Discrete Mathematics

Summary

This document contains most of the definitions, identities and theorems introduced in the course. It is intended to be a helpful reference for study and revision – however, it is important that you get a good intuitive understanding for the topics, rather than just memorise everything. Intuition will come with time and practice; until then, you can use this sheet, remembering not to rely on it too much because you won't have it in the exam!

Logic and proof theory

Boolean identities

Proof goals can often be simplified using logical identities. Before starting a formal proof, it is always useful to check if the statement can be simplified or rewritten in a way that will ease reasoning. Some of these equivalences form the basis of proof patterns discussed below.

This table exemplifies an important concept in logic and many other fields of mathematics: *duality*. Most of the properties below come in two 'flavours', corresponding to dual operations such as conjunction and disjunction, or universal and existential quantification. Remember that duality is different from negation: 'there do not exist negative primes' ($\neg\exists x.\ P(x)$) is definitely not the same as 'all primes are negative' ($\forall x.\ P(x)$), but it *is* the same as 'all primes are not negative' ($\forall x.\ \neg P(x)$). The precise connection between dual operators and negation is given by the de Morgan laws.

In this table, $P \cong Q$ means that the logical statement P is equivalent to Q – that is, $P \iff Q$ holds.

Identity	$P \wedge \top \cong P$	$P \vee \bot \cong P$
Annihilation	$P \wedge \bot \cong \bot$	$P \vee \top \cong \top$
Idempotence	$P \wedge P \cong P$	$P \vee P \cong P$
Commutativity	$P \wedge Q \cong Q \wedge P$	$P \vee Q \cong Q \vee P$
Associativity	$(P \wedge Q) \wedge R \cong P \wedge (Q \wedge R)$	$(P \vee Q) \vee R \cong P \vee (Q \vee R)$
Distributivity	$(P \vee Q) \wedge R \cong (P \wedge R) \vee (Q \wedge R)$	$(P \wedge Q) \vee R \cong (P \vee R) \wedge (Q \vee R)$
Absorption	$P \wedge (P \vee Q) \cong P$	$P \vee (P \wedge Q) \cong P$
Double negation	$\neg\neg P \cong P$	
Complement	$P \wedge \neg P \cong \bot$	$P \vee \neg P \cong \top$
de Morgan	$\neg(P \wedge Q) \cong \neg P \vee \neg Q$	$\neg(P \vee Q) \cong \neg P \wedge \neg Q$
Material implication	$P \Longrightarrow Q \cong \neg P \vee Q$	$\neg(P \Longrightarrow Q) \cong P \wedge \neg Q$
Contrapositive	$P \Longrightarrow Q \cong \neg Q \Longrightarrow \neg P$	
Boolean assumption	$\top \Longrightarrow P \cong P$	$\bot \Longrightarrow P \cong \top$
Boolean goal	$P \Longrightarrow \top \cong \top$	$P \Longrightarrow \bot \cong \neg P$
Compound assumption	$(P \wedge Q) \Rightarrow R \cong P \Rightarrow (Q \Rightarrow R)$	$(P \vee Q) \Rightarrow R \cong (P \Rightarrow Q) \wedge (Q \Rightarrow R)$
Compound goal	$P \Rightarrow (Q \wedge R) \cong (P \Rightarrow Q) \wedge (P \Rightarrow R)$	$P \Rightarrow (Q \vee R) \cong (P \Rightarrow Q) \vee (P \Rightarrow R)$
Bi-implication	$(P \Leftrightarrow Q) \cong (P \Rightarrow Q) \wedge (Q \Rightarrow P)$	$\neg(P \Leftrightarrow Q) \cong (\neg P) \Leftrightarrow Q$
Quantifier distrib.	$\forall x.\ P(x) \wedge Q(x) \cong (\forall x.\ P(x)) \wedge (\forall x.\ Q(x))$	$\exists x.\ P(x) \vee Q(x) \cong (\exists x.\ P(x)) \vee (\exists x.\ Q(x))$
Univ. scope extension	$(\forall x.\ P(x)) \wedge Q \cong \forall x.\ P(x) \wedge Q$	$(\forall x.\ P(x)) \vee Q \cong \forall x.\ P(x) \vee Q$
Exis. scope extension	$(\exists x.\ P(x)) \wedge Q \cong \exists x.\ P(x) \wedge Q$	$(\exists x.\ P(x)) \vee Q \cong \exists x.\ P(x) \vee Q$
Quantified assumption	$(\forall x.\ P(x)) \Rightarrow Q \cong \exists x.\ P(x) \Rightarrow Q$	$(\exists x.\ P(x)) \Rightarrow Q \cong \forall x.\ P(x) \Rightarrow Q$
Extraction	$\forall x.\ P(x) \cong (\forall x.\ P(x)) \wedge P(a)$	$\exists x.\ P(x) \cong (\exists x.\ P(x)) \vee P(a)$
Generalised de Morgan	$\neg(\forall x.\ P(x)) \cong \exists x.\ \neg P(x)$	$\neg(\exists x.\ P(x)) \cong \forall x.\ \neg P(x)$

