Discrete Mathematics

Computer Science Tripos – Part IA

Comprehensive Tripos Revision Guide

University of Cambridge – Computer Science Tripos, Paper 1 & Paper 2

Part	Topics	Key Skills
I	Proofs & Logic	Implication, quantifiers, contradiction, contrapositive
II	Numbers	Division, gcd, Euclid, modular arithmetic, induction
III	Sets & Relations	Relations, functions, cardinality, Cantor, well-foundedness
IV	Languages & Automata	Regular expressions, NFAs/DFAs, Kleene, pumping lemma

How to use this guide

Each section gives the **definitions**, **key theorems with proof sketches**, **worked examples**, and **common pitfalls** examiners exploit. Proofs you should be able to reproduce *verbatim* under exam conditions are marked **[EXAMINABLE PROOF]**. Tripos-style practice questions are interleaved at the end of each part – attempt them *before* reading the hints.

0 Contents

0 PART I: PROOFS

Mathematical statements – Quantifiers – Proof strategies – Contradiction

1 The Language of Mathematics

1.1 Statements and jargon

A **statement** is a sentence that is either true or false. A **predicate** is a statement depending on free variables, e.g. $P(n) :\equiv$ "n is even". The hierarchy of results: *axiom* (assumed), *definition* (a name for a concept), *theorem* (a major proved result), *proposition* (a proved result of lesser importance), *lemma* (a result proved chiefly to support another), *corollary* (an easy consequence of a previous result), *conjecture* (a statement believed but not yet proved).

Why precision matters

"The product of two odd integers is odd" hides a universally quantified implication: *For all integers m, n, if m and n are odd then mn is odd.* Tripos credit is awarded for making such structure explicit, not just for getting the right answer.

1.2 Logical connectives – proof patterns

Connective	To prove	Standard strategy
$P \Rightarrow Q$	assume P, show Q	*direct proof*
$P \Rightarrow Q$	assume $\neg Q$, show $\neg P$	*proof by contrapositive*
$P \Leftrightarrow Q$		prove $P \Rightarrow Q$ and $Q \Rightarrow P$ separately
$\forall x.\, P(x)$	fix arbitrary x, show $P(x)$	*universal generalisation*
$\exists x.\, P(x)$	exhibit a witness x_0, show $P(x_0)$	*constructive existence*
$P \wedge Q$		prove P and prove Q
$P \vee Q$	assume $\neg P$, show Q	*disjunction elimination*
$\neg P$	assume P, derive a contradiction	*proof by contradiction*

De Morgan's Laws for Quantifiers

$$\neg(\forall x.\, P(x)) \iff \exists x.\, \neg P(x), \qquad \neg(\exists x.\, P(x)) \iff \forall x.\, \neg P(x)$$
$$\neg(P \wedge Q) \iff \neg P \vee \neg Q, \qquad \neg(P \vee Q) \iff \neg P \wedge \neg Q$$

Trap: existential witnesses must be checked

When proving $\exists x.\, P(x)$, you must *verify* that your witness satisfies P, not merely assert it does. Many lost marks come from skipping this verification step.

1.3 Proof by contradiction

To prove P, assume $\neg P$ and derive a statement known to be false (often $0 = 1$ or $Q \wedge \neg Q$ for some Q).

Example 1.1 (Irrationality of $\sqrt{2}$)**.** *Suppose for contradiction $\sqrt{2} = p/q$ in lowest terms ($\gcd(p, q) = 1$). Then $p^2 = 2q^2$, so p is even, say $p = 2k$; then $4k^2 = 2q^2$ so $q^2 = 2k^2$, so q is also even – contradicting* $\gcd(p, q) = 1$.

Infinitude of primes (Euclid) – [EXAMINABLE PROOF]

Suppose for contradiction there are finitely many primes $p_1, \ldots, p_n$. Let $N = p_1 p_2 \cdots p_n + 1$. Then $N > 1$ so N has a prime factor p. But $p \neq p_i$ for any i (else $p \mid N$ and $p \mid p_1 \cdots p_n$ forces $p \mid 1$, impossible). So p is a prime not in our list – contradiction.

2 Divisibility and Congruences

Definition 2.1 (Divisibility)**.** *For $a, b \in \mathbb{Z}$, $a \mid b$ ("a divides b") iff $\exists k \in \mathbb{Z}.\ b = ak$.*

Properties of divisibility

- $a \mid b \wedge a \mid c \implies a \mid (bx + cy)$ for all $x, y \in \mathbb{Z}$ (linearity)
- $a \mid b \wedge b \mid c \implies a \mid c$ (transitivity)
- $a \mid b \wedge b \mid a \implies a = \pm b$ (antisymmetry up to sign)

The Division Theorem

For $a \in \mathbb{Z}, b \in \mathbb{Z}_{>0}$, there exist *unique* $q, r \in \mathbb{Z}$ with

$$a = bq + r, \qquad 0 \leq r < b.$$

q is the *quotient*, r is the *remainder*, written $r = a \bmod b$.

