $s \lor t, \sim s, \therefore t$ (disjunctive syllogism)

Hence the argument form is valid.

9. A detective established that one person in a gang comprised of 4 members $A$, $B$, $C$ and $D$ killed a person named $E$. The detective obtained the following statements from the gang members ($S_i$ denotes the statement made by $i$, that is $S_A$ denotes the statement made by $A$, and so on):
(i) $S_A$: B killed E.
(ii) $S_B$: C was playing pool with A when E was knocked off.
(iii) $S_C$: B didn’t kill E.
(iv) $S_D$: C didn’t kill E.

The detective was then able to conclude that all but one were lying. Can you decide who killed E?

**Solution.** Let us consider the following four statements denoted (1) to (4). The 1st two, (1) and (2) below, are true due to the detective’s work.

(1): Only one of the statements $S_A, S_B, S_C, S_D$ is true.
(2): One of A, B, C and D killed E.

From the (content of the) statements $S_A$ and $S_C$ we know

(3): $S_A \rightarrow S_D$ is true because if $S_A$ is true, then B killed E which implies C didn’t kill E due to (2), implying $S_D$ is also true.

and from (1)

(4): $S_A \rightarrow \neg S_B \land \neg S_C \land \neg S_D$ is true.

Let us examine the following sequence of statements.

(a) $S_A \rightarrow S_D$ $S_A$ true implies $S_D$ true, i.e. $(S_A, \therefore S_D)$ (from (3))
(b) $S_A \rightarrow \neg S_B \land \neg S_C \land \neg S_D$ $S_A$ true implies $S_B, S_C$ and $S_D$ all false (from (4))
(c) $\neg S_B \land \neg S_C \land \neg S_D \rightarrow \neg S_D$ conjunctive simplification (direct)
(d) $S_A \rightarrow \neg S_D$ hypothetical syllogism (from (b), (c))
(e) $\neg S_D \rightarrow \neg S_A$ modus tollens (from (a))
(f) $S_A \rightarrow \neg S_A$ hypothetical syllogism (from (d), (e))

We note all the statements on the sequence apart from the first two (a) and (b) are obtained from their previous statements or form the valid argument forms. However the first 2 statements (a) and (b) are both true hence the conclusion in (f) is also true. A statement sequence of this type is sometimes called a **proof sequence** with the last entry called a **theorem**. The whole sequence is called the **proof** of the theorem.

Alternatively sequence (a)–(f) can also be regarded as a valid argument form in which a special feature is that the truth of the first 2 statements will ensure that all the premises there are true.

From (3) and (4) and (a)–(f) we conclude $S_A \rightarrow \neg S_A$ is true. Hence $S_A$ must be false from the rule of contradiction (if $S_A$ were true then $\neg S_A$ would be true, implying $S_A$ is false: contradiction).
From the definition of $S_C$ we see $\sim S_A \rightarrow S_C$. From the modus ponens
\[
\sim S_A \rightarrow S_C, \sim S_A, \therefore S_C
\]
we conclude $S_C$ is true. Since (1) gives
\[
S_C \rightarrow \sim S_A \land \sim S_B \land \sim S_D,
\]
we obtain from the (conjunctive simplification) argument
\[
S_C \rightarrow \sim S_A \land \sim S_B \land \sim S_D, \sim S_A \land \sim S_B \land \sim S_D \rightarrow \sim S_D, \therefore S_C \rightarrow \sim S_D
\]
that $S_C \rightarrow \sim S_D$. Finally from modus ponens \((S_C \rightarrow \sim S_D, S_C, \therefore \sim S_D)\), we conclude $\sim S_D$ is true, that is, C killed E.

We note that in the above example, we have deliberately disintegrated our argument into smaller pieces with mathematical symbolisation. It turns out that verbal arguments in this case are much more concise. For a good comparison, we give below an alternative solution.

**Solution.** (alternative for example 6) Suppose A wasn’t lying, then A’s statement B killed E is true. Since A spoke the truth means B, C and D would be lying, hence the statement C didn’t kill E said by D would be false, implying C did kill E. But this is a contradiction to the assumption A spoke the truth. Hence A was lying, which means B didn’t kill E, which in turn implies C spoke the truth. Since only one person was not lying, D must have lied. Hence C didn’t kill E is false. Hence C killed E. ▲

**Exercises**

1. Use a truth table to show that the conditional statement $p \rightarrow q$ doesn’t imply its converse $q \rightarrow p$.

2. Let $p$, $q$ and $r$ be propositions. Is the following
\[
p, \quad q, \quad \sim p \land \sim q, \therefore r
\]
a valid argument form? How many critical rows are there in the truth table?

3. Let $p$, $q$, $r$ and $s$ be propositions. Show the following argument form
\[
p \rightarrow \sim r, \quad p \lor \sim q, \quad \sim r \lor q \lor s, \quad r, \therefore s
\]
is valid by deducing the conclusion from the premises step by step through the use of basic inference rules.

4. Let $p_1, p_2, \ldots, p_n$ and $q$ be propositions. Show that the argument form \((p_1, p_2, \ldots, p_n, \therefore q)\) is valid if and only if $p_1 \land p_2 \land \cdots \land p_n \land (\sim q)$ is a contradiction.
 Predicate Calculus

Symbolic logic can be used in a lot of areas but it has limitations; it doesn’t deal with situations that occur in mathematics where statements are qualified by phrases like “for all...” and “there exists ...” Predicate calculus fills the gap.

8.1 Predicate Quantifiers

A predicate is a sentence that contains a finite number of variables and becomes a statement when specific values are substituted for the variables.

The domain of a predicate variable is the set of all values that may be substituted in place of the variable.

In predicate calculus, the following two quantifiers are important:

- **universal quantifier:** ∀, meaning “for all”

- **existential quantifier:** ∃, meaning “there exists”

Example

1. The statement that

\[ |\sin(x)| \leq 1 \text{ for any real-valued number } x \]

can be written in either of the following forms

\[ \forall x \in \mathbb{R}, |\sin(x)| \leq 1 \]

\[ \forall x, \left( x \in \mathbb{R} \rightarrow |\sin(x)| \leq 1 \right) \]

Hence we see that the domain of a predicate could also be absorbed in the predicate itself.

Let \( P(x) \) and \( Q(x) \) be predicates (of a common domain \( D \)). Then

- A universal statement (\( \forall x \in D, P(x) \)) is true if and only if \( P(x) \) is true for every \( x \in D \).

- An existential statement (\( \exists x \in D, P(x) \)) is true iff \( P(x) \) is true for at least one \( x \in D \).

- \( P(x) \Rightarrow Q(x) \) means (\( \forall x \in D, P(x) \rightarrow Q(x) \)) is true.

- \( P(x) \Leftrightarrow Q(x) \) means (\( \forall x \in D, P(x) \leftrightarrow Q(x) \)) is true.
We note that in the case of $p$ and $q$ being propositions, then $p \Rightarrow q$ simply means $p \rightarrow q$ is a tautology. Moreover $p \iff q$ means both $p \Rightarrow q$ and $q \Rightarrow p$. We also note that the predicate variable $x$ actually represents all the predicate variables. In general a statement may involve many mathematical or logical operations. It is thus worthwhile to list a finer order of precedence

1. parentheses
2. $\sim$
3. $\land$, $\lor$
4. $\Rightarrow$
5. $\iff$
6. $\Rightarrow$, $\equiv$
7. $\forall$, $\exists$

Examples

2. Let $\mathbb{N}$ be the set of natural numbers, i.e., $\mathbb{N} = \{0, 1, 2, \ldots\}$. Represent mathematically the following statements:

(a) The sum or subtraction of any two natural numbers will remain a natural number.
(b) A natural number, if divided by a natural number, may not remain an natural number.
(c) There exist two natural numbers such that the sum of the squares of these two natural numbers can be written as the square of a natural number.
(d) The sum of the squares of any two natural numbers can be written as the square of a natural number.

Solution. We’ll explain in more details in the first two cases.

(a) This statement can be written in any of the following three forms

\[
\forall m, n \in \mathbb{N}, \quad ((m + n) \in \mathbb{N}) \land ((m - n) \in \mathbb{N}),
\]

\[
\forall m \in \mathbb{N}, \quad \forall n \in \mathbb{N}, \quad ((m + n) \in \mathbb{N}) \land ((m - n) \in \mathbb{N}),
\]

\[
\forall m \in \mathbb{N}, \quad \forall n \in \mathbb{N}, \quad ((m + n) \in \mathbb{N}) \land ((m - n) \in \mathbb{N}) .
\]

This is because essentially $(m + n) \in \mathbb{N}$ says the sum of two natural numbers is again an natural number, and $(m - n) \in \mathbb{N}$ says likewise for the subtraction of two natural numbers. We note that while the first form is perhaps the neatest, the first and the second forms can both be interpreted as an equivalent “shorthand” notation of the third form. Moreover, the third form can be regarded as an embedded predicate statement because it can be considered as

\[
\forall m \in \mathbb{N}, \quad P(m) ,
\]

where $P(m)$ for each given $m$ is again a predicate statement:

\[
\forall n \in \mathbb{N}, \quad ((m + n) \in \mathbb{N}) \land ((m - n) \in \mathbb{N}) .
\]

(b) First we rephrase the original sentence in a form closer to the mathematical language. It is easy to observe that the original sentence is equivalent to

“There exist two natural numbers such that the division of the first natural number by the second natural number is no longer a natural number’’. Written symbolically,

\[
\exists m, n \in \mathbb{N}, \quad \frac{m}{n} \notin \mathbb{N} .
\]
Obviously the above form can also be written in other equivalent variant forms similar to the case (a).

(c) \( \exists m \in \mathbb{N}, \exists n \in \mathbb{N}, \exists p \in \mathbb{N}, \ m^2 + n^2 = p^2. \)

(d) \( \forall m \in \mathbb{N}, \forall n \in \mathbb{N}, \exists p \in \mathbb{N}, \ m^2 + n^2 = p^2. \)

3. Are the two statements (a) and (b) below true?

(a) \( \exists m \in \mathbb{N}, \exists n \in \mathbb{N}, \exists p \in \mathbb{N}, \ m^2 + n^2 = p^2 \) and \( m, n, p \geq 1. \)

(b) \( \forall m \in \mathbb{N}, \forall n \in \mathbb{N}, \exists p \in \mathbb{N}, \ m^2 + n^2 = p^2. \)

Solution. We note that more than 1 quantifier are used in both (a) and (b).

(a) Though \( m = 0, n = 0 \) and \( p = 0 \) satisfy \( m^2 + n^2 = p^2 \), they do not satisfy \( m, n, p \geq 1 \). Hence this case is not able to shed much light on the validity of the statement (a).

However \( m = 3, n = 4 \) and \( p = 5 \) satisfy both \( m^2 + n^2 = p^2 \) and \( m, n, p \geq 1 \). We see that (a) is true regardless of other cases of \( m, n \) and \( p \).

(b) For \( m = 0, n = 0 \), we can choose \( p = 0 \) so that \( m^2 + n^2 = p^2 \), i.e. the predicate is true in this particular case. However, for \( m = 1 \) and \( n = 1 \), we see that if a \( p \in \mathbb{N} \) satisfies \( p^2 = m^2 + n^2 = 2 \), then \( p \) must be \( \sqrt{2} \). But no such nonnegative integers \( p \) satisfy \( p^2 = 2 \). Hence (b) is false.

4. For those who happen to know a little about taking a limit of a sequence, we just mention that \( \lim_{n \to \infty} a_n = a \) if the following statement is true

\[ \forall \varepsilon \in \{ x \in \mathbb{R}, x > 0 \}, \exists N \in \mathbb{N}, \forall n \in \{ x \in \mathbb{N}, n \geq N \}, \ |a_n - a| < \varepsilon. \]
Examples

5. What is the negation of the statement “Everyone loves someone”? 

Solution. Let \( D \) denote the set of all people and predicate \( L(x, y) \) denote that person \( x \) loves person \( y \). Then the statement

“What everyone loves someone”

Can be written as

\[
\forall x \in D, \exists y \in D, L(x, y)
\]

Or equivalently \((\forall x \in D, (\exists y \in D, L(x, y)))\). Hence the negation is

\[
\exists x \in D, \forall y \in D, \sim L(x, y)
\]

Which means “There is someone who loves no one”.

We note here the order in which the quantifiers appear is important. If for instance we change the order of quantifiers for the original statement and consider instead

\[
\exists y \in D, \forall x \in D, L(x, y)
\]

We see that this new statement is true means that “There’s someone everyone loves” which is quite different from the original statement.

6. Let \( S \) be the set of all students and let \( A \) be the set of all assignments. Let predicate \( D(x, y) \) denote that student \( x \) has done assignment \( y \), and predicate \( P(x) \) denote that student \( x \) passes this unit. Represent the predicate statement “If a student does all the assignments, then she will pass this unit”.

Solution. Note that the phrase “If a student ...” contains a hidden “for all” or “for every” because it is talking about any student.

We begin with writing out certain related simpler but useful components. We first observe that student \( x \) does all assignments can be written as

\[
\forall y \in A, D(x, y)
\]

Hence the statement “If student \( x \) does all the assignments, then she will pass this unit” can be written as

\[
(\forall y \in A, D(x, y)) \rightarrow P(x)
\]

It is now straightforward to see that the original statement “If a student does all the assignments, then she will pass this unit” can thus be simply denoted by

\[
\forall x \in S, (\forall y \in A, D(x, y)) \rightarrow P(x)
\]
For any statement \((\forall x \in D, P(x) \rightarrow Q(x))\), we define

- its **contrapositive** by \((\forall x \in D, \sim Q(x) \rightarrow \sim P(x))\),
- its **converse** by \((\forall x \in D, Q(x) \rightarrow P(x))\) and
- its **inverse** by \((\forall x \in D, \sim P(x) \rightarrow \sim Q(x))\).

Then the original universal conditional statement is equivalent to the contrapositive, and the converse is equivalent to the inverse.

The converse and inverse, as in the propositional logic, are in general not equivalent to the original statement. The fallacies due to wrongly assumed such equivalences are termed **converse error** and **inverse error**, respectively.

**Example**

7. Give the contrapositive, converse and inverse of the statement
   “If a real number is greater than 3 then its square is greater than 4”.

**Solution.** The statements can be written as

- **original:** \(\forall x \in \mathbb{R}, (x > 3) \rightarrow (x^2 > 4)\) \hspace{1cm} (a)
- **contrapositive:** \(\forall x \in \mathbb{R}, (x^2 \leq 4) \rightarrow (x \leq 3)\) \hspace{1cm} (b)
- **converse:** \(\forall x \in \mathbb{R}, (x^2 > 4) \rightarrow (x > 3)\) \hspace{1cm} (c)
- **inverse:** \(\forall x \in \mathbb{R}, (x \leq 3) \rightarrow (x^2 \leq 4)\) \hspace{1cm} (d)

It is easy to see that (a) and (b) are both true, while (c) and (d) are both wrong. For example, if we take \(x = 2.5\) in (c) we have \(x^2 = 2.5^2 > 4\) but \(x > 3\) is false. Hence (c) is false.

In predicate calculus, an argument is **valid** if no matter what particular variable values are substituted for the predicate symbols in its premises, the conclusion is true whenever the resulting premise statements are all true. We say

“\(\forall x, P(x)\) is a **sufficient condition** for \(Q(x)\)” if \((\forall x, P(x) \rightarrow Q(x))\) is true, and

“\(\forall x, P(x)\) is a **necessary condition** for \(Q(x)\)” if \((\forall x, Q(x) \rightarrow P(x))\) is true. Equivalently,

“\(\forall x, P(x)\) is a **necessary condition** for \(Q(x)\)” if \((\forall x, \sim P(x) \rightarrow \sim Q(x))\) is true.

If \(\forall x, P(x)\) is both a sufficient and a necessary condition for \(Q(x)\), then we say

“\(\forall x, P(x)\) **if and only if** (iff) \(Q(x)\)”.

**Examples**

8. Statement “One has to drink water in order to survive”, means for any person, drinking water is a necessary condition for survival, i.e.

\[
\forall \text{person } x, (x \text{ drinks water}) \rightarrow \sim(x \text{ survives}).
\]
It is obviously not a sufficient condition because drinking water alone is not enough to guarantee the survival of a person, i.e. the statement
\[ \forall \text{ person } x, \ (x \text{ drinks water}) \rightarrow (x \text{ survives}) \]
is not true.
That is, water is a necessary condition for survival but not sufficient.

9. Prove that the statement \((\forall n \in \mathbb{N}, \ \log_2(n + 1000) \geq n)\) is not true.

Solution. To prove \((\forall n \in \mathbb{N}, \ \log_2(n + 1000) \geq n)\) is not true is to prove that its negation, 
\[ \sim (\forall n \in \mathbb{N}, \ \log_2(n + 1000) \geq n) \]
\[ \equiv (\exists n \in \mathbb{N}, \ \sim (\log_2(n + 1000) \geq n)) \]
\[ \equiv (\exists n \in \mathbb{N}, \ \log_2(n + 1000) < n) , \]
we just need to show that there exists a nonnegative integer \(n_0\) such that \(\log_2(n_0 + 1000) < n_0\). If we try the first few integers in \(\mathbb{N}\), we see that \(n_0 = 10\) will suffice our purpose because \(\log_2(10 + 1000) < 10\), i.e. \(1010 < 2^{10} = 1024\). Obviously such \(n_0\) is not unique, and the uniqueness of such \(n_0\) is in fact irrelevant.

The rules in predicate logic can be represented as axioms symbolically. For example, we can easily convince ourselves that the followings are true

\[ P \Rightarrow (Q \rightarrow P) , \]
\[ (\sim Q \rightarrow \sim P) \Rightarrow (P \rightarrow Q) , \]
\[ (\forall x, P(x) \rightarrow Q(x)) \Rightarrow (\forall x, P(x)) \rightarrow (\forall x, Q(x)) . \]

Exercises
1. Let \(D\) denote the set of all people, and predicate \(L(x, y)\) denote that person \(x\) loves person \(y\). Represent mathematically the statement “love between two people is mutual”.
2. Let \(\mathbb{R}\) be the set of all real numbers. What is the negation of the statement
\[ \forall x \in \mathbb{R}, \ (x > 3) \rightarrow (x^2 > 4) ? \]
Write down this negation as a sentence. Is this negated statement true?
3. Suppose a student will pass this unit if he or she does all the assignments. According to this statement, is there a sufficient condition for a student to pass this unit? If yes, what is it? Is there a necessary condition for a student to pass this unit? If yes, what is it?
9 Graphs and Their Basic Types

The type of graphs that you will learn about in the following notes have an important role to play in computer science and areas of mathematics like networks analysis and operations research. They are also an important component of mathematics in their own right.

9.1 Graphs and Directed Graphs

Definition 9. A graph $G$ consists of

$\triangleright$ a non-empty set $V$ containing all vertices of $G$,

$\triangleright$ a set $E$ containing all edges of $G$,

$\triangleright$ an edge-endpoint function $\sigma$ on $G$ which associates to each edge a unique pair of end points on $V$.

It is denoted by $G = (V, E, \sigma)$, or often simply by $G = (V, E)$. Also, for a given graph $G$, we use $V(G)$ and $E(G)$ to denote respectively the corresponding vertex set $V$ and edge set $E$.

A picture of the graph $G$ is a diagram consisting of points corresponding to the vertices and lines corresponding to edges.

Notes.

$\bullet$ A graph is by default an undirected graph, which means a typical edge $e$ is represented by the set $\{v, w\}$ of its vertices through the edge-endpoint function $\sigma$, i.e. $\sigma(e) = \{v, w\}$.

$\bullet$ A graph is called a directed graph, or digraph, if a typical edge in $E$ is represented by an ordered pair $(v, w)$ of its vertices, i.e. $\sigma(e) = (v, w)$. An edge in a digraph is sometimes also called a directed edge or an arc.

$\bullet$ Two vertices in a graph $G$ are called adjacent iff there is an edge (arc) connecting them.

Examples

1. In the graph below heavy dots denote vertices, lines or arcs denote edges,

The vertex set is $V = \{v_1, v_2, v_3, v_4, v_5\}$, the edge set is $E = \{e_1, e_2, e_3, e_4, e_5\}$ and the edge-endpoint function $\sigma$ is given by

<table>
<thead>
<tr>
<th>edge $e$</th>
<th>endpoints set $\sigma(e)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_1$</td>
<td>${v_1}$</td>
</tr>
<tr>
<td>$e_2$</td>
<td>${v_1, v_2}$</td>
</tr>
<tr>
<td>$e_3$</td>
<td>${v_2, v_3}$</td>
</tr>
<tr>
<td>$e_4$</td>
<td>${v_2, v_3}$</td>
</tr>
<tr>
<td>$e_5$</td>
<td>${v_2, v_3}$</td>
</tr>
</tbody>
</table>
Note. The $e_1$ type edge is called a loop, and $e_3$, $e_4$, $e_5$ type edges are called multiple edges or parallel edges.

2. In the directed graph below

the vertex set is $V = \{v_1, v_2, v_3\}$, the edge set is $E = \{e_1, e_2, e_3, e_4\}$, and the edge-endpoint function $\sigma$ is given by

<table>
<thead>
<tr>
<th>edge $e$</th>
<th>endpoints pair $\sigma(e)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_1$</td>
<td>$(v_1, v_1)$</td>
</tr>
<tr>
<td>$e_2$</td>
<td>$(v_1, v_2)$</td>
</tr>
<tr>
<td>$e_3$</td>
<td>$(v_3, v_2)$</td>
</tr>
<tr>
<td>$e_4$</td>
<td>$(v_2, v_3)$</td>
</tr>
</tbody>
</table>

9.2 Basic Types and Features of Graphs

- A simple graph is a graph that has neither loops nor parallel edges. In such cases, the identification of an edge $e$ with its endpoints $\sigma(e)$ will not cause confusion. Thus an edge with endpoints $v$ and $w$ may be denoted by $\{v, w\}$ in simple graphs.

- A complete graph is a simple graph in which each pair of vertices are joined by an edge.
Example

3. The complete graph with \( n \) vertices is denoted by \( K_n \). The first few are

\[
\begin{array}{c}
\text{K1} & \text{K2} & \text{K3} & \text{K4} \\
\end{array}
\]

A graph is **complete bipartite** if it is a simple graph, and its vertices can be put into two groups so that any pair of vertices from different groups are joined by an edge while no pair of vertices are joined by an edge if they are from a same group. Such a graph is denoted by \( K_{m,n} \) if the two groups contains exactly \( m \) and \( n \) vertices respectively.

Example

4. Some complete bipartite graphs are

\[
\begin{array}{c}
\text{K1,2} & \text{K2,3} & \text{K3,3} \\
\end{array}
\]

A graph \( H \) is a **subgraph** of graph \( G \) iff

(i) \( V(H) \subseteq V(G) \),
(ii) \( E(H) \subseteq E(G) \),
(iii) every edge in \( H \) has same endpoints as in \( G \).

Examples

5. Suppose \( V = \{v_1, v_2, v_3\} \), \( E = \{e\} \) and let \( \sigma_1(e) = \{v_1, v_2\} \), \( \sigma_2(e) = \{v_2, v_3\} \), then graph \( G_1(V, E, \sigma_1) \) and graph \( G_2 = (V, E, \sigma_2) \) can be drawn as
Although it is obvious that
\[ V(G_2) \subseteq V(G_1) = V, \quad E(G_2) \subseteq E(G_1) = \{e\}, \]
the endpoints of \( e \) in \( G_2 \) are different from those of \( e \) in \( G_1 \). Hence \( G_2 \) is not a subgraph of \( G_1 \).

**Note.** Intuitively we see from the diagram that \( e = \{v_2, v_3\} \) in \( G_2 \) is not an “edge” of \( G_1 \) breaks the condition (ii) in this definition of a subgraph. This argument is however based on the implicit edge association with the corresponding end points. In other words, condition (iii) may be dropped, as in some texts, if the condition (ii) is extended in this sense.

6. For graph \( G \), graph \( H \) and graph \( F \) are both subgraphs of \( G \):

**Note.** An **ordered edge list** is a list of the edges using the vertices to define the edges and to give a direction along these edges. For example, in graph \( G \) above, an ordered edge list would be \( \{v_1v_2, v_1v_4, v_2v_3, v_3v_4\} \).

- The complement (inverse) of a graph \( G \) is a graph \( H \) on the same vertices such that two vertices of \( H \) are adjacent if and only if they are not adjacent in \( G \). That is, to generate the complement of a graph
  - fill in all the missing edges required to form a complete graph
  - remove all that were previously there

The simplest non-trivial complementary graphs are the 4-vertex path graph and the 5-vertex cycle graphs:
The complement of an edgeless graph is a complete graph and vice-versa.

For any vertex $v$ of a graph $G$, its degree is the number of incidences of edges at the vertex, and is denoted by $\delta(v)$, or $\deg(v)$. Notice that a loop is a special case and adds two to the degree.

**Example**

7. For graph

![Graph Diagram]

we have $\delta(v_1) = 5$, $\delta(v_2) = 4$, $\delta(v_3) = 1$ and $\delta(v_4) = 0$.

**Note.** For any graph $G$, if we denote by $E$ the total number of edges, and by $N$ the sum of the total degrees of all the vertices, then $N = 2E$. This is because every edge will supply exactly 1 to the degrees of each of its two end vertices. Hence $N$ must be always even. For the graph in the above example, for instance, we have $E = 5$ and $N = \delta(v_1) + \delta(v_2) + \delta(v_3) + \delta(v_4) = 5 + 4 + 1 + 0 = 10$, i.e., the condition $N = 2E$ is satisfied.

Let $v$ and $w$ be vertices of a graph $G$. Then a walk from $v$ to $w$ is a sequence of the form

$$v_0, e_1, v_1, e_2, \cdots, v_{n-1}, e_n, v_n$$

or alternatively

$$v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \cdots \xrightarrow{e_{n-1}} v_{n-1} \xrightarrow{e_n} v_n,$$

or simply $v_0, e_1, v_2, e_2, \cdots, v_{n-1}, e_n, v_n$, such that $v_0 = v$, $v_n = w$ and $v_{i-1}$ and $v_i$ are the endpoints of $e_i$, for $i = 1, \ldots, n$. The number of edges, $n$, is called the length of the walk. A trivial walk consists of a single vertex, and has thus a zero length.

A graph $G$ is connected iff every pair of vertices of $G$ is joined by a walk; otherwise, the graph is disconnected. A bridge is an edge whose removal will cause the graph to become disconnected.

**Note.** Some texts use the word path instead of the word walk, while others may use path to denote a walk with distinct edges.
Example

8. Graph $G$ is disconnected, because we can’t walk from vertex $v_4$ to (for instance) vertex $v_1$, i.e., there is no walk from $v_4$ to $v_1$.

![Graph diagram]

Obviously $G$ can be decomposed into three connected components $G_1$, $G_2$ and $G_3$.

- $V(G_1) = \{v_1, v_2, v_3\}$, $E(G_1) = \{e_1, e_2\}$
- $V(G_2) = \{v_4\}$, $E(G_2) = \emptyset$ (the empty set)
- $V(G_3) = \{v_5, v_6, v_7\}$, $E(G_3) = \{e_3, e_4, e_5\}$

This way each of $G_1$, $G_2$ and $G_3$ is a connected subgraph.

9.3 Planar Graphs

A graph is planar iff it can be drawn in a 2-dimensional plane without any accidental crossing. We say that a graph can be embedded in the plane, if it is planar. A planar graph divides the plane into regions (bounded by the edges), called faces. The following planar graph has four faces:

![Planar graph diagram]

We introduce without proof the following important results.

**Theorem 10. (Euler's formula)** For any connected graph $G = (V, E)$, with $n$ vertices ($|V| = n$), $m$ edges ($|E| = m$) and $f$ faces

$$n - m + f = 2$$

**Theorem 11. (Kuratowski’s Theorem)** A graph is nonplanar if and only if it can be obtained from either $K_5$ or $K_{3,3}$ by adding some, or no, vertices and edges.
Examples

9. Graph (a) is planar because it can be drawn as (b) but graph (c) is nonplanar because, after removing edges \( e \) and \( f \), and vertex \( v \), the graph becomes \( K_{3,3} \). Hence Kuratowski’s Theorem implies the original graph is nonplanar. We note that removal of vertex like \( v \) (and then connecting the end points \( v_1 \) and \( v_2 \)) is called **series reduction**. If a graph is planar, then obviously any of its subgraphs is also planar, even if series reductions are further performed. Hence Kuratowski’s Theorem can be rephrased as saying a graph is nonplanar iff it contains a subgraph which, after series reduction, is \( K_5 \) or \( K_{3,3} \).

The algorithmic testing of planarity of a graph is actually very complicated and relies on various other results. For reference pointers, see the book by Narsingh Deo, *Graph Theory with Applications to Engineering and Computer Science*, Prentice-Hall, 1974.

Note. In some texts an edge \( e \) in a graph \( G \) is actually implicitly regarded as an entity containing the edge label \( e \) and the corresponding endpoints, i.e. \((e, \sigma(e))\) where \( \sigma \) is the edge-endpoint function. Since for a given graph \( G \) a label \( e \) is sufficient to determine uniquely the edge entity \((e, \sigma(e))\), a graph \( G \) may be regarded as consisting of only its vertices \( V(G) \) and its edges \( E(G) \) in this generalised sense, as the endpoint function \( \sigma \) is now absorbed into the edges. This way a graph \( H \) is a subgraph of \( G \), for instance, iff \( V(H) \subseteq V(G) \) and all edges of \( H \) are also edges of \( G \).

Exercises

1. Construct a connected graph of 4 vertices of degrees 1, 2, 3 and 4 respectively, assuming the graph has no multiple edges.

2. Let \( n \geq 1 \). How many edges are there in \( K_{n,2n} \)? What is the length of a longest walk in \( K_{n,2n} \) from one vertex to another without walking on any edge and any vertex more than once?
10 Eulerian and Hamiltonian Circuits

10.1 Two Practical Problems

In order to see how graphs can be used to denote, interpret, or even solve some practical problems, we first give below two well-known cases.

Seven bridge problem (Bridges of Königsberg). The city of Königsberg occupied both banks and two islands of a river. The islands and the riverbanks were connected by seven bridges, as indicated in Figure 7. The question is if there is a walk around the city that crosses each bridge exactly once.

![Figure 7: The city of Königsberg](image)

Is it possible to walk over each bridge exactly once, starting from one of the locations $rb_1, rb_2, i_1, i_2$ and ending at the same location? If we use vertices to denote the locations and edges to denote bridges, then the map is simplified to the graph below:

![Figure 8: The graph of the city of Königsberg bridge system](image)
Travelling salesman problem. Given a collection of cities and the cost of travel between each pair of them, the traveling salesman problem, or TSP for short, is to find the cheapest way of visiting all of the cities and returning to your starting point.

Suppose the distances between each pair of the cities A, B, C and D are given, and suppose a salesman must travel to each city exactly once, starting and ending at city A. In the standard version we study, the travel costs are symmetric in the sense that traveling from city A to city B costs just as much as traveling from B to A. Which route from city to city will minimize the total travelling distance? If we use vertices to denote cities, and put the distance between any two cities on the edge joining them, then we can represent the given knowledge by the following weighted graph:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>30</td>
<td>50</td>
<td>40</td>
</tr>
<tr>
<td>B</td>
<td>30</td>
<td></td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>30</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Before considering the above two problems in detail, we first introduce below a few related basic terminology.

10.2 Eulerian and Hamiltonian Paths and Circuits

Let G be a connected graph.

- A **circuit** is a walk that starts and ends at a same vertex, and contains no repeated edges.
- An **Eulerian circuit** in a graph G is a circuit that includes all vertices and edges of G. A graph which has an Eulerian circuit is an **Eulerian graph**.
- An **Eulerian path** in a graph G is a walk from one vertex to another, that passes through all vertices of G and traverses exactly once every edge of G. An Eulerian path is therefore not a circuit.
- A **Hamiltonian circuit** in a graph G is a circuit that includes every vertex (except first/last vertex) of G exactly once.
- A **Hamiltonian path** in a graph G is a walk that includes every vertex of G exactly once. A Hamiltonian path is therefore not a circuit.
Examples

1. In the following graph

(a) walk $v_1e_1v_2e_3v_3e_4v_1$, loop $v_2e_2v_2$ and vertex $v_3$ are all circuits, but vertex $v_3$ is a trivial circuit.

(b) $v_1e_1v_2e_2v_3v_3e_4v_1$ is an Eulerian circuit but not a Hamiltonian circuit.

(c) $v_1e_1v_2e_2v_3e_4v_1$ is a Hamiltonian circuit, but not an Eulerian circuit.

2. $K_3$ is an Eulerian graph, $K_4$ is not Eulerian.

3. The following graph has an Eulerian path but is not Eulerian:

Theorem 12. (Euler’s Theorem) Let $G$ be a connected graph.

(i) $G$ is Eulerian, i.e. has an Eulerian circuit, if and only if every vertex of $G$ has even degree.

(ii) $G$ has an Eulerian path, but not an Eulerian circuit, if and only if $G$ has exactly two vertices of odd degree. The Eulerian path in this case must start at any of the two odd-degree vertices and finish at the other vertex.

Proof. We only consider the case (i).

(a) We first show $G$ is Eulerian implies all vertices have even degree.

Let $C$ be an Eulerian (circuit) path of $G$ and $v$ an arbitrary vertex. Then each edge in $C$ that enters $v$ must be followed by an edge in $C$ that leaves $v$. Thus the total number of edges incident at $v$ must be even.
(b) We then show by induction that $G$ is Eulerian if all of its vertices are of even degree.

Let $S_n$ be the statement that a connected graph of $n$ vertices must be Eulerian if every vertex has even degree.

For $n = 1$, $G$ is either a single vertex or a single vertex with loops. Hence $S_1$ is true because an Eulerian circuit can be obtained by traversing all loops (if any) one by one.

For induction we now assume $S_k$ is true, and $G$ has $k + 1$ vertices. Select a vertex $v$ of $G$. We form a subgraph $G'$ with one vertex less as follows: remove all loops attached to $v$ and break all remaining edges incident at $v$; remove $v$ and connect in pairs the broken edges in such a way $G$ remains connected. Since the degrees of the vertices remain even when $G$ is reduced to $G'$, the induction assumption implies the existence of an Eulerian circuit of $G'$. The Eulerian circuit of $G$ can thus be constructed by traversing all loops (if any) at $v$ and then the Eulerian circuit of $G'$ starting and finishing at $v$. Hence $G$ is Eulerian and $S_{k+1}$ is true, implying $S_n$ is true for all $n \geq 1$.

Examples

4. The following figure illustrates Euler’s theorem. Pictures (a) and (b) have no nodes of odd degree and must, therefore, have an Euler circuit. For instance, the circuit $B, C, D, A, C, A, B$ for picture (b). (note that the A-C-A portion of the circuit can be traversed in either of two ways). Pictures (c) and (d) have exactly two nodes of odd degree, thus they must contain an Euler path. For instance, $A, B, D, E, C, A, F, C, B, E, F$ is an Euler path for picture (d). It is impossible to find an Euler path that does not begin at $A$ and terminate at $F$ (or vice-versa). Finally, pictures (e) and (f) have four nodes of odd degree, and thus neither an Euler circuit nor an Euler path can be found on either. Picture (f) is actually the representation of the Königsberg bridges problem.
5. As for the travelling salesman problem, we need to find all the Hamiltonian circuits for the graph, calculate the respective total distance and then choose the shortest route.

<table>
<thead>
<tr>
<th>route</th>
<th>total distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A \rightarrow B \rightarrow C \rightarrow D \rightarrow A$</td>
<td>$30 + 30 + 25 + 40 = 125$</td>
</tr>
<tr>
<td>$A \rightarrow B \rightarrow D \rightarrow C \rightarrow A$</td>
<td>140</td>
</tr>
<tr>
<td>$A \rightarrow C \rightarrow B \rightarrow D \rightarrow A$</td>
<td>155</td>
</tr>
<tr>
<td>$A \rightarrow C \rightarrow D \rightarrow B \rightarrow A$</td>
<td>140</td>
</tr>
<tr>
<td>$A \rightarrow D \rightarrow B \rightarrow C \rightarrow A$</td>
<td>155</td>
</tr>
<tr>
<td>$A \rightarrow D \rightarrow C \rightarrow B \rightarrow A$</td>
<td>125</td>
</tr>
</tbody>
</table>

Hence the best route is either $ABCDA$ or $ADCBA$.

10.3 Fleury’s Algorithm for Finding an Eulerian Path or Circuit

**Algorithm 10.1 Fleury’s Algorithm**

(i) If there are odd degree vertices (there then must be exactly two if an Eulerian path is to exist), choose one. Travel over any edge whose removal will not result in breaking the graph into disconnected components.

(ii) Rub out the edge (or colour the edge if you like) you have just traversed, and then travel over any remaining edge whose removal will not result in breaking the remaining subgraph into disconnected components.

(iii) Repeat (ii) until other edges are rubbed out or coloured.


**Example**

6. Find an Eulerian path for the graph $G$ below by using Fleury’s algorithm:
We start at $v_5$ because $\delta(v_5) = 5$ is odd. We can’t choose edge $e_5$ to travel next because the removal of $e_5$ breaks $G$ into 2 connected parts. However we can choose $e_6$ or $e_7$ or $e_9$. We choose $e_6$. One Eulerian path is thus

$$v_5 \to v_5 \to v_5 \to v_7 \to v_8 \to v_9 \to v_3 \to v_1 \to v_2 \to v_4$$

**Exercises**

1. Is $K_4$ Eulerian? Is $K_{2,3}$ Eulerian?
2. Find an Eulerian circuit for $K_{2,4}$.
3. Does $K_{2,3}$ contain a Hamiltonian circuit? Does it contain a Hamiltonian path?
11 Graph Isomorphism and Matrix Representations

Here will look at when two graphs are the same - the idea equivalence cropping up again. Also we consider how a graph can be represented by a matrix which allows for computers to be used to analyse graphs.