Proof patterns

The standard proof techniques used in logic – you can use them systematically to prove most of the logical propositions you encounter.

Proposition: P

- **To assume**: Assume P holds.
- **To prove**:
  - *Direct proof*: Prove the statement directly.
  - *Proof by contradiction*: Assume $\neg P$ and derive a contradiction.

Negation: $\neg P$

- **To assume**: Assume $\neg P$. If P is also an assumption, we have a contradiction.
- **To prove**
  - *Reexpress as positive statement*: Use logical identities (such as the de Morgan laws) to 'push the negation in' and proceed with the proof from there.
  - *Proof by negation*: Assume P and derive a contradiction.

Implication: $P \implies Q$

- **To assume**
  - *Modus ponens*: Assume $P \implies Q$. If P is also an assumption, we can assume Q.
- **To prove**
  - *Deduction*: Assume P and prove Q.
  - *Proof by contrapositive*: Prove $\neg Q \implies \neg P$.

Conjunction: $P \wedge Q$

- **To assume**:
  - *Direct proof*: Assume P and Q independently.
  - *Partial assumption*: Assume P and prove that Q implies the conclusion. Alternatively, assume Q and prove that P implies the conclusion.
- **To prove**: Prove P and Q independently.

Disjunction: $P \vee Q$

- **To assume**
  - *Proof by cases*: In the first case, assume P is true and prove the conclusion. In the second case, assume Q is true and prove the conclusion
  - *Disjunctive syllogism*: Assume $P \vee Q$. If $\neg P$ is also an assumption, we can assume Q. Similarly, if $\neg Q$ is also an assumption, we can assume P.
- **To prove**
  - *Direct proof*: Prove P or Q independently.
  - *Proof by cases*: Find a way to split the proof in two cases, proving, in each case, either P or Q.

  - *Disjunctive syllogism*: Assume $\neg P$ and prove Q. Alternatively, assume $\neg Q$ and prove P. (This follows from a case split on the truth of P: if P is true then $P \vee Q$ is true, so only need to provide a proof for $\neg P \implies Q$.)

Bi-implication: $P \iff Q$

Bi-implication equivalent to $(P \implies Q) \wedge (Q \implies P)$. Correspondingly:

- **To assume**: Assume $P \implies Q$ and $Q \implies P$.
- **To prove**: Prove $P \implies Q$ and $Q \implies P$.

Multiple equivalence: $P_1 \iff P_2 \iff \cdots \iff P_n$

- **To assume**: Assume either P_i and freely convert to a different P_j as needed.
- **To prove**: Prove $P_1 \implies P_2$, $P_2 \implies P_3$, …, $P_{n-1} \implies P_n$ and finally $P_n \implies P_1$.

Universal quantification: $\forall x.\, P(x)$

- **To assume**: Assume $\forall x.\, P(x)$. If we also have a value a in our assumptions, we can *instantiate* the the universal quantification to deduce $P(a)$.
- **To prove**: Assume x is an *arbitrary* value and prove $P(x)$. Ensure that x is a new variable name, i.e. it isn't already used somewhere in the proof.

Existential quantification: $\exists x.\, P(x)$

- **To assume**: Assume there is a *witness* a for which $P(a)$ holds, where a is a new variable name.
- **To prove**: Find a value a for which we can prove that $P(a)$ holds.

Unique existential quantification: $\exists! x.\, P(x)$

Unique existence defined as $\exists x.\, P(x) \wedge (\forall y.\, P(y) \implies y = x)$. Correspondingly:

- **To assume**: Assume there is a *witness* a for which $P(a)$ holds; in addition, assume that this a is *unique*, i.e. if there is any other b for which $P(b)$ holds, a must equal b.
- **To prove**: Prove existence and uniqueness: find a value a for which $P(a)$ holds, and establish that any other b for which $P(b)$ holds must equal a.