Definition 2.2 (Congruence)**.** *$a \equiv b \pmod{n}$ iff $n \mid (a - b)$. This is an equivalence relation on $\mathbb{Z}$ (reflexive, symmetric, transitive), and is compatible with $+$ and $\times$:*

$$a \equiv b,\ c \equiv d \pmod{n} \implies a + c \equiv b + d,\ ac \equiv bd \pmod{n}.$$

Trap: cancellation needs coprimality

$ac \equiv bc \pmod{n}$ does *not* imply $a \equiv b \pmod{n}$ unless $\gcd(c, n) = 1$. E.g. $2 \cdot 3 \equiv 2 \cdot 0 \pmod{6}$ but $3 \not\equiv 0 \pmod{6}$.

2.1 Greatest common divisor and Euclid's algorithm

Definition 2.3 (GCD)**.** *$\gcd(a, b)$ is the largest d with $d \mid a$ and $d \mid b$ (taking $\gcd(0, 0) = 0$).*

Euclid's Algorithm

Key recursive fact: $\gcd(a, b) = \gcd(b, a \bmod b)$. Iterate until remainder 0; the gcd is the last nonzero remainder.

$$\gcd(a, 0) = a, \qquad \gcd(a, b) = \gcd(b,\ a \bmod b) \quad (b \neq 0)$$

Termination: the second argument strictly decreases each step and is a natural number, so by well-foundedness of $<$ on $\mathbb{N}$ the algorithm halts.

Extended Euclidean Algorithm & Bézout's Identity

Running Euclid's algorithm "backwards" produces $x, y \in \mathbb{Z}$ such that

$$ax + by = \gcd(a, b).$$

Practical method: keep track of the pairs (s_i, t_i) such that $r_i = s_i a + t_i b$ alongside each remainder r_i in the Euclidean algorithm, using the same recurrence as the r_i.

Example 2.4 (Worked extended Euclid). *Find* $\gcd(99,78)$ *and integers* x, y *with* $99x+78y = \gcd(99,78)$.

$$99 = 1\cdot 78 + 21, \quad 78 = 3\cdot 21 + 15, \quad 21 = 1\cdot 15 + 6, \quad 15 = 2\cdot 6 + 3, \quad 6 = 2\cdot 3 + 0$$

So $\gcd = 3$. *Back-substitute:* $3 = 15 - 2\cdot 6 = 15 - 2(21-15) = 3\cdot 15 - 2\cdot 21 = 3(78 - 3\cdot 21) - 2\cdot 21 = 3\cdot 78 - 11\cdot 21 = 3\cdot 78 - 11(99-78) = 14\cdot 78 - 11\cdot 99$. *So* $x = -11,\ y = 14$.

Coprimality results – [EXAMINABLE PROOFS]

Euclid's Lemma: if p is prime and $p \mid ab$ then $p \mid a$ or $p \mid b$. *Proof:* if $p \nmid a$ then $\gcd(p,a) = 1$ (as p is prime), so by Bézout $\exists x, y.\ px + ay = 1$. Multiply by b: $pbx + aby = b$. Since $p \mid pbx$ and $p \mid aby$ (as $p \mid ab$), we get $p \mid b$.
Multiplicativity: if $\gcd(b,c) = 1$ then $\gcd(a,bc) = \gcd(a,b)\cdot\gcd(a,c)$ (a common 2022 Tripos question – prove directly from Bézout without invoking unique factorisation).

2.2 Modular arithmetic and Fermat's Little Theorem

Fermat's Little Theorem – [EXAMINABLE PROOF]

If p is prime and $p \nmid a$, then $a^{p-1} \equiv 1 \pmod p$.
Proof. Consider $a, 2a, \ldots, (p-1)a \pmod p$. Claim these are a permutation of $1, \ldots, p-1 \pmod p$: they are nonzero mod p (since $p \nmid a$, $p \nmid i$ for $1 \le i \le p-1$, and p prime), and distinct (if $ia \equiv ja$ then $p \mid (i-j)a$, and since $p \nmid a$, Euclid's Lemma gives $p \mid (i-j)$, so $i = j$ as $|i-j| < p$). Hence

$$\prod_{i=1}^{p-1}(ia) \equiv \prod_{i=1}^{p-1} i \pmod p \implies a^{p-1}(p-1)! \equiv (p-1)! \pmod p.$$

Since $\gcd((p-1)!, p) = 1$, cancel to get $a^{p-1} \equiv 1 \pmod p$. ■

Application: modular exponentiation & RSA/Diffie–Hellman

FLT underlies cryptographic protocols. Diffie–Hellman key exchange: both parties agree a prime p and generator g; each picks a secret a (resp. b), exchanges $g^a \bmod p$ (resp. $g^b \bmod p$) publicly, and computes the shared secret $g^{ab} \bmod p = (g^b)^a = (g^a)^b$. Security relies on the hardness of the discrete logarithm problem.

2.3 The Principle of Induction

Three forms of induction

Simple induction: to prove $\forall n \in \mathbb{N}.\, P(n)$: show $P(0)$ (base case) and $\forall n.\, P(n) \Rightarrow P(n+1)$ (step).
Induction from a basis k: show $P(k)$ and $\forall n \ge k.\, P(n) \Rightarrow P(n+1)$; concludes $\forall n \ge k.\, P(n)$.
Strong induction: show $\forall n.\, \big[(\forall m < n.\, P(m)) \Rightarrow P(n)\big]$; concludes $\forall n.\, P(n)$. (Note: the base case $P(0)$ is the vacuous instance, since there is no $m < 0$.)