11.1 Graph Isomorphism and Isomorphic Invariants

We recall that \( V(G) \) and \( E(G) \) denote the vertex set and the edge set of the graph \( G \), respectively.

**Definition 13.** Two graphs \( G_1 \) and \( G_2 \) are **isomorphic**, written \( G_1 \cong G_2 \), if there exist bijections \( g: V(G_1) \to V(G_2) \) and \( h: E(G_1) \to E(G_2) \) such that for any \( v \in V(G_1) \) and any \( e \in E(G_1) \), \( v \) is an endpoint of \( e \) if and only if \( g(v) \) is an endpoint of \( h(e) \).

The pair of functions \( g \) and \( h \) is called a **graph isomorphism**. There are many applications of graph isomorphism, for instance in organic chemistry: determining if two molecule representations are identical.

It is easy to see that a graph is always isomorphic to itself. The obvious isomorphism is given by the identity maps. Just map each vertex to itself and each edge to itself. An isomorphism from a graph to itself is called an **automorphism**.

Roughly speaking, graphs \( G_1 \) and \( G_2 \) are isomorphic to each other if they are “essentially” the same. More intuitively, if graphs are made of elastic bands (edges) and knots (vertices), then two graphs are isomorphic to each other if and only if one can stretch, shrink and twist one graph so that it can sit right on top of the other graph, vertex to vertex and edge to edge. The isomorphism functions \( g \) and \( h \) will thus provide the one-to-one correspondences for the vertices and the edges respectively.

**Example**

1. Show graphs \( G_1 \) and \( G_2 \) below are isomorphic.

   ![Graphs G1 and G2](image)

   **Solution.** How to find isomorphism function \( g \) and \( h \) in general will be clearer when we introduce the concept of isomorphism invariants later on. But at this stage it is mostly...
guesswork. Let \(g\) and \(h\) be given by

\[
\begin{align*}
g(v_1) &= w_2, & g(v_2) &= w_1, & g(v_3) &= w_4, & g(v_4) &= w_3, \\
h(e_1) &= f_2, & h(e_2) &= f_1, & h(e_3) &= f_4, & h(e_4) &= f_3, & h(e_5) &= f_5
\end{align*}
\]

which can be alternatively represented in the diagrams below

\[
\begin{align*}
V(G_1) \xrightarrow{g} V(G_2) & & E(G_1) \xrightarrow{h} E(G_2)
\end{align*}
\]

We now verify the preservation of endpoints under \(g\) and \(h\)

\[
\begin{align*}
e_1 = \{v_1, v_2\} & \rightarrow f_2 = \{w_1, w_2\} = \{g(v_1), g(v_2)\} \\
e_2 = \{v_2, v_4\} & \rightarrow f_1 = \{w_1, w_3\} = \{g(v_2), g(v_4)\} \\
e_3 = \{v_2, v_3\} & \rightarrow f_4 = \{w_1, w_4\} = \{g(v_2), g(v_3)\} \\
e_4 = \{v_1, v_4\} & \rightarrow f_3 = \{w_2, w_3\} = \{g(v_1), g(v_4)\} \\
e_5 = \{v_3, v_4\} & \rightarrow f_5 = \{w_3, w_4\} = \{g(v_3), g(v_4)\}
\end{align*}
\]

where we used \(e = \{v, w\}\) to indicate that edge \(e\) has endpoints \(\{v, w\}\). Since \(g\) and \(h\) are obviously one-to-one and onto, the pair \(g\) and \(h\) thus constitute an isomorphism of graphs \(G_1\) and \(G_2\), i.e. \(G_1\) and \(G_2\) are isomorphic.

Isomorphic graphs are “same” in shapes, so properties on “shapes” will remain invariant for all graphs isomorphic to each other. More precisely, a property \(P\) is called an isomorphic invariant if and only if given any graphs isomorphic to each other, all the graphs will have property \(P\) whenever any one of the graphs does. There are many isomorphic invariants, e.g.

\[
\begin{align*}
\text{(a)} & \quad \text{vertices of a given degree}, & \text{(e)} & \quad \text{number of loops at a vertex}, \\
\text{(b)} & \quad \text{number of edges}, & \text{(f)} & \quad \text{number of sets of parallel edges}, \\
\text{(c)} & \quad \text{number of connected components}, & \text{(g)} & \quad \text{has a Hamiltonian circuit}. \\
\text{(d)} & \quad \text{has a circuit of given length},
\end{align*}
\]

Incidentally, an isomorphic invariant is sometimes also referred to as an isomorphism invariant.
Examples

2. Graphs $G_1$ and $G_2$ below are not isomorphic to each other because vertex $v$ of $G_1$ has degree 5 while no vertices of $G_2$ have degree 5.

3. Graphs $G_1$ and $G_2$ below are isomorphic to each other because $|V(G_1)| = |V(G_2)| = 5$, $|E(G_1)| = E(G_2)| = 8$, each graph has a vertex of degree 4, and other four vertices of degree 3, etc.

4. Back to example 1. We now explain briefly how we found the isomorphism functions $g$ and $h$ there.

First, since $v_2, v_4$ in $G_1$ and $w_1$ and $w_3$ in $G_2$ are the only vertices of degree 3, $g$ must map $v_2, v_4$ to $w_1, w_3$ or $w_3, w_1$ respectively. We thus choose $g(v_2) = w_1$ and $g(v_4) = w_3$. Since $\{v_2, v_4\}$ are the endpoints of $e_2$, we must have $h(e_2) = f_1$ so that the endpoints $\{v_2, v_4\}$ of edge $e_2$ are preserved because $f_1$ has endpoints $\{w_1, w_3\} = \{g(v_2), g(v_4)\}$.

Next we need to map $v_1, v_3$ to $w_2, w_4$ or $w_4, w_2$ respectively. If we choose $w_2 = g(v_1)$ we must have $w_4 = g(v_3)$. Thus the edge $e_1$ joining $v_1$ and $v_2$ should be mapped to the edge $f_2$ joining $g(v_1) = w_2$ and $g(v_2) = w_1$, i.e. $f_2 = h(e_1)$. The rest of $g$ and $h$ can be determined similarly.

11.2 Matrices

A matrix is a rectangular array of numbers (sometimes symbols or expressions) placed at the intersections of rows with columns. These numbers are called the elements of the matrix. A $m$–by–$n$ matrix has $m$ rows and $n$ columns and $m \times n$ is called the size of the matrix. A squared matrix has the number of rows equal to the number of columns, for instance $n \times n$. Usually, the elements of a matrix are denoted by a variable with two subscripts, one for its row and one for
its column. For instance, $a_{35}$ represents the element at the third row and fifth column of a matrix $A = (a_{ij})_{m \times n}$.

Some operations can be defined on matrices:

- Matrices of the same size can be added $(A + B)$ or subtracted $(A - B)$ element by element.
- The scalar multiplication $kA$ of a matrix $A$ and a number $k$ is given by multiplying every element of $A$ by $k$.
- Multiplication of two matrices is defined only if the number of columns of the left matrix is the same as the number of rows of the right matrix, that is given $A_{m \times n}$ and $B_{n \times p}$ then $A_{m \times n} \times B_{n \times p} = C_{m \times p}$. The elements of $C = (c_{ij})$ are given by dot product (sum of element by element products) of the corresponding row ($i$th) of $A$ and the corresponding column ($j$th) of $B$, that is if

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1j} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2j} & \cdots & b_{2p} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{n1} & b_{m2} & \cdots & b_{nj} & \cdots & b_{np} \end{bmatrix}$$

then matrix $C = A \times B$ has $m$ rows and $p$ columns

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1j} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2j} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{i1} & c_{i2} & \cdots & c_{ij} & \cdots & c_{ip} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mj} & \cdots & c_{mp} \end{bmatrix}$$

where $c_{ij} = (a_{i1}, a_{i2}, \cdots, a_{in}) \cdot (b_{1j}, b_{2j}, \cdots, b_{nj}) = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj}$ or, using summation notation, this can be written much more concisely as

$$c_{ij} = \sum_{k=1}^{n} a_{ik}b_{kj}.$$ 

Note that if $A = (a_{ij})$ and $B = (b_{ij})$ are two $n \times n$ matrices, the product of $A$ and $B$, i.e. $AB$, is always possible and it is another $n \times n$ squared matrix $C = (c_{ij})$ in which $c_{ij} = \sum_{k=1}^{n} a_{ik}b_{kj}$. To brush up on the matrix multiplications, we look at two simple examples below.
Examples

5. Let $2 \times 2$ (i.e. 2 by 2) matrices $A$ and $B$ be given respectively by

$$A = \begin{bmatrix} 2 & -1 \\ 5 & 3 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 7 & 4 \\ 1 & -2 \end{bmatrix}. $$

Find $A + B$, $AB$ and $A^2$.

Solution.

$$A + B = \begin{bmatrix} 2 + 7 & -1 + 4 \\ 5 + 1 & 3 - 2 \end{bmatrix} = \begin{bmatrix} 9 & 3 \\ 6 & 1 \end{bmatrix}$$

$$AB = \begin{bmatrix} 2 & -1 \\ 5 & 3 \end{bmatrix} \times \begin{bmatrix} 7 & 4 \\ 1 & -2 \end{bmatrix} = \begin{bmatrix} 2 \times 7 + (-1) \times 1 & 2 \times 4 + (-1) \times (-2) \\ 5 \times 7 + 3 \times 1 & 5 \times 4 + 3 \times (-2) \end{bmatrix} = \begin{bmatrix} 13 & 10 \\ 38 & 14 \end{bmatrix}$$

$$A^2 = \begin{bmatrix} 2 & -1 \\ 5 & 3 \end{bmatrix} \times \begin{bmatrix} 2 & -1 \\ 5 & 3 \end{bmatrix} = \begin{bmatrix} 2 \times 2 + (-1) \times 5 & 2 \times (-1) + (-1) \times 3 \\ 5 \times 2 + 3 \times 5 & 5 \times (-1) + 3 \times 3 \end{bmatrix} = \begin{bmatrix} -1 & -5 \\ 25 & 4 \end{bmatrix}$$

6. Let $3 \times 3$ matrices $A$ and $B$ be given by

$$A = \begin{bmatrix} 2 & -1 & 13 \\ 5 & 3 & -6 \\ 11 & 0 & 10 \end{bmatrix} \quad B = \begin{bmatrix} 7 & 4 & -8 \\ 1 & -2 & 0 \\ 9 & -4 & -3 \end{bmatrix}$$

Then the product matrix $AB$ is

$$AB = \begin{bmatrix} 2 & -1 & 13 \\ 5 & 3 & -6 \\ 11 & 0 & 10 \end{bmatrix} \times \begin{bmatrix} 7 & 4 & -8 \\ 1 & -2 & 0 \\ 9 & -4 & -3 \end{bmatrix}$$

$$= \begin{bmatrix} 2 \times 7 + (-1) \times 1 + 13 \times 9 & 2 \times 4 + (-1) \times (-2) + 13 \times (-4) & 2 \times (-8) + (-1) \times 0 + 13 \times (-3) \\ 5 \times 7 + 3 \times 1 + (-6) \times 9 & 5 \times 4 + 3 \times (-2) + (-6) \times (-4) & 5 \times (-8) + 3 \times 0 + (-6) \times (-3) \\ 11 \times 7 + 0 \times 1 + 10 \times 9 & 11 \times 4 + 0 \times (-2) + 10 \times (-4) & 11 \times (-8) + 0 \times 0 + 10 \times (-3) \end{bmatrix}$$

$$= \begin{bmatrix} 130 & -42 & -55 \\ -16 & 38 & -22 \\ 167 & 4 & -118 \end{bmatrix}$$
11.3 The Adjacency Matrix

Given a directed or undirected graph $G$ of $n$ vertices $v_1, \ldots, v_n$, we can represent the graph by an $n \times n$ matrix $A$ over $\mathbb{N}$, i.e.,

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}, \quad a_{ij} \in \mathbb{N}$$

in which the element $a_{ij}$ is the number of arrows from $v_i$ to $v_j$ if $G$ is a directed graph, or $a_{ij}$ is the number of edges connecting $v_i$ to $v_j$ if $G$ is an undirected graph. This matrix $A$ is then said to be the adjacency matrix of the graph $G$.

We note that a matrix is essentially just a table, and an adjacency matrix basically represents a table of nonnegative integers which correspond to the number of edges between different pairs of vertices.

For a simple graph, the adjacency matrix must have 0s on the diagonal. For an undirected graph, the adjacency matrix is symmetric.

Examples

7. The adjacency matrix of digraph on the right is

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 2 \\ 1 & 0 & 0 \end{bmatrix}$$

8. The adjacency matrix of graph on the right is

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 0 \end{bmatrix}$$

Theorem 14. Let $G$ be a directed or undirected graph of $n$ vertices $v_1, v_2, \ldots, v_n$ and $A$ be the adjacency matrix of $G$. Then for any positive integer $m$, the $(i, j)$-th entry of $A^m$ is equal to the number of walks of length $m$ from $v_i$ to $v_j$, where $i, j = 1, 2, \ldots, n$.

Proof. Let $S_m$ denote the statement that

$(i, j)^{th}$ entry of $A^m$ is equal to the number of walks of length $m$ from $v_i$ to $v_j$. 


Then for \( m = 1 \), \( S_1 \) is true because the adjacency matrix \( A \) is defined that way. For the induction purpose we now assume \( S_k \) is true with \( k \geq 1 \). Let \( B = (b_{ij}) \) def \( = A^k \), then \( b_{sj} = \) the number of walks of length \( k \) from \( v_s \) to \( v_j \) due to the induction assumption. Hence

\[
(i, j)-\text{th element of } A^{k+1} = (i, j)-\text{th element of } AB = a_{i1}b_{1j} + a_{i2}b_{2j} + \ldots + a_{in}b_{nj}
\]

the number of walks of length \( k + 1 \) from \( v_i \) to \( v_j \) that have \( v_s \) as their 2nd vertex

\[
= \text{the number of walks of length } k + 1 \text{ from } v_i \text{ to } v_j
\]

(taking any vertex as the 2nd vertex)

i.e. \( S_{k+1} \) is true. Hence \( S_m \) is true for all \( m \geq 1 \), and the proof of the theorem is thus completed.

**Example**

9. Consider the digraph on the right, which has the adjacency matrix \( A \) below

\[
A = \begin{bmatrix}
1 & 1 & 0 \\
1 & 0 & 0 \\
1 & 1 & 0
\end{bmatrix}
\]

Since \( A^2 = \)

\[
\begin{bmatrix}
1 & 1 & 0 \\
1 & 0 & 0 \\
1 & 1 & 0
\end{bmatrix} \times \begin{bmatrix}
1 & 1 & 0 \\
1 & 0 & 0 \\
1 & 1 & 0
\end{bmatrix} = \begin{bmatrix}
1 \times 1 + 1 \times 1 + 0 \times 1 & 1 \times 1 + 0 \times 0 + 0 \times 1 & 1 \times 0 + 1 \times 0 + 0 \times 0 \\
1 \times 1 + 0 \times 1 + 0 \times 1 & 1 \times 1 + 0 \times 0 + 0 \times 1 & 1 \times 0 + 0 \times 0 + 0 \times 0 \\
1 \times 1 + 1 \times 1 + 0 \times 1 & 1 \times 1 + 1 \times 0 + 0 \times 1 & 1 \times 0 + 1 \times 0 + 0 \times 0
\end{bmatrix} = \begin{bmatrix}
2 & 1 & 0 \\
1 & 1 & 0 \\
2 & 1 & 0
\end{bmatrix}
\]

we see there are exactly 2 walks of length 2 that start at \( v_3 \) and end at \( v_1 \). Since

\[
A + A^2 + A^3 = \begin{bmatrix}
1 & 1 & 0 \\
1 & 0 & 0 \\
1 & 1 & 0
\end{bmatrix} + \begin{bmatrix}
1 & 1 & 0 \\
1 & 1 & 0 \\
1 & 1 & 0
\end{bmatrix} + \begin{bmatrix}
2 & 1 & 0 \\
1 & 1 & 0 \\
2 & 1 & 0
\end{bmatrix} = \begin{bmatrix}
6 & 4 & 0 \\
4 & 2 & 0 \\
6 & 4 & 0
\end{bmatrix}
\]

has only 0’s in the third column, we conclude that no vertex can reach \( v_3 \) via a walk of length 1, 2, or 3. This is because the least number of edges which must be traversed to go from vertex \( i \) to vertex \( j \) is given by considering the matrix \( S_k = A + A^2 + \cdots + A^k \).

The \((i, j)\)-entry in this matrix gives the number of ways to go from vertex \( i \) to vertex \( j \) in \( k \) steps or less. It means that we have to continue to compute \( S_k \) as \( k \) increases until its \((i, j)\)-entry is non-zero.
Exercises

1. Is graph $K_4$ isomorphic to graph $K_5$?

2. Does $K_5$ have a subgraph that is isomorphic to $K_4$?

3. Let undirected graphs $G_1$ and $G_2$ be represented by their corresponding adjacency matrices $A_1$ and $A_2$ respectively, with

\[
A_1 = \begin{pmatrix}
0 & 1 & 0 & 1 \\
1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 \\
1 & 1 & 1 & 1
\end{pmatrix}, \quad A_2 = \begin{pmatrix}
0 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 \\
1 & 1 & 1 & 0
\end{pmatrix}.
\]

Are graphs $G_1$ and $G_2$ isomorphic? If yes, give the vertex correspondence of the isomorphism. How many walks of length 2 are there in graph $G_1$?

4. Show the following 2 graphs are isomorphic by providing the vertex correspondence.

```plaintext
\text{G1} \quad \text{G2}
```
12 Trees

A tree is a special type of graph which is important in both theory and application. Many powerful algorithms in computer science and software engineering are tree based algorithms. You may have come across a tree when you looked at the way directories or folders are organised in a computer.

12.1 Trees, Rooted Trees and Binary Trees

- A **nontrivial circuit** (a cycle) is a circuit with at least one edge.
- A **tree** is a connected graph without nontrivial circuits.
- A **forest** is composed of one tree or some disconnected trees.
- A **terminating vertex** (or a **leaf**) in a tree is a vertex of degree 1.
- An **internal vertex** (or a **branch vertex**) in a tree is a vertex of degree greater than 1.
- Vertices are sometimes referred to as **nodes**, particularly when dealing with graph trees.

Examples

1. The following two graphs are trees:

2. This graph is not a tree:

Lemma 15. *Any tree with at least one edge must have at least one vertex of degree 1.*
Proof. We start from any vertex $v_0$ and walk along the edges not walked before, thus arriving at vertices $v_1, v_2, \cdots, v_n, \cdots$ in sequence. If all vertices of the tree had degree $\geq 2$, then such walking wouldn’t terminate at any vertex (because its degree $\geq 2$) without going back to one of the vertices walked over before. Since the walking has to terminate because we are only allowed to walk on the (finite number of) edges not walked before, we must go back to at least one of the vertices already walked over. A nontrivial circuit would thus be found. This would contradict the definition of a tree. Hence not all vertices have degree $\geq 2$, i.e., the Lemma is true.

Theorem 16. For any $n \geq 1$, a connected graph with $n$ vertices is a tree if and only if it has exactly $n - 1$ edges.

Proof. The theorem is equivalent to the following two parts:

(i) Statement $S_n$: A tree $T_n$ with $n$ vertices has $n - 1$ edges.
(ii) A connected graph with $n$ vertices and $n - 1$ edges is a tree.

Proof of (i) by induction on statement $S_n$. For $n = 1$, $S_1$ is true because $T_1$ has 1 vertex and 0 edges. Now assume $S_k$ is true. From the Lemma for $n = k + 1$, we can find a vertex $v_0$ of the tree $T_{k+1}$ such that $\delta(v_0) = 1$, see figure on the left. We then removed $v_0$ and its edge to obtain $T_k$, which is obviously still a tree (with $k$ vertices). Hence $T_k$ has $k - 1$ edges because of the induction assumption $S_k$ is true. Thus $T_{k+1}$ has $1 + (k - 1) = k$ edges, i.e., $S_{k+1}$ is also true, proving (i).

Proof of (ii). Let $G$ be a connected graph with $n$ vertices and $n - 1$ edges. We show $G$ is a tree by showing it has no nontrivial circuits. Assume otherwise, i.e., $G$ has a nontrivial circuit $H$. We show that this will lead to a contradiction. Since the removal of an edge from circuit $H$ won’t disconnect $G$, we can remove from $G$ sufficient edges ($\geq 1$) so that the resulting subgraph $G'$ has no nontrivial circuits while remaining connected with $n$ vertices. $G'$ now by definition is a tree, and should have $n - 1$ edges from (i). Hence $G$ must have more than $n - 1$ edges when the removed edges are added back. Thus $G$ has no nontrivial circuit. Consequently $G$ is a tree, proving (ii).
Example

3. List all (up to isomorphism) trees of 4 vertices.

From the Lemma, there exists a vertex, call it \( v_0 \), of degree 1. We denote by \( v_1 \) the only vertex which is connected to \( v_0 \). Since there are exactly 3 edges in a tree of 4 vertices, the highest degree \( v_1 \) can attain is 3. It is also obvious that \( v_1 \) must have at least 2 edges because otherwise edge \( \{v_0, v_1\} \) would be disconnected from the remaining vertices, which would contradict the definition of a tree. Hence we conclude \( 2 \leq \delta(v_1) \leq 3 \) and will enumerate the two cases below.

(a) \( \delta(v_1) = 3 \).

(b) \( \delta(v_1) = 2 \). This means \( v_1 \) is connected to another vertex \( v_2 \). Since only one extra edge is needed, the edge has to be attached to \( v_2 \).

Hence there are exactly 2 different trees, which are (a) and (b) respectively.

A **rooted tree** is a tree in which one vertex is designated as the **root** and has no parent. Every other node has exactly one parent. The **level** of a vertex is the number of edges in the unique walk between the vertex and the root. The **height** (or **depth**) of a tree is the maximum level of any vertex there.

A **binary tree** is a rooted tree in which each vertex has at most two children. Each child there is designated either a **left child** or a **right child**.
Algebraic expressions with binary operators have an inherent tree-like structure: the terminal nodes (leaves) of the tree are the variables or constants in the expression (a, b, c, d, ...), while the non-terminal nodes are operators (+, −, ×, /).

Example

4. Draw a binary tree to represent \((a - b) \cdot c + (d/e)\).

Solution.

The expression is made up of two parts separated by +, so that + becomes the root. Look at the expression on the left of the +, order of operation is important, and again see that the expression is made up of two parts separated by a ÷ or multiplication. This operation becomes the left child of +. Continue in this fashion to complete the left side of the tree then do the same for the right side.

12.2 Traversal of Binary Trees

To traverse a tree means to visit every one of its nodes once. At each step, we distinguish a root (the current node) and its left and right subtrees. These two subtrees are visited recursively, and the distinguishing feature is when the root is visited. Let r, R, L denote the root, the right
subtree and the left subtree, respectively. There are three classical ways of traversing a binary tree:

**Preorder traversal** (rLR)
- Visit the root;
- Traverse left subtree in pre-order;
- Traverse right subtree in pre-order.

**Inorder traversal** (LrR)
- Traverse left subtree in in-order;
- Visit the root;
- Traverse right subtree in in-order.

**Postorder traversal** (LRr)
- Traverse left subtree in post-order;
- Traverse right subtree in post-order;
- Visit the root.

**Examples**
5. For binary tree

The traversals are as follows.
- Preorder: \( a, b, d, c, e, f, g \)
- Inorder: \( d, b, a, f, e, g, c \)
- Postorder: \( d, b, f, g, e, c, a \)
6. The binary tree representation of \((a + b \cdot c)/d\) is easily seen as

\[
\begin{array}{c}
/ \\
+ \\
a \\
- \\
b \\
c \\
d
\end{array}
\]

But how is the calculation actually done? It may be processed via a postorder traversal

\[a, b, c, \cdot, +, d, /\] (†)

by interpreting each binary operation as acting on the 2 quantities immediately to its left. That is, \(\alpha, \beta, \gamma, \delta, *, \cdots\) means \(\alpha, \beta, (\gamma * \delta), \cdots\), if \(\alpha, \beta, \gamma, \delta\) are terms and \(*\) is any binary operation. This way, the above formula is processed successively in following sequence

\[
\begin{align*}
& a, b, c, \cdot, +, d, / \\
& a, b \cdot c, +, d, / \\
& a + b \cdot c, d, / \\
& (a + b \cdot c)/d
\end{align*}
\]

The sequence (†) is said to be in the **Polish postfix notation**.

7. Use a binary tree to sort the following list of numbers

\[15, 7, 24, 11, 27, 13, 18, 19, 9\]

We note that when a binary tree is used to sort a list, the inorder traversal will be automatically assumed in this unit.

**Solution.**

To sort a list, just create a binary tree by adding to the tree one item after another, because the insertion is essentially a sorting process. The binary tree constructed during the course of sorting the above list then reads
It is constructed in the order of A to I depicted below.

The sorted list is thus
7, 9, 11, 13, 15, 18, 19, 24, 27.

This is because, according to the inorder traversal, we have to visit the left subtree of (the root) 15 containing vertices 7, 11, 9 and 13 before visiting 15 itself and then its right subtree. However, to visit the subtree containing exactly the vertices 7, 11, 9 and 13 we have to visit first the (empty) left subtree of 7, then 7, then the right subtree of 7.
containing vertices 11, 9 and 13. Hence 7 is the very first vertex to be visited. To visit
the right subtree of 7 containing vertices 11, 9 and 13 we visit first 9, then 11 and then
13. Hence the first 4 vertices visited are 7, 9, 11 and 13. The rest is similar. We note that
a better tree sorting algorithm will involve balancing the trees. However such additional
features fall beyond the scope of the current unit.

Note. One of the immediate applications of binary trees is for the binary coding, in particular
for the Huffman coding. For further details, interested readers may consult the book Mathematical Structures for Computer Science by Judith L. Gersting, 3rd edition, W H Freeman

Exercises

1. Let \( n \geq 0 \) be an integer. Can a forest have \( n + 2 \) vertices and \( n \) edges? If yes, how many
   connected components will there be?

2. Construct, up to isomorphisms, all trees of 5 vertices.

3. Use a binary tree to sort alphabetically the list \( B, I, N, A, R, Y, L, U, S, T, F, O, E \).

4. For the binary tree below, list the vertices according to the preorder, inorder and postorder
   traversals respectively.

5. Prove by induction on the number of faces Euler’s formula

\[
n - m + f = 2
\]
13 Spanning Trees

Sometimes a graph contains "too" much information - it may be that all that is required is that the vertices be connected to each other by some path. An example of this would be how to work out the electric or optical fibre cabling to connect various locations in the least expensive way. This is where a spanning tree is useful.

13.1 The Concept of Spanning Trees

A spanning tree for a graph $G$ is a subgraph of $G$ that contains all vertices of $G$ and is a tree. Obviously every connected graph $G$ has a spanning tree. In fact, if one keeps breaking any remaining nontrivial circuit of $G$, or the intermediate subgraphs after such steps, by removing an edge from the circuit, then the final resulting subgraph will be a spanning tree.

Examples

1. Graph $G$ from (a) becomes (b) after removing edge $\{v_2, v_3\}$ from $G$ to break the cycle $v_5 \rightarrow v_2 \rightarrow v_3 \rightarrow v_5$, and becomes a spanning tree (c) after removing edge $\{v_4, v_3\}$ to break the last remaining cycle $v_5 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5$. Of course, there can be different spanning trees for the same connected graph $G$.

2. Suppose we want to establish a communications network among four centres $A$, $B$, $C$ and $D$, and due to geographical problems direct communications (via cables for instance) may only be established between the two centres connected by an edge is the graph. Suppose each communication link (edge) costs $1000$ to install. Since communications may be relayed via intermediate centres, we may choose a spanning tree to lay the communication
links so as to minimise the total cost. We now list below all possible spanning trees each will thus only cost $3000 to build, in comparison with the $5000 (5 edges) required for the original graph $G$.

3. Back to the graph $G$ in example 1, see the graph below

Recall that a spanning tree was obtained there by breaking all the cycles in the graph. However, we can also construct a spanning tree in the opposite way, that is, by gradually collecting edges from $G$, and making sure that the resulting subgraph is always connected and contains no cycles. For instance, a spanning tree $T_4$, different from that obtained in example 1, can derived from the following 4 steps.
Notice that neither $e_5$ nor $e_6$ can be added because that would create cycles. We stop the procedure when consideration for all edges are exhausted.

A technical problem in step 4 of example 3 is how do we know that connecting vertices $v_2$ and $v_5$ won’t create a cycle! This is not trivial, particularly when there are a great many vertices. Hence we need an effective algorithm.

### 13.2 Spanning Tree Algorithm

We may use the following procedure to find a spanning tree. It is a procedure which analyses edges one by one to see if it can be collected into the resulting spanning tree.

**Algorithm 13.1 Spanning Tree Algorithm**

1. Let $v_1, v_2, \cdots, v_n$ be all the vertices of a connected graph $G$.
2. List all edges in a sequence.
3. Attach a label (or an indicator) $L(i)$ to each vertex $v_i$, and initiate $L(i)$ to distinct values by setting $L(i) = i$ $(i = 1, 2, \cdots, n)$.
4. Let the resulting graph $T$ be initially empty.
5. Pick next available edge $e$ in the edge list in (2). Let $v_{i_0}$ and $v_{j_0}$ be the vertices of $e$.
   a. If $L(i_0) \neq L(j_0)$ (implying adding $e$ to $T$ will not create cycles),
      then add edge $e$ to $T$ and change all $L(k)$ which are equal to $L(i_0)$ or $L(j_0)$ to the minimum value $\min\{L(i_0), L(j_0)\}$.
   b. If $L(i_0) = L(j_0)$, ignore edge $e$.
6. Do (5) until all edges are considered.

**Note.** It is not difficult to see that step (6) above may also be replaced by

(6'). Do (5) until all labels $L(i)$ are equal to each other.

We also note that the values of the labels $L(i)$ are to be altered in step (5) above to keep track of the connectivity of any pair of vertices: two vertices, say $v_{i_0}$ and $v_{j_0}$, are connected through the edges already collected to the resulting spanning tree $T$ if and only if the corresponding labels have exactly the same value, i.e. $L(i_0) = L(j_0)$. In the algorithm, moreover, the resulting spanning tree $T$ will gradually grow from the empty tree $\emptyset$ at the very beginning to the full spanning tree at the end of the procedure.
Example

4. Find a spanning tree for the graph

\[ \begin{align*}
\text{Solution.} & \quad \text{Since the graph is already provided with a vertex list and an edge list, we might just as well use these lists as the outcome of steps (i) and (ii) though we could use lists of different orders. Steps (iii)--(vi) can be summarised in the following table.} \\
\begin{array}{|c|c|c|c|c|c|c|c|c|c|}
\hline
\text{step} & \text{edges} & \text{labels of the vertex list} & \text{edge added} \\
\hline
0 & \emptyset & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \text{initial values} \\
\hline
1 & e_1 & 1 & 2 & 2 & 4 & 5 & 6 & 7 & e_1 \\
2 & e_2 & 1 & 2 & 2 & 4 & 5 & 5 & 7 & e_2 \\
3 & e_3 & 1 & 2 & 2 & 4 & 5 & 5 & 5 & e_3 \\
4 & e_4 & 1 & 2 & 2 & 2 & 5 & 5 & 5 & e_4 \\
5 & e_5 & 1 & 2 & 2 & 2 & 5 & 5 & 5 & e_5 \\
6 & e_6 & 1 & 2 & 2 & 2 & 5 & 5 & 5 & e_7 \\
7 & e_7 & 1 & 1 & 1 & 1 & 5 & 5 & 5 & e_7 \\
8 & e_8 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & e_9 \\
9 & e_9 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & e_9 \\
10 & e_{10} & \text{no need to consider because all labels are now equal in value} & \text{edge added to tree} \\
\hline
\end{array}
\end{align*} \]

Notice that at step 1, since \( e_1 \) has vertices \( v_2 \) and \( v_3 \) and \( L(2) = 2 \neq L(3) = 3 \), we thus change \( L(2), L(3) \) both to 2 and add \( e_1 \) to the spanning tree. At step 5, since \( e_5 \) has vertices \( v_2 \) and \( v_4 \) and \( L(2) = L(4) = 2 \) (at the end of step 4), we know from (5) that we can’t add \( e_5 \) to the tree (because the addition of \( e_5 \) would create a cycle. At step 7, since \( e_7 \) has vertices \( v_1 \) and \( v_2 \) whose labels are 1 and 2 respectively, we change in this step all the labels with value either 1 or 2 to the min(1, 2)=1. At step 10, nothing needs to be considered because all labels are now equal at the end of step 9, implying the spanning tree is already obtained. The resulting spanning tree, collecting the edges in the last column, has edges \( \{e_1, e_2, e_3, e_4, e_7, e_9\} \), or
13.3 Kruskal’s Algorithm

A weighted graph $G$ is a graph for which each edge $e$ has an associated real number called weight. A minimal spanning tree of a weighted graph is a spanning tree that has the least possible total weight. We often use $w(e)$ to denote the weight of edge $e$ and $w(G)$, the total weight of $G$. Also, for simplicity here all weights are assumed to be non-negative.

The Spanning Tree Algorithm given earlier on in this lecture can be used to find a minimal spanning tree if we change step (2) there to the following step

2’. List all edges in the order of increasing weights.

Then the algorithm becomes Kruskal’s algorithm:

**Algorithm 13.2 Kruskal’s Algorithm**

1. Let $v_1, v_2, \cdots, v_n$ be all the vertices of a connected graph $G$.
2’. List all edges in the order of increasing weights.
3. Attach a label (or an indicator) $L(i)$ to each vertex $v_i$, and
   initiate $L(i)$ to distinct values by setting $L(i) = i$ ($i = 1, 2, \cdots, n$).
4. Let the resulting graph $T$ be initially empty.
5. Pick next available edge $e$ in the edge list in (2). Let $v_{i_0}$ and $v_{j_0}$ be the vertices of $e$.
   a. If $L(i_0) \neq L(j_0)$ (implying adding $e$ to $T$ will not create cycles),
      then add edge $e$ to $T$ and
      change all $L(k)$ which are equal to $L(i_0)$ or $L(j_0)$ to the minimum value $\min\{L(i_0), L(j_0)\}$.
   b. If $L(i_0) = L(j_0)$, ignore edge $e$.
6. Do (5) until all edges are considered.

**Examples**

5. Find a minimal spanning tree for the following weighted graph, in which the number on the edges indicates the corresponding weight.

**Solution.** Since this is a simple graph, we may use a pair of vertices to represent an
edge. One possible list of edges ordered in increasing weight is

\[ e_1 = \{v_2, v_3\}, \quad w(e_1) = 1; \quad e_6 = \{v_5, v_7\}, \quad w(e_6) = 4; \]
\[ e_2 = \{v_5, v_6\}, \quad w(e_2) = 2; \quad e_7 = \{v_1, v_2\}, \quad w(e_7) = 4; \]
\[ e_3 = \{v_6, v_7\}, \quad w(e_3) = 3; \quad e_8 = \{v_1, v_4\}, \quad w(e_8) = 5; \]
\[ e_4 = \{v_3, v_4\}, \quad w(e_4) = 3; \quad e_9 = \{v_3, v_5\}, \quad w(e_9) = 5; \]
\[ e_5 = \{v_2, v_4\}, \quad w(e_5) = 4; \quad e_{10} = \{v_3, v_7\}, \quad w(e_{10}) = 7. \]

We note that this edge list happens to be the same as that in example 4. The Kruskal’s algorithm on this example can be performed/summarised in the following table:

<table>
<thead>
<tr>
<th>edges</th>
<th>L(1)</th>
<th>L(2)</th>
<th>L(3)</th>
<th>L(4)</th>
<th>L(5)</th>
<th>L(6)</th>
<th>L(7)</th>
<th>extra edge</th>
<th>extra weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\emptyset)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(e_1)</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(e_1)</td>
<td>1</td>
</tr>
<tr>
<td>(e_2)</td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>5</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(e_3)</td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>(e_3)</td>
<td>3</td>
</tr>
<tr>
<td>(e_4)</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>(e_4)</td>
<td>3</td>
</tr>
<tr>
<td>(e_5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(e_6)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(e_7)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>(e_7)</td>
<td>4</td>
</tr>
<tr>
<td>(e_8)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(e_9)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>(e_9)</td>
<td>5</td>
</tr>
<tr>
<td>(e_{10})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

in which the numbers conveniently represented by the bar “—” are the same as the numbers in the previous step. You may of course use the actual numbers instead if you so prefer. Hence a minimal spanning tree can be formed by the edges \(\{e_1, e_2, e_3, e_4, e_7, e_9\}\). The corresponding total weight is 18 (=1+2+3+3+4+5).

6. Is there perhaps a quick’n dirty way to solve the problem in the previous example?

**Solution.** When a weighted graph is not too complicated, the following procedure tends to be quite intuitive and manually quick. If we apply this procedure to the weighted graph in the previous example, the resulting minimal spanning tree can be obtained instantly. Minimal spanning trees can also be found through the following procedure.
Algorithm 13.3 Coloured Minimal Spanning Tree Algorithm

(i) Assume that the weighted graph is initially not coloured.
(ii) Pick an edge that has the smallest weight among all the remaining *uncoloured* edges (pick any one if there are more than one such edges).
(iii) Colour the picked edge into red if, in doing so, the *red-coloured* edges will not form any nontrivial circuits. Otherwise colour the edge into green.
(iv) If all edges have been coloured into either red or green, then the red-coloured edges form a minimal spanning tree. Otherwise go back to step (ii) to process the remaining uncoloured edges.

7. (a) Find a minimal spanning tree for the following weighted graph $G$

![Graph Image]

where the numbers each represent the weight of the corresponding edge. What is the total weight of the minimal spanning tree?

(b) Give a Hamiltonian path, if any, for the graph in (a).

(c) Can we find an Eulerian path for the graph in (a)? Explain why.

Solution.