Induction

Proof by induction is a powerful, general proof technique for proving properties about elements of possibly infinite sets. Depending on how the set is defined, the induction principle can be presented in different ways, but the basic principle is always the same: prove the property for some base elements of the set, then prove it for the general elements, having assumed it for the 'smaller' elements (this can be made more formal with the notion of a *well-founded relation*). Below are the three most common induction principles used in computer science.

Structural induction

We can use *structural induction* to prove properties about recursively defined structures that can be represented as *trees*: lists, binary trees, expressions, etc. These are usually defined with some recursive *grammar* (e.g. BNF notation) or an *algebraic data type*.

To prove a property $P(t)$ for every tree t of some grammar or ADT, it is enough to:

1. prove $P(l)$ for every non-recursive leaf node l;
2. for every branch node b with subtrees t_i, assuming $P(t_i)$ for subtrees t_i, prove $P(b(t_1, \ldots, t_n))$.

$$[\forall l.\, P(l)] \land [\forall b(t_1, \ldots, t_n).\, P(t_1) \land \cdots \land P(t_n) \Rightarrow P(b(t_1, \ldots, t_n))] \implies \forall t.\, P(t)$$

Rule induction

We can use *rule induction* to prove properties about subsets inductively defined by a collection of axioms and rules:

$$\frac{}{a} \qquad\qquad \frac{h_1 \quad h_2 \quad \cdots \quad h_n}{c}$$

To prove a property $P(e)$ for every element e of an inductively defined set, it is enough to:

1. prove $P(a)$ for every axiom $\overline{a}$;
2. for every rule $\frac{h_1 \cdots h_i}{c}$, assuming $P(h_i)$ for every hypothesis h_i, prove $P(c)$ for the conclusion c.

$$[\forall \overline{a}.\, P(a)] \land \left[\forall \frac{h_1 \cdots h_i}{c}.\, P(h_1) \land \cdots \land P(h_n) \Rightarrow P(c)\right] \implies \forall e.\, P(e)$$

Mathematical induction

We can use *mathematical induction* to prove properties about natural numbers.

To prove a property $P(n)$ for all $n \in \mathbb{N}$, it is enough to:

1. prove $P(0)$;
2. for every $k \in \mathbb{N}$, assuming $P(k)$, prove $P(k+1)$.

$$P(0) \land [\forall k \in \mathbb{N}.\, P(k) \Rightarrow P(k+1)] \implies \forall n \in \mathbb{N}.\, P(n)$$

It is worth noting that mathematical induction is just a special case of structural induction on the data type `type nat = zero | succ of nat`, or rule induction on the set inductively defined by:

$$\frac{}{zero} \qquad\qquad \frac{n}{succ(n)}$$

Strong induction from a basis

Induction hypotheses may be strengthened by assuming the property for every element 'smaller' than the general case, not just the immediate predecessor. In addition, we can explicitly specify the base case if the property only holds for elements above a certain size. Specialised to mathematical induction, we get the following strong induction principle:

To prove a property $P(n)$ for $n \in \mathbb{N}$ with $\ell \leq n$ for some basis $\ell \in \mathbb{N}$, it is enough to:

1. prove $P(\ell)$;
2. for every $k \in \mathbb{N}$, assuming $P(m)$ for all $\ell \leq m \leq k$, prove $P(k+1)$.

Algebra and order theory

Abstract algebra is a branch of mathematics that aims to systematise and abstractly analyse the various structures that are encountered in mathematics. The idea is that by recognising common operations, definitions and properties in different mathematical fields, new theorems and constructions based on the common properties can be applied uniformly to the distinct fields. A good analogy is *polymorphism* encountered in functional and object-oriented programming: an abstract polymorphic function (e.g. sorting) can be applied at any type that supports some property (e.g. that the elements can be compared). While this field is not explicitly covered in the course, you *are* introduced to many of its underlying concepts, and several ideas in the course can be reinterpreted in an abstract algebraic framework. Once you understand how these concepts fit together, there is immediately a lot less to memorise: instead of independently remembering one set of laws for set theory and another for logic, you can memorise the common structure and apply it to the two fields.

