

Name,DoB: Answer Key

IT University of Copenhagen
Discrete Mathematics, MSc SD
Re-Exam

4 Mar, 2020

Instructions (Read Carefully)

Contents: The exam contains 13 questions for a total of 100 points. The exam is divided into two parts: The first part has 9 multiple choice questions and the second part has 4 open ended questions.

What to check: In the multiple-choice questions, there is one and only one correct answer. You should only check 1 box.

Definitions and theorems: At the end of this document (page 8) you can find some definitions and theorems that could be useful for answering some of the questions.

Info about you: Write *clearly* your full name and your date of birth (DoB) on *every page* (top-right).

—IMPORTANT—

*Only information written on the pages 1–7 will be evaluated.
Anything else that you hand in will NOT be considered for the final evaluation!*

Part I. Answer the following multiple choice questions.

- S** 1. (6 pts) Which of the following statements is **true** for all sets A ?

☐ **A** $A \subseteq \emptyset$.

☐ **B** $A \in A$.

☒ **C** $A \in P(A)$.

☐ **D** $A \cup \emptyset = \emptyset$.

- F** 2. (6 pts) Let the function $f : \mathbb{Z} \rightarrow \mathbb{Z}$ be defined as follows:

$$f(n) = 3n + 2 \quad \text{for all } n \in \mathbb{Z}$$

Which of the following statements is **true**?

☐ **A** f is onto.

☒ **B** f is one-to-one.

☐ **C** $(f \circ f)(n) = 9n + 4$.

☐ **D** The range of f is a finite set.

Solution: Let n, m such that $f(m) = f(n)$, that is, $3m + 2 = 3n + 2$. By subtracting 2 on both sides of the equation we obtain $3m = 3n$, and by dividing by 3 on both sides of this new equation we obtain $m = n$.

- R** 3. (6 pts) Let A be an arbitrary set. Which of the following statements is **true** for all binary relations R and S on A ?

☐ **A** If R and S are transitive, then also $R \cup S$ is transitive.

☒ **B** If R and S are symmetric, then also $R \cup S$ is symmetric.

☐ **C** If R is transitive and $S \subseteq R$, then also S is transitive.

☐ **D** If R is symmetric and $S \subseteq R$, then also S is symmetric.

Solution: Let R, S be symmetric, and let $(x, y) \in R \cup S$. Then $(x, y) \in R$ or $(x, y) \in S$. Since both relations are symmetric, we obtain $(y, x) \in R$ or $(y, x) \in S$. Consequently, $(y, x) \in R \cup S$.

- CP** 4. (6 pts) A train company has found that 10% of their trains are delayed. Furthermore, their statistics show that 60% of the delayed trains travelled during rain, while of the punctual trains (i.e. trains that were on time) only 20% travelled during rain. Given that it is raining, what is the probability that a train is delayed?

☒ 25%

☐ 30%

☐ 60%

☐ 75%

Solution: Let D be the event that a given train is delayed and R the event that a given train travels during rain. We are given that $P(D) = 0.1$, $P(R|D) = 0.6$, and $P(R|D^c) = 0.2$. We need to calculate the conditional probability $P(D|R)$. To this end, we apply Bayes' Theorem:

$$\begin{aligned} P(D|R) &= \frac{P(R|D) \cdot P(D)}{P(R|D) \cdot P(D) + P(R|D^c) \cdot P(D^c)} \\ &= \frac{0.6 \cdot 0.1}{0.6 \cdot 0.1 + 0.2 \cdot 0.9} \\ &= 0.25 \end{aligned}$$

Here we also used Theorem 8 to calculate $P(D^c)$ as follows:

$$P(D^c) = 1 - P(D) = 1 - 0.1 = 0.9$$

- CP** 5. (6 pts) A team of seven software developers have to divide a set of roles among them. They need three *backend programmers*, two *testers*, and two *UI programmers*. One of the members of the team does not have the experience to be a backend programmer, but otherwise all other assignments of roles are allowed. In how many ways can they constitute their team?

☐ 720

☒ 120

☐ 100

☐ 54

Solution: There are $\binom{6}{3}$ ways of choosing the three backend programmers. After we choose the 3 backend programmers there are 4 people left, so there are $\binom{4}{2}$ ways of

choosing 2 testers. The remaining two people are the UI programmers. Hence, there are $\binom{6}{3} \cdot \binom{4}{2} = 120$ different ways to constitute the team.