(a) We use Kruskal’s algorithm to obtain a minimal spanning tree:

![Minimal Spanning Tree Image]

The procedure can be summarised in the following table
### Kruskal’s Algorithm

<table>
<thead>
<tr>
<th>edge</th>
<th>weight</th>
<th>L(A)</th>
<th>L(B)</th>
<th>L(C)</th>
<th>L(D)</th>
<th>L(E)</th>
<th>L(F)</th>
<th>take?</th>
</tr>
</thead>
<tbody>
<tr>
<td>initial</td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>BC</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>yes</td>
</tr>
<tr>
<td>DE</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>yes</td>
</tr>
<tr>
<td>EF</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>yes</td>
</tr>
<tr>
<td>DF</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>no</td>
</tr>
<tr>
<td>AB</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>yes</td>
</tr>
<tr>
<td>AD</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>yes</td>
</tr>
<tr>
<td>AE</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>stop: all labels are now equal</td>
</tr>
</tbody>
</table>

i.e. the spanning we obtained is the above one and its minimal weight is $3 + 3 + 4 + 5 + 6 = 21$.

(b) The minimal tree found in part (a) in the above is a Hamiltonian path.

(c) No. Because there are more than 2 vertices of odd degree.

### Exercises

1. Let $n \geq 1$ be an integer. How many spanning trees, up to isomorphisms, are there for $K_{2,n}$?

2. Find a minimal spanning tree for the following weighted graph $G$

![Graph Image]

where the numbers each represent the weight of the corresponding edge. What is the total weight of the minimal spanning tree?

3. Prove that if a connected graph is not a tree, then it must have more than 1 spanning trees.

4. If a weighted graph has different weight on each edge, can there be more than 1 minimal spanning trees?
14  Number Bases

14.1  Frequently Used Number Systems

The numbers used most frequently in our daily life are written in the standard decimal number system, e.g.,

$$7086 = 7 \times 10^3 + 0 \times 10^2 + 8 \times 10^1 + 6 \times 10^0 .$$

The base number in the decimal system is 10, whose special role is obvious in the above example, implying there are exactly 10 digits required for the system. These 10 elementary digits are 0, 1, 2, ..., 9. In general, a nonnegative number in a base $p$ number system is denoted by

$$(a_n a_{n-1} \ldots a_1 a_0, b_1 b_2 \ldots b_N \ldots)_p ,$$

representing the value of

$$a_n p^n + a_{n-1} p^{n-1} + \ldots + a_1 p^1 + a_0 p^0 + b_1 p^{-1} + b_2 p^{-2} + \ldots + b_N p^{-N} + \ldots$$

We note that $a_0, \ldots, a_n$ and $b_1, b_2 \ldots$ are all just digits, coming from a set of $p$ elementary digits for the number system. In other words, If base number is $p$, we use $p$ elementary digits to represent all numbers in base $p$. The elementary digits represent values from 0 to $(p - 1)$, respectively.

Examples

1. When base number $p = 2$, the system is called the binary number system. Two digits are needed, and these two digits are chosen to be “0” and “1”, consistently representing the values of 0 and 1 respectively. For example,

$$(1011.1)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2 + 1 + \frac{1}{2} = (11.5)_{10} ,$$

i.e., 1011.1₂ is equal to 11.5 in decimal. Note that the brackets can be dropped.

2. When base $p = 8$, the system is called the octal number system. Naturally, we still conveniently use the existing first eight digits in the decimal system, i.e., 0, 1, 2, ..., 7, to denote the corresponding values for the new number system. For example,

$$13.7_8 = 1 \times 8^1 + 3 \times 8^0 + 7 \times 8^{-1} = 11.875 ,$$

i.e. $13.7_8$ represents the decimal number 11.875.

3. When $p = 16$, the system is called the hexadecimal number system. 16 digits are needed in this system, representing values from 0 to 15. We shall still make use of the ten digits 0, 1, ..., 9 from the decimal number system, and use $A, B, C, D, E, F$ as symbols to make up for the rest of the digits. Hence the digits for the hexadecimal system are

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F$$
representing (in base 10) 0 to 15, respectively. For example,

\[\begin{align*}
A7F_{16} &= A \times 16^2 + 7 \times 16^1 + F \times 16^0 \\
&= 10 \times 16^2 + 7 \times 16 + 15 \times 1 = 2687
\end{align*}\]

4. When \( p = 10 \), the system is our standard decimal number system. Thus for instance \((36429)_{10}\) is just 36429.

### 14.2 Conversion to Numbers of Different Bases

Recall that for all integers \( a \) and \( b \) \((a, b \in \mathbb{Z})\), we say that \( b \) divides \( a \) if and only if (iff) there exists an integer \( c \) \((\exists c \in \mathbb{Z})\) such that \( a = bc \). When \( b \) divides \( a \), we call \( b \) a divisor of \( a \). If \( b \) divides \( a \) \((a = bc)\), then so does \( c \), so divisors always come in pairs. If \( a \) and \( b \) are positive integers, then if \( a \) divides \( b \) and \( b \) divides \( a \), then \( a = b \). (prove it!)

Conversion of a number in one base system to another base system can be done from the definitions, e.g., to convert \((45)_6\) into base 4:

\[\begin{align*}
(45)_6 &= 4 \times 6^1 + 5 \times 6^0 = 29 \quad \text{(first converted to the decimal system)} \\
&= 20 + 8 + 1 \\
&= 4 \times 5 + 2 \times 4 + 1 \\
&= 4 \times (4 + 1) + 2 \times 4 + 1 \\
&= 1 \times 4^2 + 3 \times 4^1 + 1 \times 4^0 \\
&= (131)_4
\end{align*}\]

What we were doing was find how many 4's were "hiding" in 29 and it involved a form of repeated division. However, the procedure in the above example doesn’t seem systematic and may be cumbersome to perform in general to say the least. Hence we shall introduce below an algorithm, involving repeated division, to convert numbers from one base system to another. But first, we need some theoretical basis for our algorithm. The next theorem is stating that if you divide two integers, you get a quotient and a remainder where the remainder is less than the divisor. For example, 11 divided by 4 gives a quotient of 2 with remainder 3, \(11 = 2 \times 4 + 3\).

**Theorem 17. (Remainder Theorem)** For any integers \( p \) and \( q \) with \( p > 0 \), there exist integers \( s \) and \( r \) such that

\[q = sp + r, \quad 0 \leq r < p\] (31)

**Proof.** Let’s start by proving that if \( s \) and \( r \) exist, then they are unique. So we know that \( q = sp + r \), and \( 0 \leq r < p \) for some \( s \) and \( r \) in \( \mathbb{Z} \). Then

\[\frac{q}{p} = s + \frac{r}{p} \quad \text{and} \quad 0 \leq \frac{r}{p} < 1\]
Therefore the definition of flooring implies that \( s = \left\lfloor \frac{q}{p} \right\rfloor \). Thus if (31) holds, then \( s \) is uniquely determined as the floor of \( q/p \). And from (31) it follows that \( r = q - sp \) is also unique.

Now we prove the existence of \( s \) and \( r \) satisfying (31). We set

\[
\begin{align*}
  s &:= \left\lfloor \frac{q}{p} \right\rfloor \\
  r &:= q - sp := q - \left\lfloor \frac{q}{p} \right\rfloor p
\end{align*}
\]

Obviously, \( s \) and \( r \), so defined, are integers and they satisfy equation (31). It only remains to show that \( r \) satisfies the inequalities in (31). We know that

\[
\left\lfloor \frac{q}{p} \right\rfloor < \frac{q}{p} \leq 1 + \left\lfloor \frac{q}{p} \right\rfloor
\]

Multiplying both inequalities by \( p > 0 \), we get

\[
p \left\lfloor \frac{q}{p} \right\rfloor \leq q < p + p \left\lfloor \frac{q}{p} \right\rfloor
\]

then by multiplying the first inequality by \(-1\) we get

\[
0 \leq q - p \left\lfloor \frac{q}{p} \right\rfloor := r, \quad \text{so} \quad 0 \leq r
\]

and from the second one we get

\[
r := q - p \left\lfloor \frac{q}{p} \right\rfloor < p, \quad \text{so} \quad r < p
\]

This completes the proof of Theorem 17.

The Remainder Theorem is used in the repeated division algorithm to convert a decimal integer \( a \in \mathbb{N} \) to a number of base \( p \). It produces the digits \( r_0, r_1, r_2, \ldots \) of the number \( a \) in base \( p \).

**Algorithm 14.1** Repeated Division by \( p \)

\[
\begin{align*}
  (1) & \quad a = pq_0 + r_0, \quad 0 \leq r_0 < \ p, \quad \text{Remainder Theorem on } a \\
  (2) & \quad q_0 = pq_1 + r_1, \quad 0 \leq r_1 < \ p, \quad \text{R.T. on } q_0 \\
  (3) & \quad q_1 = pq_2 + r_2, \quad 0 \leq r_2 < \ p, \quad \text{R.T. on } q_1 \\
  \vdots & \quad \ldots \\
  (m+1) & \quad q_{m-1} = pq_m + r_m, \quad 0 \leq r_m < \ p, \\
  & \quad \quad \quad \text{and} \quad q_m = 0 \quad \text{Termination Condition}
\end{align*}
\]

Then \( a' = (r_mr_{m-1} \cdots r_1r_0)_p \) is the representation in the base \( p \) number system.

To convert a non-integer number \( b \) with \( 0 < b < 1 \) to a base \( p \) number system, the procedure is as follows
Algorithm 14.2 Conversion of the decimal part of a number

(1) \[ bp = s_1 + t_1, \quad s_1 = \lfloor bp \rfloor, \quad s_1 \text{ is the integer part of } bp \]
(2) \[ t_1 p = s_2 + t_2, \quad s_2 = \lfloor t_1 p \rfloor, \quad s_2 \text{ is the integer part of } t_1 p \]
(3) \[ t_2 p = s_3 + t_3, \quad s_3 = \lfloor t_2 p \rfloor, \quad s_3 \text{ is the integer part of } t_2 p \]
\[ \vdots \]
(m) \[ t_{m-1} p = s_m + t_m, \quad s_m = \lfloor t_{m-1} p \rfloor, \quad \text{procedure terminates at } m \text{ if } t_m = 0 \]
\[ \vdots \]

Then \[ b' = (0.s_1 s_2 \cdots s_m \cdots)_p \] is the representation in the base \( p \) number system.

We note that while the algorithm for converting an integer into a different base will always terminate in finite number of steps, the conversion of a non-integer may not terminate at all, although we usually can still stop in finite number of steps after observing a repeating pattern. The proofs of the above 2 algorithms are essentially no more than the re-interpretation of

\[ a = r_m p^m + r_{m-1} p^{m-1} + \cdots + r_1 p + r_0, \quad b = \frac{s_1}{p} + \frac{s_2}{p^2} + \cdots + \frac{s_m}{p^m} + \cdots. \]

Examples

5. Convert 45 to base 4.

(1) \[ 45 = 11 \times 4 + 1 \]
(2) \[ 11 = 2 \times 4 + 3 \]
(3) \[ 2 = 0 \times 4 + 2 \]

\[ \text{Hence } 45 = (231)_4. \]

That is, we first divide 45 by 4. The quotient 11 is then divided by 4, with the new quotient 2 being again divided by 4. This repeating process is then stopped because the latest quotient has reached 0. The remainders obtained at these division steps, when listed in the reverse order, finally give the desired representation of 45 in base 4.

6. Convert 123 into binary.

Solution. We illustrate here an alternative approach to number conversions. It uses exactly the same algorithm as before, and the only difference lies in the organisation or presentation of the derivation. Obviously you don’t have to bother with this approach if you don’t like it: the approach adopted in the previous example is already neat enough.
14.2 Conversion to Numbers of Different Bases

\[(1111011)_2\]

\[
\begin{array}{c}
1|123 \div 2 \rightarrow 1|123 \\
1|61 \div 2 \rightarrow 61 \\
0|30 \div 2 \rightarrow 123 = 61 \times 2 + 1 \\
1|15 \div 2 \\
1|7 \div 2 \\
1|3 \div 2 \\
1|1 \div 2 \\
\uparrow \\
\text{successive divisions} \\
0 \\
\end{array}
\]

Hence \[123 = 1111011_2\].

7. Convert 0.1 to base 2.

\[
\begin{align*}
(1) & \quad 0.1 \times 2 = 0 + 0.2 \\
(2) & \quad 0.2 \times 2 = 0 + 0.4 \\
(3) & \quad 0.4 \times 2 = 0 + 0.8 \\
(4) & \quad 0.8 \times 2 = 1 + 0.6 \\
(5) & \quad 0.6 \times 2 = 1 + 0.2 \quad \leftarrow 0.2 \text{ here is repeating the outcome of step (1)}
\end{align*}
\]

If \(p = q^k\) and \(p, q\) and \(k\) are all positive integers, then the conversion of \((a_n \ldots a_1 a_0 b_1 b_2 \ldots)_p\) into base \(q\) can be done digitwise, with each digit \(a_i\) or \(b_j\) being mapped to a block of \(k\) consecutive digits in the base \(q\) number system. This is called the blockwise conversion.

Examples

8. Convert \((7A.1)_{16}\) into binary format.

Solution. Since \(16 = 2^4\), 1 digit in hexademical will correspond to exactly 4 digits in binary. Since

\[
7_{16} = 0111_2, \quad A_{16} = 1010_2, \quad 1_{16} = 0001_2,
\]

we have

\[
(7A \cdot 1)_{16} = (01110100.0001)_{16}.
\]

9. Convert the binary number \((1111010.0001)_2\) to base 8.

Solution. Since \(8 = 2^3\) implies every 3 consecutive digits, going left or right starting from the fractional point, in binary will correspond to a single digit in base 8. By grouping consecutive digits into blocks of 3, appending proper 0’s if necessary, we have

\[
(1111010.0001)_2 = (001 \ 111 \ 010 \ 000 \ 100)_2 = (172.04)_8
\]

because \(001_2 = 1_8, \ 111_2 = 7_8, \ 010_2 = 2_8\) and \(100_2 = 4_8\).
Basic direct operations such as additions and subtractions can be performed in any number systems in a similar way to what we would do in the standard decimal system.

**Examples**

10. Find \(343_5 + 114_5\).

\[
\begin{array}{cccc}
3 & 4 & 3 \\
+ & 1 & 1 & 4 \\
\hline
1 & 0 & 1 & 2
\end{array}
\]

\(3 + 4 = 7 = (12)_5\)

i.e., \(343_5 + 114_5 = 1012_5\). We note that the addition is performed digitwise from the rightmost digit towards the left. On the rightmost digit, \(3_5 + 4_5 = 7_{10} = (12)_5\) gives the resulting digit 2 and the carry 1 to the next digit on the left. On the 2nd rightmost digit, we have thus \(4_5 + 1_5 + (\text{carry}) = 1_5 = 6_{10} = (11)_5\), resulting to digit 1 plus a new carry 1. As the last step, we then have \(3_5 + 1_5 + 1_5 = 5_{10} = (10)_5\). This completes our addition.

11. Find \(10011_2 - 101_2\).

\[
\begin{array}{cccc}
1 & 0 & 0 & 1 & 1 \\
- & 1 & 0 & 1 \\
\hline
1 & 1 & 1 & 0
\end{array}
\]

Hence \(10011_2 - 101_2 = 1110_2\). The subtraction is also performed digitwise from the rightmost digit towards the left. On the rightmost digit, \(1_2 - 1_2 = 0_2\) gives the resulting digit 0. On the 2nd rightmost digit, \(1_2 - 0_2 = 1_2\) gives the result 1. On the 3rd rightmost digit, we **borrow** 1 from the digit on the left; actually we have to borrow 1 further on the left because the digit on the immediate left is 0 and therefore can’t be borrowed. The subtraction on the 3rd rightmost digit thus becomes \((10)_2 - 1_2 = 2_{10} - 1_{10} = 1_{10} = 1_2\). Finally, the 4th rightmost digit, after borrowed 1 from the left and being borrowed 1 from the right, is left with the remaining 1. This completes the subtraction.

**Exercises**

1. Convert 36, to base 2.


3. Find \(A8.F_{16} + 11011.1_2\) and give the result in the hexademical number system.
15 Relations

Relations are fundamental to modern database systems and artificial intelligence. The relational database systems widely used in corporations of all levels are modeled on the concept of relations.

15.1 Binary Relations

Definition 18. A binary relation from set A to set B is a subset R of the cartesian product of sets A and B, \( R \subseteq A \times B \). An n-ary relation between sets \( A_1, A_2, \ldots, A_n \) is a subset \( R \subseteq A_1 \times A_2 \times \ldots \times A_n \), that is \( R \subseteq \prod_{i=1}^{n} A_i \).

Thus for any pair \((x, y)\) in \( A \times B \), \( x \) is related to \( y \) by \( R \), written \( xRy \), if and only if \((x, y) \in R\). Likewise \((x, y) \not\in R\) may be similarly denoted by \( x \not R y \).

A binary relation can be thought of as a table that lists the relationship of elements from one set to elements of another set, e.g.,

<table>
<thead>
<tr>
<th>Student</th>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brian</td>
<td>CS</td>
</tr>
<tr>
<td>Lena</td>
<td>Maths</td>
</tr>
<tr>
<td>John</td>
<td>History</td>
</tr>
<tr>
<td>Dave</td>
<td>Art</td>
</tr>
<tr>
<td>Brian</td>
<td>Art</td>
</tr>
<tr>
<td>Joe</td>
<td>Sport</td>
</tr>
</tbody>
</table>

If Brian is taking CS and Art, Lena is taking Maths, etc. Then we say that Brian is related to CS and Art, Lena is related to Maths, etc.

Of course the table on the left is just a set of ordered pairs and, in general, we define a relation to be a set of ordered pairs. In the case of binary relations, we consider the first element of the ordered pair to be related to the second element of the ordered pair.

Examples

1. Let \( A = \{1, 3\} \) and \( B = \{2, 5\} \). Then we ask how elements in \( A \) are related to elements in \( B \) via the inequality “\( \leq \)”.

The answer is

\[
1 \leq 2, \ 1 \leq 5, \ 3 \not\leq 2, \ 3 \leq 5 .
\]

In other words, 3 pairs \((1, 2), (1, 5)\) and \((3, 5)\) observe the “\( \leq \)” relationship. Hence by collecting them in \( R \), i.e.,

\[
R = \{(1,2), (1,5), (3,5)\},
\]

we know relation \( R \) describes precisely the relationship between elements of \( A \) and elements of \( B \) under “\( \leq \)”.

For example,

\[
3 R 5 \iff (3, 5) \in R \iff 3 \text{ is related to 5 by } R \iff 3 \leq 5 ,
\]

\[
3 \not R 2 \iff (3, 2) \not\in R \iff 3 \text{ is not related to 2 by } R \iff 3 \not\leq 2 .
\]
2. Let $A = \{1, 2\}$ and $B = \{1, 2, 3\}$ and define a binary relation $R$ from $A$ to $B$ by

$$\text{for any } (x, y) \in A \times B: \quad (x, y) \in R \iff x - y \text{ is even}$$

(a) Give $R$ explicitly by its elements.
(b) Is $1R1$? Is $2R3$? Is $1R3$?

**Solution.**

(a) For any $(x, y)$ pair in $A \times B = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (2, 3)\}$, we must check if $xRy$ or if $x - y$ is even. This is done in the following table

<table>
<thead>
<tr>
<th>$(x, y)$</th>
<th>property of $x - y$</th>
<th>conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(1, 1)$</td>
<td>$1 - 1$ even</td>
<td>$(1, 1) \in R$</td>
</tr>
<tr>
<td>$(1, 2)$</td>
<td>$1 - 2$ odd</td>
<td>$(1, 2) \notin R$</td>
</tr>
<tr>
<td>$(1, 3)$</td>
<td>$1 - 3$ even</td>
<td>$(1, 3) \in R$</td>
</tr>
<tr>
<td>$(2, 1)$</td>
<td>$2 - 1$ odd</td>
<td>$(2, 1) \notin R$</td>
</tr>
<tr>
<td>$(2, 2)$</td>
<td>$2 - 2$ even</td>
<td>$(2, 2) \in R$</td>
</tr>
<tr>
<td>$(2, 3)$</td>
<td>$2 - 3$ odd</td>
<td>$(2, 3) \notin R$</td>
</tr>
</tbody>
</table>

Hence $R = \{(1, 1), (1, 3), (2, 2)\}$.

(b) Yes. $1R1$ since $(1, 1) \in R$.
No. $2R3$ since $(2, 3) \notin R$.
Yes. $1R3$ since $(1, 3) \in R$.

**Notes.**

1. Given a binary relation $R \subseteq A \times B$, the set $\{x \in A \mid (x, y) \in R \text{ for some } y \in B\}$ is the **domain** of $R$, the set $\{y \in B \mid (x, y) \in R \text{ for some } x \in A\}$ is the **range** of $R$. If the relation is given as a table, the domain consists of the (unique) members of the first column, while the range consists of the (unique) members of the second column.

2. The trivial relation is the empty set, $\emptyset \subseteq A \times B$, $\forall A, B$. Its domain and range are both equal to empty set.

3. The Cartesian product $A \times B$ is the largest binary relation from $A$ to $B$; its domain is $A$ and its range is $B$. By the definition of a binary relation, it contains every relation from $A$ to $B$.

4. For a **binary relation on a set** $A$, i.e., a binary relation from $A$ to $A$, the relation may be represented by a directed graph. The representation is simple: each element in $A$ is denoted by a vertex, and each $(x, y) \in R$, i.e., $x$ is related to $y$, is denoted by an arrow from $x$ to $y$. 

120
5. A function \( f : A \rightarrow B \) may be regarded as, or represented by, a special relation \( F \) from \( A \) to \( B \) that satisfies

(i) \( \forall x \in A, \exists y \in B \) such that \((x, y) \in F\),
(ii) \( \forall x \in A, \forall y, z \in B \), if \((x, y) \in F\), \((x, z) \in F\), then \( y = z \).

The identification of \( f \) with \( F \) is via \( y = f(x) \) iff \((x, y) \in F\). Notice that (i) describes a function property that every element in \( A \) should be mapped somewhere on \( B \), and (ii) implies a single element in \( A \) shouldn’t be mapped to two different elements in \( B \).

Example

3. Function \( f: \{1, 2, 3\} \rightarrow \{1, 2, 3, 4, 5, 6\} \) with \( f(n) = \frac{n(n+1)}{2} \) is equivalent to the relation

\[ F = \{(1, 1), (2, 3), (3, 6)\} \]

in which, for instance, \((2, 3)\) corresponds to \(3 = f(2)\).

4. Let \( A = \mathbb{R} \) be the set of all real numbers, and a binary relation \( S \) on the set \( A \) be defined by

\[ S = \{(x, y) \in \mathbb{R} \times \mathbb{R} \mid (x - 10)^2 + y^2 \leq 16, \ \text{or} \ (x - 5)^2 + y^2 \leq 4\} \]

Graph \( S \) in the Cartesian plane.

Solution. Recall from analytic geometry that a Cartesian plane is a plane with the normal \( x \) and \( y \) axis, and a circle of radius \( r \) centred at the coordinates \((a, b)\) is given precisely by the equation

\[(x - a)^2 + (y - b)^2 = r^2.\]

Hence equation \((x - 10)^2 + y^2 = 16 (= 4^2)\) represents a circle of radius 4 centred at the coordinates \((10, 0)\), and \((x - 5)^2 + y^2 = 4\) is a circle of radius 2 centred at the coordinates \((5, 0)\). Thus \( S \) is the union of these 2 disks, and is represented by the shaded area in the graph below.
5. Let $A = \{1, 2, 3\}$ and a binary relation $E$ be defined by $(x, y) \in E$ iff $x - y$ is even and $x, y \in A$. Then $E = \{(1,1), (2,2), (3,3), (1,3), (3,1)\}$ can be represented by the following digraph

\[(1,1): \quad 1 - 1 = 0 \text{ even, hence the loop at vertex labelled by } 1 \in A.\]
\[(2,2): \quad 2 - 2 = 0 \text{ even, hence the loop at vertex labelled by } 2 \in A.\]
\[(3,3): \quad 3 - 3 = 0 \text{ even, hence the loop at vertex labelled by } 3 \in A.\]
\[(1,3): \quad 1 - 3 = -2 \text{ even, hence the arrow from vertex 1 to vertex 3.}\]
\[(3,1): \quad 3 - 1 = 2 \text{ even, hence the arrow from vertex 3 to vertex 1.}\]

15.2 Reflexivity, Symmetry and Transitivity

Let $R$ be a binary relation on a set $A$.

- $R$ is reflexive if for all $x \in A$, $xRx$.
- $R$ is symmetric if for all $x, y \in A$, if $xRy$, then $yRx$.
- $R$ is transitive if for all $x, y, z \in A$, if $xRy$ and $yRz$, then $xRz$.
- $R$ is an equivalence relation if $A$ is nonempty and $R$ is reflexive, symmetric and transitive.

In terms of digraphs for the representation of binary relations, reflexivity is equivalent to having a loop on each vertex; symmetry means any arrow from one vertex to another will always be accompanied by another arrow in the opposite direction; and transitivity is the same as saying there must be a direct arrow from one vertex to another if one can walk from that vertex to the other through a list of arrows, travelling always along the direction of the arrows.

Examples

6. Let $A = \{0, 1, 2, 3\}$ and a relation $R$ on $A$ be given by

$$R = \{(0,0), (0,1), (0,3), (1,0), (1,1), (2,2), (3,0), (3,3)\}.$$  

Is $R$ reflexive? symmetric? transitive?

Solution. We’ll make use of the digraph for $R$ on the right.
15.2 Reflexivity, Symmetry and Transitivity

(a) $R$ is reflexive, i.e. there is a loop at each vertex.

(b) $R$ is symmetric, i.e. the arrows joining a pair of different vertices always appear in a pair with opposite arrow directions.

(c) $R$ is not transitive. This is because otherwise the arrow from 1 to 0 and arrow from 0 to 3 would imply the existence of an arrow from 1 to 3 (which doesn’t exist). In other words $(1, 0) \in R$, $(0, 3) \in R$ and $(1, 3) \notin R$ imply $R$ is not transitive.

**Note.** It is equally easy to show these properties without resorting to the digraph.

7. Let $m, n$ and $d$ be integers with $d \neq 0$. Then if $d$ divides $(m - n)$, denoted by $d \mid (m - n)$, i.e., $m - n = dk$ for some integer $k$, then we say $m$ is congruent to $n$, written simply as

\[ m \equiv n \pmod{d}. \]

Let $R$ be the relation of congruence modulo 3 on the set $\mathbb{Z}$ of all integers, i.e.

\[ m R n \iff m \equiv n \pmod{3} \iff 3 \mid (m - n) \]

Show $R$ is an equivalence relation.

**Solution.** We just need to verify that $R$ is reflexive, symmetric and transitive.

(a) Reflexive: for any $n \in \mathbb{Z}$ we have $n R n$ because 3 divides $n - n = 0$.

(b) Symmetric: for any $m, n \in \mathbb{Z}$ if $m R n$, i.e., $m \equiv n \pmod{3}$ then there exists a $k \in \mathbb{Z}$ such that $m - n = 3k$. This means $n - m = 3(-k)$, i.e., $n \equiv m \pmod{3}$, implying finally $n R m$. For example, $7 R 4$ is equivalent to $4 R 7$ can be seen from

\[ 7 R 4 \iff 7 \equiv 4 \pmod{3} \iff 7 - 4 = 3 \times 1 \iff 4 - 7 = 3 \times (-1) \iff 4 \equiv 7 \pmod{3} \iff 4 R 7 \]

(c) Transitive: for any $m, n, p \in \mathbb{Z}$, if $m R n$ and $n R p$, then there exist $r, s \in \mathbb{Z}$ such that $m - n = 3r$ and $n - p = 3s$. Hence $m - p = (m - n) + (n - p) = 3(r + s)$, i.e., $m R p$.

We thus conclude that $R$ is an equivalence relation.

8. Let $R$ be the ”son to father” relation on the set of males, $M$. Then $R$ is non reflexive, non-symmetric and non-transitive.
Exercises

1. Let a binary relation $R$ on $\mathbb{R}$ be defined by

$$R = \{(x, y) \in \mathbb{R} \times \mathbb{R} \mid \sqrt{3}x + y \geq 1\}.$$ 

Graph $R$ in the Cartesian plane.

2. Let function $f(n) = n^2$ be defined on $\mathbb{Z}$. Represent the function $f$ by a binary relation $F$ on $\mathbb{Z}$.

3. Let a digraph $G$ be given by

```
\begin{center}
\begin{tikzpicture}
\coordinate (v1) at (0,0); \coordinate (v2) at (2,0); \coordinate (v3) at (2,2); \coordinate (v4) at (0,2);
\draw (v1) -- (v2); \draw (v1) -- (v4); \draw (v2) -- (v3); \draw (v3) -- (v4);
\filldraw[black] (v1) circle (2pt) node[below] {$v_1$}; \filldraw[black] (v2) circle (2pt) node[below] {$v_2$}; \filldraw[black] (v3) circle (2pt) node[below] {$v_3$}; \filldraw[black] (v4) circle (2pt) node[below] {$v_4$};
\end{tikzpicture}
\end{center}
```

We define a binary relation $R$ on the set of the vertices of the digraph $G$ by

$$ (v, w) \in R \text{ iff there exists a walk (following the directed edges) from vertex } v \text{ of } G \text{ to vertex } w \text{ of } G $$

Draw the digraph that represents the binary relation $R$. Is $R$ reflexive, symmetric and transitive?

4. Let $\mathcal{F}$ be the set of all real-valued functions on $\mathbb{R}$, $\mathcal{F} = \{f \mid f : \mathbb{R} \to \mathbb{R}\}$. We define a binary relation $R$ on $\mathcal{F}$ by

$$ (f, g) \in R \text{ iff } f(x) = \Theta(g(x)). $$

Show that $R$ is reflexive and transitive.
16 Equivalence Relations

16.1 Equivalence Classes and Partitions

We recall that a binary relation $R$ on a set $A$ is an equivalence relation if and only if the following 3 conditions are all true:

- **Reflexivity:** $\forall x \in A$, $xRx$
- **Symmetry:** $\forall x, y \in A$, $xRy \Rightarrow yRx$
- **Transitivity:** $\forall x, y, z \in A$, $xRy \land yRz \Rightarrow xRz$

**Example**

1. Let set $A = \{2, 4, 6, 8, 10\}$ and $R$ be a binary relation on $A$ defined by

   $$\forall m, n \in A, (m, n) \in R \iff m \equiv n \pmod{4}$$

   Recall that $m \equiv n \pmod{4}$ is equivalent to saying 4 divides $(m - n)$ or $(m - n)$ is an integer multiple of 4. Thus it is easy to see

   \[
   \begin{array}{c|c|c|c}
   (m, n) & m - n & \text{is multiple of 4?} & \text{conclusion} \\
   \hline
   (2,2) & 2 - 2 = 0 & \text{yes, 0 is a multiple of 4} & (2, 2) \in R \\
   (2,4) & 2 - 4 = -2 & \text{no, -2 is not a multiple of 4} & (2, 4) \notin R \\
   (2,6) & 2 - 6 = -4 & -4: \text{yes} & (2, 6) \in R \\
   (2,8) & 2 - 8 = -6 & -6: \text{no} & (2, 8) \notin R \\
   (2,10) & 2 - 10 = -8 & -8: \text{yes} & (2, 10) \in R \\
   (4,2) & 4 - 2 = 2 & 2: \text{no} & (4, 2) \notin R \\
   (4,4) & 4 - 4 = 0 & 0: \text{yes} & (4, 4) \in R \\
   (4,6) & 4 - 6 = -2 & -2: \text{no} & (4, 6) \notin R \\
   \vdots & \vdots & \vdots & \vdots \\
   \end{array}
   \]

   which gives explicitly

   $$R = \{(2, 2), (2, 6), (2, 10), (4, 4), (4, 8), (6, 6), (6, 10), (8, 8), (10, 10), (6, 2), (10, 2), (8, 4), (10, 6)\}$$

   This relation $R$ can be drawn as
and it is obviously an equivalence relation (cf. examples in the previous lecture). Notice that there are two “connected” components, one containing elements 4 and 8 and the other containing elements 2, 6 and 10. Here the “connection” is made through certain walks along the directions of the arrows. These two components are just the two distinct equivalence classes under the equivalence relation $R$. We may use any element in an equivalence class to represent that particular class which basically contains all elements that are connected to the arbitrarily chosen representative element. Hence, in this case, we may choose 2 to represent the class $\{2\}$, written simply $[2] = \{2\}$, and choose, for instance, 8 to represent the other class $\{4, 8\}$, that is, $[8] = \{4, 8\}$. Since any member of an equivalence class can be used to represent that class, $[6]$ and $[10]$ will be representing exactly the same equivalence class as $[2]$. Hence


are the two distinct equivalence classes.

The definition of equivalence classes and the related properties as those exemplified above can be described more precisely in terms of the following lemma.

**Lemma 19.** Let $A$ be a set and $R$ an equivalence relation on $A$. For any $a \in A$ we define the equivalence class of $a$, written $[a]$, by

$$[a] = \{x \in A : xRa\}.$$

Then

(i) For any $a \in A$, $a \in [a]$.

(ii) For any $a, b \in A$, $a R b$ iff $[a] = [b]$.

(iii) For any $a, b \in A$, either $[a] \cap [b] = \emptyset$, or $[a] = [b]$.

(iv) For an equivalence class $S$, any element $a \in S$ can be used as a representative of $S$, guaranteeing $[a] = S$.

**Proof.** First, if $R$ is an equivalence relation, then $R$ is reflexive, symmetric and transitive.

(i) Since $R$ is reflexive implies $aRa$ for any $a \in A$, hence $a \in [a]$.

(ii) (a) **Necessity.** Let $aRb$. For any $x \in [a]$ we have $xRa$. From $xRa$ and $aRb$, we derive from $R$’s transitivity that $xRb$, i.e., $x \in [b]$. Hence $[a] \subseteq [b]$. Likewise $[b] \subseteq [a]$ because $bRa$ is implied by $aRb$ and $R$ is symmetric. Hence $[a] = [b]$.

(b) **Sufficiency.** Let $[a] = [b]$. Since (i) implies $a \in [a]$ and thus $a \in [b]$, we have by definition of $[b]$ that $aRb$.

(iii) If $[a] \cap [b] \neq \emptyset$, then there exists an $x \in A$ such that $x \in [a]$ and $x \in [b]$. This means $xRa$ and $xRb$, i.e., $aRx$ and $xRb$ due to the symmetry of $R$, implying $aRb$ due to the transitivity. (ii) thus implies $[a] = [b]$.

(iv) $S$ is an equivalence class means $S = [b]$ for some $b \in A$. Thus for any $a \in S = [b]$, we have $aRb$ and hence from (ii) $[a] = [b] = S$. 

126
Examples

2. For the binary relation $R$ defined in Example 1, let us verify the results of the above lemma.

Solution. Actually this is more like explanations.

(a) (i) of the lemma in this case implies $2 \in [2] = \{2, 6, 10\}$, $6 \in [6]$, $10 \in [10]$, $4 \in [4]$ and $8 \in [8] = \{4, 8\}$ which is obviously true due to the explicit formula at the end of Example 1.

(b) (ii) in this case is the same as saying $[2] = [6] = [10]$ and $[4] = [8]$, which are already shown in Example 1.

(c) (iii) is consistent with the fact that any 2 of the equivalence classes $[2]$, $[4]$, $[6]$, $[8]$ and $[10]$ are either disjoint (e.g., $[2]$ and $[4]$ have no elements in common) or exactly the same (e.g., $[2] = [6]$).

(d) (iv) for the equivalence class $\{2, 6, 10\}$ implies we can use either 2 or 6 or 10 to represent that same class, which is consistent with $[2] = [6] = [10]$ observed in Example 1. Similar observations can be made to the equivalence class $\{4, 8\}$.

3. Show that the distinct equivalence classes in Example 1 form a partition of the set $A$ there.

Solution. In Example 1 we have shown that $[2] = \{2, 6, 10\}$ and $[4] = \{4, 8\}$ are the only distinct equivalence classes. Since $A$ in Example 1 is given by $A = \{2, 4, 6, 8, 10\}$, we can easily verify


From the definition of the set partition (see the earlier lecture for sets) we conclude that $\{[2], [4]\}$ is a partition of set $A$.

The above example is in fact a special case of the more universal properties to be summarised in the following theorem.

Theorem 20. Let $A$ be a nonempty set.

(i) If $R$ is an equivalence relation on $A$, then the distinct equivalence classes of $R$ form a partition of $A$. This partition is said to be induced by the equivalence relation $R$.

(ii) Suppose $A$ has a partition $\{A_1, \cdots, A_n\}$, i.e., $A = \bigcup_{i=1}^n A_i$ and $A_i \cap A_j = \emptyset$ for any $i \neq j$. If a relation $R$ on $A$ is defined by

$\forall x, y \in A : x R y \iff \exists A_i$ of the partition such that $x \in A_i$ and $y \in A_i$ (*),

then $R$ is an equivalence relation, and the distinct equivalence classes of $R$ form the original partition $\{A_1, \cdots, A_n\}$. 

127
Proof.

(i) Let $A_i$ for $i = 1, \ldots, m$ be all the distinct equivalence classes of $R$. For any $x \in A$, since 
$x \in [x] \subseteq \bigcup_{i=1}^{m} A_i$. Hence $A \subseteq \bigcup_{i=1}^{m} A_i$, implying $A = \bigcup_{i=1}^{m} A_i$ because $A_i \subseteq A$ for any $i = 1, \ldots, m$. Notice that $A_i \cap A_j = \emptyset$ for any $i \neq j$ is obvious from Lemma (iii).

(ii) (a) Reflexive: $\forall x \in A = \bigcup_{i=1}^{n} A_i$, $\exists i_0$ such that $x \in A_{i_0}$. Hence $xRx$ from the definition

(*) of $R$.

(b) Symmetric: $\forall x, y \in A$ such that $xRy$, $\exists i_0$ such that $x, y \in A_{i_0}$, i.e., $y, x \in A_{i_0}$, implying $yRx$ from (*).