Trap: stating the right inductive hypothesis

A frequent Tripos error is invoking $P(n)$ when proving $P(n+1)$ but actually needing $P(n-1)$ too (e.g. Fibonacci-style recurrences) – this requires *strong* induction, or induction on *two* preceding values explicitly stated as the hypothesis.

Example 2.5 (Sum identity by induction). $\sum_{i=0}^{n-1} i = \binom{n}{2}$. *Base* $n = 0$*: empty sum* $= 0 = \binom{0}{2}$. *Step: assume true for* n*; then* $\sum_{i=0}^{n} i = \binom{n}{2} + n = \frac{n(n-1)}{2} + n = \frac{n(n+1)}{2} = \binom{n+1}{2}$.

Practice – Part I

1995/P1/Q1. Use $(1+\alpha)^r(1+\alpha)^s = (1+\alpha)^{r+s}$ and generating functions to derive Vandermonde's convolution $\sum_{k\geq 0}\binom{r}{k}\binom{s}{r-k} = \binom{r+s}{r}$; reprove combinatorially by counting ways to choose r people from r men and s women.

2022/P2/Q8(a). Without using the Fundamental Theorem of Arithmetic, prove that for positive integers a, b, c, if $\gcd(b, c) = 1$ then $\gcd(a, bc) = \gcd(a, b)\gcd(a, c)$. *Hint: Bézout both factors and multiply the identities.*

Generic practice. Prove by induction that $n^3 - n$ is divisible by 6 for all $n \in \mathbb{N}$; prove $2^n > n^2$ for $n \geq 5$ by simple induction; prove every $n \geq 2$ has a prime factor by strong induction.

2 PART II: SETS, RELATIONS, FUNCTIONS

Relations – matrices – preorders – functions – cardinality – Cantor's theorem

3 Sets and Basic Constructions

Core set vocabulary

Extensionality: $A = B \iff \forall x.\,(x \in A \Leftrightarrow x \in B)$. **Subset:** $A \subseteq B \iff \forall x \in A.\, x \in B$. **Powerset:** $\mathcal{P}(A) = \{S \mid S \subseteq A\}$, and $\#\mathcal{P}(A) = 2^{\#A}$ for finite A. **Russell's Paradox:** the "set" $R = \{x \mid x \notin x\}$ leads to $R \in R \iff R \notin R$ – contradiction; this is why naive unrestricted comprehension is unsound, motivating restricted *separation* $\{x \in A \mid \phi(x)\}$.

Boolean algebra of $\mathcal{P}(A)$

$(\mathcal{P}(A), \cup, \cap, {}^c, \emptyset, A)$ satisfies commutativity, associativity, distributivity, De Morgan's laws ($\overline{X \cup Y} = \bar{X} \cap \bar{Y}$), and idempotence, mirroring propositional logic under the dictionary $\cup \leftrightarrow \vee$, $\cap \leftrightarrow \wedge$, ${}^c \leftrightarrow \neg$.

4 Relations

Definition 4.1 (Relation)**.** *A (binary) relation R from A to B, written $R : A \not\to B$, is a subset $R \subseteq A \times B$; we write aRb for $(a, b) \in R$.* ***Relational extensionality:*** $R = S \iff \forall a, b.\,(aRb \Leftrightarrow aSb)$.

Named relations

Empty $\emptyset : A \not\to B$ relates nothing. **Full/total** $A \times B$ relates everything. **Identity** $\mathrm{id}_A : A \not\to A$, $a\,\mathrm{id}_A\,a' \iff a = a'$.

4.1 Relational composition

Definition 4.2 (Composite)**.** *For $R : A \not\to B$, $S : B \not\to C$:*

$$S \circ R = \{(a, c) \mid \exists b \in B.\ aRb \wedge bSc\}.$$

Composition is a monoid – [EXAMINABLE PROOF]

Composition is **associative**: $(T \circ S) \circ R = T \circ (S \circ R)$, proved by unfolding both sides to $\exists b, c.\, aRb \wedge bSc \wedge cTd$ and using associativity of $\exists/\wedge$.
Identity is neutral: $R \circ \mathrm{id}_A = R = \mathrm{id}_B \circ R$.
Consequently $(\mathrm{Rel}(A, A), \mathrm{id}_A, \circ)$ – the **endo-relations** on A – form a **monoid**.

4.2 Relations as matrices

A relation $R : [m] \not\to [n]$ over finite sets corresponds to an $(m \times n)$ **boolean matrix** M with $M_{i,j} = 1 \iff iRj$. This correspondence (Lemma: $\mathrm{mat}_{\mathbb{B}}, \mathrm{rel}_{\mathbb{B}}$ are mutually inverse on boolean matrices) turns:

- **relational composition** into **matrix multiplication**: $(N \otimes M)_{i,j} = \bigvee_k (N_{k,j} \wedge M_{i,k})$, i.e. $\mathrm{rel}(N \otimes M) = \mathrm{rel}(N) \circ \mathrm{rel}(M)$;
- **union of relations** into **pointwise matrix addition** (boolean OR): $\mathrm{rel}(M \oplus N) = \mathrm{rel}(M) \cup \mathrm{rel}(N)$.