- L** 6. (6 pts) Let the sequence a_1, a_2, a_3, \dots be recursively defined by

$$\begin{aligned}a_k &= a_{k-1} + k \quad \text{for all } k > 1 \\a_1 &= 1\end{aligned}$$

Which of the following equations is **true** for all $n > 0$?

☐ **A** $a_n = n(n+1)$

☒ $a_n = \sum_{i=1}^n i$

☐ **C** $a_n = 2^{n-1}$

☐ **D** $a_n = \prod_{i=1}^n i$

Solution: A cannot be true since $a_1 = 1$ but $1(1+1) = 2$; C cannot be true since $a_2 = 1+2 = 3$ but $2^{2-1} = 2$; D cannot be true since $a_2 = 3$ but $\prod_{i=1}^2 i = 1 \cdot 2 = 2$.

- A** 7. (6 pts) Consider the following six strings:

$w_1 = \lambda$	$w_4 = baba$
$w_2 = aaa$	$w_5 = baaba$
$w_3 = bbb$	$w_6 = bbababaa$

Which of those strings belong to the language defined by the regular expression $(a^*b)^*(\lambda \mid aa)$?

☐ **A** w_2, w_3, w_6

☒ w_1, w_3, w_6

☐ **C** w_1, w_4, w_5

☐ **D** w_1, w_5, w_6

- A** 8. (6 pts) Consider the following grammar:

$$\begin{aligned}S &\rightarrow AB \\A &\rightarrow aAb \\A &\rightarrow a \\B &\rightarrow Bab \\B &\rightarrow \lambda\end{aligned}$$

Which one of the strings below **does not** belong to the language generated by this grammar?

☐ A *aaabbab*

☐ B *aabab*

☒ C *aabbab*

☐ D *aaabb*

Solution: The three other strings can be generated as follows:

$$S \Rightarrow AB \Rightarrow aAbB \Rightarrow aaAbbB \Rightarrow aaabbB \Rightarrow aaabbBab \Rightarrow aaabbab$$

$$S \Rightarrow AB \Rightarrow aAbB \Rightarrow aabB \Rightarrow aabBab \Rightarrow aabab$$

$$S \Rightarrow AB \Rightarrow aAbB \Rightarrow aaAbbB \Rightarrow aaabbB \Rightarrow aaabb$$

G 9. (6 pts) Consider the following three descriptions of a graph:

- (A) A simple graph with 6 vertices: one vertex of degree 0, one of degree 1, one of degree 2, one of degree 3, one of degree 4 and one of degree 5.
- (B) A tree with 5 vertices and an Euler trail.
- (C) A tree with 5 vertices and a total degree of 10.

Which statement is **true**?

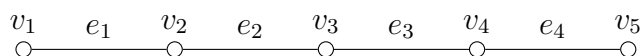
☐ A (B) and (C) exist, but (A) does not.

☒ B (B) exists, but neither (A) nor (C) do.

☐ C (A) and (B) exist, but (C) does not.

☐ D (C) exists, but neither (A) nor (B) do.

Solution: (A) not exist. Such a graph would have total degree of $0+1+2+3+4+5 = 15$. However, according to the Handshake Theorem, every graph has an even total degree. (B) does exist:



(C) does not exist. According to the Handshake Theorem, a graph with total degree 10 must have exactly 5 edges. However, according to Theorem 14 on page 14, a tree with 5 vertices must have exactly 4 edges.

Part II. Answer the following questions. Be brief but precise. Your correct use of mathematical notation is an important aspect of your answer.

- L 10.** (12 pts) Prove by mathematical induction that $n^3 + 5n$ is divisible by 3 for all integers $n \geq 0$.

Solution: We want to prove the statement

$$3 \mid n^3 + 5n$$

for all $n \geq 0$.

Basis step: Let $n = 0$. We have

$$0(0^2 + 5) = 0$$

and 3 divides 0 since $0 \cdot 3 = 0$.