(c) Transitive: $\forall x, y, z \in R$ such that $xRy$ and $yRz$, $\exists i_0$ and $j_0$ such that $x, y \in A_{i_0}$ and $y, z \in A_{j_0}$. Since $y \in A_{i_0}$, $y \in A_{j_0}$ and $A_i$'s are disjoint for different $i$, we must have $i_0 = j_0$, hence $x, z \in A_{i_0}$, implying $xRz$.

From (a), (b), (c) we conclude $R$ is an equivalence relation.

Since by definition of $x \in A_i$ means $[x] = \{y \in A : yRx\} = \{y \in A : \exists i_0 \text{ such that } x, y \in A_{i_0}\}$, if $x \in A_i$ for some $i$ then $i_0$ must be equal to $i$ because $A_j \cap A_k = \emptyset$ for any $j \neq k$. We thus conclude that all $A_i$'s are just the distinct equivalence classes, and $[x]$ for any $x \in A$ is just one of $A_i$'s. Therefore we see that $R$ will induce the original partitions $\{A_1, \ldots, A_n\}$.

Example

4. Let $R$ be the equivalence relation defined on $\mathbb{Z}$ by $R = \{(m, n) : m, n \in \mathbb{Z}, m \equiv n \pmod{3}\}$, see examples in the previous lecture. Give the partition of $\mathbb{Z}$ in terms of the equivalence classes of $R$.

Solution.

(a) Pick any element in $\mathbb{Z}$, say $0$, we have

$$[0] = \{x \in \mathbb{Z} : x R 0\} = \{\cdots, -6, -3, 0, 3, 6, \cdots\}.$$

(b) Pick any element in $\mathbb{Z}\setminus[0]$, say $1$, we have

$$[1] = \{x \in \mathbb{Z} : x R 1\} = \{\cdots, -5, -2, 1, 4, 7, \cdots\}.$$

(c) Pick any element in $\mathbb{Z}\setminus([0] \cup [1])$, say $5$, we have

$$[5] = \{x \in \mathbb{Z} : x R 5\} = \{\cdots, -4, -1, 2, 5, 8, \cdots\}.$$

(d) Since $\mathbb{Z}\setminus([0] \cup [1] \cup [5]) = \emptyset$, all distinct equivalence classes are now exhausted. Hence the partition of $\mathbb{Z}$ under $R$ is $\{[0], [1], [5]\}$. 

128
16.2 Property Closure

Let $R$ be a binary relation on a set $A$. A binary relation $R'$ on $A$ is said to be the closure of $R$ with respect to property $P$ if

(i) $R'$ has property $P$,

(ii) $R \subseteq R'$,

(iii) If $S$ is any relation with property $P$ and $R \subseteq S$, then $R' \subseteq S$.

That is, a new relation is built on the old so that the new one has property $P$. For example, a relation may not satisfy the transitive property, so the problem is to create a new relation out of the old so that transitivity is satisfied.

Example

5. Let $A = \{0, 1, 2, 3\}$ and $R = \{(0, 1), (1, 2), (2, 3)\}$. From the graph for $R$

we see that a transitive closure, or closure w.r.t. the transitivity, can be obtained by adding arrows $(0, 2)$ due to $(0, 1)$ and $(1, 2)$, $(1, 3)$ due to $(1, 2)$ and $(2, 3)$, and $(0, 3)$ due to $(0, 1)$ and newly obtained $(1, 3)$. Hence the directed graph $R'$ will be

i.e., $R' = \{(0, 1), (1, 2), (2, 3), (0, 2), (1, 3), (0, 3)\}$

Note. We may make an equivalence relation out of $R$ by building the reflexive, symmetric and transitive closures as follows
Exercises

1. Suppose $R$ is an equivalence relation on a set $A$. If $aRb$ for all $a, b \in A$, how many equivalence classes are there?

2. Let $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. We define a binary relation $R$ on $A$ by

   $$(m, n) \in R \iff m \equiv n \pmod{2}.$$ 

   List all the distinct equivalence classes.

3. Let $A = \{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 1), (2, 2)\}$. We define a binary relation $R$ on the set $A$ by

   $$(m, n)R(j, k) \iff m - n = j - k.$$ 

   Is $R$ an equivalence relation? If yes, give representatives for all the distinct equivalence classes.
17 Partial Order Relations

Partial orderings have important applications, e.g., the analysis of computer programs. Trees, that you met earlier, have an inherent partial order. In addition, partial orders play an important role in aspects of pure mathematics.

17.1 Partial Orderings

Let $R$ be a binary relation on a set $A$.

- $R$ is **antisymmetric** if for all $x, y \in A$, if $xRy$ and $yRx$, then $x = y$.
- $R$ is a **partial order relation** if $R$ is reflexive, antisymmetric and transitive.
- A partially ordered set (poset) consists of a set together with a partial order relation on it.

In terms of the digraph of a binary relation $R$, the *antisymmetry* is tantamount to saying there are no arrows in opposite directions joining a pair of (different) vertices.

**Example**

1. Let $A = \{0, 1, 2\}$ and $R = \{(0, 0), (0, 1), (0, 2), (1, 1), (1, 2), (2, 2)\}$ and $S = \{(0, 0), (1, 1), (2, 2)\}$ be two relations on $A$. Show
   
   (i) $R$ is a partial order relation.
   
   (ii) $S$ is an equivalence relation.

**Solution.** We choose to use digraphs to make the explanations in this case.

(i) The digraph for $R$ on the right implies the relation $R$ is
   - **Reflexive**: loops on every vertex.
   - **Transitive**: if you can travel from vertex $v$ to vertex $w$ along consecutive arrows of the same direction, then there is also a single arrow pointing from $v$ to $w$.
   - **Antisymmetric**: no $\leftrightarrow$ type of arrows.

(ii) The digraph for $S$ on the right is reflexive due to loops on every vertex, and is symmetric and transitive because no no-loop arrows exist.
Note. In Example 1, \( R \) and \( S \) are built on \( A \) from \( \leq \) and \( = \) respectively by
\[
R = \{(x, y) : x, y \in A, x \leq y \},
\]
\[
S = \{(x, y) : x, y \in A, x = y \}.
\]
Hence partial order relation and equivalence relation can be in general regarded as “generalisation” of \( \leq \) and \( = \), respectively. For the same reasons, they are often denoted by
\[
\text{\( \triangleright x \preceq y \) if \( xR_1 y \) and \( R_1 \) is a partial order relation,}
\]
\[
\text{\( \triangleright x \sim y \) if \( xR_2 y \) and \( R_2 \) is an equivalence relation.}
\]

Let \( R \) be a partial order relation on a set \( A \).

- For any elements \( a, b \in A \), “\( aRb \)” can be alternatively denoted by “\( a \leq b \)” meaning element \( a \) precedes element \( b \) under the partial order relation \( R \).
- Two elements \( a, b \in A \) are comparable if either \( aRb \) or \( bRa \) (i.e., either \( a \leq b \) or \( b \leq a \)), or if \( a = b \).
- If all elements of \( A \) are comparable with each other, then the partially ordered set \( A \) (w.r.t. \( R \)) is said to be a totally ordered set, and the relation \( R \) is also said to be a total order relation. A total order let us arrange the elements of set \( A \) in order as thought on a line. Hence a total ordering is also called a linear ordering.
- An element \( a \in A \) is a maximal element of \( A \) if \( b \preceq a \) holds for every \( b \in A \) whenever \( b \) and \( a \) are comparable.
- An element \( a \in A \) is a greatest element (top element) of \( A \) if \( b \preceq a \) holds for all \( b \in A \).
- An element \( a \in A \) is a minimal element of \( A \) if \( a \preceq b \) holds for every \( b \in A \) whenever \( b \) and \( a \) are comparable.
- An element \( a \in A \) is a least element (bottom element) of \( A \) if \( a \preceq b \) holds for all \( b \in A \).

Example

2. Let \( A \) be the set of all subsets of set \( \{a, b, c\} \). Show the “subset” relation \( \subseteq \) on \( A \), i.e.,
\[
\forall u, v \in A, u \leq v \text{ or } uRv, \text{ iff } u \subseteq v,
\]
is a partial order relation. Find a minimal element and a greatest element.

Solution. It is easy to verify that “\( \subseteq \)” is a partial ordering. Since \( \emptyset \) is a subset of any \( u \in A \), i.e., \( \emptyset \leq u \), we see \( \emptyset \) is not only a minimal element, it is also a least element of \( A \). Since for any \( u \in A \) one has \( u \subseteq \{a, b, c\} \), i.e., \( u \leq \{a, b, c\} \), we see that \( \{a, b, c\} \) is a greatest element of \( A \).
A greatest (top) element is always a maximal element, but a maximal element needs not be a greatest, not even if the maximal element is unique. However, an unique maximal element in a finite partial ordered set is a top.

*Mutatis mutandis* for bottom:

A least (bottom) element is always a minimal element, but a minimal element needs not be a least, not even if the minimal element is unique. However, an unique minimal element in a finite partial ordered set is a bottom.

**Theorem 21.** *A poset can have at most one top.*

**Proof.** Let \((A, R)\) be a poset. If \(x\) and \(y\) are tops then \(yRx\) and \(xRy\), so \(x = y\) (by antisymmetry of \(R\)). Same for bottoms, of course.

17.2 **Hasse Diagrams**

*Hasse diagrams* are meant to present partial order relations in equivalent but somewhat simpler forms by removing certain deducible “noncritical” parts of the relations. For better motivation and understanding, we’ll introduce it through the following examples.

**Examples**

3. The relation in Example 2 can be drawn as

![Hasse Diagram](image)

It is somewhat “messy” and some arrows can be derived from transitivity anyway. If we

- omit all loops,
- omit all arrows that can be inferred from transitivity,
- draw arrows without heads,
- understand that all arrows point upwards,
then the above graph simplifies to

![Hasse diagram](image)

This type of graph is called a **Hasse diagram** and it is often used to represent a partially ordered set. If a digraph of a partial order relation is simplified according to the steps under the four “▶” symbols in the above, then the graph becomes a Hasse diagram.

4. Let \( A = \{1, 2, 3, 9, 18\} \) and for any \( a, b \in A, \ a \leq b \iff a \mid b \). Draw the Hasse diagram of the relation. Give a maximal element and a greatest element if any.

**Solution.** First it is easy to verify that the relation \( \leq \) defined above is a partial ordering. The directed graph of relation \( \leq \) is drawn on the right. By removing all the loops, deleting all the edges that can be derived transitively, and making sure all the retained edges will point upwards implicitly, we obtain the following Hasse diagram

![Hasse diagram](image)

This is because the edge from vertex 1 to vertex 18, for instance, can be derived from the edge from 1 to 2 and the edge from 2 to 18. Hence the edge from 1 to 18 shouldn’t be kept in the Hasse diagram as it can be induced from the transitivity.

Since 18 divides no elements in the set \( A \) other than itself, no elements in \( A \) is preceded by 18 other than 18 itself because of the definition of the partial order relation \( \leq \). Hence 18 is a maximal element. Since all elements in \( A \) divide 18 means \( a \leq 18 \) holds for all elements \( a \in A \), 18 is also a greatest element.

Finally we remark that the vertices in the digraphs in this and the previous examples happen to be conveniently placed for creating the corresponding Hasse diagrams, in the sense
that all arrows are already pointing upwards. If this is not the case, then one may have to “twist” the digraph to make it so, ready for the transformation into a Hasse diagram.

**Remember:**

- In a partial order relation, all *greatest* elements are also *maximal* elements and all *least* elements are also *minimal* elements.

- In terms of a Hasse diagram, all *local top* vertices are maximal elements, and all *local bottom* vertices are minimal elements.

- Two elements are *comparable* if there is a path connecting them and that the walk along the path does not change direction vertically on *any* edge of the path.

- In terms of a Hasse diagram
  - \( a \in A \) is a maximal element of \( A \) if there is no \( y \in A \) such that \( y \) is above \( a \).
  - \( a \in A \) is a minimal element of \( A \) if there is no \( y \in A \) such that \( y \) is below \( a \).
  - \( a \in A \) is the greatest (top) element of \( A \) if \( a \) is above every other element of \( A \).
  - \( a \in A \) is the least (bottom) element of \( A \) if \( a \) is below every other element of \( A \).

- If a Hasse diagram has only *finite* number of vertices, then a maximal element is also a greatest element if there is only 1 maximal element, and a minimal element is also a least element if there is only 1 minimal element.

**Example**

5. (a) Let set \( A \) be given by \( A = \{3, 4, 5, 6, 10, 12\} \) and a binary relation \( R \) on \( A \) be defined by

\[
(x, y) \in R \text{ if and only if } x \text{ divides } y
\]

\[
(x, y) \in R \text{ iff } x \mid y
\]

Give \( R \) explicitly in terms of its elements and draw the corresponding Hasse diagram. List all the maximal, minimal, greatest, and least elements.

(b) Let a new binary relation \( R' \) on the set \( A \) given in (a) be defined by

\[
(x, y) \in R' \text{ if and only if either } x \mid y \text{ or } y \mid x
\]

and \( R'' \) be the transitive closure of \( R' \). Use directed graphs to represent \( R, R' \) and \( R'' \) respectively. Which of the three relations \( R, R' \) and \( R'' \) is an equivalence relation? For the equivalence relation, give all the distinct equivalence classes.
Solution.

(a) Among all the elements of set \( A = \{3, 4, 5, 6, 10, 12\} \), obviously \( 3 \in A \), for instance, divides 3, 6 and 12. Hence, by the definition of the relation \( R \) specified by the question, we conclude \((3, 3), (3, 6)\) and \((3, 12)\) are all elements of the relation \( R \). Likewise we can show that \((4, 4), (4, 12), (5, 5), (5, 10), (6, 6), (6, 12), (10, 10), (12, 12)\) are all elements of \( R \). In fact we have

\[
R = \{ (3, 3), (3, 6), (3, 12), (4, 4), (4, 12), (5, 5), \\
(5, 10), (6, 6), (6, 12), (10, 10), (12, 12) \}.
\]

Hence the digraph for \( R \) is

From the Hasse diagram, we observe that there are 2 local top vertices, 12 and 10, and 3 local bottom vertices, 3, 4 and 5. Hence 12 and 10 are maximal elements, and 3, 4 and 5 are minimal elements. Since the Hasse diagram has only finite number of vertices, the fact that the Hasse diagram has more than 1 local top vertices means there are no greatest elements. Likewise there are no least elements either.

(b) The digraphs for \( R \) is already given in (a). As for \( R' \) we observe that, according to the definition of the relation \( R' \), if \((x, y)\) is in \( R \) so will \((y, x)\) (e.g., \((3, 6) \in R\) thus \((6, 3) \in R\) ). Hence we can obtain \( R' \) by adding to \( R \) the symmetric pairs like \((6, 3), (12, 3)\), etc. In terms of the digraph, such addition of elements is equivalent to drawing opposite arrows to each existing (non-loop) arrows. The digraph of \( R' \) thus takes the following form
Since relation $R''$ is the transitive closure of $R'$, we can derive $R''$ from $R'$ by connecting 3 to 4 and 4 to 6 and connecting 4 to 3 and 6 to 4. We connect 3 to 4 via an direct arrow because we can travel from 3 to 12 and then from 12 to 4 all along the arrows, and we connect 4 to 3 because we can travel from 4 to 12 then to 3 all along the arrows too. Similar reasons are applicable for the arrows from 4 to 6 and 6 to 4. Hence $R''$ can be drawn as

Since there is no arrow from element 12 to element 3 in the digraph of $R$ despite the existence of an arrow from 3 to 12, relation $R$ is not symmetric hence is not an equivalence relation. Since relation $R'$ is not transitive (because its transitive closure $R''$ is not the same as $R'$ itself), relation $R'$ is not an equivalence relation either. As for the relation $R''$, it is obviously reflexive, symmetric and transitive. Hence $R''$ is an equivalence relation.

Since elements 3, 4, 6 and 12 are all related (connected) to each other through the arrows of the digraph $R''$ and none of these 4 elements are related to any other elements, they must form a single equivalence class. Hence we have

$$[3] = \{3, 4, 6, 12\}.$$
Likewise we can derive another equivalence class

\[ [5] = \{5, 10\} . \]

Because any element of \( A \) is either in the equivalence class \([3]\) or in the equivalence class \([5]\), these two classes are all the distinct equivalence classes.

### 17.3 Topological Sorting

A partial order relation can be used to do a topological sorting, which may find applications in areas such as syntax tree evaluation in compiler construction, or as schedule serialisation in concurrent programming or database management. For further details, interested readers may consult the book *Discrete Mathematics with Applications* by Susanna S Epp, 3rd edition, Thomson Brooks/Cole, 2004.

For any finite set \( A \) and a partial order relation \( \preceq \) on the set, the purpose of topological sorting is to sort all the elements of the set \( A \) into an ordered list such that its sequential order preserves the partial order dictated by the relation \( \preceq \). In other words, if \( a \) and \( b \) are two arbitrary elements of \( A \) and \( a \) precedes \( b \), i.e., \( a \preceq b \), then \( a \) must appear before \( b \) in the resulting topologically sorted list. More precisely, topologically sorting a set \( A \) of \( n \) elements with respect to (w.r.t.) a partial order relation \( \preceq \) is to find an ordered enumeration, \( a_1, a_2, \ldots, a_n \), such that for all \( i \) and \( j \), \( a_i \preceq a_j \) implies \( i \leq j \), where \( a_k \) for \( k = 1, \ldots, n \) denotes the element placed to the \( k \)-th position of the resulting list. We note that the topologically sorted list \( a_1, \ldots, a_n \) can also be characterised by another partial order relation \( \preceq' \) defined by (recall a binary relation on \( A \) is a subset of the Cartesian product \( A \times A \))

\[ \preceq' \overset{\text{def}}{=} \{(a, b) \in A \times A | \exists i, j \text{ such that } a = a_i, b = a_j, i \leq j \} . \]

We observe that the relation \( \preceq' \) is in fact a total order relation and is defined so that \( a_1 \preceq' a_2 \), \( a_2 \preceq' a_3 \), \ldots, \( a_{n-1} \preceq' a_n \). Hence if we sort the set \( A \) w.r.t. the ordering \( \preceq' \) using an usual sorting algorithm such as the insertion sort, we will then obtain exactly the same resulting list \( a_1, \ldots, a_n \).

Two partial order relations \( R_1 \) and \( R_2 \) on the same set \( A \) are said to be compatible if, whenever \( a \) and \( b \) are comparable under both \( R_1 \) and \( R_2 \), we have \( (a, b) \in R_1 \) iff \( (a, b) \in R_2 \). Hence the topological sorting of a set \( A \) w.r.t. a partial order relation \( \preceq' \) can be regarded as the construction of a total order relation \( \preceq' \) such that \( \preceq' \) is compatible with the existing partial order \( \preceq \). In general, a topological sorting doesn’t produce a unique result, unless the existing partial order relation \( \preceq \) is in fact also a total order relation.

#### Example

6. Let \( A = \{F, M, D, S\} \) denote a set of family members, \( F \) (father), \( M \) (mother), \( D \) (daughter) and \( S \) (son). Suppose the family have just acquired a computer game and all wish to play it as soon as possible. In what order can the family take turns to play the game, if
the family tradition that children be given priority when it comes to playing games is to be observed?

**Solution.** It is obvious that there are four acceptable solutions. They are (i) \(D, S, M, F\); (ii) \(D, S, F, M\); (iii) \(S, D, M, F\) and (iv) \(S, D, F, M\). In obtaining any of the above four solutions, we have implicitly done a topological sorting! In fact, the family tradition that children be given priority can be precisely represented by a partial order relation \(\leq\) where all the comparable pairs are list below

\[
D \leq M, \quad D \leq F, \quad S \leq M, \quad S \leq F.
\]

Let us now check that the list (i) is indeed a topological sorting of the set \(A\). That is, the list \(D, S, F, M\) preserves the partial order relation \(\leq\).

First we compare \(D\) with \(S, M\) and \(F\). Since \(D\) and \(S\) are not comparable because neither has the priority, the order these 2 appear in the resulting list is not relevant. Since \(D\) comes ahead of both \(M\) and \(F\) is consistent with the existing partial order, \(D \leq M\) and \(D \leq F\) respectively, we conclude that the first element, \(D\), observes the existing partial order \(\leq\).

We then compare the second element, \(S\), with all of its later elements, \(M\) and \(F\), in the list (i). We can show likewise that \(S\) also preserves the existing partial order. Similarly it can be verified that all elements in the list (i) are ordered consistently with the family tradition characterised by the relation \(\leq\).

The **topological sorting algorithm** for a (nonempty) finite set \(A\) with respect to a partial order relation \(\leq\) is in fact straightforward. The algorithm is as follows.

**Algorithm 17.1** Topological Sorting

(i) Set the resulting list to empty initially.

(ii) Pick any minimal element in \(A\). Append the element to the end of the resulting list and remove the element from the set \(A\).

(iii) Go back to step (ii) if \(A\) is still nonempty. Otherwise the algorithm terminates.

With the assistance of Hasse diagrams, the above topological sorting algorithm can be simplified to the following steps.

**Algorithm 17.2** Topological Sorting with Hasse Diagrams

(i’) Set the resulting list to empty initially.

(ii’) Pick any minimal element of the Hasse diagram. Append the element to the end of the resulting list and remove the element, along with all the edges that are directly connected to it, from the Hasse diagram.

(iii’) Go back to step (ii’) if the Hasse diagram is still nonempty. Otherwise the algorithm terminates.
We note that minimal elements in a Hasse diagram are those bottom vertices in the diagram. By a bottom vertex we mean a vertex that is not downwardly connected to any other vertices in the Hasse diagram.

**Examples**

7. Let $A = \{1, 2, 3, 9, 18\}$ and, for any $a, b \in A$, $a \leq b$ iff $a \mid b$. Construct a topological sorting for the relation $\leq$ on the set $A$.

**Solution.** The diagrams below depict the Hasse diagrams created and utilised in the topological sorting algorithm. To start with, we first set the resulting list $T$ to an empty list, and observe that the Hasse diagram in (A) shows “1” is the only minimal element in $A$. Hence we choose “1” as the 1st element of the resulting list $T$, and remove “1” and the 2 edges that are directly connected to “1”. The resulting Hasse diagram for the new set $\{2, 3, 9, 18\}$, the original set $A$ after the removal of “1”, is then given in (B). There are 2 minimal elements in this case, “2” and “3”. We choose “3” although we could also choose “2”. By removing the selected minimal element “3” and its directly connected edge from the Hasse diagram (B), we obtain the new $T$ and the new Hasse diagram for $\{2, 9, 18\}$ in (C). There are again 2 minimal elements, “2” and “9”. This time we pick “2”. The resulting list $T$ and the shrunk Hasse diagram for $\{9, 18\}$ then become those in (D). We proceed similarly until all the elements of the Hasse diagram have been moved to the resulting list $T$, see (F). The list produced by the topological sorting is thus $1, 3, 2, 9, 18$. We note that the topological sorting is not unique in this case. For instance, the list $1, 3, 9, 2, 18$ is another valid topological sorting.

<table>
<thead>
<tr>
<th>$T : \emptyset$</th>
<th>$T : 1$</th>
<th>$T : 1, 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Diagram A]</td>
<td>![Diagram B]</td>
<td>![Diagram C]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$T : 1, 3, 2$</th>
<th>$T : 1, 3, 2, 9$</th>
<th>$T : 1, 3, 2, 9, 18$</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Diagram D]</td>
<td>![Diagram E]</td>
<td>![Diagram F]</td>
</tr>
</tbody>
</table>
8. Let $A$ be the set of all subsets of set $\{a, b, c\}$, i.e.,

$$A = \{ \emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\} \},$$

and a partial order relation $\leq$ on $A$ be defined by $u \leq v$ iff $u \subseteq v$. Construct a topological sorting for the relation $\leq$ on the set $A$.

**Solution.** The intermediate results, obtained by carrying out steps (i)-(iii) in the topological sorting algorithm, are summarised in the table below.

<table>
<thead>
<tr>
<th>elements of set $A$</th>
<th>minimal elements</th>
<th>pick</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset, {a}, {b}, {c}, {a, b}, {a, c}, {b, c}, {a, b, c}$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>${a}, {b}, {c}, {a, b, c}$</td>
<td>${a}, {b}, {c}$</td>
<td>${b}$</td>
</tr>
<tr>
<td>${a}, {b}, {c}, {a, b, c}$</td>
<td>${a}$</td>
<td>${a}$</td>
</tr>
<tr>
<td>${a, b}, {a, c}, {b, c}, {a, b, c}$</td>
<td>${a, b}$</td>
<td>${a, b}$</td>
</tr>
<tr>
<td>${a, c}, {b, c}, {a, b, c}$</td>
<td>${c}$</td>
<td>${c}$</td>
</tr>
<tr>
<td>${a, c}, {b, c}, {a, b, c}$</td>
<td>${a, c}, {b, c}$</td>
<td>${a, c}$</td>
</tr>
<tr>
<td>${b, c}$</td>
<td>${b, c}$</td>
<td>${b, c}$</td>
</tr>
<tr>
<td>${a, b, c}$</td>
<td>${a, b, c}$</td>
<td>${a, b, c}$</td>
</tr>
<tr>
<td>none</td>
<td>none</td>
<td>stop</td>
</tr>
</tbody>
</table>

Hence the topological sorting for the set $A$ gives

$$\emptyset, \{b\}, \{a\}, \{a, b\}, \{c\}, \{a, c\}, \{b, c\}, \{a, b, c\}.$$ 

We note that the construction of the above table may, or may not, make use of the following list of Hasse diagrams for locating the minimal elements.
9. Find an alternative topological sorting for the problems in the previous two Examples, respectively.

10. Construct a topological sorting for the set $A = \{3, 4, 5, 6, 10, 12\}$ with a partial order relation $R$ on $A$ given by the following Hasse diagram.

![Hasse diagram](image)

**Solution.** One possible topological sorting is 5, 10, 4, 3, 6, 12.

**Exercises**

1. Suppose a partial order relation $R$ on the set $A = \{a, b, c, d, e, f\}$ has the following Hasse diagram

![Hasse diagram](image)

Give the maximal, minimal, greatest and least elements respectively whenever they exist.

2. Let $R$ be a binary relation on $\mathbb{R}$ defined by $xRy$ iff $x \leq y$. Show $R$ is a partial order relation.

3. Let $R$ be a binary relation on $\mathbb{R}$ defined by $xRy$ iff $x < y$. Is $R$ a partial order relation?

4. Suppose $T$ is a rooted tree and $V$ is the set of all the vertices of $T$. For any two vertices $u$ and $v$ of the tree, we define $u \leq v$ if and only if $u$ is a vertex of a subtree of $v$. Show that $\leq$ is a partial order relation on $V$ and that the Hasse diagram is in fact isomorphic to the tree $T$ itself.
18 Switching Circuits and Boolean Algebra

Switching circuits are a way of describing pictorially the symbolic logic that you met earlier. Boolean algebras are abstract mathematical constructions that unify the apparently different concepts of sets, symbolic logic and switching systems. This may be your first encounter with abstract algebra, which includes such things as groups, rings and fields.

18.1 Switching Circuits

A switch is a device which is attached to a point in an electrical circuit. The switch can be in either of two states, open or closed:

- in the open state the switch does not allow current to flow through the point;
- in the closed state the switch does allow current to flow through the point.

We shall indicate a switch by means of the symbols

\[ \begin{array}{c}
\text{open} \\
\text{x}
\end{array} \quad \begin{array}{c}
\text{closed} \\
\text{x}
\end{array} \]

In principle, "x" indicates a sentence such that the associated switch is closed when x is true and it is open when x is false. Another notation is \(-x\).

Two points (available to the outside) are connected by a switching circuit if and only if they are connected by wires on which a finite collection of switches are located.

For example, the following

\[
\begin{array}{c}
\text{switches} \\
\text{y}
\end{array} \quad \begin{array}{c}
\text{switching system} \\
\text{switching system}
\end{array} \
\begin{array}{c}
\text{x} \\
\text{battery}
\end{array} \quad \begin{array}{c}
\text{z} \\
\text{light}
\end{array}
\]

is a switching circuit, making use of an energy source (battery) an output (light) as well as a switching system. If switches x and z are open while switch y is closed, then the state of the switching system may be represented by...
In a switching system, switches may be connected with one another

- **in parallel**: current flows between points $a$ and $b$ iff $x \lor y$ is true

- **in series**: current flows between points $a$ and $b$ iff $x \land y$ is true

- **through the use of complementary switches**: for any given switch $x$, the corresponding complementary switch, denoted by $x'$, is always in the opposite state to that of $x$.

Easy generalisation to the case of a finite number of switches $x_1, x_2, \ldots, x_n$:

- connected in parallel: current flows through the circuit iff $x_1 \lor x_2 \lor \ldots \lor x_n$ is true.
- connected in series: current flows through the circuit iff $x_1 \land x_2 \land \ldots \land x_n$ is true.

In order to describe switching systems formally and mathematically, we denote open and closed states by 0 and 1, respectively. It’s obvious that the state space $S$ for any switch or switching system is composed of two states: 0 (open) and 1 (closed), i.e., $S = \{0, 1\}$.

Furthermore, for any two switches $x$ and $y$, we use $x + y$ and $x \cdot y$ to denote their parallel and series connections, respectively. The symbols “+” and “·” here shouldn’t be confused with those in the arithmetic, although there exist some similarities. With the above introduced notations, we can represent the state of, or the effect of switches in parallel, in series and so on by the following tables.
18.2 Boolean Algebra

A Boolean algebra is essentially a mathematical abstraction and extension of switching systems and/or symbolic logic. A Boolean algebra is the foundation of electronic design of computers. It is a branch of mathematics developed around 1850 by George Boole as rules of algebra for logical thinking. Two preliminary definitions:

- **A binary operation** on a set $A$ takes two elements of the set and produces a third. Formally, it is a mapping $f: A \times A \rightarrow A$, i.e., $f(a, b) \in A$ for any pair $(a, b)$ with $a$ and $b$ both in $A$.

- **A unary operation** on a set $A$ takes one element of the set and produces another. Formally, it is a mapping $f: A \rightarrow A$, i.e., $f(a) \in A$ for any $a$ in $A$.

**Example**

1. For switching systems with state space $S=\{0, 1\}$, the “$+$” and “$\cdot$” operations are binary and the “$'$” operation is unary.

   **Solution.** This is because for any switching systems $x$ and $y$, we have that $x + y$, $x \cdot y$ and $x'$ are all still switching systems with the same state space $S$. 

   **Note.** Binary operator or operation has nothing to do with binary numbers.
More precisely, a Boolean algebra is a set $S$ on which are defined two binary operations $+$ and $\cdot$ and one unary operation $'$ and in which there are at least two distinct elements 0 and 1 such that the following properties hold for all $a, b, c \in S$:

**B1.**

\[
\begin{align*}
  a + b &= b + a \\
  a \cdot b &= b \cdot a
\end{align*}
\]  

\{ commutativity \}

**B2.**

\[
\begin{align*}
  (a + b) + c &= a + (b + c) \\
  (a \cdot b) \cdot c &= a \cdot (b \cdot c)
\end{align*}
\]  

\{ associativity \}

**B3.**

\[
\begin{align*}
  a + (b \cdot c) &= (a + b) \cdot (a + c) \\
  a \cdot (b + c) &= (a \cdot b) + (a \cdot c)
\end{align*}
\]  

\{ distributivity \}

**B4.**

\[
\begin{align*}
  a + 0 &= a \\
  a \cdot 1 &= a
\end{align*}
\]  

\{ identity relations \}

**B5.**

\[
\begin{align*}
  a + a' &= 1 \\
  a \cdot a' &= 0
\end{align*}
\]  

\{ complementation \}

The “0” and “1” in the above are just a notation, two special elements of $S$. They are symbols rather than normal numerical values. Likewise the operators “+”, “·” and “′” are also symbols, each representing the designated special roles. A Boolean algebra is thus often represented by a tuple $(S, +, \cdot, ', 0, 1)$ which carries all the relevant components.

A variable $x \in S$ is called a Boolean variable. A combination of some elements of $S$ (variables) via the connectives $+$ and $\cdot$ and the complement $'$ is a Boolean expression.

Among the 3 operations “′”, “·” and “+” in a Boolean expression, the “′” operation has the highest precedence, then comes the “·” operation, and then the “+” operation. For example, $a + b' \cdot c$ in fact means $a + ((b') \cdot c)$.

Each property in the definition of a Boolean algebra has its dual as part of the definition. The dual is obtained by interchanging $+$ with $\cdot$ and 0 with 1. Unary operation $'$ remains unchanged.

**Theorem 22.** Let $(S, +, \cdot, ', 0, 1)$ be a Boolean algebra. Then the following properties hold for all $a, b, c \in S$.
P1. \( a + a = a; \quad a \cdot a = a \). (idempotent laws)

P2. \( a + 1 = 1; \quad a \cdot 0 = 0 \). (dominance laws)

P3. \((a')' = a\). (double complement)

P4. \( a + a \cdot b = a \). (absorption law)

P5. If \( a + c = 1 \), \( a \cdot c = 0 \), then \( c = a' \). (uniqueness of inverses)

P6. If \( a \cdot c = b \cdot c \), \( a \cdot c' = b \cdot c' \), then \( a = b \). (cancellation law)

P7. If \( a + c = b + c \), \( a + c' = b + c' \), then \( a = b \). (cancellation law)

**Proof.** For P1, the first half is derived from B3–B5 by

\[
 a + a \overset{B4}{=} (a + a) \cdot 1 \overset{B5}{=} (a + a) \cdot (a + a') \overset{B3}{=} a + (a \cdot a') \overset{B5}{=} a + 0 \overset{B4}{=} a ,
\]

where the names on the equality sign indicate the property being used, while the second half is derived by

\[
 a \overset{B4}{=} a \cdot 1 \overset{B5}{=} a \cdot (a + a') \overset{B3}{=} a \cdot a + a \cdot a' \overset{B5}{=} a \cdot a + 0 \overset{B4}{=} a \cdot a .
\]

For P2, we need to observe

\[
 a + 1 \overset{B5}{=} a + (a + a') \overset{B2}{=} (a + a) + a' \overset{P1}{=} a + a' \overset{B5}{=} 1 , \quad a \cdot 0 \overset{B5}{=} a \cdot (a \cdot a') \overset{B2}{=} (a \cdot a) \cdot a' \overset{P1}{=} a \cdot a' \overset{B5}{=} 0 .
\]

The proof of P3 is

\[
 a' \overset{B4}{=} a' \cdot 1 \overset{B5}{=} a' \cdot (a + a') \overset{B3}{=} a' \cdot a + a' \cdot a' \overset{B5}{=} a' \cdot a + 0 \overset{B5}{=} a' \cdot a + a' \cdot a \overset{B3}{=} (a' + a') \cdot a \overset{B5}{=} 1 \cdot a \overset{B4}{=} a .
\]

The proof of P4 is

\[
 a + a \cdot b \overset{B4}{=} a + 1 + a \cdot b \overset{B3}{=} a + (1 + b) \overset{B1, P2}{=} a \cdot 1 \overset{B4}{=} a .
\]

The other properties, P5–P7, can be derived similarly.

**Note.** An immediate corollary of P5 in the above theorem is that \( 0' = 1 \) and \( 1' = 0 \) hold on any Boolean algebra \((S, +, \cdot, \cdot', 0, 1)\).

**Examples**

2. Switching system \((S, +, \cdot, \cdot', 0, 1)\) with \( S = \{0, 1\} \) is a Boolean algebra.

**Solution.** We need to show B1 – B5, by letting “+”, “.” and “’” specifically denote switches in parallel, in series and in complementation respectively.

The identities in B1 are valid because
The proof of B2 – B5 is elementary. Hence we’ll simply show only the first half of B3, i.e., \( a + (b \cdot c) = (a + b) \cdot (a + c) \). We’ll show the identity by the use of the evaluation table below

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

all corresponding values exactly same

3. Let \( \mathbb{R} \) be the set of real numbers, \(+, \times\) be the normal addition and multiplication, \(0, 1 \in \mathbb{R}\) be the normal numbers. If we define \( \gamma \) by \( \gamma' = 1 - \gamma \) for any \( \gamma \in \mathbb{R} \), then is \((\mathbb{R}, +, \times', 0, 1)\) a Boolean algebra?

**Solution.** No. Because neither the first half of B3 nor the second half of B5 is satisfied. For example,

\[
1 + (2 \times 3) \neq (1 + 2) \times (1 + 3).
\]

4. Suppose \( S = \{1, 2, 3, 5, 6, 10, 15, 30\} \). Let \( a+b \) denote the least common multiple of \( a \) and \( b \), \( a \cdot b \) denote the greatest common divisor of \( a \) and \( b \), and \( a' = \frac{30}{a} \). Prove \((S, +, \gamma', 1, 30)\) is a Boolean algebra.