Algorithmic payoff: reachability via matrix powers

For $R : [n] \not\to [n]$ with adjacency matrix M, define $M_0 = I_n$, $M_{k+1} = I_n \oplus (M \otimes M_k)$. Then $M_n = \mathrm{mat}_{\mathbb{B}}(R^{\circ *})$, the adjacency matrix of the **path relation** (reachability). Intuition: $M_k = \bigoplus_{l \le k} M^l$ accumulates all paths of length $\le k$; since a simple path in an n-vertex graph has

length $< n$, M_n captures all reachability.

4.3 Directed graphs, paths, and closure

Definition 4.3. *A directed graph is (A, R) with $R : A \not\to A$. The n-fold iterate $R^{\circ n}$ is defined by $R^{\circ 0} = \mathrm{id}_A$, $R^{\circ(n+1)} = R \circ R^{\circ n}$. The **path relation** is $R^{\circ *} = \bigcup_{n\in\mathbb{N}} R^{\circ n}$.*

Preorders

$(P, \sqsubseteq)$ is a **preorder** iff $\sqsubseteq$ is **reflexive** ($\forall x.\, x \sqsubseteq x$) and **transitive** ($x \sqsubseteq y \wedge y \sqsubseteq z \Rightarrow x \sqsubseteq z$). A preorder is a **partial order** when additionally **antisymmetric**: $x \sqsubseteq y \wedge y \sqsubseteq x \Rightarrow x = y$.
Examples: $(\mathbb{R}, \leq)$, $(\mathcal{P}(A), \subseteq)$, $(\mathbb{Z}, |)$ (divisibility).

Reflexive-transitive closure – [EXAMINABLE PROOF SKETCH]

Theorem. $R^{\circ *}$ is the *smallest* preorder containing R, i.e. $R^{\circ *} = \bigcap \mathcal{F}_R$ where $\mathcal{F}_R = \{Q \mid R \subseteq Q,\ Q \text{ a preorder}\}$.
Proof idea. ($\bigcap \mathcal{F}_R \subseteq R^{\circ *}$): show $R^{\circ *} \in \mathcal{F}_R$ – it contains R (a path of length 1) and is a preorder (concatenate paths for transitivity, empty path for reflexivity), so it's the smallest element of $\mathcal{F}_R$ by definition of intersection. ($R^{\circ *} \subseteq \bigcap \mathcal{F}_R$): fix any preorder $Q \supseteq R$; show $R^{\circ n} \subseteq Q$ by induction on n (base: reflexivity of Q; step: transitivity of Q glues an R-step onto an $R^{\circ n} \subseteq Q$ path).

5 Functions

Definition 5.1 (Partial function)**.** *$R : A \not\to B$ is **functional** iff $\forall a, b_1, b_2.\ aRb_1 \wedge aRb_2 \Rightarrow b_1 = b_2$. A functional relation is a **partial function** $f : A \rightharpoonup B$; write $f(a)\downarrow$ for "defined at a". f is **total** (a **function** $f : A \to B$) when $\mathrm{dom}(f) = A$.*

Cardinality of function spaces

For finite A, B: $\#(A \rightharpoonup B) = (\#B + 1)^{\#A}$ $\#(A \to B) = (\#B)^{\#A}$.
Reasoning: each of the $\#A$ inputs independently picks one of $\#B$ outputs (function case) or one of $\#B$ outputs *or* "undefined" (partial case).

Closure: identity and partial functions/functions are closed under composition – functionality of $g \circ f$ follows because each stage has at most one output; totality of $g \circ f$ (for total f, g) because every input to f yields some $b = f(a)$, which (by totality of g) yields some $c = g(b)$.

5.1 Sections, retractions, isomorphisms

Definition 5.2. *(s, r) with $s : B \to A$, $r : A \to B$ is a **section-retraction pair** when $r \circ s = \mathrm{id}_B$. A function $f : A \to A$ is **idempotent** when $f \circ f = f$.*

Idempotents $\leftrightarrow$ section-retraction pairs – [EXAMINABLE PROOF]

($\Rightarrow$) If s is a section of r, then $s \circ r$ is idempotent: $(s \circ r)(s \circ r) = s \circ (r \circ s) \circ r = s \circ \mathrm{id}_B \circ r = s \circ r$.
($\Leftarrow$, splitting idempotents) Given idempotent $f : A \to A$, let $B = \mathrm{Im}(f) = \{f(a) \mid a \in A\}$. Define $r : A \to B$ by $r(x) = f(x)$ and $s : B \hookrightarrow A$ the inclusion. Then $r(s(b)) = f(s(b)) = f(f(s(b)))$... more directly, since $b = f(a_0)$ for some a_0, $r(s(b)) = f(b) = f(f(a_0)) = f(a_0) = b$ using idempotence.

Definition 5.3 (Isomorphism)**.** *$f : A \to B$ is an isomorphism iff $\exists g : B \to A$ with $g \circ f = \mathrm{id}_A$ and $f \circ g = \mathrm{id}_B$ (necessarily unique, written f^{-1}). Write $A \cong B$.*

Counting isomorphisms / bijections

$$\#\mathrm{Iso}(A,B) = \begin{cases} 0 & \#A \neq \#B \\ n! & \#A = \#B = n \end{cases}$$

Combinatorial argument: build the bijection by successively assigning images; round k has $n-k$ remaining choices, giving $n! = \prod_{0 \leq k < n}(n-k)$.
$\cong$ is an equivalence relation on sets: **reflexive** (id_A), **symmetric** (invert f), **transitive** (compose isomorphisms – the inverse of $g \circ f$ is $f^{-1} \circ g^{-1}$).