Inductive step:

Suppose that $k \geq 0$ and that $P(k)$ holds, that is,

$$3 \mid k^3 + 5k$$

We must show that $P(k+1)$ holds, that is,

$$3 \mid (k+1)^3 + 5(k+1)$$

We have that

$$\begin{aligned}(k+1)^3 + 5(k+1) &= k^3 + 3k^2 + 3k + 1 + 5k + 5 \\ &= k^3 + 5k + 3k^2 + 3k + 6 \\ &= k^3 + 5k + 3(k^2 + k + 3)\end{aligned}$$

By induction hypothesis we have that 3 divides $k^3 + 5k$, and by definition 3 divides $3(k^2 + k + 3)$. Hence, by Theorem 1, 3 also divides $k^3 + 5k + 3(k^2 + k + 3)$. By the above calculation the latter is equal to $(k+1)^3 + 5(k+1)$, which is therefore also divisible by 3.

- L 11.** (12 pts) Using the logical equivalences on page 8, prove the following logical equivalence:

$$p \rightarrow (q \rightarrow r) \equiv (q \wedge p) \rightarrow r$$

In each step, indicate (by number) which equivalence from page 8 you used (only use one per step).

Solution:

$$p \rightarrow (q \rightarrow r) \quad (12)$$

$$\equiv p \rightarrow (\sim q \vee r)$$

$$\equiv \sim p \vee (\sim q \vee r) \quad (12)$$

$$\equiv (\sim p \vee \sim q) \vee r \quad (2)$$

$$\equiv \sim(p \wedge q) \vee r \quad (9)$$

$$\equiv (p \wedge q) \rightarrow r \quad (12)$$

$$\equiv (q \wedge p) \rightarrow r \quad (1)$$

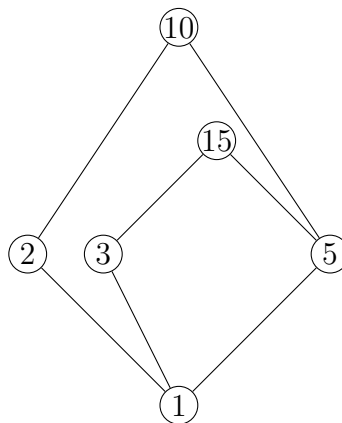
- R 12.** (12 pts) Consider the “divides” relation on the set $A = \{1, 2, 3, 5, 10, 15\}$. That is, consider the relation R defined as follows for all $n, m \in A$:

$$m R n \iff m \mid n$$

The relation R is a partial order relation.

- (a) Draw the Hasse diagram for R .

Solution:



- (b) Find all minimal elements of R . Write 'None' if there aren't any.

Solution: 1

- (c) Find all maximal elements of R . Write 'None' if there aren't any.

Solution: 10, 15

- (d) Find all least elements of R . Write 'None' if there aren't any.

Solution: 1

- (e) Find all greatest elements of R . Write 'None' if there aren't any.

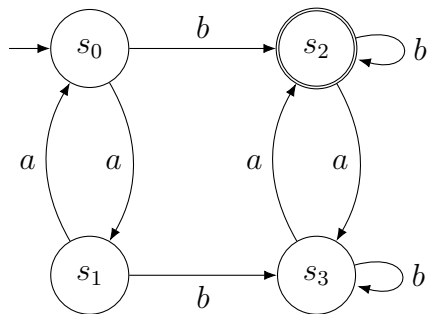
Solution: None

- A 13.** (10 pts) Construct a finite-state automaton A with input alphabet $\{a, b\}$ that recognises the set of all strings that have an even number of occurrences of a and at least one occurrence of b . That is, A must satisfy

$$L(A) = \{w \mid |w|_a \text{ even} \wedge |w|_b \geq 1\}$$

Describe the automaton A using a next-state table *or* a transition diagram.

Solution:



Definitions and theorems

Logic

The truth table for a number of logical operators is given below.

p	q	$\sim p$	$p \vee q$	$p \wedge q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	F	T	T	T	T
T	F	F	T	F	F	F
F	T	T	T	F	T	F
F	F	T	F	F	T	T

A compound proposition is called a *tautology* if it is always true no matter what the truth values of the propositional variables are. A compound proposition that is always false is called a *contradiction*.

The compound propositions p and q are called *logically equivalent* if $p \leftrightarrow q$ is a tautology. The notation $p \equiv q$ denotes that p and q are logically equivalent.