Algebraic structures

Abstract algebra focuses on the analysis of *algebraic structures*. An algebraic structure consists of an underlying set (*carrier*) with a collection of *operators* and *distinguished elements* which obey a collection of *laws*. An algebraic structure can be built on top of another one by adding new operations, elements or laws, thereby constructing a hierarchy of algebraic structures. Some examples that appear in the course:

Monoid A set A with a binary operator $\bullet : A \times A \to A$ (monoidal *multiplication*) and element $\varepsilon \in A$ (*unit* or *neutral element*) that obey the following laws:

- Left and right unit: $\forall a \in A.\ a \bullet \varepsilon = a = \varepsilon \bullet a$
- Associativity: $\forall a, b, c \in A.\ a \bullet (b \bullet c) = (a \bullet b) \bullet c$

Commutative monoid A monoid $(A, \bullet, \varepsilon)$ with the additional law of *commutativity* for the multiplication $\bullet$: $\forall a, b \in A.\ a \bullet b = b \bullet a$. For example, $(\mathbb{N}, +, 0)$ and $(\mathbb{N}, \cdot, 1)$ are commutative monoids.

Group A monoid $(A, \bullet, \varepsilon)$ with an additional unary operation $(\cdot)^{-1} : A \to A$ such that for any $a \in A$, a^{-1} is the *inverse* element of a obeying the laws:

$$a \bullet a^{-1} = \varepsilon = a^{-1} \bullet a$$

Commutative (or abelian) group A group $(A, \bullet, \varepsilon, (\cdot)^{-1})$ where the binary operation is commutative. For example, $(\mathbb{Z}, +, 0, -)$ and $(\mathbb{Q}, \times, 1, 1/(\cdot))$ are commutative groups.

Semiring A set A with an *additive* commutative monoidal structure $(A, \oplus, \mathbf{0})$ and a *multiplicative* monoidal structure $(A, \otimes, \mathbf{1})$ where $\otimes$ *distributes* over $\oplus$ on both sides:

$$\forall a, b, c \in A.\ a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c) \qquad (b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$$

and the additive unit serves as an *annihilator* for the multiplicative identity:

$$\forall a \in A.\ a \otimes \mathbf{0} = \mathbf{0} = \mathbf{0} \otimes a$$

If the multiplicative operation is commutative, we have a *commutative semiring*. Examples are $(\mathbb{N}, +, 0, \times, 1)$ and $(\mathbb{B}, \vee, \bot, \wedge, \top)$.

Ring A semiring whose additive commutative monoid is a commutative group. That is, we have an *additive inverse* for every element of A. Again, we have a *commutative ring* if the multiplicative operation is commutative. An example is $(\mathbb{Z}, +, 0, -, \times, 1)$.

Field A commutative ring which has multiplicative inverse for every non-$\mathbf{0}$ element of A. That is, a field $(A, \oplus, \mathbf{0}, \ominus, \otimes, \mathbf{1}, (\cdot)^{-1})$ is a structure where $(A, \oplus, \mathbf{0}, \ominus)$ is an abelian (additive) group and $(A \setminus \{\mathbf{0}\}, \otimes, \mathbf{1}, (\cdot)^{-1})$ is an abelian (multiplicative) group. Examples are $(\mathbb{Q}, +, 0, -, \times, 1, (\cdot)^{-1})$ or $(\mathbb{R}, +, 0, -, \times, 1, (\cdot)^{-1})$. Fields are sets for which addition, multiplication, subtraction and division are defined, so a field resembles (and generalises) rational or real numbers – many operations and theorems defined for real numbers can be interpreted more abstractly for fields, and therefore applied to any particular instance of a field (such as polynomials or integers modulo a prime number).

Category An algebraic characterisation of structures that have an 'source' and 'target'. Instead of a single carrier set, a category is defined on a collection of sets (or, more generally, *objects*) and a collection of *arrows* or *morphisms* between the objects. A typical example is the category **Set** of all sets and all functions between them: the set of morphisms between two sets A and B is the set $A \Rightarrow B$. A category also requires an operation of *composition* of $f : A \to B$ and $g : B \to C$ to get $g \circ f : A \to C$, and an *identity* morphism $\mathrm{id}_A : A \to A$ for every object A, satisfying:

- Left and right unit: $\forall f : A \to B.\ f \circ \mathrm{id}_A = f = \mathrm{id}_B \circ f$
- Associativity: $\forall f : A \to B, g : B \to C, h : C \to D.\ h \circ (g \circ f) = (h \circ g) \circ f$

From the similarity of the laws, we can see that categories are a generalisation of monoids: indeed, a one-object category (where there is no distinction between the source and target) is a monoid (cf. Corollary 110, stating that $(\mathrm{Rel}(A), \circ, \mathrm{id}_A)$ is a monoid for any set A). Any time you are asked to prove that some relational structure (partial functions, surjections, etc.) has a unital identity and an associative composition operation, you prove that it is a category. The field of *category theory* builds some very complex results from this simple idea, establishing surprising formal connections between seemingly disparate mathematical concepts and deriving powerful, abstract results that can be applied to a range of mathematical domains.