**Solution.** Please try it yourself.
18.3 Algebraic Equivalence

Switching systems and symbolic logic are essentially the same. Furthermore they both form a Boolean algebra \((S, +, \cdot, ', 0, 1)\) on a set \(S = \{0, 1\}\), where 0 is open in the switching systems and is \(F\) in the symbolic logic, while 1 is closed in the switching systems and is \(T\) in the symbolic logic. The correspondence can be seen in the following table:

<table>
<thead>
<tr>
<th>Boolean algebra</th>
<th>Switching systems</th>
<th>Symbolic logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S = {0, 1})</td>
<td>{open, closed}</td>
<td>{(F, T)}</td>
</tr>
<tr>
<td>+</td>
<td>(x + y) (in parallel)</td>
<td>(p \lor q) (“or”)</td>
</tr>
<tr>
<td>(\cdot)</td>
<td>(x \cdot y) (in series)</td>
<td>(p \land q) (“and”)</td>
</tr>
<tr>
<td>'</td>
<td>(x') (complement)</td>
<td>(\sim p) (“not”)</td>
</tr>
<tr>
<td>0</td>
<td>circuit open</td>
<td>contradiction</td>
</tr>
<tr>
<td>1</td>
<td>circuit closed</td>
<td>tautology</td>
</tr>
</tbody>
</table>

Hence properties B1–B5 and P1–P7 will also hold in symbolic logic, when “+”, “−”, “′”, “0” and “1” are replaced by “\(\lor\)”, “\(\land\)”, “\(\sim\)”, contradiction and tautology, respectively. Hence, for example,

\[
\begin{align*}
\quad a \lor (b \land c) & \equiv (a \lor b) \land (a \lor c), \quad a \land (b \lor c) \equiv (a \land b) \lor (a \land c), \\
\quad a \lor a & \equiv a, \quad a \land a \equiv a, \quad \sim (\sim a) \equiv a, \quad a \lor \bot \equiv a, \quad a \land \top \equiv a, \\
\quad a \lor (a \land b) & \equiv a, \quad a \land (\sim a) \equiv \bot, \quad a \lor (\sim a) \equiv \top, \quad a \land \bot \equiv \bot, \quad a \lor \top \equiv \top
\end{align*}
\]

hold for any propositions \(a, b\) and \(c\), where \(\bot\) represents a contradiction and \(\top\) represents a tautology.

**Examples**

5. Convert \((p \lor q) \rightarrow r\) into the corresponding Boolean expression.

**Solution.** Since \(p \rightarrow q\) is equivalent to \((\sim p) \lor q\), we see that \((p \lor q) \rightarrow r\) is equivalent to \((\sim(p \lor q)) \lor r\) which is thus converted to \((p + q)' + r\).

6. (De Morgan’s Laws) Let \((S, +, \cdot, ', 0, 1)\) be a Boolean algebra, then for any \(x, y \in S\)

\[
(x + y)' = x' \cdot y', \quad (x \cdot y)' = x' + y'.
\]

**Solution.** Proof obvious from the theorem in the previous section.
7. Suppose \((T, \ast, \circ, \ominus, \cup, \cap)\) is a Boolean algebra. Show \((\cup \cap)\cup = \cap\).

**Solution.** Recall that when we say \((T, \ast, \circ, \ominus, \cup, \cap)\) is a Boolean algebra, the tuple means that \(\ast, \circ, \ominus\) correspond respectively to the “+”, “·” and “′” operations entailed by a Boolean algebra, and that \(\cup\) and \(\cap\) correspond respectively to the “0” and “1” elements possessed by the Boolean algebra. Hence \((\cup \cap)\cup = \cap\) is the same as \((0′) + 0 = 1\) and is thus obviously true.

8. Let \(B = \{0, 1\}\) and \((B, +, \cdot, ′, 0, 1)\) be a Boolean algebra. Let \(B^n\) denote the set of all the tuples \((x_1, x_2, \ldots, x_n)\) with \(x_1, \ldots, x_n\) being any elements of \(B\), i.e. \(B^n \overset{def}{=} \{(x_1, \ldots, x_n) | x_1 \in B, \ldots, x_n \in B\}\). Then \((B^n, +, \cdot, ′, 0, 1)\) is also a Boolean algebra if

\[
\begin{align*}
0 &= (0, \ldots, 0), \\
1 &= (1, \ldots, 1), \\
x + y &= (x_1 + y_1, \ldots, x_n + y_n), \\
x \cdot y &= (x_1 \cdot y_1, \ldots, x_n \cdot y_n), \\
x′ &= (x_1′, \ldots, x_n′).
\end{align*}
\]

\[\blacksquare\]

### 18.4 Sets connection with Boolean Algebra

If we make the correspondence between

\(\emptyset, U, \cup, \cap, ′\) and \(0, 1, +, \cdot, ′\)

for sets and Boolean algebra, respectively, we see that properties S1–S5 are exactly those B1–B5 for the definition of Boolean algebra.

Hence for any nonempty set \(S, (\mathcal{P}(S), \cup, \cap, ′, \emptyset, S)\) is a Boolean algebra.

**Theorem 23.** If \(\mathcal{B}\) is a Boolean algebra with exactly \(n\) elements, then \(n = 2^m\) for some \(m\). Furthermore, \(\mathcal{B}\) and \((\mathcal{P}\{1, 2, \ldots, m\}, \cup, \cap, ′, \emptyset, \{1, 2, \ldots, m\})\) essentially represent the same Boolean algebra.

**Example**

9. Let \(U = \{2, 3\}\) and \(S = \mathcal{P}(U)\). Then \((S, \cup, \cap, ′, \emptyset, U)\) is a Boolean algebra. Notice that the set \(S\) in this case contains more than 2 elements.
Exercises

1. Represent the Boolean expression \((a \cdot b + c) \cdot (b + d)\) as a switching system. Is the switching system conductive when switches \(a\) and \(b\) are closed and switches \(c\) and \(d\) are open?

2. Let \(\mathbb{Z}\) be the set of all integers and let \(\mathbb{N}\) be the set of all nonnegative integers. Is the mapping \(f : \mathbb{Z} \rightarrow \mathbb{Z}\) with \(f(n) = n^2\) exactly the same as the mapping \(g : \mathbb{Z} \rightarrow \mathbb{N}\) with \(g(n) = n^2\)?
19  Boolean Functions

Given a set $B = \{0, 1\}$ and a positive integer $n$, a **Boolean function** (a.k.a. **truth function**, **switching function**) of $n$ variables, $f(x_1, \cdots, x_n)$, is a mapping on $B^n$ into $B$,

$$f : B^n \to B$$

The most elementary way to define a Boolean function is to provide its **truth table**, a complete list of all the points in $B^n$ together with the value of the function at each point. We say that the Boolean function is defined explicitly and it is completely determined by the corresponding truth table and vice versa. In general, for $f : B^n \to B$, there are $2^n$ rows!

19.1  Convert Boolean Function to Boolean Expression

As explicit definition of a Boolean function grows exponentially on $n$, a better way to represent it is implicitly, that is – as a **sum of product terms**. Let $B = \{0, 1\}$ and $f : B^n \to B$ be a Boolean function. If $f$ is not already explicitly defined by a truth table, we can easily establish one first. The procedure is as follows

(i) List all possible input states and the corresponding function values, i.e., list the truth table

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$\cdots$</th>
<th>$x_{n-1}$</th>
<th>$x_n$</th>
<th>$f(x_1, \cdots, x_n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>*</td>
</tr>
</tbody>
</table>

(ii) remove all rows whose output state is 0, i.e., $f(x_1, \cdots, x_n) = 0$.

(iii) replace 0 and 1 in $x_i$–columns by $x'_i$ and $x_i$, respectively, in the Boolean expression.

(iv) Sum up all the (input state) product terms.

To get rid of parenthesis when writing Boolean expressions,

(a) keep in mind the order of precedence $\land \lor$

(b) discard any parenthesis that become redundant as a consequence of associativity of $\land$ and $\lor$. 

152
Example

1. Let \( f: \{0, 1\}^3 \rightarrow \{0, 1\} \) be given by \( f(x, y, z) = xyz + \min(x, (y + z)') \). Here \( \min \) means minimum. Write \( f \) as a Boolean expression, i.e., as a sum of product terms.

   **Solution.** From the following truth table

   \[
   \begin{array}{ccc|c}
   x & y & z & f(x, y, z) \\
   \hline
   0 & 0 & 0 & 0 \\
   0 & 0 & 1 & 0 \\
   0 & 1 & 0 & 0 \\
   0 & 1 & 1 & 0 \\
   1 & 0 & 0 & 1 \\
   1 & 0 & 1 & 0 \\
   1 & 1 & 0 & 0 \\
   1 & 1 & 1 & 1 \\
   \end{array}
   \]

   We see that only two rows will actually contribute to the final Boolean expression

   \[
   f(x, y, z) = xy'z' + xyz
   \]

   where the binary operator \( \cdot \) is omitted for simplicity.

19.2 Canonical Form

Once we know that all Boolean functions can be expressed as a sum of product terms, it will be meaningful to introduce the concept of canonical representation so that each function will have a unique canonical form.

Let \( (\mathcal{B}, +, \cdot, \cdot, 0, 1) \) be a Boolean algebra and \( x, y \) and \( z \) be any variables taking values in \( \mathcal{B} \), then

- **_literals** are individual variables or their complements, e.g. \( x, x', y, y' \)

- **product terms** are expressions made up of literals and the binary operation \("\cdot\) although the symbol \("\cdot\) is optional, e.g. \( x, x', x \cdot y'z, x \cdot y \cdot x'y \cdot z \)
Boolean expression is a sum of product terms, e.g. \( x + x' \cdot y(z + y') \)

standard product term of an expression is a product in which each variable appears in the product exactly once, e.g. \( xy' \) is a standard product of \( xy' + x + x'y' \), but is not a standard product of \( xy' + xyz \) because variable \( z \) is missing from the product.

a Boolean expression is canonical if it is a sum of distinct standard product terms, e.g., \( xy + x'y' \) is canonical, but \( xy + x'y' + z \) is not canonical.

Given \( n \) Boolean variables, one can obtain \( 2^n \) distinct standard product terms. The procedure to convert a Boolean function into its canonical form is as follows:

Algorithm 19.1 Boolean Canonical Form

1. expand the expression into a sum of standard product terms. Use the distributive law \( a \cdot (b + c) = a \cdot b + a \cdot c \) for this.

2. inspect each term to see if it contains all the variables.

3. if a term misses one or more variables, it is multiplied with an expression such as \( a + a' ( = 1) \) where \( a \) is one of the missing variables.

Two Boolean expressions are equivalent if they represent the same Boolean function. Two Boolean expressions are equivalent if they either

- have the same truth table, or
- can be transformed from one to another through the use of the axioms B1 – B5 in the previous lecture, and their induced properties such as the De Morgan’s Laws, or
- have the same canonical form.

Example

2. Show Boolean expressions \( x \) and \( xy + xy' \) are equivalent.

   (a) By truth table
(b) By properties of Boolean algebra

\[ xy + xy' = x(y + y') = x \cdot 1 = x \]

(c) By canonical form

\[
\begin{align*}
  x &= x \cdot (y + y') \\
  &= xy + xy' \\
  xy + xy' &= xy + xy'
\end{align*}
\]

\[
\text{same canonical form}
\]
19.3 Minimal Representation

Among equivalent Boolean functions or expressions, it is natural that we often want a simpler or even “simplest” form. To serve this purpose we call a Boolean expression $E$ a **minimal representation** of a Boolean function $f$ if $E$ represents $f$ and

(i) $E$ is a sum of product terms;

(ii) if $F$ is any sum of product terms representing $f$, then the number of product terms in $E$ is no greater than that in $F$;

(iii) if $F$ is any sum of product terms representing $f$ and having the same number of product terms as in $E$, then the total number of literals in $E$ is no greater than that in $F$.

**Examples**

3. Find the minimal representation for $f = a' \cdot b' + a' \cdot b$.

**Solution.** First we observe

$$f = a' \cdot b' + a' \cdot b = a' \cdot (b' + b) = a' \cdot 1 = a'$$

Since $f$ is not identical to 0 or 1, and its equivalent expression $x$ has only 1 product term and 1 literal in total, this equivalent Boolean expression must be a minimal representation of $f$.

4. Find the minimal representation for $f = xy + xy + xx' + xy'$.

**Solution.** First we observe

$$f = (xy + xy) + xx' + xy' = xy + 0 + xy' = xy + xy' = x(y + y') = x \cdot 1 = x$$

Since $f$ is not identical to 0 or 1, and its equivalent expression $x$ has only 1 product term and 1 literal in total, this equivalent Boolean expression must be a minimal representation of $f$.

5. Show $f = xyz' + xy'z + xy'z' + x'y'z + x'yz' + x'yz$ is equivalent to any of the four Boolean expressions below
(a) \(xy' + x'y + xyz'\)  
(b) \(xy' + x'y + xz'\)

(c) \(xy' + x'y + yz'\)  
(d) \(xy' + x'yz + yz'\)

and (a) and (d) are not minimal.

**Solution.** Since \(f\) is already in canonical form, once we convert (a)–(d) into canonical forms we should see that the newly obtained canonical forms are in exactly the same form as \(f\) itself. For (a), for instance, we have

\[
xy' + x'y + xyz' = xy'(z + z') + x'y(z + z') + xyz' \\
= xy'z + xy'z' + x'yz + x'yz' + xyz'
\]

which is same as \(f\). We can also derive (a)–(d) directly from \(f\). For (c), for instance, we have

\[
f = xyz' + xy'z + xy'z' + x'yz + x'yz' \\
= (xy'z + xy'z') + (xyz' + x'yz') + (x'yz + x'yz') \\
= xy'(z + z') + (x + x')yz' + x'y(z + z') \\
= xy' + yz' + x'y.
\]

It is easy to see that expressions (a)–(d) all have 3 product terms. However (a) and (d) both have 7 literals while (b) and (c) both have only 6 literals. Hence neither (a) nor (d) is a minimal representation. We finally note that (b) and (c) in Example 4 are minimal representations.

6. The Boolean expression \(f\) in Example 4 may be drawn as a switching circuit below according to (b)

7. The keys to Boolean minimisation lie in the following theorems:

\[
a + a \cdot b = a \quad a \cdot (a + b) = a \quad \text{(Absorption)}
\]

\[
a + a' \cdot b = a + b \quad a \cdot (a' + b) = a \cdot b
\]

\[
a \cdot b + a \cdot b' = a \quad (a + b) \cdot (a + b') = a \quad \text{(Logic Adjacency)}
\]
The proof is immediate, for instance:

\[ a + a' \cdot b = a + a \cdot b + a' \cdot b = a + (a + a') \cdot b = a + 1 \cdot b = a + b \]
\[ a \cdot b + a \cdot b' = a \cdot (b + b') = a \cdot 1 = a \]

In general, it is difficult to show a Boolean expression is minimal from the very definitions. We shall instead claim that the simplified forms provided by Karnaugh maps, to be given shortly, are minimal representations.

### 19.4 Logic Gates

A **logic circuit** is a computer switching circuit that performs some processing or controlling function, that is some logical operations on data. It can be seen as a **black box**, with some inputs and some outputs.

![Logic circuit diagram]

One can distinguish between the **analysis problem** – given a logic circuit, determine binary outputs for each combination of inputs, and the **design problem** – given a task, develop a circuit that accomplishes the task.

Claude Shannon (1938) realised that Boolean algebra could be used in electronic design by modeling logic gates. Electronically, logic circuits are typically implemented through the use of a collection of **logic gates**. A logic gate is implemented by means of a Boolean function or expression and is, as usual, characterised by their input-output correspondence. Physically, every terminal (input or output) in a logic gate is in one of the two binary conditions, represented by different voltage levels:

- **low** (0) \( \approx \) zero volts (0 V)
- **high** (1) \( \approx \) five volts (+5 V)

As the circuit processes data, the logical state of a circuit changes often. This is a good example of a discrete approximation of an analogue phenomenon. The most popular five logic gates are as follows.

- **A NOT-gate**, taking \( p \) as the input, will output the logical value \( p' \):

  \[
  \begin{array}{cc}
  p & p' \\
  0 & 1 \\
  1 & 0 \\
  \end{array}
  \]
An **AND-gate**, taking \( p \) and \( q \) as the input, will output the logical value \( pq \):

\[
\begin{array}{ccc}
p & q & pq \\
0 & 0 & 0 \\
0 & 1 & 0 \\
1 & 0 & 0 \\
1 & 1 & 1 \\
\end{array}
\]

An **OR-gate**, taking \( p \) and \( q \) as the input, will output the logical value \( p + q \):

\[
\begin{array}{ccc}
p & q & p+q \\
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 1 \\
\end{array}
\]

An **NAND-gate**, taking \( p \) and \( q \) as the input, will output the logical value \((pq)’\); or it can be built through the use of an AND-gate and a NOT-gate:

\[
\begin{array}{ccc}
p & q & (pq)’ \\
0 & 0 & 1 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\end{array}
\]

An **NOR-gate**, taking \( p \) and \( q \) as the input, will output the logical value \((p + q)’\); or it can be built from an OR-gate and a NOT-gate:

\[
\begin{array}{ccc}
p & q & (p+q)’ \\
0 & 0 & 1 \\
0 & 1 & 0 \\
1 & 0 & 0 \\
1 & 1 & 0 \\
\end{array}
\]

**Examples**

6. Draw a gate implementation for the Boolean expression \( xy’z’ + xyz \).

**Solution.**
The idea here is to work backwards, that is, from right to left, building up the diagram. A simple example is given in Tutorial 8.

7. Show that any Boolean expression can be represented by NAND-gates alone.

**Solution.** Since every Boolean expression can be represented by some AND-gates, OR-gates and NOT-gates, if we can construct these 3 types of gates with NAND-gates alone, then we have shown that any Boolean expression can be presented by just the NAND-gates. Since we can construct NOT-gates and AND-gates respectively via

\[
\begin{align*}
(p'q')' &= p + q
\end{align*}
\]

because \((p'q')' = p + q\), we conclude that NAND-gates alone are enough to represent any Boolean expressions.

**Exercises**

1. Use Boolean algebra properties to show

   (a) \(x + x + x + yz + xx' + yz = x + yz\).
   
   (b) \(x = xyz + xy'z + xyz' + xy'z'\).
2. Let a Boolean function $f(x, y, z)$ be given by the table

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
<th>$f(x, y, z)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Give a Boolean expression that corresponds to this function.

3. Implement the Boolean expression $xy$ through the use of NOR-gates alone.

4. Design a switching circuit for a light that has 3 switches. The light has to be able to be turned on or off from any of the 3 switches.
20 Karnaugh Maps

The Karnaugh map provides a “pictorial” technique to simplify Boolean expressions using the axiom $a + a' = 1$ and occasionally the property $a + a = a$. One of the fundamental algorithms in the field of CAD, Karnaugh maps are used for many small design problems and constitute the starting point for many other algorithms. The idea is to visualise adjacency in Boolean space by using a projection of an n-dimensional hypercube onto two-dimensional rectangle such that adjacent points in the hypercube remain adjacent in the projection.

20.1 Grid Layout for Karnaugh Maps

- Two product terms are said to be **adjacent**, if they differ by exactly one literal and if this literal appears in one product and its complement in the other. For example, $wx'y'z'$ and $wx'yz'$ are adjacent (notice $y$ and $y'$).

- Since a Boolean expression has a canonical form, we draw a grid or table such that all possible standard products have a unique position or box in the grid. For example, Boolean expression $f(x, y, z)$ may be associated with the grid:

\[
\begin{array}{cccc}
xy & xy' & x'y' & x'y \\
z & & & \\
z' & & & \\
\end{array}
\]

in which the terms on the edges of the grid serve as labels. More specifically, a special case of Boolean expression $xy + y'z$, for instance, has the canonical form

$$xyz + xyz' + xy'z + x'y'z$$

which can be represented by placing a “1” in the corresponding box in the above grid for each standard product term in the canonical form, leaving the rest of the boxes (if any) empty or filled with “0”. The finished grid here, called a Karnaugh map, reads

\[
\begin{array}{cccc}
xy & xy' & x'y' & x'y \\
z & 1 & 1 & 1 \\
z' & 1 & & \\
\end{array}
\]

The grids for Karnaugh maps should satisfy
- each standard product term is uniquely represented by a box,
- adjacent boxes (vertically or horizontally or in other legal directions) represent adjacent product terms,
- each box has exactly $n$ neighbours (adjacent product terms) if the Boolean expression has exactly $n$ variables.

**Examples**

1. For Boolean expressions with variables $x, y$ and $z$,

- $xy$  $xy'$  $x'y'$  $x'y$
- $y'x'$  $y'x$  $yx$  $yx'$

are both valid grids, and there are other possible valid grids as well.

2. For a Boolean expression with 4 variables $w, x, y$ and $z$, a typical valid grid would be

- $wx$  $wx'$  $w'x'$  $w'x$
- $yz$
- $yz'$
- $y'z'$  $\text{shaded box}$
- $y'z$

where $1, 2, 3$ and $4$ are the 4 neighbours of the shaded box.

3. For a Boolean expression with 5 variables $v, w, x, y$ and $z$, the following grid
is a typical grid and shaded box has 5 neighbours \(1, 2, \ldots, 5\).

The main feature of a Karnaugh map is that any block of boxes (of “1”) can be represented by a single product term if the number of boxes in each dimension or direction is of the form \(2^m\) for some \(m \in \mathbb{N}\). In fact we’ll reserve the word block exclusively for this sense.

- A block of \(2^M\) boxes in a Boolean expression of \(N\) variables can be represented by a single product term with \(N - M\) literals.
- If a variable changes when we move inside a block, the literals related to the variable will not exist in the reduced single product term.

**Examples**

4. The circled area in the Karnaugh map is a \(2 \times 1\) block representing

\[
f(x, y) = xy + xy' = x(y + y') = x \cdot 1 = x
\]

In other words, the block of two is simplified to a single product term \(x\) (with \(1 = N - M = 2 - 1\) literals).

5. The circled area in the Karnaugh map is a \(2 \times 2\) block representing

\[
xyz + xyz' + x'yz + x'y'z' = xy(z + z') + x'y(z + z') = xy + x'y = (x + x')y = y
\]

In other words, the block of 4 is simplified to a single product term \(y\) (with \(1 = N - M = 3 - 2\) literals)
6. The circled area in the Karnaugh map is a $2 \times 4$ block representing $z$. Notice that in a typical term $wx'w'x'w'x$ in the block, the variable names $w$, $x$, and $y$ may change inside the circled block, and are thus not present in the final product term which becomes $z$.

## 20.2 Simplification Procedure

The procedure for simplifying a Boolean expression via Karnaugh map is as follows

**Algorithm 20.1 Karnaugh Map**

(i) Write the expression in terms of standard product terms, that is, in canonical form.

(ii) Place a 1 in the box corresponding to each standard product term.

(iii) Circle isolated 1’s.

(iv) Circle 2 adjacent 1’s (pairs in any direction, or blocks of 2) such that the newly circled 1’s can’t be contained in a block of 4. Use as few circles as possible.

(v) Circle any blocks of 4 (containing at least one uncircled 1) such that the newly circled 1’s can’t be contained in a block of 8. Use as few circles as possible.

(vi) Circle blocks of $2^n (n \geq 3)$ boxes similar to (v).

Then the simplified expression can be read off the Karnaugh map with each circle representing one product term. The term that changes, from $x$ to $x'$ say, within a circle is the one that "disappears". The final result is a minimal representation of the Boolean expression.

**Examples**

7. Use a Karnaugh map to simplify $F = xyz + x'y'z'$.

**Solution.**
(a) $F = xyz + x'y'z + x'y'z' + x'y'z$ is the canonical form.

<table>
<thead>
<tr>
<th></th>
<th>xy</th>
<th>xy'</th>
<th>x'z</th>
<th>x'y</th>
<th>y'z</th>
</tr>
</thead>
<tbody>
<tr>
<td>z</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>z'</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(b) No isolated 1's.

(c) No blocks of 4. But there are pairing choices. We select least number of pairs.

<table>
<thead>
<tr>
<th></th>
<th>xy</th>
<th>xy'</th>
<th>x'z</th>
<th>x'y</th>
<th>y'z</th>
</tr>
</thead>
<tbody>
<tr>
<td>z</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>z'</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(d) No blocks of 4 means no blocks of $2^m$ for $m \geq 2$.

In the circle in the first row $x$ changes to $x'$ so it "disappears", leaving $yz$. In the other circle, $y$ changes, so it goes leaving $x'z'$. Thus the minimal representation can be read off as $yz + x'z'$.

8. Suppose a Boolean expression can be represented by the following Karnaugh map

<table>
<thead>
<tr>
<th></th>
<th>wx</th>
<th>wx'</th>
<th>w'x</th>
<th>w'x'</th>
</tr>
</thead>
<tbody>
<tr>
<td>yz</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>yz'</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y'z'</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>y'z</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Then the simplification procedure is as follows.
(a) No isolated 1’s.

(b) Only 2 possible pairs exist, that can’t be included in a block of 4 (1’s). Since a 2nd pair will not cover any uncircled 1’s that will not be covered by blocks of 4, we need to pick just 1 pair.

(c) No blocks of 8. But there are 3 blocks of 4, each covering new 1’s. The Karnaugh map now looks like

(d) Since no uncircled 1’s are left over the procedure is completed.

In the circle in the first row both w and x change so they “disappear”, leaving yz. In the circle in the top left corner, x and z change so they go, leaving wy. And so on. The minimal representation is thus read off as

\[ w'y z' + w' y + w y' z' \]

9. If one reverses the procedure and goes from big blocks to smaller ones, the results may not be minimal. For instance, with bigger block first, we would have to circle in the Karnaugh map below in the following order

which gives 5 product terms (corresponding to 5 circles). However, if we drop the block of 4 circle, we still have all the 1’s circled. But the new and equivalent expression will only have 4 product terms. With our normal procedure, nevertheless, we’ll arrive at the correct answer represented by

167
Exercises

1. Among all the product terms in the Boolean expression

\[ x'y'z' + xy'z' + xz + xy' + x'y'z + xyz + x'y'z' , \]

which are adjacent to the product term \( xy'z' \)? Write down the canonical form of the above Boolean expression.

2. Simplify the Boolean expression \( x + x'yz' \).

3. Give an alternative minimal representation, if any, for the Boolean expression \( yz' + y'z + wx' \).

4. Use a Karnaugh map to simplify the Boolean expression

\[ vwy + vw'yz + v'y'z + vwz' + v'wx'z' + vw'x'y'z' \]

Is the minimal representation unique in this case?
21 Recurrence Relations

Recurrence relations are important in quite a large number of areas of mathematics. In particular, they crop up in the analysis of algorithms (see section 5.3 and tutorial 1) because many algorithms, particularly divide and conquer algorithms, have time complexities which are naturally modeled by recurrence relations. The famous Mandelbrot set arises out of a recurrence relation.

21.1 Recurrence Relations

A sequence is an ordered list of objects or events. The number of elements (terms, members) is called the length of the sequence. A sequence can be finite (e.g., alphabet letters) or infinite (e.g., positive integers). In a sequence, an element can appear multiple times in different positions. Elements of a sequence are usually identified by their position (the first, the third, the 6\textsuperscript{th}, etc.).

The easiest way to create an ordered list of objects or events is by using a function defined on the set of natural numbers with values in the set, say $A$, of given objects or events. Then a sequence is the image of this function and we denote it by \( \{a_i\}_{i \in \mathbb{N}} \) such that $i \mapsto a_i, \ a_i \in A$.

Any equation involving several terms of a sequence is called a recurrence relation. Because sequences are ordered, a recurrence relation for a given sequence $\{a_i\}_{i \geq m}$ is an equation that expresses each $a_n$ in terms of one or more of the previous terms of the sequence (namely $a_0, a_1, \ldots, a_{n-1}$) for all integers $n$ with $n \geq m$. The first few elements $(a_0, a_1, \ldots, a_{m-1})$ in the sequence that can not be related to each other by the recurrence relation are often determined by the initial conditions. A recurrence relation is sometimes also called a difference equation. Formally, a recurrence relation is an equation of the type

$$F(n, a_n, a_{n+1}, \ldots, a_{n+m}) = 0,$$

where $m \in \mathbb{N}$ is fixed. To solve a recurrence relation means to eliminate recursion from the function definition.

The order of a recurrence relation is the difference between the greatest and the lowest subscripts of the terms of the sequence in the equation. For example, in the above equation, the order is $(n + m) - (n) = m$.

A sequence is called a solution of a recurrence relation if its terms satisfy the recurrence relation. A recurrence relation can have multiple solutions but a recurrence relation with initial conditions has an unique solution.
Examples

1. For the sequence \( \{a_i\}_{i \in \mathbb{N}} \), the following formula

\[
a_n = 7a_{n-1} - 5a_{n-2}
\]

is a recurrence relation valid for \( n \geq 2 \). The elements in the sequence that are not related by the above formula are \( a_0 \) and \( a_1 \). Hence \( a_0 \) and \( a_1 \) can be determined by the initial conditions. Once the values of \( a_0 \) and \( a_1 \) are specified, the whole sequence \( \{a_i\}_{i \geq 0} \) is completely specified by the recurrence relation.

2. Let a sequence \( \{a_i\}_{i \in \mathbb{N}} \) be determined by the recurrence relation

\[
a_n = 3a_{n-1} + 2a_{n-2}
\]

and the initial conditions

\[
a_0 = 1, \quad a_1 = 2.
\]

Calculate \( a_4 \) recursively first. Then calculate \( a_4 \) again iteratively.

**Solution.** Recursively, we use (*) repeatedly (“topdown”) to decrease the indices involved until they all reach the initial ones. Hence

\[
\begin{align*}
a_4 &= 3a_3 + 2a_2 \quad \text{used (*) for } n = 4 \\
    &= 3(3a_2 + 2a_1) + 2a_2 \quad \text{used (*) for } n = 3 \\
    &= 11a_2 + 6a_1 \\
    &= 11(3a_1 + 2a_0) + 6a_1 \quad \text{used (*) for } n = 2 \\
    &= 39a_1 + 22a_0 \\
    &= 39 \times 2 + 22 \times 1 = 100. \quad \text{used (**)}
\end{align*}
\]

Iteratively, we use (*) repeatedly (“building-up”) to derive more and more known elements until the desired index is reached. Hence

\[
\begin{align*}
a_0 &= 1, \\
a_1 &= 2, \\
a_2 &= 3a_1 + 2a_0 = 8, \quad \text{used (*) for } n = 2 \\
a_3 &= 3a_2 + 2a_1 = 28, \quad \text{used (*) for } n = 3 \\
a_4 &= 3a_3 + 2a_2 = 100. \quad \text{used (*) for } n = 4
\end{align*}
\]
3. Let $f(n)$ for $n \in \mathbb{N}$ be given by the recurrence relation

$$f(n) = nf(n-1), \quad n \in \mathbb{N}, n \geq 1$$

$$f(0) = 1 \quad \text{(initial condition)}$$

Find the solution $f(n)$.

**Solution.** We first derive

$$f(n) = n f(n-1) \quad \text{if } n \geq 1$$

$$= n(n-1)f(n-2) \quad \text{if } n \geq 2$$

$$= n(n-1)(n-2)f(n-3) \quad \text{if } n \geq 3$$

$$= \ldots$$

$$= n(n-1)(n-2) \cdots 2 \cdot 1 f(0)$$

$$= n! f(0) = n! \cdot 1 = n!$$

Then we can show inductively $f(n) = n!$ for $n \geq 0$.

### 21.2 Basic Concepts Related to Recurrence Relations

A **linear equation** is an algebraic equation in which each term is either a constant or the product of a constant and the first power of a single variable.

A recurrence relation of order $m$ is said to be **linear** if it is linear in $a_n, a_{n+1}, \ldots, a_{n+m}$. Otherwise, the recurrence equation is said to be non-linear (usually, very hard to solve).

The general linear recurrence relation of order $m$ has the form

$$s_m(n)a_{n+m} + s_{m-1}(n)a_{n+m-1} + \cdots + s_1(n)a_{n+1} + s_0(n)a_n = g(n), \quad n \geq 0$$

where $s_0(n), s_1(n), \ldots, s_m(n), g(n)$ are given functions. If these ”s“ functions are constants (they don’t depend on the index $n$ explicitly), say $s_i(n) = c_i$ for all $n \in \mathbb{N}, i = 0, 1, 2, \ldots, m$, then an **$m$–th order linear, constant coefficient recurrence relation** on a sequence $\{a_n\}_{n \geq 0}$ is a recurrence relation which can be written in the form

$$c_m a_{n+m} + c_{m-1} a_{n+m-1} + \cdots + c_1 a_{n+1} + c_0 a_n = g(n), \quad n \geq 0 \quad \text{(***)}$$

where $c_0, \cdots, c_m$ are constants, $c_0 c_m \neq 0$ (why?), and $g(n)$ is a function of $n$.

If furthermore $g(n) = 0$ for all $n$, then the relation is said to be **homogeneous**

$$c_m a_{n+m} + c_{m-1} a_{n+m-1} + \cdots + c_1 a_{n+1} + c_0 a_n = 0, \quad n \geq 0$$
Examples

4. $a_n = 5a_{n-1} + 2a_{n-2} + 3^n$, $n \geq 2$, is a second order linear, constant coefficient recurrence relation and is non-homogeneous. This is because we can equivalently rewrite it as

$$a_{k+2} - 5a_{k+1} - 2a_k = 3^{k+2}, \quad k \geq 0.$$

In terms of the notation in (**) we have in this case $m = 2$ and

$$c_m = c_2 = 1, \quad c_{m-1} = c_1 = -5, \quad c_{m-2} = c_0 = -2, \quad g(n) = 3^{n+2}.$$

5. $B_{n+2} = \sin(B_{n+1}), n \geq -1$, is not a linear recurrence relation because $\sin(B_{n+1})$ is not a linear function of the dependent function $B_{n+1}$.

6. $a_{n+1} = na_n, n \geq 0$, is not a constant coefficient recurrence relation, though it is linear and homogeneous.

7. $f(n + 1) = 3f(n - 5) + 4f(n - 2), n \geq 5$, is a homogeneous, 6th order, linear, constant coefficient recurrence relation. Observe that the difference of the highest index ($n + 1$) and the lowest index ($n - 5$) is exactly the order 6 of the recurrence relation.

8. $a_{n+1} = ra_n$ with constant $r$ and $n \geq 0$ induces a geometric sequence $\{a_n\}_{n \geq 0}$ with

$$a_n = ra_{n-1} = r(ra_{n-2}) = r^2a_{n-2} = \cdots = r^na_0,$$

i.e., $a_n = r^na_0$ as its solution.

The characteristic equation of an order $m$, linear, constant coefficient recurrence relation

$$c_ma_{n+m} + c_{m-1}a_{n+m-1} + \cdots + c_1a_{n+1} + c_0a_n = g(n), \quad n \geq 0$$

with $c_mc_0 \neq 0$ is the following polynomial equation

$$c_ml^m + c_{m-1}l^{m-1} + \cdots + c_1l + c_0 = 0,$$

where $\lambda$ (pronounced as lambda), is just an unknown variable. Notice that homogeneity does not play any role when building characteristic equation.

Examples

9. Recurrence relation $a_{n+2} + 3a_{n+1} + 2a_n = 0, n \geq 0$, has the characteristic equation

$$\lambda^2 + 3\lambda + 2 = 0.$$

10. Recurrence relation $f(n + 1) = 3f(n - 2) + n^2 + 5, n \geq 2$ has the characteristic equation

$$\lambda^3 - 3\lambda = 0$$

because the recurrence relation can be written via $n = k + 2$ as

$$f(k + 3) - 3f(k) = (k + 2)^3 + 5.$$  In fact we have in this case $c_m \equiv c_3 = 1$, $c_2 = 0$, $c_1 = 0$ and $c_0 = -3$ in (***), and thus the characteristic equation $c_3\lambda^3 + c_2\lambda^2 + c_1\lambda + c_0 = 0$ becomes simply $\lambda^3 - 3\lambda = 0$.

Note. An alternative way to construct the characteristic equation: Since the highest index in the recurrence relation is $n + 1$ while the lowest index is $n - 2$, their difference $(n + 1) - (n - 2) = 3$ must be the order $m$, i.e., $m = 3$. Likewise the characteristic equation can also be obtained in following way.
(i) Remove the non-homogeneous terms \( g(n) \). This way the recurrence relation \( f(n + 1) = 3f(n - 2) + n^2 + 5 \) becomes the reduced recurrence relation 
\[
f(n + 1) = 3f(n - 2).
\]

(ii) Find the lowest index \( L \). Here we thus have \( L = n - 2 \).

(iii) For each term in the reduced recurrence relation, if its index is \( K \) then replace the term by \( \lambda^{K-L} \). For the term \( f(n + 1) \) we see \( K = n + 1 \) hence 
\[
K - L = (n + 1) - (n - 2) = 3.
\]
Hence \( f(n + 1) \) is to be replaced by \( \lambda^3 \). Likewise for the term \( f(n - 2) \) on the r.h.s. we see \( K = n - 2 \) hence \( K - L = 0 \), implying that the term \( f(n - 2) \) is to be replaced by \( \lambda^0 \) which is simply 1. This way the reduced recurrence relation is finally changed into the characteristic equation \( \lambda^3 = 3 \).

### 21.3 Simplest Case of General Solutions

A solution \( a_n \) of a recurrence relation (\(*\ *)\) is said to be a general solution, typically containing some arbitrary constants in the solution expression for \( a_n \), if any particular solution of the recurrence relation (\(*\ *)\) can be obtained as a special case of the general solution.

For example, it is easy to verify that \( a_n = A2^n + B3^n \) for arbitrary constants \( A \) and \( B \) solves the recurrence relation \( a_{n+2} - 5a_{n+1} + 6a_n = 0 \). Likewise we can show that \( a_n = 5 \times 2^n \) is also a (particular) solution. Obviously the particular solution \( a_n = 5 \times 2^n \) is included in the more general solution expression \( a_n = A2^n + B3^n \) if we choose \( A = 5 \) and \( B = 0 \). In fact one can show that all the solutions of \( a_{n+2} - 5a_{n+1} + 6a_n = 0 \) are embraced by the solution expression \( a_n = A2^n + B3^n \), which is hence the general solution.

An alternative way to determine if a solution is the general solution of an \( m \)-th order linear, constant coefficient recurrence relation is to see if the solution expression contains exactly \( m \) independent arbitrary constants. The word independent here roughly means that none of the arbitrary constants can be made redundant.

From this perspective, we can also conclude that \( a_n = A2^n + B3^n \) is the general solution of the second order recurrence relation \( a_{n+2} - 5a_{n+1} + 6a_n = 0 \) because the solution contains \( A \) and \( B \) as the two independent arbitrary constants.