Given any propositional variables p, q, r , a tautology **t** and a contradiction **c**, the following logical equivalences hold:

- Commutative laws: $p \wedge q \equiv q \wedge p$ $p \vee q \equiv q \vee p$
- Associative laws: $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$ $(p \vee q) \vee r \equiv p \vee (q \vee r)$
- Distributive laws: $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$ $p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$
- Identity laws: $p \wedge \mathbf{t} \equiv p$ $p \vee \mathbf{c} \equiv p$
- Negation laws: $p \vee \sim p \equiv \mathbf{t}$ $p \wedge \sim p \equiv \mathbf{c}$
- Double negative law: $\sim(\sim p) \equiv p$
- Idempotent laws: $p \wedge p \equiv p$ $p \vee p \equiv p$
- Universal bound laws: $p \vee \mathbf{t} \equiv \mathbf{t}$ $p \wedge \mathbf{c} \equiv \mathbf{c}$
- De Morgan's laws: $\sim(p \wedge q) \equiv \sim p \vee \sim q$ $\sim(p \vee q) \equiv \sim p \wedge \sim q$
- Absorption laws: $p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$
- Negations of **t** & **c**: $\sim \mathbf{t} \equiv \mathbf{c}$ $\sim \mathbf{c} \equiv \mathbf{t}$
- Conditional: $\sim p \vee q \equiv p \rightarrow q$ $p \rightarrow q \equiv \sim q \rightarrow \sim p$
- Biconditional: $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$ $p \leftrightarrow q \equiv \sim p \leftrightarrow \sim q$

Sets

A *set* is an (unordered) collection of objects, called *elements* or *members*. We write $a \in A$ to indicate that a is an *element of* A .

A set A is a *subset* of a set B , written $A \subseteq B$, if for all a , $a \in A$ implies $a \in B$.

Two sets A and B are equal if $A \subseteq B$ and $B \subseteq A$.

The *power set* of a set A , denoted $\mathcal{P}(A)$, is the set of all subsets of A . That is, $\mathcal{P}(A) = \{B \mid B \subseteq A\}$.

The *union* of two sets A and B is the set $A \cup B = \{x \mid x \in A \vee x \in B\}$.

The *intersection* of A and B is the set $A \cap B = \{x \mid x \in A \wedge x \in B\}$.

The *difference* of two sets A and B is the set $A - B = \{x \mid x \in A \wedge x \notin B\}$.

The *complement* of a set A is the set $A^c = \{x \in \mathcal{U} \mid x \notin A\}$, where \mathcal{U} is the *universal set*.

The *Cartesian product* of A and B is the set $A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$. That is, $A \times B$ is the set of all pairs (a, b) with $a \in A$ and $b \in B$.

Functions

Given two sets A and B , a *function* f from A to B (written $f : A \rightarrow B$) is an assignment of exactly one element of B to each element of A .

A function $f : A \rightarrow B$ is *onto* if for every element $b \in B$ there is an element $a \in A$ such that $f(a) = b$.

A function $f : A \rightarrow B$ is *one-to-one* if $f(a) = f(b)$ implies $a = b$ for all a and b in the domain of f .

A function f is a *one-to-one correspondence* if it is both one-to-one and onto.

Given two functions $g : A \rightarrow B$ and $f : B \rightarrow C$, the *composition* of f and g (written $f \circ g : A \rightarrow C$) is given by

$$(f \circ g)(a) = f(g(a)) \quad \text{for all } a \in A$$

Let $f : X \rightarrow Y$. If $f(x) = y$ we say that y is the *image* of x and x is an *inverse image* (or *preimage*) of y . The *range* (or *image*) of f is the set of all values of f , i.e.

$$\text{range}(f) = \{y \in Y \mid \exists x \in X. f(x) = y\}$$

The *inverse image* of an element $y \in Y$ is the set of all inverse images of y , i.e.

$$\text{inverse image of } y = \{x \in X \mid f(x) = y\}$$

Relations

Let A and B be sets. A *binary relation* from A to B is a subset of $A \times B$.

A *relation on a set* A is a binary relation from A to A .

A relation R on a set A is called *reflexive* if $(a, a) \in R$ for every element $a \in A$.

A relation R on a set A is called *symmetric* if $(b, a) \in R$ whenever $(a, b) \in R$, for all $a, b \in A$.

A relation R on a set A is called *antisymmetric* if whenever $(a, b) \in R$ and $(b, a) \in R$, then $a = b$, for all $a, b \in A$.