Useful finite isomorphisms

$[m] \times [n] \cong [mn]$ (via $I(i,j) = mj+i$); $[m]+[n] \cong [m+n]$ (lay strips end to end); $\mathcal{P}(X+[1]) \cong \mathcal{P}(X)+\mathcal{P}(X)$ (via indicator functions, see below).

5.2 Indicator functions and comprehension

Definition 5.4. *A **predicate** on A is $\varphi : A \to \{0,1\}$. The **indicator function** of $S \subseteq A$ is $\chi_S(a) = 1$ iff $a \in S$. The **comprehension** of φ is $[\varphi] = \{a \mid \varphi(a) = 1\}$.*

Universal property of indicators

$\chi_{(-)}$ and $[-]$ are mutually inverse, giving $\mathcal{P}(A) \cong (A \to \{0,1\})$. This is the categorical underpinning for treating subsets and boolean-valued functions interchangeably – a powerful proof technique (e.g. deriving $\mathcal{P}(X+[1]) \cong \mathcal{P}(X)+\mathcal{P}(X)$ via maps out of a disjoint union splitting as a pair of maps).

5.3 Equivalence relations and partitions

Definition 5.5. *$E : A \not\to A$ is an **equivalence relation** iff reflexive, symmetric, and transitive. The **equivalence class** of a is $[a]_E = \{x \mid xEa\}$. A **partition** $P \subseteq \mathcal{P}(A)$ has non-empty pairwise-disjoint blocks with $\bigcup P = A$.*

Equivalence relations $\leftrightarrow$ partitions – [EXAMINABLE PROOF]

Theorem. $\Phi : \mathrm{EqRel}(A) \to \mathrm{Part}(A)$, $\Phi(E) = \{[a]_E \mid a \in A\}$ is a bijection, with inverse $\Phi^{-1}(P)$ sending (x,y) to "$\exists b \in P.\ x,y \in b$".
Key step: since P is a partition, every a lies in *exactly one* block of P – no fewer (blocks cover A), no more (blocks disjoint). This single fact drives both halves of the bijection proof.

5.4 Finite and infinite sets

Definition 5.6. *A is **finite** of cardinality n when $A \cong [n]$. A is **infinite** otherwise. The **axiom of infinity** asserts $\mathbb{N}$ is a set (not derivable from the other ZF axioms).*

6 Surjections, Injections, and Counting Infinity

Logical characterisation of isomorphisms – [EXAMINABLE PROOF]

For $f : A \to B$, TFAE: (1) f is invertible; (2) $\forall b.\, \exists! a.\, f(a) = b$; (3) f is both surjective and injective.
Proof sketch ((1)⇒(2)): take $a = f^{-1}(b)$; uniqueness follows by applying f^{-1} to any two preimages.
((2)⇒(1)): define $f^{-1}(b)$ to be the unique witness a; check both composite identities directly from uniqueness.

Definition 6.1. *$f : A \to B$ is **surjective** ($f : A \twoheadrightarrow B$) iff $\forall b \in B.\, \exists a.\, f(a) = b$. f is **injective** ($f : A \hookrightarrow B$) iff $\forall a_1, a_2.\, f(a_1) = f(a_2) \Rightarrow a_1 = a_2$. **Isomorphism = surjective + injective.***

Both classes are closed under identity and composition (standard "chase the witnesses" proofs – practice writing these out, they recur constantly in Tripos).

Counting injections

$$\#\mathrm{Inj}(A,B) = \begin{cases} \binom{\#B}{\#A} \cdot (\#A)! & \#A \le \#B \\ 0 & \text{otherwise} \end{cases}$$

Choose the image subset $U \subseteq B$ ($\binom{\#B}{\#A}$ ways), then a bijection $A \to U$ ($(\#A)!$ ways).

6.1 Enumerability, countability, choice

Definition 6.2. *A is **enumerable** iff $\exists$ surjection $\mathbb{N} \twoheadrightarrow A$. A is **countable** iff empty or enumerable.* ***Axiom of Choice:*** *every surjection has a section.*

Closure properties of countability – key results

- Non-empty subsets of enumerable sets are enumerable (compose with a "default value" patch map).
- A countable $\iff$ $A + [1]$ enumerable.
- Products and finite unions of countable sets are countable. *Proof idea for products:* given enumerations e_A, e_B of A, B and any enumeration f of $\mathbb{N} \times \mathbb{N}$ (zigzag/diagonal), $(e_A \times e_B) \circ f : \mathbb{N} \twoheadrightarrow A \times B$.
- $\mathbb{Z}$ is enumerable via the explicit "zigzag" bijection $e(i) = 0, i/2, -(i+1)/2$ according to parity of i.

7 Images, Diagonalisation, Well-Foundedness

Definition 7.1 (Image factorisation)**.** *Any $f : A \to B$ factors as $f = m_f \circ \tilde{f}$ where $\tilde{f} : A \twoheadrightarrow \mathrm{Im}(f)$ is surjective and $m_f : \mathrm{Im}(f) \hookrightarrow B$ is injective. Every function is "epi followed by mono".*

Definition 7.2 (Relational images)**.** ***Direct image:*** $R_*X = \{b \mid \exists x \in X.\, xRb\}$. ***Inverse image:*** $R^*Y = \{a \mid \forall b.\, aRb \Rightarrow b \in Y\}$.