To better prepare ourselves for the rest of the topics, we recall that a polynomial \( f(\lambda) \) has \( \lambda_0 \) as one of its roots means, precisely, \( f(\lambda_0) = 0 \). For example, if \( f(\lambda) = \lambda^2 - 5\lambda + 6 \), then \( \lambda_0 = 3 \) is one of its roots because \( f(\lambda_0) = f(3) = 3^2 - 5 \times 3 + 6 = 0 \). In other words, 3 is a root of the equation \( \lambda^2 - 5\lambda + 6 = 0 \). In general, a polynomial equation of order \( n \) will have exactly \( n \) roots, some of which may be distinct while others may be repeated, some of which may be real while others may be complex numbers. Recall that a second order polynomial equation

\[
a\lambda^2 + b\lambda + c = 0, \quad a \neq 0
\]
has two roots, \( \lambda_1 \) and \( \lambda_2 \), given by
\[
\lambda_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \quad \lambda_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a},
\]
see also Preliminary Mathematics at the beginning of these notes. If \( \lambda_1 = \lambda_2 \), then the two roots are the repeated roots and \( \lambda_1 \), which is the same as \( \lambda_2 \), is a root of multiplicity 2. If \( \sqrt{b^2 - 4ac} \geq 0 \) the two roots are real numbers else if \( \sqrt{b^2 - 4ac} < 0 \) the two roots are complex numbers.

For a polynomial equation \( f(\lambda) = 0 \), a value \( \lambda_0 \) is said to be a root of multiplicity \( m \) if \( f(\lambda) \) can be written as \( f(\lambda) = (\lambda - \lambda_0)^m g(\lambda) \) such that \( g(\lambda) \) is again a polynomial with \( g(\lambda_0) \neq 0 \). For example, the polynomial equation \( \lambda^2 - 6\lambda + 9 = 0 \) has a root 3 of multiplicity 2. This is because \( \lambda^2 - 6\lambda + 9 = (\lambda - 3)^2 \times 1 \), in which \( f(\lambda) = \lambda^2 - 6\lambda + 9 \), \( g(\lambda) = 1 \), \( \lambda_0 = 3 \) and \( m = 2 \). If we solve the equation \( \lambda^2 - 6\lambda + 9 = 0 \) through the use of the above root formula for \( \lambda_1 \) and \( \lambda_2 \), we see that both \( \lambda_1 \) and \( \lambda_2 \) are equal to the same value 3. This also explains why 3 is a root of multiplicity 2 for the equation \( \lambda^2 - 6\lambda + 9 = 0 \).

**Note.** A polynomial equation \( f(\lambda) = 0 \) has a root \( \lambda_0 \), i.e., \( f(\lambda_0) = 0 \), if and only if \( f(\lambda) = (\lambda - \lambda_0) \cdot g(\lambda) \) for another nonzero polynomial \( g(\lambda) \). If \( \lambda_0 \) is furthermore a root of multiplicity \( m > 1 \) of \( f(\lambda) = 0 \), then \( \lambda_0 \) must be a root of multiplicity \( m - 1 \) of \( g(\lambda) = 0 \). A root of multiplicity 1 is called a simple root. A simple root is thus not a repeated root.

**Theorem 24.** If the characteristic equation of an \( m \)-th order homogeneous, linear, constant coefficient recurrence relation \( c_ma_{n+m} + c_{m-1}a_{n+m-1} + \cdots + c_1a_{n+1} + c_0a_n = 0 \), \( c_m \neq 0 \), \( n \geq 0 \) has \( m \) distinct roots \( \lambda_1, \lambda_2, \ldots, \lambda_m \), then
\[
a_n = A_1\lambda_1^n + A_2\lambda_2^n + \cdots + A_m\lambda_m^n
\]
with arbitrary constants \( A_1, \ldots, A_m \) is the general solution of the recurrence relation.

**Proof.** First we show \( a_n = \sum_{i=1}^{m} A_i\lambda_i^n \) is a solution. For this purpose we substitute the expression for \( a_n \) into the recurrence relation and obtain
\[
c_ma_{n+m} + \cdots + c_1a_{n+1} + c_0a_n = c_m(A_1\lambda_1^{n+m} + \cdots + A_m\lambda_m^{n+m}) + \cdots + c_1(A_1\lambda_1^{n+1} + \cdots + A_m\lambda_m^{n+1})
\]
\[
+ c_0(A_1\lambda_1^n + \cdots + A_m\lambda_m^n)
\]
\[
= A_1(c_m\lambda_1^{n+m} + \cdots + c_1\lambda_1^{n+1} + c_0\lambda_1^n) + A_2(c_m\lambda_2^{n+m} + \cdots + c_1\lambda_2^{n+1} + c_0\lambda_2^n)
\]
\[
+ \cdots + A_m(c_m\lambda_m^{n+m} + \cdots + c_1\lambda_m^{n+1} + c_0\lambda_m^n)
\]
\[
= \sum_{i=1}^{m} A_i\lambda_i^n (c_m\lambda_i^m + \cdots + c_1\lambda_i + c_0) = 0.
\]

Hence \( a_n = \sum_{i=1}^{m} A_i\lambda_i^n \) is a solution. Since the solution involves \( m \) arbitrary constants \( A_1, \ldots, A_m \), it is in fact a general solution. Alternatively we can also argue that for any initial values of
21.3 Simplest Case of General Solutions

If we multiply the first relation by $a_k$, the expression $a_n = \sum_{i=1}^{m} A_i \lambda_i^k$ is indeed a general solution.

**Note.** If the roots $\lambda_1, \cdots, \lambda_m$ are not distinct, i.e., there are repeated roots, then $a_n = \sum_{i=1}^{m} A_i \lambda_i^n$ is still a solution but is not a general solution. The general solution for such cases will be dealt with in the next section.

**Corollary 25.** Given two solutions $\{x_n\}$ and $\{y_n\}$ of an $m$-th order homogeneous, linear, constant coefficient recurrence relation, any linear combination of them, $z_n = Ax_n + By_n$, where $A, B$ are constants, is also a solution of the same recurrence relation.

**Proof.** Given $c_m a_{m+n} + c_{m-1} a_{n+m-1} + \cdots + c_1 a_{n+1} + c_0 a_n = 0$, $c_m c_0 \neq 0$, $n \geq 0$, if $\{x_n\}$ and $\{y_n\}$ are solutions, then

\[
\begin{align*}
&c_m x_{n+m} + c_{m-1} x_{n+m-1} + \cdots + c_1 x_{n+1} + c_0 x_n = 0 \quad \mid \cdot A \\
&c_m y_{n+m} + c_{m-1} y_{n+m-1} + \cdots + c_1 y_{n+1} + c_0 y_n = 0 \quad \mid \cdot B
\end{align*}
\]

If we multiply the first relation by $A$, the second by $B$ and then sum them up, we get

\[
c_m [Ax_{n+m} + By_{n+m}] + c_{m-1} [Ax_{n+m-1} + By_{n+m-1}] + \cdots + c_1 [Ax_{n+1} + By_{n+1}] + c_0 [Ax_n + By_n] = 0
\]

that is

\[
c_m z_{n+m} + c_{m-1} z_{n+m-1} + \cdots + c_1 z_{n+1} + c_0 z_n = 0
\]

which shows that $z_n$ is a solution for the same recurrence relation.

**Corollary 26.** Given two solutions $\{x_n\}$ and $\{y_n\}$ of an $m$-th order, non-homogeneous, linear, constant coefficient recurrence relation, then the difference of them is a solution of the homogeneous version of the recurrence relation.

**Proof.**

Given $c_m a_{m+n} + c_{m-1} a_{n+m-1} + \cdots + c_1 a_{n+1} + c_0 a_n = g(n)$, $c_m c_0 \neq 0$, $n \geq 0$, if $\{x_n\}$ and $\{y_n\}$ are solutions, then they satisfy

\[
\begin{align*}
&c_m x_{n+m} + c_{m-1} x_{n+m-1} + \cdots + c_1 x_{n+1} + c_0 x_n = g(n) \\
&c_m y_{n+m} + c_{m-1} y_{n+m-1} + \cdots + c_1 y_{n+1} + c_0 y_n = g(n)
\end{align*}
\]

and their difference gives

\[
c_m [x_{n+m} - y_{n+m}] + c_{m-1} [x_{n+m-1} - y_{n+m-1}] + \cdots + c_1 [x_{n+1} - y_{n+1}] + c_0 [x_n - y_n] = g(n) - g(n) = 0
\]

\[
c_m z_{n+m} + c_{m-1} z_{n+m-1} + \cdots + c_1 z_{n+1} + c_0 z_n = 0
\]
Example

11. Find the general solution of

\[ a_{n+2} - 5a_{n+1} + 6a_n = 0, \quad n \geq 0. \]

Give also the particular solution satisfying \( a_0 = 0 \) and \( a_1 = 1 \).

Solution. Since the associated characteristic equation is

\[ \lambda^2 - 5\lambda + 6 = 0 \]

and has 2 distinct roots \( \lambda_1 = 2 \) and \( \lambda_2 = 3 \), i.e., \( 2^2 - 5 \times 2 + 6 = 0 \) and \( 3^2 - 5 \times 3 + 6 = 0 \), the general solution for the recurrence relation, according to the theorem earlier on, is \( a_n = A_1 2^n + A_2 3^n, \quad n \geq 0 \), where \( A_1 \) and \( A_2 \) are two arbitrary constants. To find the particular solution, we need to determine \( A_1 \) and \( A_2 \) explicitly using the initial conditions \( a_0 = 0 \) and \( a_1 = 1 \). Hence we require

\[
\begin{align*}
a_0 &= A_1 2^0 + A_2 3^0 = 0, & A_1 + A_2 &= 0, \\
a_1 &= A_1 2^1 + A_2 3^1 = 1. & 2A_1 + 3A_2 &= 1.
\end{align*}
\]

If we convert the first equation \( A_1 + A_2 = 0 \) into \( A_2 = -A_1 \) and substitute it into the second equation \( 2A_1 + 3A_2 = 1 \), we get \( 2A_1 + 3(-A_1) = 1 \) and hence \( A_1 = -1 \). Substitute \( A_1 = -1 \) back to the first equation \( A_1 + A_2 = 0 \), we get \(-1 + A_2 = 0 \) which gives \( A_2 = 1 \). Thus the particular solution is \( a_n = -2^n + 3^n \) for \( n \geq 0 \). For more examples on how to solve a set of linear equations, see Preliminary Mathematics at the beginning of these notes.

Note. It is often a good practice to check your answers, even if just partially. To check the above particular solution, we see

- \( a_0 = -2^0 + 3^0 = 0 \)
- \( a_1 = -2^1 + 3^1 = 1 \)
- \[ a_{n+2} - 5a_{n+1} + 6a_n = (-2^{n+2} + 3^{n+2}) - 5(-2^{n+1} + 3^{n+1}) + 6(-2^n + 3^n) \]
  \[ = -2^n (2^2 - 5 \times 2 + 6) + 3^n (3^2 - 5 \times 3 + 6) = 0. \]

i.e., all conditions are satisfied.

Exercises

1. Find all the roots of \((\lambda^2 + 9\lambda + 14)(\lambda^2 + 7\lambda + 10) = 0\).

2. Suppose a sequence \( \{a_n\}_{n \geq 0} \) is determined by the recurrence relation \( 3a_n a_{n+1} + a_{n+2} = 1 \) and the initial conditions \( a_0 = 1 \) and \( a_1 = 2 \). Find \( a_3 \) both iteratively and recursively.
3. Is the recurrence relation $a_{n+2} + na_{n+1} + 5a_n = 0$ linear and homogeneous? Is it a constant coefficient recurrence relation?

4. Find the general solution of the recurrence relation $a_{n+1} - a_{n-1} = 0$. 
22 Solution of Linear Homogeneous Recurrence Relations

22.1 General Solutions for Homogeneous Problems

If the characteristic equation associated with a given $m$-th order linear, constant coefficient, homogeneous recurrence relation has some repeated roots, then the solution given by $\sum A_i \lambda^n_i$ will not have $m$ arbitrary constants. To see this, we assume for instance $\lambda_1 = \lambda_2$, i.e., root $\lambda_1$ is repeated. Then the solution $a_n = \sum_{i=1}^m A_i \lambda^n_i = (A_1 + A_2) \lambda^n_1 + A_3 \lambda^n_2 + \cdots + A_m \lambda^n_m$ has less than $m$ arbitrary constants because $A_1 + A_2$ comprises essentially only one arbitrary constant. To make up for the missing ones, we introduce the following more general theorem.

**Theorem 27.** Suppose the characteristic equation of the linear, constant coefficient recurrence relation

$$c_m a_{n+m} + c_{m-1} a_{n+m-1} + \cdots + c_1 a_{n+1} + c_0 a_n = 0, \quad c_m c_0 \neq 0, \ n \geq 0$$

has the following roots (all roots accounted for)

$$\lambda_1, \cdots, \lambda_{m_1}, \lambda_2, \cdots, \lambda_{m_2}, \cdots, \lambda_k, \cdots, \lambda_k$$

such that $\lambda_1, \cdots, \lambda_k$ are distinct, $m_1, \cdots, m_k \geq 1$ and $m_1 + \cdots + m_k = m$. Then the general solution of the recurrence relation is

$$a_n = \sum_{i=1}^k \left( \sum_{j=0}^{m_i-1} A_{i,j} n^j \right) \lambda^n_i,$$

i.e.,

$$a_n = \left( A_{1,0} + A_{1,1} n + \cdots + A_{1,m_1-1} n^{m_1-1} \right) \lambda^n_1 + \left( A_{2,0} + A_{2,1} n + \cdots + A_{2,m_2-1} n^{m_2-1} \right) \lambda^n_2 + \cdots + \left( A_{k,0} + A_{k,1} n + \cdots + A_{k,m_k-1} n^{m_k-1} \right) \lambda^n_k$$

where $A_{i,j}$ for $i = 1, \cdots, k$ and $j = 0, \cdots, m_i - 1$ are $m$ arbitrary constants.

Each root is multiplied by a polynomial of degree one less the root’s multiplicity. So, if the characteristic equation has a root of, say, $\lambda = 3$ that occurs twice (called having multiplicity 2) then in the solution $3^n$ is multiplied by a polynomial of degree one less that the multiplicity. That is of degree one, a linear polynomial: $(An + B)3^n$.

If the root was $\lambda = 4$ of multiplicity 3 then $4^n$ is multiplied by a polynomial of degree 2: $(An^2 + Bn + C)4^n$.
Examples

1. Find a particular solution of

\[ f(n + 2) + 4f(n + 1) + 4f(n) = 0, \quad n \geq 0 \]

with initial conditions \( f(0) = 1 \) and \( f(1) = 2 \).

**Solution.** For clarity, we artificially split the solution procedure into three steps below.

(a) The associated characteristic equation is

\[ \lambda^2 + 4\lambda + 4 \equiv (\lambda + 2)^2 = 0 \]

which has a repeated root \( \lambda_1 = -2 \). In other words, all of its roots counting the multiplicity are \( \lambda_1, \lambda_1, \) i.e., \( m_1 = 2, k = 1 \).

(b) The general solution from the theorem in this lecture is thus

\[ f(n) = (A_{1,0} + A_{1,1}n)\lambda_1^n = (B_0 + B_1n)(-2)^n, \]

where \( B_0 \) and \( B_1 \), denoting \( A_{1,0} \) and \( A_{1,1} \) respectively, are arbitrary constants.

(c) Constants \( B_0 \) and \( B_1 \) are to be determined from the initial conditions

\[
\begin{align*}
    f(0) &= (B_0 + B_1 \times 0)(-2)^0 = 1 \quad \Leftrightarrow \quad B_0 = 1 \\
    f(1) &= (B_0 + B_1 \times 1)(-2)^1 = 2 \quad \Rightarrow \quad -2(B_0 + B_1) = 2.
\end{align*}
\]

They are thus \( B_0 = 1 \) and \( B_1 = -2 \). Hence the requested particular solution is

\[ f(n) = (1 - 2n)(-2)^n, \quad n \geq 0. \]

2. Find the general solution of

\[ a_{n+3} - 3a_{n+2} + 4a_n = 0, \quad n \geq 0. \]

**Solution.** The associated characteristic equation is \( F(\lambda) \overset{\text{def}}{=} \lambda^3 - 3\lambda^2 + 4 = 0 \). Although there are formulae to give explicit roots for a third order polynomial in terms of its coefficients, we are taking a shortcut here by guessing one root \( \lambda_1 = -1 \). We can check that \( \lambda_1 = -1 \) is indeed a root by verifying \( F(-1) = (-1)^3 - 3(-1)^2 + 4 = 0 \). To find the remaining roots, we first factorise \( F(\lambda) \) by performing a long division.
which implies \( \lambda^3 - 3\lambda^2 + 4 = (\lambda + 1)(\lambda^2 - 4\lambda + 4) = (\lambda + 1)(\lambda - 2)^2 \).

Therefore all the roots, counting the corresponding multiplicity, are \(-1, 2, 2\), i.e., \(\lambda_1 = -1, m_1 = 1\) and \(\lambda_2 = 2, m_2 = 2\).

Hence the general solution reads (with \(A = A_{1,0}, B = A_{2,0}, C = A_{2,1}\))

\[
a_n = A(-1)^n + (B + Cn)2^n, \quad n \geq 0.
\]

3. Find the particular solution of

\[
u_{n+3} + 3u_{n+2} + 3u_{n+1} + u_n = 0, \quad n \geq 0
\]

satisfying the initial conditions \(u_0 = 1, u_1 = 1\) and \(u_2 = -7\).

**Solution.**

(a) The associated characteristic equation \(\lambda^3 + 3\lambda^2 + 3\lambda + 1 \equiv (\lambda + 1)^3 = 0\) has roots \(\lambda_1 = -1\) with multiplicity 3, i.e., \(m_1 = 3\).

(b) The general solution then reads for arbitrary constants \(A, B\) and \(C\)

\[
u_n = (A + Bn + Cn^2)(-1)^n, \quad n \geq 0.
\]

(c) To determine \(A, B\) and \(C\) through the use of the initial conditions, we set \(n\) in the
solution expression in (b) to 0, 1 and 2 respectively, and then observe

\[ u_0 \text{ gives } A = 1 \]
\[ u_1 \text{ gives } (A + B + C)(-1) = 1 \]
\[ u_2 \text{ gives } A + 2B + 4C = -7. \]

The solution of these 3 equations,

\[ A = 1, \quad B = 0, \quad C = -2. \]

finally produces the required particular solution

\[ u_n = (1 - 2n^2)(-1)^n, \quad n \geq 0. \]

### 22.2 Ideas Behind the Theorem

We now briefly show that the general solution given in the theorem indeed satisfies the recurrence relation. Let us first define a step operation \( \Delta \) by \( \Delta F(n) = F(n + 1) \) for any function \( F \). Let

\[ f(\lambda) \overset{\text{def}}{=} c_m\lambda^m + \cdots + c_1\lambda + c_0 \]

and for any \( s \in \mathbb{N} \)

\[ P_s(\lambda) = \{(b_0 + b_1n + \cdots + b_sn^s)\lambda^n \mid b_i \in \mathbb{C}, \ i = 0, 1, \ldots, s\}. \]

Since the binomial expansion gives

\[ (\Delta - \lambda)(n^i\lambda^n) = (n + 1)^i\lambda^{n+1} - \lambda n^i\lambda^n \]
\[ = \lambda^{n+1}\left[\binom{i}{1}n^{i-1} + \binom{i}{2}n^{i-2} + \cdots + \left(\frac{i}{i-1}\right)n + \left(\frac{i}{i}\right)\right], \]

then we have

\[ (\Delta - \lambda)P_s(\lambda) \subseteq P_{s-1}(\lambda), \]
\[ (\Delta - \lambda)^2P_s(\lambda) \subseteq (\Delta - \lambda)P_{s-1} \subseteq P_{s-2}, \]
\[ \vdots \]
\[ (\Delta - \lambda)^sP_s \subseteq P_0, \]
\[ (\Delta - \lambda)^{s+1}P_s \subseteq \{0\}. \]

Because \( \lambda_i \)'s are roots of \( f(\lambda) \) with multiplicity \( m_i \), for any such given \( i \) there exists a polynomial \( g(\lambda) \) such that \( f(\lambda) = (\lambda - \lambda_i)^m g(\lambda) \) and \( g(\lambda_i) \neq 0, g(\lambda_j) = 0, \forall j \neq i \). Hence for any \( i \) with
1 \leq i \leq k$, we have

\[
\begin{align*}
 f(\Delta) \left( \sum_{j=0}^{m_i-1} A_{ij} n_j^i \right) \lambda_i^n & \in f(\Delta) P_{m_i-1}(\lambda_i) \\
 & = g(\Delta)(\Delta - \lambda_i)^m P_{m_i-1}(\lambda_i) \\
 & \subseteq g(\Delta)[0] \subseteq \{0\}.
\end{align*}
\]

Thus

\[
\sum_{i=1}^{k} \left( \sum_{j=0}^{m_i-1} A_{ij} n_j^i \right) \lambda_i^n = f(\Delta) \left( \sum_{j=0}^{m_i-1} A_{ij} n_j^i \right) \lambda_i^n + \sum_{s=1}^{k} f(\Delta) \left( \sum_{j=0}^{m_s-1} A_{sj} n_j^s \right) \lambda_s^n = 0,
\]

i.e., \( a_n = \sum_{i=1}^{k} \left( \sum_{j=0}^{m_i-1} A_{ij} n_j^i \right) \lambda_i^n \) is a solution of the recurrence relation \( f(\Delta) a_n = 0 \).

It is easy to see that this solution expression for \( a_n \) has exactly \( m \) arbitrary constants \( A_{i,j} \).

\[ \square \]

**Exercises**

1. Find all the roots, and their corresponding multiplicity, of the equation

\[
\lambda^3 - 4\lambda^2 + 5\lambda - 2 = 0,
\]

assuming we already know that \( \lambda = 1 \) is one of the roots.

2. Find the general solution of the recurrence relation \( a_{n+2} + 14a_{n+1} + 49a_n = 0 \).

3. Solve the recurrence relation \( a_{n+1} - 6a_n + 9a_{n-1} = 0 \) for \( n \geq 1 \) with the initial conditions \( a_0 = 5 \) and \( a_1 = -6 \).
23 Basics of Linear Nonhomogeneous Recurrence Relations

23.1 Connection between Homogeneous and Nonhomogeneous Problems

The solutions of linear nonhomogeneous recurrence relations are closely related to those of the corresponding homogeneous equations. First of all, remember Corollary 3, Section 21:

If $v_n$ and $w_n$ are two solutions of the nonhomogeneous equation (*) then $\varphi_n = w_n - v_n$, $n \geq 0$ is a solution of the homogeneous equation (**).

**Theorem 28.** Consider the following linear constant coefficient recurrence relation

$$c_m a_{n+m} + \cdots + c_1 a_{n+1} + c_0 a_n = g(n), \quad c_0 c_m \neq 0, \quad n \geq 0$$

and its corresponding homogeneous form

$$c_m a_{n+m} + \cdots + c_1 a_{n+1} + c_0 a_n = 0.$$ (**)  

If $u_n$ is the general solution of the homogeneous equation (**), and $v_n$ is any particular solution of the nonhomogeneous equation (*), then

$$a_n = u_n + v_n, \quad n \geq 0$$

is the general solution of the nonhomogeneous equation (*).

**Proof.** For $a_n = u_n + v_n$, we have

$$c_m a_{n+m} + \cdots + c_1 a_{n+1} + c_0 a_n = \sum_{i=0}^{m} c_i a_{n+i} = \sum_{i=0}^{m} c_i (u_{n+i} + v_{n+i})$$

$$= \sum_{i=0}^{m} c_i u_{n+i} + \sum_{i=0}^{m} c_i v_{n+i} = g(n),$$

i.e., $a_n$ satisfies the non-homogeneous recurrence relation (*). Since the general solution $u_n$ of the homogeneous problem has $m$ arbitrary constants thus so is $a_n = u_n + v_n$. Hence $a_n$ is the general solution of (*). More precisely, for any solution $w_n$ of (*), since $\varphi_n = w_n - v_n$ satisfies (**), $\varphi_n$ will just be a special case of the general solution $u_n$ of (**). Hence $w_n = \varphi_n + v_n$ is included in the solution $a_n = u_n + v_n$. Therefore $a_n = u_n + v_n$ is the general solution of the nonhomogeneous problem (*).

23.2 Some Simple Examples

In the following, we’ll give a few simple examples for finding particular solutions for non-homogeneous problems. The arguments will be intuitive and some underneath subtleties are
thus deliberately hidden at this stage to keep the main idea simple. Therefore the readers are urged not to make simplistic conclusion or generalisation based on the possible incomplete arguments in these examples, until you have read thoroughly the rest of the subject in the next section.

Examples

1. Find a particular solution of \( a_{n+2} - 5a_n = 2 \times 3^n \) for \( n \geq 0 \).

**Solution.** As the r.h.s. is \( 2 \times 3^n \), we try the special solution in the form of \( a_n = C3^n \), with the constant \( C \) to be determined. The substitution of \( a_n = C3^n \) into the recurrence relation thus gives

\[
a_{n+2} - 5a_n = C \cdot 3^{n+2} - 5 \cdot C \cdot 3^n = 3^2 \cdot C \cdot 3^n - 5 \cdot C \cdot 3^n = 2 \times 3^n,
\]

i.e., \( 4C = 2 \) or \( C = \frac{1}{2} \). Hence \( a_n = \frac{1}{2} \times 3^n \), for \( n \geq 0 \) is a particular solution.

2. Find a particular solution of \( f_{n+1} - 2f_n + 3f_{n-4} = 6n, n \geq 4 \).

**Solution.** As the r.h.s. is \( 6n \), we try the similar form

\[
f_n = An + B,
\]

with constants \( A \) and \( B \) to be determined. Hence \( f_n \) be a solution requires

\[
6n = f_{n+1} - 2f_n + 3f_{n-4}
= (A(n + 1) + B) - 2(An + B) + 3(A(n - 4) + B)
= 2An + (2B - 11A)
\]

i.e.,

\[
2A = 6 \quad \Leftrightarrow \quad A = 3
\]

\[
2B - 11A = 0 \quad \Leftrightarrow \quad B = \frac{33}{2}
\]

Therefore our particular solution is \( f_n = 3n + \frac{33}{2} \).

3. Find the particular solution of \( a_{n+3} - 7a_{n+2} + 16a_{n+1} - 12a_n = 4^n n \) with \( a_0 = -2, \quad a_1 = 0, \quad a_2 = 5 \).

**Solution.** We first find the general solution \( u_n \) for the corresponding homogeneous problem. Then we look for a particular solution \( v_n \) for the nonhomogeneous problem without concerning ourselves with the initial conditions. Once these two are done, we obtain the general solution \( a_n = u_n + v_n \) for the nonhomogeneous recurrence relation, and we just need to use the initial conditions to determine the arbitrary constants in the general solution \( a_n \) so as to derive the final particular solution.
(a) The associated characteristic equation \( \lambda^3 - 7\lambda^2 + 16\lambda - 12 = 0 \) can be shown to admit the following roots \( \lambda_1 = 3 \), \( m_1 = 1 \), (simple root), \( \lambda_2 = 2 \), \( m_2 = 2 \), (double root).

We can easily verify this by first guessing \( \lambda_1 = 3 \) is a root and then factorising \( \lambda^3 - 7\lambda^2 + 16\lambda - 12 \) into \( (\lambda - 3)(\lambda^2 - 4\lambda + 4) \) to find the remaining roots. If you are concerned that you might not be able to guess a first root like \( \lambda_1 \) in similar circumstances, then you can leave your worry behind because we won’t expect you to make such a guess (see next section).

The general solutions for the corresponding homogeneous problem thus reads

\[
\begin{align*}
u_n &= A3^n + (B + Cn)2^n, \quad n \geq 0.
\end{align*}
\]

That is, \( u_n \) solves \( a_{n+3} - 7a_{n+2} + 16a_{n+1} - 12a_n = 0 \).

(b) Since the r.h.s. of the nonhomogeneous recurrence relation is \( 4^n \cdot n \), which fits into the description of \( 4^n \times (\text{first order polynomial in } n) \), we'll try a particular solution in a similar form, i.e., \( v_n = 4^n(Dn + E) \).

The substitution of \( v_n \) into the original recurrence relation then gives

\[
\begin{align*}4^n \cdot n &= v_{n+3} - 7v_{n+2} + 16v_{n+1} - 12v_n \\
&= 4^{n+3}(D(n + 3) + E) - 7 \times 4^{n+2}(D(n + 2) + E) \\
&\quad + 16 \times 4^{n+1}(D(n + 1) + E) - 12 \times 4^n(Dn + E),
\end{align*}
\]
i.e.,

\[
\begin{align*}n &= 64(Dn + 3D + E) - 112(Dn + 2D + E) + 64(Dn + D + E) - 12(Dn + E) \\
&= 4Dn + 4E + 32D.
\end{align*}
\]

Hence we have

\[
4D = 1, \quad 4E + 32D = 0 \quad \Leftrightarrow \quad D = \frac{1}{4}, \quad E = -2
\]

and consequently

\[
v_n = 4^n \left( \frac{n}{4} - 2 \right).
\]

(c) The general solution for the nonhomogeneous problem is then given by \( a_n = u_n + v_n \), i.e.,

\[
a_n = 4^n \left( \frac{n}{4} - 2 \right) + A3^n + (B + Cn)2^n, \quad n \geq 0.
\]

(d) We now determine \( A, B, C \) by the initial conditions and the use of the solution expression in (c)
23.2 Some Simple Examples

<table>
<thead>
<tr>
<th>Initial Conditions</th>
<th>Induced Equations</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_0 = -2$</td>
<td>$A + B - 2 = -2$</td>
<td>$A = 1$</td>
</tr>
<tr>
<td>$a_1 = 0$</td>
<td>$3A + 2B + 2C - 7 = 0$</td>
<td>$B = -1$</td>
</tr>
<tr>
<td>$a_2 = 5$</td>
<td>$9A + 4B + 8C = 29$</td>
<td>$C = 3$</td>
</tr>
</tbody>
</table>

Finally, the particular solution satisfying both the nonhomogeneous recurrence relations and the initial conditions is given by

$$a_n = 4^n \left( \frac{n}{4} - 2 \right) + 3^n + (3n - 1)2^n, \quad n \geq 0.$$  

Note.

1. In all the examples in this lecture, it is easy to verify that the $g(n)$ function in (*) is in the form of
   $$g(n) = \mu^n (\alpha_1 n^k + \cdots + \alpha_1 n + \alpha_0),$$
   where $\mu$ is not a root of the associated characteristic equation. If this were not the case, we would have to use different forms to try for the particular solutions. These will be the topics of the next lecture.

2. If $g(n) = \mu_1^n n + \mu_2^n (3n^2 + 1)$, for instance, with $\mu_1$ and $\mu_2$ neither being a root of the characteristic equation, then the particular solution should be tried in the form
   $$v_n = \mu_1^n (A_1 n + A_0) + \mu_2^n (B_2 n^2 + B_1 n + B_0).$$

3. If $g(n) = \cos(\alpha n) \cdot n$, for another instance, then we can treat it as
   $$g(n) = \frac{\cos(n) + \cos(n)}{2} n = \frac{n}{2} \times \mu_1^n + \frac{n}{2} \times \mu_2^n$$
in which $\mu_1 = e^{i\alpha}$ and $\mu_2 = e^{-i\alpha}$. Alternatively, we could try the particular solution in the form
   $$v_n = \sin(\alpha n)(A_1 n + A_0) + \cos(\alpha n)(B_1 n + B_0).$$

Exercises

1. Suppose both $u_n$ and $v_n$ are some solutions of the recurrence relation $a_{n+15} - 2a_{n+1} + a_n = 1$, i.e., both $u_{n+15} - 2u_{n+1} + u_n = 1$ and $v_{n+15} - 2v_{n+1} + v_n = 1$ are satisfied. Is $u_n + v_n$ also a solution of the same nonhomogeneous recurrence relation?

2. Find the general solution of the recurrence relation $a_{n+2} - 9a_{n+1} + 14a_n = 6n$.

3. Find the particular solution of the recurrence relation $a_{n+2} - 9a_{n+1} + 14a_n = 6n$, satisfying the initial conditions $a_0 = \frac{1}{6}$ and $a_1 = \frac{31}{6}$.  

186
24 Solution of Linear Nonhomogeneous Recurrence Relations

24.1 A Case for Thought

We already mentioned that finding a particular solution for a nonhomogeneous problem can be more involved than those exemplified in the previous lecture. Let us first highlight our point with the following example.

Example

1. Solve \( a_{n+2} + a_{n+1} - 6a_n = 2^n \) for \( n \geq 0 \).

Solution. First we observe that the homogeneous problem

\[
u_{n+2} + u_{n+1} - 6u_n = 0
\]

has the general solution \( u_n = A2^n + B(-3)^n \) for \( n \geq 0 \) because the associated characteristic equation \( \lambda^2 + \lambda - 6 = 0 \) has 2 distinct roots \( \lambda_1 = 2 \) and \( \lambda_2 = -3 \). Since the r.h.s. of the nonhomogeneous recurrence relation is \( 2^n \), if we formally follow the strategy in the previous lecture, we would try \( v_n = C2^n \) for a particular solution. But there is a difficulty: \( C2^n \) fits into the format of \( u_n \) which is a solution of the homogeneous problem. In other words, it can’t be a particular solution of the nonhomogeneous problem. This is really because 2 happens to be one of the two roots \( \lambda_1 \) and \( \lambda_2 \). However, we suspect that a particular solution would still have to have \( 2^n \) as a factor, so we try \( v_n = Cn2^n \). Substituting it to \( v_{n+2} + v_{n+1} - 6v_n = 2^n \), we obtain

\[
C(n + 2)2^{n+2} + C(n + 1)2^{n+1} - 6Cn2^n = 2^n,
\]

i.e., \( 10C2^n = 2^n \) or \( C = \frac{1}{10} \). Hence a particular solution is \( v_n = \frac{n}{10}2^n \) and the general solution of our nonhomogeneous recurrence relation is

\[
a_n = A2^n + B(-3)^n + \frac{n}{10}2^n, \quad n \geq 0.
\]

In general, it is important that a correct form, often termed ansatz in physics, for a particular solution is used before we fix up the unknown constants in the solution ansatz. The following theorem can help.

Theorem 29. (The Rational Roots Test). Let \( P(x) = a_nx^n + a_{n-1}x^{n-1} + \cdots + a_1x + a_0 \) be a polynomial with real coefficients and \( a_n \neq 0 \). If \( P(x) \) has rational roots, they are of the form \( \pm \frac{p}{q} \) where \( p \mid a_0 \) and \( q \mid a_n \).

The choice of the form of a particular solution, covering the cases in this current lecture as well as the previous one, can be summarized below.
24.2 Method of Undetermined Coefficients

Consider a linear, constant coefficient recurrence relation of the form

\[ c_m a_{n+m} + \cdots + c_1 a_{n+1} + c_0 a_n = g(n), \quad c_0 c_m \neq 0, \quad n \geq 0. \]  

(\*)

Suppose function \( g(n) \), the nonhomogeneous part of the recurrence relation, is of the following form

\[ g(n) = \mu^n(b_0 + b_1 n + \cdots + b_k n^k), \]

(\**)

where \( k \in \mathbb{N}, \mu, b_0, \ldots, b_k \) are constants, and \( \mu \) is a root of multiplicity \( M \) of the associated characteristic equation

\[ c_m \lambda^m + \cdots + c_1 \lambda + c_0 = 0. \]

Then a particular solution \( v_n \) of (\*) should be sought in the form

\[ v_n = \left[ \sum_{j=0}^{k} B_j n^j \right] \mu^n = \mu^n \left( B_0 + B_1 n + \cdots + B_k n^k \right) n^M \]

(\***)

where constants \( B_0, \ldots, B_k \) are to be determined from the requirement that \( a_n = v_n \) should satisfy the recurrence relation (\*). Obviously, the \( v_n \) in (\***) is composed of two parts: one is the \( \mu^n (B_0 + B_1 n + \cdots + B_k n^k) \) which is of the same form of \( g(n) \) in (\**). The other is the \( n^M \) which is a necessary adjustment for the case when \( \mu \), appearing in \( g(n) \) in (\**), is also a root (of multiplicity \( M \)) of the characteristic equation of the associated homogeneous recurrence relation.

Notes.

1. If \( \mu \) is not a root of the characteristic equation, then just choose \( M = 0 \) (so \( n^M = 1 \)), implying alternatively that \( \mu \) is a "root" of 0 multiplicity. ▼

2. We can also try \( \tilde{v}_n = \mu^n \left( \sum_{j=0}^{k+M} A_j n^j \right) \). If we rewrite \( \tilde{v}_n \) as \( \mu^n \sum_{j=0}^{k+M} A_j n^j + \mu^n \sum_{j=0}^{M-1} A_j n^j \), then the first part is essentially (\***), while the second part is just a solution of the homogeneous problem. It is however obvious that \( v_n \) in (\***) is simpler than \( \tilde{v}_n \).

3. We briefly hint why \( v_n \) is chosen in the form of (\***). Let \( \Delta \), \( f(\lambda) \) and \( P_k(\lambda) \) be defined in the same way as we did in the derivation hints of the theorem in the lecture just before the previous one, and we’ll also make use of some intermediate results there. Recall that (\*) can be written as \( f(\Delta) a_n = g(n) \) and (\**) implies \( g(n) \in P_k(\mu) \).

(i) If \( f(\mu) \neq 0 \), then \( f(\Delta) P_k(\mu) \subseteq P_k(\mu) \). Hence if we try \( v_n = (B_0 + B_1 n + \cdots + B_k n^k) \mu^n \in P_k(\mu) \), then we can derive a set of exactly \( (k+1) \) linear equations in \( B_0, \ldots, B_k \), which can be used to determine these \( B_i \)'s.
(ii) If \( \mu \) is a root of \( f(\lambda) = 0 \) with multiplicity \( M \geq 1 \), then
\[
f(\Delta)P_{M-1}(\mu) \subseteq \{0\}, \quad f(\Delta)P_{M+\epsilon}(\mu) \subseteq P_k(\mu).
\]
Hence if we try \( v_n = n^M(B_0 + \cdots + B_k n^k)\mu^n \in P_{M+k}(\mu) \), we'll again have a set of exactly \((k+1)\) linear equations as the coefficients of the terms \( \mu^n, \mu^n n, \ldots, \mu^n n^k \). The \((k+1)\) constants \( B_0, \ldots, B_k \) can thus be determined from these linear equations. \( \Box \)

**Examples**

2. Find the general solution of \( f(n+2) - 6f(n+1) + 9f(n) = 5 \times 3^n, n \geq 0 \).

**Solution.** Let \( f(n) = u_n + v_n \), with \( u_n \) being the general solution of the homogeneous problem and \( v_n \) a particular solution.