Order theory

A branch of mathematics which investigates the notion of *ordering* and *comparison* using homogeneous binary relations. Like with algebraic structures, we start with a *carrier set* P and some binary relation $\sqsubseteq \in \mathrm{Rel}(P)$ satisfying some laws.

Preorder A *preorder* $(P, \sqsubseteq)$ consists of a carrier set P and a relation $\sqsubseteq : P \nrightarrow P$ satisfying:

- Reflexivity: $\forall a \in P.\ a \sqsubseteq a$
- Transitivity: $\forall a, b, c \in P.\ (a \sqsubseteq b \wedge b \sqsubseteq c) \implies a \sqsubseteq c$

Partial order A *partially ordered set* or *poset* is a preorder $(P, \sqsubseteq)$ which further satisfies *antisymmetry*: $\forall a, b \in P.\ (a \sqsubseteq b \wedge b \sqsubseteq a) \implies a = b$.

Total order A *totally ordered set* is a poset $(P, \sqsubseteq)$ which further satisfies *totality* (or *linearity*): $\forall a, b \in P.\ a \sqsubseteq b \vee b \sqsubseteq a$.

Strict order A *strict order* $(P, \sqsubset)$ can be constructed from any partial order $(P, \sqsubseteq)$ by defining $\sqsubset$ as $a \sqsubset b \triangleq a \sqsubseteq b \wedge a \neq b$.

The the following definitions are concerned with special constructions within a fixed poset $(P, \sqsubseteq)$.

Least and greatest element An element $\bot \in P$ is the *least* (or *bottom*) *element* of the poset P if it is below every element in P: $\forall a \in P.\ \bot \sqsubseteq a$. An element $\top \in P$ is the *greatest* (or *top*) *element* if it is above every element in P: $\forall a \in P.\ a \sqsubseteq \top$. Top and bottom elements are unique, if they exist.

Minimal and maximal element An element $m \in P$ is *minimal* if it has no elements below it: $\forall a \in P.\ a \sqsubseteq m \implies a = m$. An element $n \in P$ is *maximal* if it has no elements above it: $\forall a \in P.\ n \sqsubseteq a \implies n = a$. If a preorder has no least element, it may still have several minimal elements which are lower than any other non-minimal element but not related to each other (and similarly for maximal elements).

Lower and upper bound Given a subset $S \subseteq P$, an element $l \in P$ is a *lower bound* for S if it is below every element of S: $\forall a \in S.\ l \sqsubseteq a$. An element $u \in P$ is an *upper bound* for S if it is above every element of S: $\forall a \in S.\ a \sqsubseteq u$. Bounds do have to exist for every subset, and if they do, they might not be unique.

Least upper bounds and greatest lower bound Given a subset $S \subseteq P$, the *least upper bound* (also called *join* or *supremum*), denoted $\bigvee S$, is the least element of the set of upper bounds of S; that is, the element uniquely determined (if it exists) by the properties:

- An upper bound: $\forall a \in S.\ a \sqsubseteq \bigvee S$
- Below any upper bound: $\forall u \in P.\ (\forall a \in S.\ a \sqsubseteq u) \implies \bigvee S \sqsubseteq u$

Dually, the *greatest lower bound* (also called *meet* or *infimum*), denoted $\bigwedge S$, is the greatest element of the set of lower bounds of S:

- A lower bound: $\forall a \in S.\ \bigwedge S \sqsubseteq a$
- Above any lower bound: $\forall l \in P.\ (\forall a \in S.\ l \sqsubseteq a) \implies l \sqsubseteq \bigwedge S$.