Cantor's Theorem – [EXAMINABLE PROOF, very high yield]

Theorem. For any set A, there is no surjection $A \twoheadrightarrow \mathcal{P}(A)$.
Proof (diagonalisation). Suppose $e : A \twoheadrightarrow \mathcal{P}(A)$ is surjective. Let $U = \{x \in A \mid x \notin e(x)\} \in \mathcal{P}(A)$. By surjectivity, $\exists a.\, e(a) = U$. Then

$$a \in e(a) \iff a \in U \iff a \notin e(a),$$

a contradiction. ■
Corollary: $\mathbb{R}$ is uncountable, since $\mathbb{R} \cong \mathcal{P}(\mathbb{N})$ (via $\mathbb{R} \cong [0,1] \cong (\mathbb{N} \to \{0,1\}) \cong \mathcal{P}(\mathbb{N})$) and $\mathcal{P}(\mathbb{N})$ admits no surjection from $\mathbb{N}$.

Lawvere's fixed point theorem (the categorical Cantor)

If $\exists$ surjection $A \twoheadrightarrow (A \to X)$, then every $f : X \to X$ has a fixed point. *Proof:* let $e : A \twoheadrightarrow (A \to X)$, define $h(a) = f(e(a)(a))$; surjectivity gives a_h with $e(a_h) = h$, and then $e(a_h)(a_h) = f(e(a_h)(a_h))$ is a fixed point. Cantor's theorem follows by taking $X = \{0,1\}$: a surjection $A \twoheadrightarrow \mathcal{P}(A) \cong (A \to \{0,1\})$ would force every $f : \{0,1\} \to \{0,1\}$ to have a fixed point, but the swap map doesn't – contradiction.

7.1 Well-foundedness and induction

Definition 7.3. *m is a **minimal** element of $S \subseteq A$ under $\prec$ when $\neg\exists x \in S.\, x \prec m$. $\prec$: $A \not\to A$ is **well-founded** when every non-empty subset of A has a minimal element. Equivalently: no infinite descending chain $a_0 \succ a_1 \succ a_2 \succ \cdots$.*

Well-founded induction – [EXAMINABLE PROOF]

Theorem. For well-founded $\prec$ on A, $S = A$ iff $\forall x \in A.\, [(\forall y \prec x.\, y \in S) \Rightarrow x \in S]$.
Proof (hard direction). Suppose the bracketed condition holds; suppose for contradiction $A \setminus S \neq \emptyset$. By well-foundedness it has a minimal element m. Then $\forall y \prec m.\, y \notin A \setminus S$, i.e. $y \in S$. By assumption (instantiated at $x = m$), $m \in S$ – contradicting $m \in A \setminus S$. **Strong induction on** $\mathbb{N}$ is exactly this theorem for $\prec = <$.

Practice – Part II

2022/P2/Q8(c). Given P with projections $p : P \to A$, $q : P \to B$ satisfying the universal property of a product, define maps $P \to A \times B$ and $A \times B \to P$; show $u\langle p, q\rangle = \mathrm{id}_P$; deduce $P \cong A \times B$. *(This is the categorical/universal-property route to products – compare with the concrete pairing construction.)*
Generic practice. Prove Cantor–Bernstein–Schröder (if $\exists$ injections $A \hookrightarrow B$ and $B \hookrightarrow A$ then $A \cong B$) is consistent with your understanding of cardinality without assuming choice. Prove $\mathbb{N} \times \mathbb{N}$ is enumerable by exhibiting an explicit bijection (Cantor pairing $\pi(i, j) = \frac{(i+j)(i+j+1)}{2} + j$) and verify it directly. Practice writing out, from scratch, the proof that the composite of two injections is an injection and the composite of two surjections is a surjection.

7 PART III: FORMAL LANGUAGES & INDUCTIVE DEFINITIONS

Inductively defined functions – rule induction – syntax

8 Inductively Defined Functions

Definition 8.1. *Given $a \in A$, $f : \mathbb{N} \times A \to A$, the function **inductively defined** from a, f is the unique $\rho_{a,f} : \mathbb{N} \to A$ with*

$$\rho_{a,f}(0) = a, \qquad \rho_{a,f}(n+1) = f(n, \rho_{a,f}(n)).$$

*$R : \mathbb{N} \not\to A$ is (a,f)-**closed** when $0Ra$ and $nRx \Rightarrow (n+1)Rf(n,x)$.*

Existence & uniqueness of $\rho_{a,f}$ – [EXAMINABLE PROOF, structure only]

Define $\rho_{a,f}$ as the *intersection of all (a,f)-closed relations.* Three things to show:

1. The intersection is itself (a,f)-closed (intersections preserve closure conditions).
2. It is **functional** – proved by induction on n, using a clever "patch" relation in each case to force uniqueness at stage n from uniqueness at all earlier stages.
3. It is **total** – proved by an easy induction on n using the closure condition directly.