(a) Find \( u_n \): The associated characteristic equation \( \lambda^2 - 6\lambda + 9 = 0 \) has a repeated root \( \lambda = 3 \) with multiplicity 2. Hence the general solution of the homogeneous problem \( u_{n+2} - 6u_{n+1} + 9u_n = 0, \quad n \geq 0 \) is \( u_n = (A + Bn)3^n \).

(b) Find \( v_n \): Since the r.h.s. of the recurrence relation, the nonhomogeneous part, is \( 5 \times 3^n \) and 3 is a root of multiplicity 2 of the characteristic equation (i.e., \( \mu = 3, k = 0, M = 2 \)), we try due to \((\ast \ast \ast)\) \( v_n = B_0 \mu^n \times n^M \equiv Cn^23^n \): we just need to observe that \( Cn^2 \) is of the form \( 5 \times 3^n \) and that the extra factor \( \frac{5}{18} \) is due to \( \mu = 3 \) being a double root of the characteristic equation. Thus
\[
5 \times 3^n = v_{n+2} - 6v_{n+1} + 9v_n = C(n+2)^23^{n+2} - 6C(n+1)^23^{n+1} + 9Cn^23^n = 18C3^n.
\]
Hence \( C = \frac{5}{18} \) and \( v_n = \frac{5}{18}n^23^n \). Therefore our general solution reads
\[
f(n) = (A + Bn + \frac{5}{18}n^2)3^n, \quad n \geq 0.
\]

3. Find the particular solution of \( a_{n+4} - 5a_{n+3} + 9a_{n+2} - 7a_{n+1} + 2a_n = 3, \quad n \geq 0 \)
satisfying the initial conditions \( a_0 = 2, \quad a_1 = -\frac{1}{2}, \quad a_2 = -5, \quad a_3 = -\frac{31}{2} \).

**Solution.** We first find the general solution \( u_n \) for the homogeneous problem. We then find a particular solution \( v_n \) for the nonhomogeneous problem without considering the initial conditions. Then \( a_n = u_n + v_n \) would be the general solution of the nonhomogeneous problem. We finally make use of the initial conditions to determine the arbitrary constants in the general solution so as to arrive at our required particular solution.

(a) Find \( u_n \): Since the associated characteristic equation \( \lambda^4 - 5\lambda^3 + 9\lambda^2 - 7\lambda + 2 = 0 \) has the sum of all the coefficients being zero, i.e. \( 1 - 5 + 9 - 7 + 2 = 0 \), it must have a root \( \lambda = 1 \). After factorising out \( (\lambda - 1) \) via
\[ \lambda^4 - 5\lambda^3 + 9\lambda^2 - 7\lambda + 2 = (\lambda - 1)(\lambda^3 - 4\lambda^2 + 5\lambda - 2), \]
the rest of the roots will come from \( \lambda^3 - 4\lambda^2 + 5\lambda - 2 = 0 \). Notice that \( \lambda^3 - 4\lambda^2 + 5\lambda - 2 = 0 \) can again be factorised by a factor \( (\lambda - 1) \) because \( 1 - 4 + 5 - 2 = 0 \). This way we can derive in the end that the roots are

\[ \lambda_1 = 1 \text{ with multiplicity } m_1 = 3, \text{ and } \]
\[ \lambda_2 = 2 \text{ with multiplicity } m_2 = 1. \]

Thus the general solutions for the homogeneous problem is

\[ u_n = (A + Bn + Cn^2)1^n + D2^n, \]
or simply \( u_n = A + Bn + Cn^2 + D2^n \) because \( 1^n = 1 \).

(b) Find \( v_n \): Notice that the nonhomogeneous part is a constant 3 which can be written as \( 3 \times 1^n \) when cast into the form of \({**}\), and that 1 is in fact a root of multiplicity 3. In other words, we have in \({**}\) \( \mu = 1, k = 0 \) and \( M = 3 \). Hence we try a particular solution \( v_n = En^3 \cdot 1^n = En^3 \). The substitution of \( v_n \) into the nonhomogeneous recurrence equations then gives, using a formula in the subsection Binomial Expansions in the Preliminary Mathematics at the beginning of these notes,

\[
\begin{align*}
3 &= v_{n+4} - 5v_{n+3} + 9v_{n+2} - 7v_{n+1} + 2v_n \\
&= E(n + 4)^3 - 5E(n + 3)^3 + 9E(n + 2)^3 - 7E(n + 1)^3 + 2En^3 \\
&= E(n^3 + 3n^2 \times 4 + 3n \times 4^2 + 4^3) - 5E(n^3 + 3n^2 \times 3 + 3n \times 3^2 + 3^3) \\
&\quad+ 9E(n^3 + 3n^2 \times 2 + 3n \times 2^2 + 2^3) - 7E(n^3 + 3n^2 \times 1 + 3n \times 1^2 + 1^3) + 2En^3 \\
&= -6E
\end{align*}
\]
i.e., \( E = -\frac{1}{2} \). Hence \( v_n = -\frac{n^3}{2} \).

Note. Should you find it very tedious to perform the expansions in the above, you could just substitute, say, \( n = 0 \) into

\[ 3 = E(n + 4)^3 - 5E(n + 3)^3 + 9E(n + 2)^3 - 7E(n + 1)^3 + 2En^3 \]
to obtain readily \( 3 = EA^3 - 5EA^3 + 9EA^3 - 7E = -6E \). Incidentally you don’t have to substitute \( n = 0 \); you can in fact substitute any value for \( n \) because the equation is valid for all \( n \). Obviously this alternative technique also applies even if there are more than 1 unknowns in the equation; we just need to substitute sufficiently many distinct values for \( n \) to collect enough equations to determine the unknowns. The drawback of this technique is that you have to make sure that the form you have proposed for \( v_n \) is absolutely correct through the use of the proper theory, otherwise an error in the form for \( v_n \) will go undetected in this alternative approach.

(c) The general solution of the nonhomogeneous problem is thus

\[ a_n = u_n + v_n = A + Bn + Cn^2 + D2^n - \frac{n^3}{2}. \]
(d) We now ask the solution in (c) to comply with the initial conditions.

<table>
<thead>
<tr>
<th>Initial Conditions</th>
<th>Induced Equations</th>
<th>Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_0 = 2$</td>
<td>$A + D = 2$</td>
<td>$A = 3$</td>
</tr>
<tr>
<td>$a_1 = -1/2$</td>
<td>$A + B + C + 2D = 0$</td>
<td>$B = -2$</td>
</tr>
<tr>
<td>$a_2 = -5$</td>
<td>$A + 2B + 4C + 4D = -1$</td>
<td>$C = 1$</td>
</tr>
<tr>
<td>$a_3 = -31/2$</td>
<td>$A + 3B + 9C + 8D = -2$</td>
<td>$D = -1$</td>
</tr>
</tbody>
</table>

Hence our required particular solution takes the following final form

$$a_n = 3 - 2n + n^2 - \frac{n^3}{2} - 2^n, \quad n \geq 0.$$ 

4. Find the general solution of $a_{n+1} - a_n = n2^n + 1$ for $n \geq 0$.

**Solution.**

(a) The general solution for homogeneous problem is $u_n = A$ because the only root of the characteristic equation is $\lambda_1 = 1$.

(b) Since $n2^n + 1 = 2^n \times n + 1^n$ is of the form $\mu_1^n(b_1n + b_0) + \mu_2^n c_0$ and $\mu_2 = 1$ is a simple root of the characteristic equation, we try the similar form $v_n = 2^n(B + Cn) + Dn$ for a particular solution. Substituting $v_n$ into the recurrence relation, we have

$$n2^n + 1 = v_{n+1} - v_n = 2^{n+1}(B + C(n + 1)) + D(n + 1) - 2^n(B + Cn) - Dn$$

$$= 2^n(Cn + B + 2C) + D,$$

i.e.,

$$2^n [ (C - 1)n + (B + 2C) ] + (D - 1) = 0.$$ 

In order the above equation be identically 0 for all $n \geq 0$, we require all its coefficients to be 0, i.e.,

$$C - 1 = 0, \quad B + 2C = 0, \quad D - 1 = 0.$$  

Hence $B = -2$, $C = 1$ and $D = 1$ and the particular solution is $v_n = 2^n(n - 2) + n$.

(c) The general solution is $u_n + v_n$ and thus reads

$$a_n = 2^n(n - 2) + n + A, \quad n \geq 0.$$  

\[\Box\]
5. Let \( m \in \mathbb{N} \) and \( S(n) = \sum_{i=0}^{n} i^m \) for \( n \in \mathbb{N} \). Convert the problem of finding \( S(n) \) to a problem of solving a recurrence relation.

**Solution.** We first observe

\[
S(n + 1) = (n + 1)^m + \sum_{i=0}^{n} i^m = S(n) + (n + 1)^m.
\]

Since the general solution will contain just 1 arbitrary constant, one initial condition should suffice. Hence \( S(n) \) is the solution of

\[
S(n + 1) - S(n) = (n + 1)^m, \quad \forall n \in \mathbb{N}
\]

\[
S(0) = 0.
\]

**Note.** A similar procedure for solving linear, constant coefficient nonhomogeneous recurrence relations can be found in the book by Stephen B Maurer and A Ralston, *Discrete Algorithmic Mathematics*, Addison-Wesley, 1991.

**Summary**

1. **Homogeneous case**

   First solve the characteristic equation.

   (a) Distinct roots

   **Example** \( \lambda = 1, \quad \lambda = 2 \)

   **Solution** \( a_n = A(1)^n + B(2)^n = A + B2^n \)

   (b) Repeated roots

   **Example** \( \lambda = 3 \) with multiplicity 2

   **Solution** \( a_n = (An + B)3^n \)

   **Example** \( \lambda = 2 \) with multiplicity 3

   **Solution** \( a_n = (An^2 + Bn + C)2^n \)

   **Example** \( \lambda = -1 \) with multiplicity 2

   **Solution** \( a_n = A2^n + (Bn + C)(-1)^n \)

If there are initial conditions use them to find the values for the constants \( A, B, \) etc.
2. Nonhomogeneous case

Solve the characteristic equation and obtain a solution to the homogeneous case as in (1). Call it \( u_n \). If there are initial conditions leave these for the moment.

Next find a particular solution, \( v_n \), to the homogeneous case.

(a) Solution to the homogeneous case not like the right hand side of the recurrence relation.

Try substituting something that looks like the right hand side (RHS).

<table>
<thead>
<tr>
<th>Example</th>
<th>RHS</th>
<th>( v_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( 2^n )</td>
<td>( C2^n )</td>
</tr>
<tr>
<td></td>
<td>( 4n )</td>
<td>( An + B )</td>
</tr>
<tr>
<td></td>
<td>( 3^n n )</td>
<td>( 3^n (An + B) )</td>
</tr>
<tr>
<td></td>
<td>( n^2 )</td>
<td>( An^2 + Bn + C )</td>
</tr>
</tbody>
</table>

(b) Solution to the homogeneous case similar to the RHS of the recurrence relation.

(i) Distinct roots

Try a solution with an extra \( n \).

| Example | \( \lambda = 2 \) and RHS = \( 2^n \), try \( v_n = Cn2^n \). |

(ii) Repeat Roots

Try a solution with an extra \( n^2 \) for multiplicity 2, \( n^3 \) for multiplicity 3 etc.

<table>
<thead>
<tr>
<th>Example</th>
<th>( \lambda = 3 ) with multiplicity 2 and RHS = ( 3^n ), try ( v_n = Cn^23^n ).</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \lambda = 1 ) with multiplicity 3 and RHS = ( 4 = 4 \times 1^n ), try ( v_n = Cn^3 \cdot 1^3 = Cn^3 ).</td>
</tr>
</tbody>
</table>

Once you have decided what a possible solution to the nonhomogeneous case could be, substitute it into the original recurrence relation. Then find values for the constants and put these values back into the expressions for \( v_n \).

Now add \( u_n \) and \( v_n \) to obtain the general solution. That is, \( a_n = u_n + v_n \).

If there are initial conditions, substitute them in now to find the values of the constants that were part of \( v_n \).

Finally, check your answer by substituting it into the original recurrence relation.
Example

6. (a) Find the general solution of the recurrence relation

\[ a_{n+2} - 5a_{n+1} + 6a_n = 0, \quad n \geq 0. \]

(b) Find a particular solution of the recurrence relation given in (a) such that it satisfies the initial conditions \(a_0 = 2\) and \(a_1 = 3\).

(c) Find the general solution for the recurrence relation

\[ w_{n+2} - 4w_{n+1} + 4w_n = 2^{n+2}, \quad n \geq 0. \]

Solution.

(a) The associated characteristic equation \(\lambda^2 - 5\lambda + 6 = 0\) has two (distinct) roots \(\lambda_1 = 2\) and \(\lambda_2 = 3\). Hence the general solution of the recurrence relation is

\[ a_n = A2^n + B3^n, \quad n \geq 0, \]

where \(A\) and \(B\) are arbitrary constants.

(b) From the general solution obtained in (a), the initial conditions give rise to the following two equations

\[ a_0 = A + B = 2, \quad a_1 = A \times 2 + B \times 3 = 3. \]

Solving the above two equations we obtain \(A = 3\) and \(B = -1\). Hence the particular solution satisfying the initial conditions is

\[ a_n = 3 \times 2^n - 3^n, \quad n \geq 0. \]

(c) The associated characteristic equation \(\lambda^2 - 4\lambda + 4 = (\lambda - 2)^2 = 0\) has a double root \(\lambda_1 = 2\). Hence the general solution of the corresponding homogeneous problem is \(u_n = 2^n(A + Bn)\) for \(n \geq 0\), where \(A\) and \(B\) are arbitrary constants. Since the nonhomogeneous term is \(2^{n+2} = 4 \times 2^n\), a particular solution \(v_n\) will take the form \(v_n = Cn^22^n\) because 2 is a double root of the associated characteristic equation. Hence, substituting \(v_n\) into the nonhomogeneous recurrence relation \(v_{n+2} - 4v_{n+1} + 4v_n = 2^{n+2}\), we obtain

\[ C\left[4(n + 2)^2 - 4(n + 1)^2 \times 2 + 4n^2\right] = 4, \]

which then simplifies to just \(2C = 1\). Hence we have \(C = \frac{1}{2}\) and \(v_n = n^22^{n-1}\). The general solution \(w_n\) of the nonhomogeneous recurrence relation is thus \(w_n = u_n + v_n\), and hence takes the form

\[ w_n = 2^n(A + Bn) + n^2 \cdot 2^{n-1} \]

\[ = 2^n\left[A + Bn + \frac{n^2}{2}\right], \quad n \geq 0. \]
Exercises

1. Find the general solution of the recurrence relation $a_{n+2} - 9a_{n+1} + 14a_n = 2^n$.

2. Solve the recurrence relation $a_{n+2} - 2a_{n+1} + a_n = 2$ with the initial conditions $a_0 = 1$ and $a_1 = 1$. 
25 Hints and Solutions to Supplementary Exercises

§1 1. Let $S = \{(a, x), (5, x), (6, x), (a, \emptyset), (5, \emptyset), (6, \emptyset)\}$, $\mathcal{P}(B) = \{\emptyset, \{x\}, \{\emptyset\}, \{x, \emptyset\}\}$. 

2. From the Venn diagram below we see that $(A - B) \cup (B - C) \cup (C - A) = A \cup B \cup C$ holds if and only if $A \cap B \cap C = \emptyset$.

3. Let $U$ be the universal set. We first show $(A \cap B)' \subseteq A' \cup B'$. For any $x \in (A \cap B)'$, we have $x \in U - (A \cap B)$, i.e. $x \in U$ but $x \notin A \cap B$. If $x \notin A$ then $x \in U - A = A' \subseteq A' \cup B'$. If (otherwise) $x \in A$ then we must have $x \notin B$ because $x \notin A \cap B$. Thus $x \in U - B = B' \subseteq A' \cup B'$. Hence we have shown $x \in A' \cup B'$ in both cases, implying $(A \cap B)' \subseteq A' \cup B'$. Now we show $A' \cup B' \subseteq (A \cap B)'$. For any $y \in A' \cup B'$, if $y \in A'$ then $y \notin A$, then $y \notin (A \cup B)$ and then $y \in (A \cup B)'$. If (otherwise) $y \notin A'$ then $y \in B'$ because $y \in A' \cup B'$. Thus we have likewise $y \in (A \cup B)'$. Hence $y \in (A \cup B)'$ in both cases, implying $A' \cup B' \subseteq (A \cap B)'$. From $(A \cap B)' \subseteq A' \cup B'$ and $A' \cup B' \subseteq (A \cap B)'$ we then conclude $(A \cap B)' = A' \cup B'$.

4. A simple recursive calculation shows that $S_2(5, 3) = 25$. There are $2^n$ subsets of a set with $n$ elements. A subset and its complement consist of a solution of $S_2(n, 2)$, except the empty set. Therefore, $S_2(n, 2) = 2^{n-1} - 1$.

§2 1. Let $S_n$ denote the statement $2^0 + \cdots + 2^n = 2^{n+1} - 1$. Since for $n = 1$ the statement $S_1$ is $2^0 + 2^1 = 2^2 - 1$ which is obviously true, we know $S_1$ is true. For the inductive step, we assume $S_k$ is true for some $k \geq 1$ and proceed to show that this will ensure $S_{k+1}$ is also true. Since 

$$2^0 + \cdots + 2^k + 2^{k+1} = (2^0 + \cdots + 2^k) + 2^{k+1}$$

$$= (2^{k+1} - 1) + 2^{k+1} = 2^{k+1}(1 + 1) - 1 = 2^{k+2} - 1 ,$$

where we have made use of the identity $2^0 + \cdots + 2^k = 2^{k+1} - 1$ implied by the induction assumption $S_k$, we see that $S_{k+1}$ is true whenever $S_k$ is true. Hence, from the P.M.I., we conclude that $S_n$ is true for all integers $n \geq 1$. 

196
2. Let $S_n$ denote the statement $3^n > 4n$. Obviously $S_2$ is true because the statement in this case is just $3^2 > 4 \times 2$. Assume that $S_k$ is true for some $k \geq 2$. Then $3^k > 4k$, and thus
\[ 3^{k+1} = 3^k \times 3 > 4k \times 3 = 4(k + 1) + (8k - 4) > 4(k + 1) \]
because $8k - 4 > 0$ due to $k \geq 2$. Hence we have shown $3^{k+1} > 4(k + 1)$, i.e. $S_{k+1}$ is true whenever $S_k$ is true. The P.M.I. then implies $S_n$ is true for all $n \geq 2$.

3. Let $S_n$ denote the statement $a_n = 2^{n+1} - 1$. First we observe easily that $S_0$ is true because $a_0 = 2^{0+1} - 1 = 1$ is exactly the same value given by the question. Assume $S_k$ is true for some $k \geq 0$, thus implying $a_k = 2^{k+1} - 1$. Then from the recurrence relation we have
\[ a_{k+1} = 2a_k + 1 = 2(2^{k+1} - 1) + 1 = 2^{k+2} - 2 + 1 = 2^{(k+1)+1} - 1, \]
and hence $S_{k+1}$ is true. The P.M.I. then implies $a_n = 2^{n+1} - 1$ for all integers $n \geq 0$.

§3

1. 3 multiplications are needed because
\[ 2x^3 + 5x^2 + 3x + 1 = x \times (x \times (x \times 2 + 5) + 3) + 1. \]

2. Since $|2n^3 - 9n + 1| \leq 12|n^3|$ holds for $n \geq 1$ because
\[
|2n^3 - 9n + 1| \leq |2n^3| + |9n| + |1| \quad \text{(generalised triangle inequality)}
\leq 2n^3 + 9n \times n^2 + n^3 \quad \text{(assume } n \geq 1) \\
\leq 12|n^3|,
\]
we conclude from the definition of big $O$ that $2n^3 - 9n + 1 = O(n^3)$.

3. Let $n \geq 2$. Then $|n - 1| \geq n - 1 \geq n - \frac{9}{2} = \frac{n}{2}$ and $\sqrt{|n - 1|} + 1 \geq \sqrt{|n - 1|} \geq \sqrt{n}$. Hence
\[
\left| \frac{n + 1}{\sqrt{|n - 1|} + 1} \right| = \frac{n + 1}{\sqrt{|n - 1|} + 1} \leq \frac{n + 1}{\sqrt{n}} \leq \frac{n + n}{\sqrt{n}} = 2 \sqrt{2} \sqrt{n}.
\]
In other words we have shown that $\left| \frac{n + 1}{\sqrt{|n - 1|} + 1} \right| \leq (2 \sqrt{2}) \cdot |\sqrt{n}|$ holds for all $n \geq 2$. Hence $\frac{n + 1}{\sqrt{|n - 1|} + 1} = O(\sqrt{n})$.

4. Since $f(x) = O(g(x))$ implies the existence of a constant $C > 0$ and a constant $L$ such that $|f(x)| \leq C|g(x)|$ holds for all $x \geq L$. Let $M$ be the maximum value of $C$ and $L$, i.e. $M = \max\{C, L\}$. Then for $x \geq M$ we have $x \geq L$ and thus $|f(x)| \leq C|g(x)| \leq M|g(x)| \leq |x \cdot g(x)|$.

5. Suppose that $f(x)$ is $O(g(x))$ and $g(x)$ is $O(h(x))$. By definition of $O$-notation, there exist positive numbers $C_1$ and $C_2$, and non-negative numbers $M_1$ and $M_2$ such that
\[ |f(x)| \leq C_1|g(x)| \quad \text{for all real numbers } x > M_1 \]
and
\[ |g(x)| \leq C_2|h(x)| \quad \text{for all real numbers } x > M_2. \]
Let $C = C_1 C_2$ and let $M = \max(M_1, M_2)$. Then, if $x > M$,

$$|f(x)| \leq C_1 |g(x)| \leq C_1 (C_2 |h(x)|) \leq (C_1 \cdot C_2) |h(x)| \leq C |h(x)|$$

Thus, by definition of $O$-notation, $f(x)$ is $O(h(x))$.

§4 1. Upper bound.

$$|2n^3 + 9 \log n + 1| = 2n^3 + 9 \log n + 1 \quad \text{(assume } n \geq 1\text{)}$$

$$\leq 2n^3 + 9n + 1 \quad \text{(use } \log_2 n \leq n\text{)}$$

$$\leq 2n^3 + 9n \times n^2 + n^3 = 12n^3$$

Lower bound.

$$|2n^3 + 9 \log n + 1| = 2n^3 + 9 \log n + 1 \quad \text{(assume } n \geq 1\text{)}$$

$$\geq 2n^3 = 2|n^3|$$

Hence we have

$$2|n^3| \leq |2n^3 + 9 \log n + 1| \leq 12|n^3|$$

for all $n \geq 1$, implying $2n^3 + 9 \log n + 1 = \Theta(n^3)$.

2. Upper bound.

$$\left| \frac{n + \sin n}{\sqrt{n} + 1} \right| = \frac{n + \sin n}{\sqrt{n} + 1} \quad \text{(assume } n \geq 2\text{, observe } |\sin n| \leq 1\text{)}$$

$$\leq \frac{n + \frac{n}{\sqrt{n}}}{\sqrt{n}} \leq 2 \sqrt{n}$$

Lower bound.

$$\left| \frac{n + \sin n}{\sqrt{n} + 1} \right| = \frac{n + \sin n}{\sqrt{n} + 1} \quad \text{(assume } n \geq 2\text{, observe } |\sin n| \leq 1\text{)}$$

$$\geq \frac{n + \frac{n}{\sqrt{n}}}{\sqrt{n} + \sqrt{n}} \quad (\sqrt{n} \geq 1 \text{ because } n \geq 2)$$

$$\geq \frac{n - 1}{2 \sqrt{n}} \geq \frac{n - \frac{n}{2}}{2 \sqrt{n}} \quad (\frac{n}{2} \geq 1 \text{ because } n \geq 2)$$

$$= \frac{1}{4} \sqrt{n}$$

Hence we have

$$\frac{1}{4} |\sqrt{n}| \leq \left| \frac{n + \sin n}{\sqrt{n} + 1} \right| \leq 2 |\sqrt{n}|$$

for all $n \geq 2$, implying $\frac{n + \sin n}{\sqrt{n} + 1} = \Theta(\sqrt{n})$.

3. If otherwise, i.e. $nf(n) = O(g(n))$, then there would exist a constant $C > 0$ and another constant $M$ such that $|nf(n)| \leq C|g(n)|$ whenever $n \geq M$. 

198
Since \( f(n) = \Theta(g(n)) \) implies the existence of a constant \( D > 0 \) and a constant \( N \) such that \( D|g(n)| \leq |f(n)| \) whenever \( n \geq N \), we would have

\[
D|g(n)| \leq |f(n)| \leq \frac{C}{n} \cdot |g(n)|, \quad \text{hence,} \quad \left( D - \frac{C}{n} \right) \cdot |g(n)| \leq 0,
\]

for all \( n > L = \max\{M, N, C/D, 1\} \). Due to \( D - C/n > 0 \) we would thus have to have \( |g(n)| \leq 0 \) which is impossible because \( |g(n)| \geq \frac{C}{n} \cdot |f(n)| > 0 \). Hence our initial assumption \( nf(n) = O(g(n)) \) must have been incorrect. In other words we have shown \( nf(n) \neq O(g(n)) \).

§5
1. \( \left\lceil \frac{3 - x}{2} \right\rceil = 2, \left\lceil \frac{3 - x}{3} \right\rceil = 1 \).
2. 4 comparisons. The comparisons are made with \( E, H, F \) and \( G \) sequentially.

§6
1. (a) and (b) are shown in the first truth table below, while (c) and (d) are shown in the second truth table.

<table>
<thead>
<tr>
<th>( p )</th>
<th>( q )</th>
<th>( \sim p )</th>
<th>( \sim q )</th>
<th>( p \lor q )</th>
<th>( (\sim p) \lor q )</th>
<th>( (p \lor q) \land ((\sim p) \lor q) )</th>
<th>( p \rightarrow q )</th>
<th>( (\sim q) \rightarrow (\sim p) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>( T )</td>
<td>( F )</td>
<td>( F )</td>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
</tr>
<tr>
<td>( T )</td>
<td>( F )</td>
<td>( F )</td>
<td>( T )</td>
<td>( T )</td>
<td>( F )</td>
<td>( F )</td>
<td>( F )</td>
<td>( F )</td>
</tr>
<tr>
<td>( F )</td>
<td>( T )</td>
<td>( T )</td>
<td>( F )</td>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
</tr>
<tr>
<td>( F )</td>
<td>( F )</td>
<td>( T )</td>
<td>( T )</td>
<td>( F )</td>
<td>( F )</td>
<td>( T )</td>
<td>( T )</td>
<td>( T )</td>
</tr>
</tbody>
</table>

identical columns

<table>
<thead>
<tr>
<th>( p )</th>
<th>( q )</th>
<th>( \sim p )</th>
<th>( \sim q )</th>
<th>( p \lor q )</th>
<th>( (\sim p) \lor q )</th>
<th>( (p \lor q) \land ((\sim p) \lor q) )</th>
<th>( (p \lor q) \lor ((\sim p) \lor q) )</th>
<th>( (p \lor q) \land ((\sim p) \land (\sim q)) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>( T )</td>
<td>( F )</td>
<td>( F )</td>
<td>( T )</td>
<td>( F )</td>
<td>( T )</td>
<td>( F )</td>
<td>( F )</td>
</tr>
<tr>
<td>( T )</td>
<td>( F )</td>
<td>( F )</td>
<td>( T )</td>
<td>( F )</td>
<td>( T )</td>
<td>( T )</td>
<td>( F )</td>
<td>( F )</td>
</tr>
<tr>
<td>( F )</td>
<td>( T )</td>
<td>( T )</td>
<td>( F )</td>
<td>( T )</td>
<td>( F )</td>
<td>( T )</td>
<td>( F )</td>
<td>( F )</td>
</tr>
<tr>
<td>( F )</td>
<td>( F )</td>
<td>( T )</td>
<td>( T )</td>
<td>( F )</td>
<td>( T )</td>
<td>( T )</td>
<td>( F )</td>
<td>( F )</td>
</tr>
</tbody>
</table>

all \( T \): tautology

all \( F \): contradiction

2. The truth table below shows \( (p \land q) \lor r \) and \( p \land (q \lor r) \) are not logically equivalent because their corresponding columns in the truth table are not exactly the same.
§7 1. $p \rightarrow q$ doesn’t imply $q \rightarrow p$, i.e. $(p \rightarrow q, \therefore q \rightarrow p)$ is not valid, because the argument form fails at 1 critical row shaded in the truth table below.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$r$</th>
<th>$p \land q$</th>
<th>$q \lor r$</th>
<th>$(p \land q) \lor r$</th>
<th>$p \land (q \lor r)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
<td>$F$</td>
</tr>
</tbody>
</table>

(columns not exactly the same)

2. We observe from the truth table below that there are no critical rows at all. Hence the argument form is valid.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$q$</th>
<th>$p \rightarrow q$</th>
<th>$q \rightarrow p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
</tr>
<tr>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
<td>$F$</td>
</tr>
<tr>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
</tbody>
</table>
3. The derivation is as follows:

(1) \( r, p \rightarrow \sim r, \therefore \sim p \) (modus tollens)
(2) \( \sim p, p \lor \sim q, \therefore \sim q \) (disjunctive syllogism)
(3) \( r, \sim q, \therefore r \land \sim q \) (conjunctive addition)
(4) \( \sim r \lor q \lor s, \therefore \sim (r \land \sim q) \lor s \) (De Morgan’s laws)
(5) \( r \land \sim q, \sim (r \land \sim q) \lor q, \therefore q \) (disjunctive syllogism)

Hence the argument form is valid.

4. If \((p_1, \ldots, p_n, \therefore q)\) is valid, then \(q\) is true whenever \(p_1, \ldots, p_n\) are all true, i.e. whenever \(p_1 \land \cdots \land p_n\) is true. Hence \(\sim q\) is false whenever \(p_1 \land \cdots \land p_n\) is true. That is, \(p_1 \land \cdots \land p_n \land \sim q\) is false whenever \(p_1 \land \cdots \land p_n\) is true. However, since \(p_1 \land \cdots \land p_n \land \sim q\) is also false if \(p_1 \land \cdots \land p_n\) is false, hence \(p_1 \land \cdots \land p_n \land \sim q\) is always false. In other words \(p_1 \land \cdots \land p_n \land \sim q\) is a contradiction. Conversely, if \(p_1 \land \cdots \land p_n \land \sim q\) is a contradiction, then \((p_1 \land \cdots \land p_n) \land \sim q\) is always false. Hence if \(p_1 \land \cdots \land p_n\) is true, then \(\sim q\) must be false, i.e. \(q\) must be true. This means exactly that the argument form \((p_1, \ldots, p_n, \therefore q)\) is valid.

\[1\] \(\forall x \in D, \forall y \in D, L(x, y) \leftrightarrow L(y, x)\).

2. Due to \(p \rightarrow q \equiv \sim p \lor q\) for any statements \(p\) and \(q\) we observe that the negation is

\[\sim \left( \forall x \in \mathbb{R}, \sim (x > 3) \lor (x^2 > 4) \right) \equiv \exists x \in \mathbb{R}, \sim \left( \sim (x > 3) \lor (x^2 > 4) \right) \equiv \exists x \in \mathbb{R}, (x > 3) \land \sim (x^2 > 4) \equiv \exists x \in \mathbb{R}, (x > 3) \land (x^2 \leq 4)\]
i.e., there exists an $x \in \mathbb{R}$ such that $x > 3$ and $x^2 \leq 4$. This negated statement is obviously not true because any real value $x > 3$ will not satisfy the condition $x^2 \leq 4$ simultaneously.

3. A sufficient condition for a student to pass this unit is that he or she does all the assignments. But this condition is not a necessary condition. The original statement has not specified any necessary conditions.

§9 1. There are essentially 3 different such graphs. They are

![Graphs](image)

2. $K_{n,2n}$ has $2n^2$ edges, and the length of a longest walk in $K_{n,2n}$ from one vertex to another, without walking on any edge and any vertex more than once, is $2n$.

§10 1. Neither $K_4$ nor $K_{2,3}$ are Eulerian. This is because not all vertices of $K_4$ nor of $K_{2,3}$ are of even degree. However $K_{2,3}$ does have an Eulerian path because $K_{2,3}$ has exactly 2 vertices of odd degrees.

2. One Eulerian circuit in $K_{2,4}$ is drawn below, in which the arrows indicate the walking direction and the numbers on the edges denote the walking order.

![Eulerian Circuit](image)

3. $K_{2,3}$ doesn’t contain any Hamiltonian circuits, but does contain a Hamiltonian path, see the arrowed path below.

![Hamiltonian Path](image)
§11 1. No, because $K_4$ has 4 vertices and $K_5$ has 5 vertices.

2. Yes, $K_5$ has a subgraph that is isomorphic to $K_4$. Pick any vertex from $K_5$, say v in the diagram below. If we remove that vertex and all the edges incident upon that vertex, then the remaining subgraph is isomorphic to $K_4$.

![Graph Diagram]

3. The graphs $G_1$ and $G_2$ represented by $A_1$ and $A_2$ respectively are drawn as

![Graph Diagram]

from which we see that the 2 graphs are isomorphic and the vertex correspondence of the isomorphism is

![Vertex Correspondence Diagram]

The total number of walks of length 2 in $G_1$ is the sum of all the elements of $A_1^2$. Since

$$A_1^2 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 2 & 2 \\ 1 & 3 & 1 & 3 \\ 2 & 1 & 2 & 2 \\ 2 & 3 & 2 & 4 \end{pmatrix},$$

the total number of walks of length 2 is 33.
4. The isomorphism is equivalent to following correspondence of the vertices

\[\begin{array}{cccccc}
v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \\
G_1 & G_2 & G_2 & G_2 & G_2 & G_2 \\
\end{array}\]

§12

1. Yes, it is possible to have a forest of \( n + 2 \) vertices and \( n \) edges. In fact such a forest must have exactly 2 connected components, each of which is a tree.

2. The strategy is to start with a vertex of degree 1, then gradually put back vertices and edges one by one logically and exhaustively. Remove the trees that are duplicated under graph isomorphisms. There are 3 nonisomorphic trees of 5 vertices, and they are

\[\begin{array}{ccc}
G_1 & G_2 & G_3 \\
\end{array}\]

3. The binary tree constructed during the sorting is

\[\begin{array}{c}
A \\
B \\
I \\
F \\
E \\
N \\
L \\
R \\
O \\
Y \\
S \\
T \\
\end{array}\]

The sorted list, read off the binary tree, is thus \( A, B, E, F, I, L, N, O, R, S, T, U, Y \).

4. **Preorder traversal:** \( A, B, D, F, I, K, C, E, G, H, J, L, M \).

   **Inoder traversal:** \( B, F, K, I, D, A, C, G, E, L, J, M, H \).

   **Postoder traversal:** \( K, I, F, D, B, G, L, M, J, H, E, C, A \).

5. **Basic step.** \( G \) is connected and has only one face \( (f = 1) \). It is a tree, so \( m = n - 1 \) and therefore \( n - m + f = n - (n - 1) + 1 = 2 \).
Inductive step. Suppose the formula holds for a connected graph with \( f - 1 \) faces. Prove it for \( f \).

Choose an edge \( e \) connecting two different faces of \( G \) and remove it. This removal decreases both the number of faces and edges by one. So, we can use inductive hypothesis \( n - m + f = n - (m - 1) + (f - 1) = 2 \).

\[
\begin{align*}
\text{Inductive step. Suppose the formula holds for a connected graph with } f - 1 \\
\text{faces. Prove it for } f. \\
\text{Choose an edge } e \text{ connecting two different faces of } G \text{ and remove it. This removal decreases both the number of faces and edges by one. So, we can use inductive hypothesis } n - m + f = n - (m - 1) + (f - 1) = 2.
\end{align*}
\]

§13 1. A typical spanning tree for \( K_{2,n} \) will look like

\[
\begin{array}{c}
\text{k vertices} \\
\vdots \\
\text{n–k–1 vertices}
\end{array}
\]

where the number of vertices in the 2 dashed boxes should sum up to \( n - 1 \). The dashed box with smaller number of vertices will be allowed to contain 0 up to \( \lfloor \frac{n+1}{2} \rfloor \) vertices. Hence the total number of nonisomorphic spanning trees is \( \lfloor \frac{n+1}{2} \rfloor \).

2. One of the minimal spanning trees is

\[
\begin{array}{c}
\text{A} \\
\text{E} \\
\text{D} \\
\text{F} \\
\text{C}
\end{array}
\]

The total weight of the minimal spanning tree is 21.

3. If a connected graph \( G \) is not a tree, then there exists at least 1 edge (taken from a nontrivial circuit of graph \( G \)), denoted by \( e \), such that the removal of edge \( e \) will leave the remaining graph, denoted by \( G' \), still connected. If we apply Fleury’s algorithm to \( G' \), we get a spanning tree (not containing edge \( e \)) of \( G' \), which is also a spanning tree of \( G \). If we order all edges of \( G \) in such a way that edge \( e \) is the very 1st edge and then apply Fleury’s algorithm again, we then obtain a spanning tree that will contain edge \( e \). Hence we have at least 2 different spanning trees for \( G \).
4. No. A graph with distinct weight on each edge has exactly 1 minimal spanning tree. This is because there is just 1 way of listing all the edges in the order of increasing weight.