These conditions can be equivalently presented with single bi-implications which follow from combining $a \sqsubseteq \bigvee S$ and $\bigvee S \sqsubseteq u$ via transitivity (and similarly for meets):

$$\forall u \in P.\ \bigvee S \sqsubseteq u \iff (\forall a \in S.\ a \sqsubseteq u) \qquad \forall l \in P.\ l \sqsubseteq \bigwedge S \iff (\forall a \in S.\ l \sqsubseteq a)$$

Binary joins and meets Meets and joins of two-element sets $S = \{a, b\}$ are often written as binary operators: $a \vee b \triangleq \bigvee S$ and $a \wedge b \triangleq \bigwedge S$. Given $a, b \in P$, the join $a \vee b$ has the properties:

$$a \sqsubseteq a \vee b \qquad b \sqsubseteq a \vee b \qquad \forall u \in P.\ (a \sqsubseteq u \text{ and } b \sqsubseteq u) \implies a \vee b \sqsubseteq u$$

and the meet $a \wedge b$ has the properties:

$$a \wedge b \sqsubseteq a \qquad a \wedge b \sqsubseteq b \qquad \forall l \in P.\ (l \sqsubseteq a \text{ and } l \sqsubseteq b) \implies l \sqsubseteq a \wedge b$$

Equivalently, we have the bi-implications:

$$\forall u \in P.\ a \vee b \sqsubseteq u \iff (a \sqsubseteq u \text{ and } b \sqsubseteq u) \qquad \forall l \in P.\ l \sqsubseteq a \wedge b \iff (l \sqsubseteq a \text{ and } l \sqsubseteq b)$$

Using these notions, we can describe some order-theoretic structures algebraically.

Join- and meet-semilattice A poset $(P, \sqsubseteq)$ is a *join-semilattice* if every pair of elements has a join (and, by induction, this implies that every finite subset of P has a join). This lets us treat joins as an abstract algebraic operator $\vee\colon P \times P \to P$ whose laws follow from the structure of subsets in P: they contain unique elements and are unordered, and nested sets can be collapsed into a single set. The corresponding laws are:

- Idempotence: $\forall a \in P.\ a \vee a = a$
- Commutativity: $\forall a, b \in P.\ a \vee b = b \vee a$
- Associativity: $\forall a, b, c \in P.\ a \vee (b \vee c) = (a \vee b) \vee c$.

Dually, P is a *meet-semilattice* if every pair of elements has a meet, making $\wedge\colon P \times P \to P$ into an idempotent, commutative and associative operator on P.

Lattice A poset $(P, \sqsubseteq)$ is a *lattice* if it is both a join- and a meet-semilattice, and in addition, the following *absorption laws* hold: $\forall a, b \in P.\ a \wedge (a \vee b) = a$ and $a \vee (a \wedge b) = a$.

Bounded lattice A lattice $(P, \vee, \wedge)$ is *bounded* if it has both a least and greatest element $\bot$ and $\top$ which are the units for join and meet, respectively: $\forall a \in P.\ a \vee \bot = a = a \wedge \top$.

Distributive lattice A lattice $(P, \vee, \wedge)$ is *distributive* if meets distribute over joins (and vice versa): $\forall a, b, c \in P.\ a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$. Any distributive lattice is a semiring under join and meet.

Complemented lattice A bounded lattice $(P, \vee, \wedge, \bot, \top)$ is *complemented* if every element $a \in P$ has a complement $a^\complement$ satisfying the complement laws $a \vee a^\complement = \top$ and $a \wedge a^\complement = \bot$.

Boolean algebra A Boolean algebra $(P, \vee, \wedge, \top, \bot, (\cdot)^\complement)$ is a *bounded distributive lattice*. Complements in distributive lattices are always unique. We can interpret the classic de Morgan laws in any Boolean algebra: $\forall a, b \in P.\ (a \wedge b)^\complement = a^\complement \vee b^\complement$ and $(a \vee b)^\complement = a^\complement \wedge b^\complement$.

The prototypical example of a Boolean algebra (which gives it its name and syntax) is the structure associated with Boolean propositional logic: $(\mathbb{B}, \wedge, \vee, \bot, \top, \neg)$. Explicitly: $\mathbb{B}$ is the set of truth values $\{\top, \bot\}$ with the order $\bot \sqsubseteq \top$ (and reflexivity); $\wedge$ and $\vee$ is Boolean conjunction and disjunction, respectively; $\bot$ (false) and $\top$ (true) are the least and greatest elements for the order; $\neg$ is Boolean negation. The associated Boolean algebra laws (idempotence, associativity, commutativity, distributivity, annihilation, complement) and the derived identities are exactly the ones shown in Section 1.1. The rules associated with implication follow from all these laws if we define $P \implies Q$ as $\neg P \vee Q$ (which is one of the identities): for example, $\neg P \cong P \implies \bot$ follows from $P \implies \bot = \neg P \vee \bot = \neg P$, where we used the unit law for joins and bottom elements. In addition, it is also the case that $P \sqsubseteq Q$ if and only if $P \implies Q$ (since the only ordering that doesn't hold is $\top \sqsubseteq \bot$, which corresponds to the only false line in the truth table for $\implies$), so $\implies$ acts as a so-called *internalisation* of the ordering relation.