A relation R on a set A is called *transitive* if whenever $(a, b) \in R$ and $(b, c) \in R$, then $(a, c) \in R$, for all $a, b, c \in A$.

The *transitive closure* of a relation R is the smallest transitive relation T such that $R \subseteq T$.

A relation is an *equivalence relation* if it is reflexive, symmetric and transitive.

Let R be an equivalence relation on a set A . The *equivalence class* of $a \in A$ is the set $[a]_R = \{x \in A \mid x R a\}$.

A relation is a *partial order relation* if it is reflexive, antisymmetric and transitive.

A *partially ordered set* is a pair (A, R) consisting of a set A and a partial order relation R on A .

Elements a and b of a partially ordered set (A, \preceq) are said to be *comparable* if $a \preceq b$ or $b \preceq a$. Otherwise, a and b are said to be *incomparable*.

Let (A, \preceq) be a partially ordered set. An element $a \in A$ is called

- a *greatest element* of A , if $x \preceq a$ for all $x \in A$
- a *least element* of A , if $a \preceq x$ for all $x \in A$
- a *maximal element* of A if, for all $x \in A$, either $x \preceq a$ or a and x are incomparable
- a *minimal element* of A if, for all $x \in A$, either $a \preceq x$ or a and x are incomparable

Number Theory

Given two integers a and b , with $a \neq 0$, we say that a *divides* b (written $a \mid b$) if there exist an integer c such that $b = ac$, or equivalently, if $\frac{b}{a}$ is an integer.

Theorem 1. *Let a , b , and c be any integers.*

1. *If $a \mid b$ and $a \mid c$ then $a \mid (b + c)$.*

2. If $a \mid b$ then $a \mid bc$.

3. If $a \mid b$ and $b \mid c$ then $a \mid c$.

Theorem 2 (The Quotient/Remainder Theorem). *Given any integer a and a positive integer d , there exist unique integers q and r such that*

$$a = dq + r \quad \text{and } 0 \leq r < d$$

In Theorem 2 the value d is called the *divisor*, a is the *dividend*, q is the *quotient*, and r is the *remainder*. Then div and mod are defined as $a \text{ div } d = q$, $a \text{ mod } d = r$. Remember that the remainder cannot be negative.

Let n and m be any integers and d a positive integer. We write $n \equiv m \pmod{d}$ if

$$(n \text{ mod } d) = (m \text{ mod } d)$$

Theorem 3. *Let n and m be any integers and d a positive integer then*

$$n \equiv m \pmod{d} \quad \text{if and only if} \quad d \mid (n - m)$$

Let n be a nonnegative integer and $b_r b_{r-1} \dots b_0$ a finite sequence of binary digits, i.e. $b_i \in \{0, 1\}$ for $0 \leq i \leq r$. The sequence $b_r b_{r-1} \dots b_0$ is a binary representation of n if

$$n = b_r \cdot 2^r + b_{r-1} 2^{r-1} + \dots + b_1 \cdot 2 + b_0$$

For example 101101 is the binary representation of $1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2 + 1 = 45$

The *greatest common divisor* of two integers a and b , not both zero, is denoted by $\text{gcd}(a, b)$ and is the largest integer that both divides a and divides b .

The *least common multiple* of two positive integers a and b is denoted by $\text{lcm}(a, b)$ and is the smallest positive integer c such that $a \mid c$ and $b \mid c$.

Theorem 4. *If a and b integer, not both zero, then*

$$\text{gcd}(a, b) = \text{gcd}(b, a \text{ mod } b)$$

Counting

The table below states how many different ways there are to order n distinct objects, and how many ways there are to choose r objects out of n distinct objects (depending on whether the order matters or repetition is allowed).

Order n objects (permutation)	$P(n) = n! = n \cdot (n - 1) \dots 2 \cdot 1$	
<hr/>		
Choose r objects from n	without repetition	with repetition
- order matters (r -permutation)	$P(n, r) = \frac{n!}{(n-r)!}$	n^r
- order doesn't matter (r -combination)	$\binom{n}{r} = \frac{n!}{r! (n-r)!}$	$\binom{n+r-1}{r}$

Theorem 5 (Binomial Theorem). *Given real numbers a and b , and non-negative integer n ,*

$$(a + b)^n = \sum_{k=0}^n \binom{n}{k} a^{n-k} b^k$$

Let A be a finite set. We write $N(A)$ to denote the number of elements in A . We say A has $N(A)$ elements or A is a set of size $N(A)$