This pattern – *define the desired object as the intersection of everything satisfying the closure rules, then show that intersection is itself closed and is functional/total* – is the master template for all inductive definitions in this course (including the syntax/automata material below).

9 Rule Induction

Definition 9.1 (Syntactic rule)**.** *A rule $\dfrac{u_1 \ \cdots \ u_n}{v}$ over X says: if $u_1, \ldots, u_n$ lie in the subset being defined, so does v (u_i = premises, v = conclusion; no premises = **axiom**). The closure condition it denotes is $\{S \subseteq X \mid u_1, \ldots, u_n \in S \Rightarrow v \in S\}$. For a rule set R, Cl_R is the intersection of all these closure conditions.*

Definition 9.2 (Derivation)**.** *A derivation of u from rules R is a finite tree rooted at u, where each node is the conclusion of a rule whose premises are its children, and leaves are axioms. $X_R \subseteq X$ is the set of elements with a derivation.*

Rule Induction Theorem – [EXAMINABLE PROOF, very high yield]

Theorem. $X_R = \bigcap \mathrm{Cl}_R$. Equivalently: X_R is the *smallest* subset of X closed under all rules in R.

Proof. **(i) $X_R \in \mathrm{Cl}_R$:** given a rule with premises $u_1, \ldots, u_n \in X_R$, each has a derivation; graft them under the rule's conclusion to derive v, so $v \in X_R$. **(ii) $X_R \subseteq S$ for any $S \in \mathrm{Cl}_R$:** induct on the *height* of derivations. Base case (≤ 0): vacuous, no derivation has height 0. Step: a derivation of height $\leq n+1$ with conclusion v has premises derived by sub-trees of height $\leq n$; by IH each premise lies in S; since S is closed under the rule, $v \in S$. ■

The proof recipe for rule induction (memorise this)

To prove a property $P(u)$ for all $u \in X_R$: let $S = \{u \mid P(u)\}$ and show S is closed under *every* rule in R (check axioms hold P, and each rule preserves P from premises to conclusion). Then $X_R \subseteq S$ by the theorem. **This is the single most reusable proof template in the whole course** – it appears in reflexive-transitive closure, regular expression properties, automata correctness, and beyond.

Example 9.3 (Worked rule induction)**.** *Language over $\{a,b\}$ presented by $\dfrac{}{\varepsilon}$, $\dfrac{u}{aua}$, $\dfrac{u}{bub}$, $\dfrac{u \ \ v}{uv}$. Claim: every derivable u has equally many a's and b's. Proof by rule induction: ε has $0 = 0$; if u has n each,*

aua and bub still balance the new pair; if u, v each balance, so does uv (sums add). ■

10 Formal Languages, Regular Expressions, and Parsing

Definition 10.1. *An **alphabet** Σ is a finite set of **symbols**. $\Sigma^\star$ is the set of **strings**; ε is the empty string; concatenation forms a (non-commutative) **monoid** $(\Sigma^\star, \varepsilon, \cdot)$. A **formal language** over Σ is a subset $L \subseteq \Sigma^\star$.*

Regular expressions: abstract syntax

Trees built from $\mathsf{Union}, \mathsf{Concat}$ (2 children), Star (1 child), $\mathsf{Null}, \mathsf{Never}, \mathsf{Sym}_a$ (0 children, for $a \in \Sigma$). Concrete syntax $\varepsilon, \bar{\varnothing}, r \mid s, rs, r^*$ parses to abstract syntax via a **precedence relation** $r \rightsquigarrow R@n$ resolving ambiguity (Union binds loosest, then Concat, then Star, then atoms).

Matching semantics (inductive definition)

$$\frac{}{(a, \mathsf{Sym}_a)} \quad \frac{}{(\varepsilon, \mathsf{Null})} \quad \frac{(u, R)}{(u, \mathsf{Union}(R, S))} \quad \frac{(u, S)}{(u, \mathsf{Union}(R, S))}$$

$$\frac{(v, R) \quad (w, S)}{(vw, \mathsf{Concat}(R, S))} \quad \frac{}{(\varepsilon, \mathsf{Star}(R))} \quad \frac{(u, R) \quad (v, \mathsf{Star}(R))}{(uv, \mathsf{Star}(R))}$$

$L(R) = \{u \mid u \text{ matches } R\}$.

Useful regular expression idioms over $\{a, b\}$

$(a \mid b)^*$ = every string; $b(a \mid b)^*$ = strings starting with b; $((a \mid b)(a \mid b))^*$ = even-length strings; $\varnothing b \mid a$ = matches only the symbol a (since $\varnothing b$ matches nothing).

10 PART IV: FINITE AUTOMATA & KLEENE'S THEOREM

NFA – DFA – subset construction – Kleene's theorem – pumping lemma

11 Finite Automata

Definition 11.1 (NFA). *$M = (Q, \Sigma, \Delta, s, F)$: Q finite states, Σ alphabet, $\Delta : Q \times \Sigma \not\to Q$ transition relation ($q \xrightarrow{a} q'$), $s \in Q$ start state, $F \subseteq Q$ accepting states.*

$$L(M) = \{u \in \Sigma^\star \mid \exists q \in F.\ s \xrightarrow{u*} q\}$$

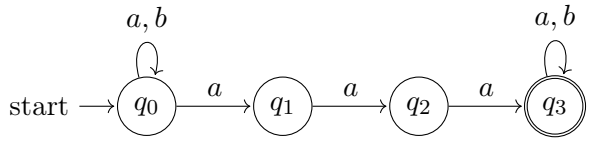
where $q \xrightarrow{\varepsilon*} q' \iff q = q'$ and $q \xrightarrow{au*} q' \iff \exists q''.\, q \xrightarrow{a} q'' \wedge q'' \xrightarrow{u*} q'$.