Number theory

Definitions

Divisibility For all integers d and n, d *divides* n if:

$$d \mid n \iff \exists k \in \mathbb{Z}.\ n = k \cdot d$$

Congruence For all $m \in \mathbb{Z}^+$ and $a, b \in \mathbb{Z}$, a and b are *congruent modulo* m if:

$$a \equiv b \pmod{m} \iff m \mid (a - b)$$

Quotient and remainder The unique pair of natural numbers $\mathrm{quo}(m, n)$ and $\mathrm{rem}(m, n)$ associated to a given $m \in \mathbb{N}$ and $n \in \mathbb{Z}^+$ by the Division Theorem, with the defining properties:

$$\mathrm{rem}(m, n) < n \ \wedge\ m = \mathrm{quo}(m, n) \cdot n + \mathrm{rem}(m, n)$$

Modulus For every $m \in \mathbb{Z}^+$ and $k \in \mathbb{Z}$, the unique natural number k *modulo* m is defined as $[k]_m \triangleq \mathrm{rem}(k + |k| \cdot m, m) \in \mathbb{N}$ with the defining properties:

$$[k]_m < m \quad \wedge \quad k \equiv [k]_m \pmod{m}$$

Modular arithmetic For every $m \in \mathbb{Z}^+$, the set of *integers modulo* m is the set $\mathbb{Z}_m \triangleq \{0, 1, \ldots, m-2, m-1\}$ of nonnegative integers strictly less than m. Modular addition and multiplication of integers $k, l \in \mathbb{Z}_m$ is defined as:

$$k +_m l \triangleq [k + l]_m \quad \wedge \quad k \cdot_m l \triangleq [k \cdot l]_m$$

Common divisors Given an $n \in \mathbb{N}$, the set of its *divisors* is defined as

$$\mathrm{D}(n) \triangleq \{\, d \in \mathbb{N} \mid (d \mid n) \,\}$$

The set of *common divisors* of $n, m \in \mathbb{N}$ is the set

$$\mathrm{CD}(m, n) \triangleq \mathrm{D}(m) \cap \mathrm{D}(n) = \{\, d \in \mathbb{N} \mid (d \mid m \wedge d \mid n) \,\}$$

Greatest common divisor The *greatest common divisor* of $m, n \in \mathbb{N}$ is the the maximum element of $\mathrm{CD}(m, n)$, the set of their common divisors.

Linear combination An integer r is said to be a *linear combination* of a pair $m, n \in \mathbb{Z}$ when there exist a pair of *coefficients* s and t such that

$$s \cdot m + t \cdot n = r$$

Theorems

Fermat's Little Theorem For all naturals i and primes p,

$$i^p \equiv i \pmod{p} \qquad p \nmid i \implies i^{p-1} \equiv 1 \pmod{p}$$

Division Theorem For all $m \in \mathbb{N}$ and $n \in \mathbb{Z}^+$,

$$\exists! q, r \in \mathbb{N}.\ r < n \ \wedge\ m = q \cdot n + r$$

Modulus and remainder For all $m \in \mathbb{Z}^+$ and naturals $k, l, n \in \mathbb{N}$,

$$k \equiv l \pmod{m} \iff \mathrm{rem}(k, m) = \mathrm{rem}(l, m) \qquad n \equiv \mathrm{rem}(n, m) \pmod{m}$$

Common divisor properties For all $n \in \mathbb{Z}^+$ and naturals $m, m' \in \mathbb{N}$,

$$m \equiv m' \pmod{n} \implies \mathrm{CD}(m,n) = \mathrm{CD}(m',n)$$
$$\mathrm{CD}(m,n) = \mathrm{CD}(\mathrm{rem}(m,n),n)$$
$$\mathrm{CD}(m,n) = \mathrm{CD}(\max(m,n) - \min(m,n), \min(m,n))$$

Euclid's Algorithm For all $m, n \in \mathbb{Z}^+$,

$$\gcd(m,n) \triangleq \begin{cases} n & \text{if } n \mid m \\ \gcd(n, \mathrm{rem}(m,n)) & \text{otherwise} \end{cases}$$

This algorithm terminates on all pairs $m, n \in \mathbb{Z}^+$ and $\gcd(m,n)$ is the greatest common divisor of m and n, satisfying the properties:

- A common divisor: $\gcd(m,n) \mid m \wedge \gcd(m,n) \mid n$
- Greatest of all common divisors: $\forall d \in \mathbb{Z}^+. d \mid m \wedge d \mid n \implies d \mid \gcd(m,n)$

The latter condition is known as the *universal property* of GCDs: a property which uniquely characterises the greatest common divisor of two numbers without explicitly defining it.