Theorem 6. *If A and B are finite sets, then*

$$N(A \cup B) = N(A) + N(B) - N(A \cap B)$$

Theorem 7 (Difference Rule). *If A is a finite set and B is a subset of A , then*

$$N(A - B) = N(A) - N(B).$$

Probability

A *sample space* is the set of all possible outcomes of a random process or experiment. An *event* is a subset of a sample space. The probability of an event E is denoted $P(E)$. If S is a finite sample space in which all outcomes are equally likely and E is an event in S , then the probability of E is

$$P(E) = \frac{N(E)}{N(S)}$$

Let S be a sample space. A *probability function* P is a function from the set of all events in S to the set of real numbers satisfying

1. $0 \leq P(A) \leq 1$ for all events A in S .
2. $P(\emptyset) = 0$ and $P(S) = 1$.
3. If A and B are disjoint events in S , i.e. $A \cap B = \emptyset$, then $P(A \cup B) = P(A) + P(B)$.

Theorem 8. *If A is any event in a sample space S then the probability of the complement event $A^c = S - A$ is*

$$P(A^c) = 1 - P(A)$$

Theorem 9. *If A and B are events in a sample space S then*

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

Suppose the sample space of an experiment or a random process is given by $S = \{a_1, \dots, a_n\}$ where a_i is a real number for all $1 \leq i \leq n$. Suppose that each a_i occur with probability p_i for $1 \leq i \leq n$. The *expected value* of the process is

$$\sum_{i=1}^{i=n} a_i p_i = a_1 p_1 + a_2 p_2 + \dots + a_n p_n$$

Let A and B be events in a sample space S . If $P(A) \neq 0$, then the *conditional probability* of B given A , denoted $P(B|A)$, is

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

Theorem 10 (Bayes' Theorem). *Suppose that a sample space S is the union of mutually disjoint events B_1, \dots, B_n where $P(B_i) \neq 0$ for all $1 \leq i \leq n$. Suppose that A is an event in S and that $P(A) \neq 0$. Then*

$$P(B_k|A) = \frac{P(A|B_k)P(B_k)}{P(A|B_1)P(B_1) + P(A|B_2)P(B_2) + \dots + P(A|B_n)P(B_n)}$$

Graphs & Trees

A *graph* is a triple (V, E, f) consisting of: a finite, nonempty set V of vertices; a finite set E of edges; and a function $f : E \rightarrow \{\{x, y\} \mid x, y \in V\}$.

An edge with only one endpoint is called a *loop*. Two edges with the same set of endpoints are called *parallel*. Vertices are called *adjacent* if they are endpoints of the same edge (if the edge is a loop, its endpoint is called *adjacent to itself*). We say that an edge is *incident on* its endpoints. Two edges with a common endpoint are called *adjacent*. A vertex on which no edge is incident is called *isolated*.

A graph is called *simple* if it has no loops or parallel edges.

Let G be a graph and v a vertex of G . The *degree* of v , denoted $\deg(v)$, is the number of edges that are incident on v , with an edge that is a loop counted twice. The *total degree* of G is the sum of the degrees of all vertices in G .

Theorem 11 (Handshake Theorem). *A graph with m edges has total degree $2m$.*

The table below gives an overview over the different kinds of walks in a graph:

	repeated edge	repeated vertex	same start & end vertex	must contain at least 1 edge
Walk	allowed	allowed	allowed	no
Trail	no	allowed	allowed	no
Path	no	no	no	no
Closed walk	allowed	allowed	yes	no
Circuit	no	allowed	yes	yes
Simple Circuit	no	first and last only	yes	yes

A *subgraph* of G is a graph (V', E', f') with $V' \subseteq V$, $E' \subseteq E$ and $f'(e) = f(e)$ for all $e \in E'$.

A graph (V, E, f) is *connected* if there exists a walk from u to v for all $u, v \in V$. A *connected component* of a graph G is a maximal connected subgraph C of G (i.e. every connected subgraph of G is either a subgraph of C or has no common vertices with C).

An *Euler circuit* of a graph G is a circuit that contains every edge and every vertex of G .

An *Euler trail* of a graph G is a trail that contains every edge and every vertex of G .