Definition 11.2 (DFA). *A **DFA** is an NFA where Δ is a total function $\delta : Q \times \Sigma \to Q$ (exactly one transition per state/symbol pair).*

Definition 11.3 (NFAε). *An NFA with an extra ε-transition relation $T : Q \not\to Q$ allowing "free" moves $q \xrightarrow{\varepsilon} q'$ that consume no input. Define $q \Rightarrow^u q'$ inductively:*

$$\frac{}{q \Rightarrow^\varepsilon q} \qquad \frac{q \xrightarrow{a} q'' \quad q'' \Rightarrow^u q'}{q \Rightarrow^{au} q'} \qquad \frac{q \xrightarrow{\varepsilon} q'' \quad q'' \Rightarrow^u q'}{q \Rightarrow^u q'}$$

$L(M) = \{u \mid \exists q \in F.\ s \Rightarrow^u q\}$.



NFA recognising "contains three consecutive a's" over $\{a, b\}$.

12 The Subset Construction

Subset construction $\mathcal{P}(M)$

For NFAε M, define DFA $\mathcal{P}(M)$:

- States: $Q_{\mathcal{P}(M)} = \mathcal{P}(Q_M)$ (subsets of M's states).
- Transition: $\delta_{\mathcal{P}(M)}(S, a) = \{q' \mid \exists q \in S.\ q \Rightarrow^a q'\}$ – a relational direct image.
- Start: $s_{\mathcal{P}(M)} = \{q \mid s_M \Rightarrow^\varepsilon q\}$ (ε-closure of s_M).
- Accepting: $F_{\mathcal{P}(M)} = \{S \mid S \cap F_M \neq \emptyset\}$.

Trap: why accepting sets aren't singletons

A natural-looking alternative definition $F_{\mathcal{P}(M)} = \{\{q\} \mid q \in F_M\}$ **fails**: it breaks completeness already on ε, since the ε-closure of s_M need not be a singleton.

Correctness of subset construction – [EXAMINABLE PROOF]

Soundness ($L(\mathcal{P}(M)) \subseteq L(M)$): given an accepted run $s_{\mathcal{P}(M)} \xrightarrow{a_1} S_1 \to \cdots \to S_n \ni q_n \in F_M$, work *backwards*: at each step find $q_{i-1} \in S_{i-1}$ with $q_{i-1} \Rightarrow^{a_i} q_i$ (by definition of $\delta_{\mathcal{P}(M)}$), chaining to a genuine run $s_M \Rightarrow^u q_n$ in M.
Completeness ($L(M) \subseteq L(\mathcal{P}(M))$): given a run $s_M \Rightarrow^{a_1} q_1 \Rightarrow \cdots \Rightarrow q_n \in F_M$, build S_i *forwards* by $S_{i+1} = \delta_{\mathcal{P}(M)}(S_i, a_{i+1})$ and show by induction $q_i \in S_i$ throughout, so the DFA's run ends in an accepting set.
Together: **NFAε, NFA, DFA all recognise the same languages.**

13 Kleene's Theorem

Definition 13.1. *A language is* ***regular*** *iff* $L = L(R)$ *for some regex* R*;* ***automatical*** *iff* $L = L(M)$ *for some finite automaton* M*.*

Regex ⇒ automaton (regular ⇒ automatical) – [EXAMINABLE PROOFS]

By **rule induction** on regex syntax, show automatical languages are closed under the regex constructors:

- **Base cases:** $\{a\}$ (2-state DFA), $\{\varepsilon\}$ (1-state NFA, no transitions, start = accept), $\emptyset$ (1-state NFA, no accepting states).
- **Union:** new start state with ε-transitions into both M_1, M_2's starts; accept = $F_{M_1} \cup F_{M_2}$.
- **Concatenation:** ε-transition from every accepting state of M_1 to the start of M_2; start = s_{M_1}, accept = F_{M_2}.
- **Star:** new start/accept state q_0 with $\varepsilon \to s_M$, and ε-transitions from every old accepting state back to q_0.

Automaton ⇒ regex (automatical ⇒ regular) – [EXAMINABLE PROOF]

Define a **region** (q, S, q'): strings taking $q \to q'$ passing only through intermediate states in S, with regional language $L_M(q, S, q')$.

Induction on $\#S$. Base ($S = \emptyset$): $L_M(q, \emptyset, q') = \{\varepsilon \mid q = q'\} \cup \bigcup\{\{a\} \mid q \xrightarrow{a} q'\}$ – a finite union, regular. Step: pick $q_0 \in S$, $S' = S \setminus \{q_0\}$; split paths by whether they ever visit q_0:

$$L_M(q, S, q') = L_M(q, S', q') \;\cup\; L_M(q, S', q_0)\, L_M(q_0, S', q_0)^\star\, L_M(q_0, S', q')$$

By the induction hypothesis all components are regular, so their union is regular.