§14
1. \(36_{10} = 27_{10} = 11011_2\).
2. Since \(16 = 4^2\), we can use the blockwise conversion. Since \(A_{16} = 22_4\), \(8_{16} = 20_4\) and \(F_{16} = 33_4\), we thus have \(A8.F_{16} = 2220.33_4\).
3. Since \(11011.1_2 = 1B.8_{16}\), we have \(A8.F_{16} + 11011.1_2 = A8.F_{16} + 1B.8_{16} = C4.7_{16}\).

§15
1. Since \(\tan(2\pi/3) = \tan 120^\circ = -\sqrt{3}\), equation \(y = 1 - \sqrt{3}x\) is a straight line of slope \(120^\circ\) that goes through the point \((0, 1)\) in the Cartesian plane. Hence \(\sqrt{3}x + y \geq 1\) describes the area on the right hand side of the straight line, including the line itself.

2. The binary relation \(F\) is \(F = \{(n, n^2) | n \in \mathbb{Z}\}\), i.e.

\[F = \{(0, 0), (1, 1), (-1, 1), (2, 4), (-2, 4), (3, 9), (-3, 9), ... \}\.

3. The digraph that represents the binary relation \(R\) is

Hence \(R\) is reflexive and transitive, but not symmetric.
4. Since $|f(x)| \leq |f(x)|$ means $f(x) = \Theta(f(x))$, we conclude immediately $(f, f) \in R$. That is, $R$ is reflexive. If $(f, g) \in R$ and $(g, h) \in R$, then there exist positive constants $C_1, D_1, C_2$ and $D_2$ and constants $M_1$ and $M_2$ such that

$$D_1|g(x)| \leq |f(x)| \leq C_1|g(x)|, \quad \forall x \geq M_1; \quad D_2|h(x)| \leq |g(x)| \leq C_2|h(x)|, \quad \forall x \geq M_2.$$ 

Hence for $x \geq M \overset{\text{def}}{=} \max\{M_1, M_2\}$ we have

$$(D_1D_2)|h(x)| \leq D_1|g(x)| \leq |f(x)| \leq C_1|g(x)| \leq (C_1C_2)|h(x)|,$$

which means $f(x) = \Theta(h(x))$, i.e. $(f, h) \in R$. Hence $R$ is transitive.

§16

1. Just 1 equivalence class.

2. There are 2 distinct equivalence classes. They are $[1] = \{1, 3, 5, 7, 9\}$ and $[2] = \{2, 4, 6, 8, 10\}$.

3. Yes, $R$ is an equivalence relation because the reflexivity, symmetry and transitivity can all be easily verified. There are 5 distinct equivalence classes. They are represented by $(2,0)$, $(1,0)$, $(0,0)$, $(0,1)$ and $(0,2)$ respectively. In the graph below, vertices in the same equivalence class are connected by some edges.

§17

1. There are 2 minimal elements, $a$ and $f$, and 1 maximal element $d$. The maximal element $d$ is also the greatest element because it can reach all other elements along paths that go downwards on every edge of the paths. However, there are no least elements because $a$ and $f$ are not comparable.

2. Let $x$, $y$ and $z$ be any elements of $\mathbb{R}$. $R$ is reflexive because $xRx$ holds for any $x$ due to $x \leq x$. $R$ is antisymmetric because $xRy$ and $yRx$ will imply $x \leq y$ and $y \leq x$, and hence $x = y$. $R$ is transitive because $xRy$ and $yRz$ will imply $x \leq y$ and $y \leq z$, and hence $x \leq z$ or, equivalently, $xRz$. Therefore $R$ is a partial order relation.

3. No, $R$ is not a partial order relation because $R$ is not reflexive. In fact, $xRx$, or $x < x$, is not true for any $x \in \mathbb{R}$. 

207
4. If we redraw the tree $T$ such that the root is at the very top and that child vertices are always drawn below their parents, then the newly drawn tree becomes also the Hasse diagram for the partial order relation $\leq$. The Hasse diagram here is obviously isomorphic to $T$.

§18

1. The switching system is

![Diagram of a switching system](image)

and is conductive when $a$ an $b$ are closed and $c$ and $d$ are open.

2. No. Because mappings $f$ and $g$ don’t have the same codomain.

3. Boolean algebra properties B1–B5, plus P2, imply

$$a = a \cdot 1 = a \cdot (1 + b) = a \cdot 1 + a \cdot b = a + a \cdot b.$$ 

§19

1. (a) $x + x + x + yz + xx' + yz = (x + x) + x + (yz + yz) + (xx') = x + x + yz + 0 = (x + x) + yz = x + yz.$

(b) $xyz + xy'z + xyz' + xy'z' = (xy + xy') \cdot (z + z') = x \cdot (y + y') \cdot (z + z') = x \cdot 1 \cdot 1 = x.$

2. $f(x, y, z) = x'y'z + xy'z + xyz'$. 

3. We obtain from $xy = (x'+y')' = ((x+x')+(y+y'))'$ the following gate implementation

![Gate Implementation](image)

4. There are two possible answers to this question. They correspond to the two tables below.
For the 2nd table (table B), for instance, the Boolean expression for the output is
\[ x'yz' + x'y'z + xy'z + xyz'. \]
This expression does not simplify any further. (Incidentally it is no coincidence that there are no adjacent terms, can you explain why?) The switching circuit thus takes the form

In fact a complete electrical circuit can be drawn as
where each switch is actually a multi-switch having just 2 states

The first state connects $A$ with $a$ and $B$ with $b$ while disconnecting $C$ from $c$ and $D$ from $d$. The second state does the opposite. If we use the readily-available toggle switches instead, then the electrical circuit can be wired as

where a toggle switch has the following 2 states
§20

1. \(xy'z', xyz\) and \(x'y'z\) are adjacent to \(xyz\). Since

\[
x'y'z + xy'z' + xz + xy' + x'y'z + xyz + x'y'z
\]

\[
= x'y'z + xy'z' + (xyz + xy'z) + (xy'z + xy'z') + x'y'z + xyz + x'y'z
\]

\[
= (x'y'z + x'y'z) + (xy'y'z + xy'y'z') + (xyz + xyz) + (xy'z + xy'z) + x'y'z
\]

\[
= x'y'z + xy'z' + xyz + xy'z + x'y'z,
\]

the canonical form is \(x'y'z + xy'z' + xyz + xy'z + x'y'z\).

2. \(x + x'y'z = (x + xyz) + x'y'z = x + (x + x'y)yz' = x + 1 \cdot yz' = x + yz'\).

3. An alternative minimal representation is \(yz' + y'z + wx'z\), as can be observed from the following Karnaugh map

4. First, the Karnaugh map grid for the given 5-variable Boolean expression

\[
vwy + vw'yz + v'y'z + vwx'z' + vw'x'y'z
\]

can be drawn as

<table>
<thead>
<tr>
<th></th>
<th>vwx</th>
<th>vwx'</th>
<th>vw'x</th>
<th>vw'x'</th>
<th>v'wx</th>
<th>v'wx'</th>
<th>v'w'x</th>
<th>v'w'x'</th>
</tr>
</thead>
<tbody>
<tr>
<td>yz</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>yz'</td>
<td>9</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>y'z'</td>
<td>12</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>y'z</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>v'w'x</th>
<th>v'w'x'</th>
<th>v'wx</th>
<th>v'wx'</th>
<th>v'y'z'</th>
<th>v'y'z'</th>
<th>v'y'z</th>
<th>v'y'z'</th>
<th>yz</th>
<th>yz'</th>
<th>y'z'</th>
<th>y'z</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>11</td>
<td>14</td>
<td>15</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
where the temporary *box labels* $[1-15]$ are placed there to facilitate our later explanations. Next, we put proper “1”s according to the given Boolean expression. One way of doing this is to first cast the expression into the canonical form and then place a corresponding “1” for each standard product term. An alternative way is to interpret each *product term*, which may or may not be a standard product term, in the Boolean expression directly as a block of 1’s. Hence the product terms $vwy, v'yz, v'yz, v'wx'z'$ and $v'x'y'z$ correspond to the blocks containing “1 2 9 10”, “3 4”, “5 6 7 8”, “9 10 12 13” and “15” respectively. Hence the Karnaugh map takes the following form

![Karnaugh map diagram]

Obviously, the block of eight “1”s and the block of four “1”s, as those circled in

![Karnaugh map diagram with circled blocks]

correspond to $yz$ and $vwy$ respectively. The block of two “1”s and another block of four “1”s, as those circled in
Correspond to \( v'w'x'z' \) and \( wx'z' \) respectively. Hence the minimal representation is
\[
yz + vwz' + wx'z' + v'w'x'z'.
\]
In fact the minimal representation is unique in this case.

§21
1. The roots are collected from both \( \lambda^2 + 9\lambda + 14 = 0 \) and \( \lambda^2 + 7\lambda + 10 = 0 \). The first equation gives two roots \(-2\) and \(-7\) while the second gives two roots \(-2\) and \(-5\). Hence all the roots, counting multiplicity, of the equation \((\lambda^2+9\lambda+14)(\lambda^2+7\lambda+10) = 0\) are \(-2\), \(-2\), \(-5\) and \(-7\). In other words, \(-5\) and \(-7\) are two simple roots while \(-2\) is a repeated root of multiplicity 2.

2. Iteratively, we have \( a_2 = 1 - 3a_0a_1 = -5 \) and \( a_3 = 1 - 3a_1a_2 = 31 \). Recursively, we have \( a_3 = 1 - 3a_1a_2 = 1 - 3a_1(1 - 3a_0a_1) = 1 - 3 \times 2 \times (1 - 3 \times 1 \times 2) = 31 \).

3. Yes, the recurrence relation \( a_{n+2} + na_{n+1} + 5a_n = 0 \) is linear and homogeneous. But it is not a constant coefficient recurrence relation because one of the coefficients, \( n \), is not a constant.

4. Since the associated characteristic equation \( \lambda^2 - 1 = 0 \) has two roots 1 and \(-1\), the general solution is thus given by \( a_n = A \times 1^n + B \times (-1)^n = A + B \times (-1)^n \), where \( A \) and \( B \) are arbitrary constants.

§22
1. \( \lambda = 1 \) is a root of the equation \( \lambda^3 - 4\lambda^2 + 5\lambda - 2 = 0 \) because, if \( \lambda = 1 \) is substituted into the equation, the equation evaluates to 0, i.e. \( 1^3 - 4 \times 1^2 + 5 \times 1 - 2 = 0 \). In fact a polynomial equation has 1 as one of its roots if and only if the summation of all the coefficients is 0. The polynomial equation has 1 as a root implies the factorisation \( \lambda^3 - 4\lambda^2 + 5\lambda - 2 = (\lambda - 1) \times (a \text{ second degree polynomial in } \lambda) \). According to the long division
the factorisation takes the form $\lambda^3 - 4\lambda^2 + 5\lambda - 2 = (\lambda - 1) \cdot (\lambda^2 - 3\lambda + 2)$. Since $\lambda^2 - 3\lambda + 2 = 0$ has two distinct roots, 1 and 2, the original equation $\lambda^3 - 4\lambda^2 + 5\lambda - 2 = 0$ has a simple root 2 and a repeated root 1 of multiplicity 2.

2. The associated characteristic equation $\lambda^2 + 14\lambda + 49 = 0$ has just one repeated root, -7, because the roots

$$-14 \pm \sqrt{14^2 - 4 \times 1 \times 49}$$

$$2 \times 1$$

both give the value -7. Hence the general solution of the recurrence relation is $a_n = (A + Bn) \times (-7)^n$, where A and B are arbitrary constants.

3. The associated characteristic equation has just one repeated root 3 of multiplicity 2. Hence the general solution of the recurrence relation is $a_n = (A + Bn) \times 3^n$ for some arbitrary constants A and B. From the initial conditions we have the following two equations

$$5 = a_0 = (A + B \times 0) \times 3^0, \quad -6 = a_1 = (A + B \times 1) \times 3^1,$$

i.e. $5 = A$ and $-6 = (A + B) \times 3$. Solving these two equations we obtain $A = 5$ and $B = -7$. Hence the final solution of the recurrence relation, satisfying also the initial conditions, is $a_n = (5 - 7n) \times 3^n$ for $n \geq 0$.

§23 1. No, $a_n = u_n + v_n$ can’t satisfy the recurrence relation $a_{n+15} - 2a_{n+1} + a_n = 1$. This is
because

\[
a_{n+1} - 2a_{n+1} + a_n = (u_{n+15} + v_{n+15}) - 2(u_{n+1} + v_{n+1}) + (u_n + v_n)
= (u_{n+15} - 2u_{n+1} + u_n) + (v_{n+15} - 2v_{n+1} + v_n)
= 1 + 1 = 2 \neq 1.
\]

2. The associated characteristic equation \( \lambda^2 - 9\lambda + 14 = 0 \) has two distinct roots, 2 and 7. Hence the general solution of the corresponding homogeneous recurrence relation is \( a_n = A2^n + B7^n \) for some arbitrary constants \( A \) and \( B \). The nonhomogeneous term \( 6n \) on the r.h.s. of the recurrence relation is of the form \( n = (1^n) \times (\text{polynomial of degree 1}) \), in which the base “1” of the power 1 is not a root of the characteristic equation. Hence a particular solution \( v_n \) of the nonhomogeneous problem will take the form \( v_n = (1^n) \times (Cn + D) = Cn + D \), where \( Cn + D \) is a general polynomial of degree 1 with \( C \) and \( D \) as undetermined constants. Substituting \( a_n = v_n \) into the nonhomogeneous recurrence relation we obtain

\[
v_{n+2} - 9v_{n+1} + 14v_n = (C(n + 2) + D) - 9(C(n + 1) + D) + 14(Cn + D) = 6n
\]

which then simplifies to \((6C - 6)n + (6D - 7C) = 0\). Hence \( C \) and \( D \) must satisfy the equations \( 6C - 6 = 0 \) and \( 6D - 7C = 0 \), implying \( C = 1 \) and \( D = \frac{7}{6} \), and thus \( v_n = n + \frac{7}{6} \). The general solution of the nonhomogeneous recurrence relation is therefore \( a_n = A2^n + B7^n + n + \frac{7}{6} \), where \( A \) and \( B \) are arbitrary constants.

3. From the solution of the previous question, the general solution of the nonhomogeneous recurrence relation, ignoring the initial conditions, is \( a_n = A2^n + B7^n + n + \frac{7}{6} \) for some arbitrary constants \( A \) and \( B \). The initial conditions thus give rise to the following 2 equations

\[
\frac{1}{6} = a_0 = A \times 2^0 + B \times 7^0 + 0 + \frac{7}{6}, \quad \frac{31}{6} = a_1 = A \times 2^1 + B \times 7^1 + 1 + \frac{7}{6}
\]

whose solutions are \( A = -2 \) and \( B = 1 \).

Hence the particular solution of the nonhomogeneous recurrence relation, satisfying the initial conditions, is \( a_n = -2^{n+1} + 7^n + n + \frac{7}{6} \).

§24 1. The characteristic equation and the general solution of the corresponding homogeneous recurrence relation have already been obtained in the questions in the previous section. The general solution of the homogeneous problem was found to be \( a_n = A2^n + B7^n \) for some arbitrary constants \( A \) and \( B \). We thus now proceed to find a particular solution for the nonhomogeneous problem.

Since the nonhomogeneous term on the r.h.s. of the recurrence relation has the form \( 2^n = 2^n \times (\text{polynomial of degree 0}) \), a particular solution \( v_n \) will take the same form, i.e. \( 2^n \times (\text{polynomial of degree 0}) \), plus an adjustment multiplier due to the base 2 of \( 2^n \) being also a root of the characteristic equation. The general form of
2^n \times (\text{polynomial of degree } 0) \text{ can be represented by } 2^n \times C \text{ for an arbitrary constant } C, \text{ and the adjustment multiplier is } n^1 \text{ because } 2 \text{ is a simple root of the characteristic equation. Hence a particular solution will take the form } v_n = (2^n C)n. \text{ Substituting } v_n \text{ into the nonhomogeneous recurrence relation we obtain}

\begin{align*}
v_{n+2} - 9v_{n+1} + 14v_n &= 2^{n+2}C(n + 2) - 9 \times 2^{n+1}C(n + 1) + 14 \times 2^n Cn = 2^n
\end{align*}

which then simplifies to \(-10C = 1\). Hence \(C = -\frac{1}{10}\), \(v_n = -\frac{1}{10} \times n2^n\), and the final general solution of the nonhomogeneous recurrence relation thus reads \(a_n = (A - \frac{n}{10})2^n + B7^n\).

2. The associated characteristic equation has only the root 1 of multiplicity 2. Hence the general solution of the corresponding homogeneous problem is just \(u_n = (A+Bn)1^n\), where \(A \text{ and } B \) are arbitrary constants. Since the nonhomogeneous term \(2 = 2 \times 1^n \) and 1 is a root of multiplicity 2 of the characteristic equation, a particular solution will take the form \(v_n = (1^n \times C) \times n^2 = Cn^2\). Substituting \(v_n\) into the nonhomogeneous recurrence relation we obtain

\begin{align*}
C(n + 2)^2 - 2C(n + 1)^2 + Cn^2 &= 2 \iff 2C = 2 \iff C = 1.
\end{align*}

Hence \(v_n = n^2\), and the general solution of the nonhomogeneous recurrence relation is thus \(a_n = u_n + v_n = A + Bn + n^2\). Since the initial conditions result in the following 2 equations

\begin{align*}
1 &= a_0 = A + B \times 0 + 0^2, \quad 1 &= a_1 = A + B \times 1 + 1^2
\end{align*}

whose solutions are \(A = 1\) and \(B = -1\), the final solution of the nonhomogeneous recurrence relation, satisfying the initial conditions, is therefore \(a_n = 1 - n + n^2\).

§A 1. For the selection sort, the least possible number of comparisons needed to sort a list of \(n\) items is \(\frac{1}{2}n(n-1)\) because every case is both the best and the worst case. For the bubble sort and the insertion sort, the best cases are when a list is already in the correct order. In such cases, \(n - 1\) comparisons are needed to sort a list of \(n\) items. For the merge sort, the best case is when the merging of every pair of lists of, say, \(p\) items and \(q\) items respectively, will take exactly the minimum \(\min\{p, q\}\) of comparisons. Hence for a list of \(n = 2^k\) items, the breakdown of the least number of comparisons is summarised in the table below, where \(t_1 = 2^{k-2}\), \(t_2 = 2^{k-1}\) and \(t_3 = 2^{k-1} + 2^{k-2}\).
Hints and Solutions to Supplementary Exercises AMTH140, Discrete Mathematics 2013

The total number of comparisons is thus

\[ 2^0 \times 2^{k-1} + 2^1 \times 2^{k-2} + \cdots + 2^{k-2} \times 2 + 2^{k-1} \times 1 = k2^{k-1} = \frac{n}{2} \log_2 n. \]

Hence in the best case of sorting a list of \( n = 2^k \) items, the merge sort needs \( \frac{1}{2} n \log_2 n \) comparisons.

2. If we always use the first item as the pivot, the sorting of a correctly ordered list is actually the worst case for the quick sort. This is rather embarrassing as we would normally expect that sorting a list that is already in the correct order should be easier. A simple way to overcome this problem is to choose a pivot randomly. Another simple method is to pick 3 items, the 1st, the last and the middlemost items, then choose for the pivot the item which is the median (the middle value) of the three. This is called the **Median-of-Three Partitioning**. For example, if we use this partitioning method to sort the list 5, 4, 1, 2, 3, 6, 7, then the intermediate steps are

<table>
<thead>
<tr>
<th>steps</th>
<th>lists</th>
<th>comparisons</th>
<th>1st, middle, last items</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5 4 1 2 3 6 7</td>
<td>0</td>
<td>5, 2, 7</td>
</tr>
<tr>
<td>1</td>
<td>4 1 2 3 5* 6 7</td>
<td>6</td>
<td>4, 1, 3 ; 6, 6, 7</td>
</tr>
<tr>
<td>2</td>
<td>1* 2* 3* 4* 5* 6* 7*</td>
<td>4</td>
<td>1, 1, 2</td>
</tr>
<tr>
<td>3</td>
<td>1* 2* 3* 4* 5* 6* 7*</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

The total number of comparisons made in this case is \( 6 + 4 + 1 = 11 \).

3. The derivation here requires the knowledge of integrations. In a typical pass by the quick sort, the pivot item, say \( a_m \), will have the probability \( \frac{1}{n} \) to be moved to position \( k \) for any \( k \) with \( 1 \leq k \leq n \),

\[
\frac{L}{a_1, \ldots, a_m, \ldots, a_n} \quad \rightarrow \quad \frac{L_1}{a_{N_1}, \ldots, a_{N_{k-1}}, a_m, a_{N_{k+1}}, \ldots, a_{N_n}} \quad \left( (k-1) \text{ items} \right) \\
\frac{L_2}{\text{(n-k) items}}
\]
requiring an average of additional \( T(k - 1) \) and \( T(n - k) \) comparisons to sort the 2 newly created sublists \( L_1 \) and \( L_2 \) respectively. Hence

\[
T(n) = (n - 1) + \frac{1}{n} \sum_{i=1}^{n} \left[ T(i - 1) + T(n - i) \right] = (n - 1) + \frac{2^{n-1}}{n} \sum_{i=0}^{n-1} T(i),
\]

where the term \("(n - 1)"\) is the number of comparisons needed to split the original list \( L \) into the 2 sublists \( L_1 \) and \( L_2 \). This means

\[
nT(n) = n(n - 1) + \left[ 2 \sum_{i=0}^{n-2} T(i) \right] + 2T(n - 1)
\]

\[
= n(n - 1) + [(n - 1)T(n - 1) - (n - 1)(n - 2)] + 2T(n - 1)
\]

\[
= (n + 1)T(n - 1) + 2(n - 1),
\]

or equivalently,

\[
\frac{T(n)}{2(n + 1)} = \frac{T(n - 1)}{2n} + \frac{2}{n + 1} - \frac{1}{n}.
\]

Hence, recursively and inductively,

\[
\frac{T(n)}{2(n + 1)} = \frac{T(n - 2)}{2(n - 1)} + \left( \frac{2}{n} - \frac{1}{n - 1} \right) + \left( \frac{2}{n + 1} - \frac{1}{n} \right) = \ldots
\]

\[
= \frac{T(1)}{2 \times 2} + \frac{2}{n + 1} + \frac{1}{n} + \frac{1}{n - 1} + \ldots + \frac{1}{3} - \frac{1}{2}.
\]

We thus have, due to \( T(1) = 0 \),

\[
\frac{T(n)}{2(n + 1)} = \frac{2}{n + 1} + \left[ \frac{1}{n} + \frac{1}{n - 1} + \ldots + \frac{1}{3} + \frac{1}{2} \right] - 1.
\]

\[
\leq \sum_{i=1}^{n-1} \int_{i}^{i+1} \frac{dx}{x} + \frac{2}{n + 1} - 1 = \int_{1}^{n} \frac{dx}{x} + \frac{1 - n}{n + 1} = \ln(n) + \frac{1 - n}{n + 1}, \quad n \geq 1.
\]

Since for \( f(n) = \ln(n) + 1 - n \) we have \( f(n) \leq 0 \) for all \( n \geq 1 \) because \( f(1) = 0 \) and \( f'(n) = 1/n - 1 \leq 0 \), we obtain

\[
T(n) \leq 2(n + 1) \ln(n) + 2(1 - n) = 2n \ln(n) + 2 \left[ \ln(n) + 1 - n \right] \leq 2n \ln(n), \quad \forall n \geq 1.
\]

Hence we conclude

\[
T(n) \leq \left( \frac{2}{\log_2 e} \right) \cdot n \log_2 n, \quad \forall n \geq 1.
\]

Likewise we have also for \( n \geq 1 \)

\[
\frac{T(n)}{2(n + 1)} = \frac{2}{n + 1} + \left[ \frac{1}{n} + \frac{1}{n - 1} + \ldots + \frac{1}{3} + \frac{1}{2} + 1 \right] - 2
\]

\[
\geq \sum_{i=1}^{n} \int_{i}^{i+1} \frac{dx}{x} + \frac{2}{n + 1} - 2 = \int_{1}^{n+1} \frac{dx}{x} - \frac{2n}{n + 1} = \ln(n + 1) - \frac{2n}{n + 1}.
\]
Hence for any $\varepsilon$ such that $0 < \varepsilon < 1$ we have for all $n \geq \lceil \exp(2/\varepsilon) \rceil$

$$T(n) \geq 2(n + 1) \ln(n + 1) - 4n$$
$$\geq 2n [\ln(n) - 2]$$
$$= 2(1 - \varepsilon)n \ln(n) + 2n [\varepsilon \ln(n) - 2]$$
$$\geq 2 \left[ (1 - \varepsilon)/ \log_2 e \right] n \log_2 n$$

because

$$\varepsilon \ln(n) - 2 \geq \varepsilon \ln[\exp(2/\varepsilon)] - 2 \geq 0.$$ 

Hence if there exist constants $D$ and $M$ such that $T(n) \leq Dn \log_2 n$ holds for all $n \geq M$, then we must have $D \geq 2(1 - \varepsilon)/ \log_2 e$. By taking the limit $\varepsilon \to 0$ we then conclude $D \geq 2/ \log_2 e = 2 \ln 2 \approx 1.386$. 


Index

$(S, +, \cdot, 0, 1)$, 146
$(x + y)^n$, 33
$A - B$, 20
$A \cap B$, 20
$A \cup B$, 20
$A \setminus B$, 20
$A \subseteq B$, 20
$A^{'}$, 21
$A_1 \times A_2 \times \cdots \times A_n$, 22
$B \supseteq A$, 19
$K_{m,n}$, 78
$K_n$, 78
$P \equiv Q$, 57
$P(x) \leftrightarrow Q(x)$, 70
$P(x) \Rightarrow Q(x)$, 70
$P \Leftrightarrow Q$, 57
$[a]$, 126
$\Theta$, 42
$\bot$, 149
$\rightarrow x \not\in$, 49
$\leftrightarrow x \notin$, 49
$\binom{n}{r}$, 32
$\emptyset$, 21
$\exists$, 70
$\forall$, 70
$\neg p$, 55
$\sim p$, 21
$\therefore$, 61
$\top$, 149
$|a|$, 11
$a \leq b$, 132
$f : A \times A \rightarrow A$, 145
$p \Leftrightarrow q$, 55, 61
$p \Rightarrow q$, 55
$p \lor q$, 55
$p \wedge q$, 55
$p \Leftrightarrow q$, 71

$p \Rightarrow q$, 59, 71
$p \rightarrow q$, 59, 63
$v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \cdots \xrightarrow{e_{n-1}} v_{n-1} \xrightarrow{e_n} v_n$, 80
$v_0, e_1, v_1, e_2, \cdots, v_{n-1}, e_n, v_n$, 80
$xRy$, 119
$E$, 80
$\mathbb{N}$, 80
$\mathbb{O}$, 37
$\mathcal{P}(S)$, 21
$\mathbb{C}$, 19
$\mathbb{N}$, 19
$\mathbb{Q}$, 19, 21
$\mathbb{R}$, 19
$\mathbb{Z}$, 19
$p_1, p_2, \ldots, p_n \therefore q$, 61
sequential search, 50

binary search, 50

absolute value, 11
absorption law, 147
addition, 118
adjacency matrix, 94
adjacent, 162
algorithm, 50
and, 61
AND-gate, 159
ansatz, 187
antisymmetric, 131
arc, 76
argument, 61
argument form, 61
arrows, 94
associativity, 146
axiom of comprehension, 24

base $p$, 113
base number, 113
big-pi notation, 26
binary number system, 113
binary operation, 145
binary relation, 119
binary relation on a set, 120
binary tree, 99
binary tree to sort, 102
binomial expansion, 33
binomial expansions, 16
binomial theorem, 33
block, 164
blockwise conversion, 117
books for reference, 7, 10, 25, 82, 87, 104, 138, 192
Boolean algebra, 146, 150
Boolean expression, 146, 154
Boolean function, 152
Boolean variable, 146
borrow, 118
branch vertex, 97
bubble sort, 53
cancellation law, 147
canonical, 154
cardinality, 22
carry, 118
Cartesian plane, 121
Cartesian product, 22
ceiling, 49
characteristic equation, 172
children, 99
circuit, 84
closed, 143
closure, 129
commutativity, 146
comparable, 132
compatible, 138
complement, 79
complementary switches, 144
complementation, 20, 146
complete bipartite, 78
complete graph, 77
complexity, 54
conclusion, 61, 63
conditional statement, 59
congruent, 123
conjunction, 55
conjunctive addition, 64
conjunctive simplification, 64
connected, 80
connected components, 81
connectives, 55
contradiction, 55, 61
contrapositive, 63, 74
contrapositive equivalence, 64
converse, 63, 74
reverse error, 74
convert numbers, 114
critical rows, 62
critical rows, 69
cycle, 97
decimal number system, 113
degree, 35, 80
depth, 99
difference, 20
difference equation, 169
digraph, 76
directed edge, 76
directed graph, 76
disjoint, 21
disjunction, 55
disjunctive addition, 64
disjunctive syllogism, 64
distributivity, 146
division into cases, 64
domain, 70
dominance laws, 147
double complement, 147
d-vertex-endpoint function, 76
edges, 76, 94
elimination method, 15
empty set, 21
<table>
<thead>
<tr>
<th>Term</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>equivalence</td>
<td>55</td>
</tr>
<tr>
<td>equivalence class</td>
<td>126</td>
</tr>
<tr>
<td>equivalence connective</td>
<td>61</td>
</tr>
<tr>
<td>equivalence relation</td>
<td>122</td>
</tr>
<tr>
<td>Euler's formula</td>
<td>81</td>
</tr>
<tr>
<td>Euler's Theorem</td>
<td>85</td>
</tr>
<tr>
<td>Eulerian circuit</td>
<td>84</td>
</tr>
<tr>
<td>Eulerian graph</td>
<td>84</td>
</tr>
<tr>
<td>Eulerian path</td>
<td>84</td>
</tr>
<tr>
<td>existential quantifier</td>
<td>70</td>
</tr>
<tr>
<td>factorise</td>
<td>179</td>
</tr>
<tr>
<td>factorising quadratic polynomials</td>
<td>14</td>
</tr>
<tr>
<td>false</td>
<td>61</td>
</tr>
<tr>
<td>first principles</td>
<td>37</td>
</tr>
<tr>
<td>Fleury's algorithm</td>
<td>87</td>
</tr>
<tr>
<td>floor</td>
<td>49</td>
</tr>
<tr>
<td>forest</td>
<td>97</td>
</tr>
<tr>
<td>function</td>
<td>22</td>
</tr>
<tr>
<td>general solution</td>
<td>173, 174, 178, 183</td>
</tr>
<tr>
<td>generalised triangle inequality</td>
<td>12</td>
</tr>
<tr>
<td>graph</td>
<td>76</td>
</tr>
<tr>
<td>graph isomorphism</td>
<td>89</td>
</tr>
<tr>
<td>greatest element</td>
<td>132</td>
</tr>
<tr>
<td>grid</td>
<td>162</td>
</tr>
<tr>
<td>Hamiltonian circuit</td>
<td>84</td>
</tr>
<tr>
<td>Hamiltonian path</td>
<td>84</td>
</tr>
<tr>
<td>Hasse diagram</td>
<td>134</td>
</tr>
<tr>
<td>height</td>
<td>99</td>
</tr>
<tr>
<td>hexadecimal number system</td>
<td>113</td>
</tr>
<tr>
<td>homogeneous</td>
<td>171</td>
</tr>
<tr>
<td>Horner's algorithm</td>
<td>36</td>
</tr>
<tr>
<td>Huffman coding</td>
<td>104</td>
</tr>
<tr>
<td>hypotheses</td>
<td>61</td>
</tr>
<tr>
<td>hypothesis</td>
<td>63</td>
</tr>
<tr>
<td>hypothetical syllogism</td>
<td>64</td>
</tr>
<tr>
<td>idempotent laws</td>
<td>147</td>
</tr>
<tr>
<td>identity relations</td>
<td>146</td>
</tr>
<tr>
<td>iff</td>
<td>57</td>
</tr>
<tr>
<td>implication</td>
<td>55, 59, 61</td>
</tr>
<tr>
<td>implies</td>
<td>59</td>
</tr>
<tr>
<td>induced</td>
<td>127</td>
</tr>
<tr>
<td>initial conditions</td>
<td>169, 184, 185</td>
</tr>
<tr>
<td>inorder traversal</td>
<td>101</td>
</tr>
<tr>
<td>insertion sort</td>
<td>53</td>
</tr>
<tr>
<td>internal vertex</td>
<td>97</td>
</tr>
<tr>
<td>intersection</td>
<td>20</td>
</tr>
<tr>
<td>inverse</td>
<td>63, 74</td>
</tr>
<tr>
<td>inverse error</td>
<td>74</td>
</tr>
<tr>
<td>isomorphic</td>
<td>89</td>
</tr>
<tr>
<td>isomorphic invariant</td>
<td>90</td>
</tr>
<tr>
<td>iteratively</td>
<td>170</td>
</tr>
<tr>
<td>Karnaugh map</td>
<td>162</td>
</tr>
<tr>
<td>Kruskal's algorithm</td>
<td>109</td>
</tr>
<tr>
<td>Kuratowski's Theorem</td>
<td>81</td>
</tr>
<tr>
<td>l.h.s.</td>
<td>28</td>
</tr>
<tr>
<td>leaf</td>
<td>97</td>
</tr>
<tr>
<td>least element</td>
<td>132</td>
</tr>
<tr>
<td>left child</td>
<td>99</td>
</tr>
<tr>
<td>legitimate logical expression</td>
<td>59</td>
</tr>
<tr>
<td>length</td>
<td>80</td>
</tr>
<tr>
<td>level</td>
<td>99</td>
</tr>
<tr>
<td>linear equations</td>
<td>15</td>
</tr>
<tr>
<td>linear, constant coefficient recurrence relation</td>
<td>171</td>
</tr>
<tr>
<td>literals</td>
<td>153</td>
</tr>
<tr>
<td>logarithm</td>
<td>13</td>
</tr>
<tr>
<td>logic circuit</td>
<td>158</td>
</tr>
<tr>
<td>logic gates</td>
<td>158</td>
</tr>
<tr>
<td>Logical Equivalences</td>
<td>58</td>
</tr>
<tr>
<td>logically equivalent</td>
<td>57</td>
</tr>
<tr>
<td>long division</td>
<td>179</td>
</tr>
<tr>
<td>loop</td>
<td>77</td>
</tr>
<tr>
<td>lower bound</td>
<td>42, 43</td>
</tr>
<tr>
<td>matrix</td>
<td>94</td>
</tr>
<tr>
<td>matrix multiplications</td>
<td>17, 92</td>
</tr>
<tr>
<td>maximal element</td>
<td>132</td>
</tr>
<tr>
<td>Median-of-Three Partitioning</td>
<td>217</td>
</tr>
</tbody>
</table>
merge sort, 53
middlemost, 51
minimal element, 132
minimal representation, 156
minimal spanning tree, 109
modus ponens, 63
modus tollens, 63
multiple edges, 77
multiplicity, 178
n-ary relation, 119
NAND-gate, 159
necessary condition, 74
negation, 55
negation of quantifiers, 72
nested evaluation, 36
nodes, 97
nonhomogeneous part, 188
nonhomogeneous problem, 183
nonhomogeneous recurrence relations, 183, 192
nontrivial circuit, 97
NOR-gate, 159
not, 61
NOT-gate, 158
number of walks, 94
octal number system, 113
open, 143
or, 61
OR-gate, 159
order of precedence, 61, 71

P.M.I., 27
parallel, 144
parallel edges, 77
parent, 99
partial order relation, 131
particular solution, 173, 183, 186
partition, 21, 127
path, 80
planar, 81
please read me, 18
Polish postfix notation, 102
polynomial, 35
postorder traversal, 101
power, 13
power set, 21
precedence, 61, 71, 146
predicate, 70
preliminary mathematics, 10
premise, 63
premises, 61
preorder traversal, 101
Principle of Mathematical Induction, 27
product terms, 153
proof, 68
proof sequence, 68
property closure, 129
proposition, 55
quick sort, 53
r.h.s., 28
recurrence relation, 169
recursively, 170
reference books, 7, 10, 25, 82, 87, 104, 138, 192
reflexive, 122
relations, 119
relative complement, 20
Remainder Theorem, 114
repeated division algorithm, 115
repeated root, 179
repeated roots, 178
representative, 126
right child, 99
root, 99
root and factor equivalence, 14
root of multiplicity, 174
root of multiplicity $M$, 188
rooted tree, 99
roots of quadratic equations, 13
rule of contradiction, 64
rules of inference, 72
Russell’s Paradox, 24
S.P.M.I., 32
sandwiched, 42
selection sort, 53
sequence, 26, 169
series, 144
series reduction, 82
set, 19
set identities, 23
siblings, 99
sigma notation, 26
simple graph, 77
simple root, 174
simplification procedure, 165
spanning tree, 105
standard product term, 154
step operation, 181
Strong Principle of M. I., 32
subgraph, 78
subset, 19
subtraction, 118
sufficient condition, 74
switch, 143
switching circuit, 143
switching function, 152
symmetric, 122
tautology, 55, 61
telescoping form, 36
terminating vertex, 97
theorem, 68
therefore, 61
topological sorting, 138
topological sorting algorithm, 139
total order relation, 132
totally ordered, 132
transitive, 122
transitive closure, 129, 135, 137
tree, 97
triangle inequalities, 37
triangle inequality, 12
trigonometric properties, 13
trivial walk, 80
true, 61
truth function, 152
truth tables, 56
tuple, 146
unary operation, 145
undirected graph, 76
union, 20
uniqueness of inverses, 147
universal modus ponens, 72
universal modus tollens, 72
universal quantifier, 70
universal set, 21
upper bound, 42, 43
vacuous truth, 72
valid, 62, 74
vertices, 76
w.r.t., 138
walk, 80
weight, 109
weighted graph, 84, 109
well-formed formulas, 59
wff, 59