Theorem 12. *A graph has an Euler circuit if and only if it is connected and every vertex has positive even degree.*

Theorem 13. *Given two distinct vertices u and v in a graph G , there exists an Euler trail from u to v if and only if G is connected, u and v have odd degree, and all other vertices have positive even degree.*

A graph is called *circuit-free* if it has no circuits. A *tree* is a connected and circuit-free graph. In a tree, a vertex is called a *leaf* if it has degree 1, and an *internal vertex* if it has degree 2 or greater. A tree which consists of only one vertex is called *trivial*.

Theorem 14. *Let n be a positive integer. A tree with n vertices has $n - 1$ edges.*

A *rooted tree* is a tree in which one vertex is distinguished from the others and is called the *root*. The *level* of a vertex in a rooted tree is the number of edges on the unique path from that vertex to the root. Given two adjacent vertices v and w such that the level of w is one greater than the level of v , then we call w a *child* of v and v the *parent* of w . In

a rooted tree, a vertex with one or more children is called an *internal vertex*, and a vertex with no children is called a *leaf*. A *binary tree* is a rooted tree in which every parent has at most two children. A *full binary tree* is a binary tree in which each parent has *exactly* two children.

Theorem 15. *A full binary tree with n internal vertices has $n + 1$ leaves.*

Automata, Regular Expressions, Grammars

An *alphabet* Σ is a finite set of *symbols*.

A *string* (or *word*) w over Σ is a finite sequence of symbols from Σ .

A *language* A is a set of strings over some Σ .

λ is the *empty string*.

The *length* of w , written $|w|$, is the number of symbols that w contains. We write $|w|_a$ for the number of times the symbol a occurs in w .

Given words $w_1 = a_1a_2 \dots a_m$ and $w_2 = b_1b_2 \dots b_n$, we write w_1w_2 to mean their *concatenation*: $a_1a_2 \dots a_mb_1b_2 \dots b_n$.

The concatenation of two languages L_1 and L_2 is defined as the language

$$L_1L_2 = \{w_1w_2 \mid w_1 \in L_1 \text{ and } w_2 \in L_2\}$$

The *Kleene closure* of a language L is the language $L^* = \{\lambda\} \cup L \cup LL \cup LLL \cup \dots$

A *finite-state automaton* is a 5-tuple (S, I, N, s_0, F) consisting of: a finite set of states S ; a finite set I (the input alphabet); a transition function $N : S \times I \rightarrow S$; an initial state s_0 ; and a set $F \subseteq S$ of accepting states.

The language accepted by an automaton A is denoted by $L(A)$:

$$L(A) = \{w \mid w \in I^* \text{ and } N^*(s_0, w) \in F\}$$

where $N^* : S \times I^* \rightarrow S$ is defined by

$$\begin{aligned} N^*(s, \lambda) &= s \\ N^*(s, aw) &= N^*(s', w) \quad \text{where } s' = N(s, a) \end{aligned}$$

A *regular expression* r over an alphabet Σ can be built using: \emptyset ; λ ; a symbol $a \in \Sigma$; concatenation rs and union $r \mid s$ of regular expressions r, s ; and Kleene closure r^* of a regular expression r .

The language $L(r)$ defined by a regular expression r is

$$\begin{array}{lll} L(\emptyset) = \emptyset & L(a) = \{a\} & L(r \mid t) = L(r) \cup L(t) \\ L(\lambda) = \{\lambda\} & L(rt) = L(r)L(t) & L(r^*) = L(r)^* \end{array}$$

A *grammar* $G = (V, T, S, P)$ consists of: A set of symbols V called *vocabulary*; a set $T \subset V$ of terminal symbols (symbols in $N = V - T$ are called *non-terminal*); a starting symbol $S \in V$; and a set P of productions of the form $z_0 \rightarrow z_1$, where $z_0, z_1 \in V^*$ and z_0 must contain at least one non-terminal symbol.

Given a grammar G and two strings $s, t \in V^*$, we write $s \Rightarrow t$, if s can be rewritten to t by applying one production from P . We write $s \xRightarrow{*} t$, and say that t is derivable from s , if s can be rewritten to t by applying several productions from P . The language defined by the grammar G is the set of strings over T derivable from S :

$$L(G) = \left\{ w \mid w \in T^* \text{ and } S \xRightarrow{*} w \right\}$$