Linear combinations and GCDs For all $m, n \in \mathbb{Z}^+$, $\gcd(m,n)$ is the *least positive linear combination* of m and n. That is:

1. It is a linear combination: $\exists k_0, l_0 \in \mathbb{Z}.\ \gcd(m,n) = k_0 \cdot m + l_0 \cdot n$ (where the coefficients $k_0 = \mathrm{lc}_1(m,n)$ and $l_0 = \mathrm{lc}_2(m,n)$ can be efficiently computed using the *Extended Euclidean Algorithm*)
2. It is the least of all linear combinations: $\forall k, l \in \mathbb{Z}.\ \gcd(m,n) \mid k \cdot m + l \cdot n$

It follows that m and n are coprime if and only if there is a way to express 1 as their linear combination: $(\exists k, l \in \mathbb{Z}.\ k \cdot m + l \cdot n = 1) \iff \gcd(m,n) = 1$

Euclid's Theorem For all $m, n \in \mathbb{Z}^+$ and primes p, $p \mid (m \cdot n) \implies p \mid m \vee p \mid n$.

Multiplicative inverses in modular arithmetic For all $m, n \in \mathbb{Z}^+$,

1. $n \cdot \mathrm{lc}_2(m,n) \equiv \gcd(m,n) \pmod{m}$
2. If $\gcd(m,n) = 1$, $[\mathrm{lc}_2(m,n)]_m$ is the multiplicative inverse of $[n]_m$ in $\mathbb{Z}_m$
3. For a prime p, every non-zero element of $\mathbb{Z}_p$ has $[i^{p-2}]_p$ as a multiplicative inverse. Hence, the algebraic structure $(\mathbb{Z}_p, +_p, 0, -_p, \times_p, 1, [(\cdot)^{p-2}]_p)$ is a field.

The divisibility lattice The structure $(\mathbb{N}, \mathrm{lcm}, \gcd, 1, 0)$, is a bounded lattice, where:

- $(\mathbb{N}, |)$ is a partial order: $m \in \mathbb{N}$ is below $n \in \mathbb{N}$ if m divides n ($m \mid n$)
- 1 is the bottom element: $\forall n \in \mathbb{N}.\ 1 \mid n$
- 0 is the top element: $\forall n \in \mathbb{N}.\ n \mid 0$
- The common divisors of two numbers $m, n \in \mathbb{N}$ is the set of lower bounds: $\forall d \in \mathrm{CD}(m,n).\ d \mid m \wedge d \mid n$
- The greatest common divisor is the meet: the characteristic property of meets

$$\forall d \in \mathbb{N}.\ (d \mid m \wedge d \mid n) \iff d \mid \gcd(m,n)$$

is exactly the universal property associated with GCDs.
- The top element is the unit for the meet: $\forall n \in \mathbb{N}.\ \gcd(0, n) = n$.
- Dually, the join of this preorder is the *least common multiple*, the bottom element of the set of common multiples of two numbers. The bottom of the preorder is the unit for the join: $\forall n \in \mathbb{N}.\ \mathrm{lcm}(1, n) = n$.

Set theory

Foundations

Definitions

Subset $A \subseteq B \iff \forall x.\ x \in A \implies x \in B$

Proper subset $A \subset B \iff A \subseteq B \wedge A \neq B$

Empty set $\forall x.\ x \notin \emptyset$

Finite set $[n] \triangleq \{0, \ldots, n-1\}$

Cardinality The size $\#A$ or $|A|$ associated with a set A. If it is a natural number, the set is *finite*. In general, the numbers used to describe the cardinality of (finite or infinite) sets are called *cardinal numbers*.

Universe The 'domain of discourse' in set theory often denoted as $\mathcal{U}$ – the collection of elements that we are concerned with, the superset of all the sets that we are investigating. When we say 'A is a set', we usually mean it is a subset of the universe $\mathcal{U}$. We need this subtlety to avoid set-theoretic paradoxes such as Russell's paradox. However, we often leave the universe implicit: writing x for an element implicitly assumes that $x \in \mathcal{U}$.

Set comprehension A notation for defining a set by specifying the property that its elements must satisfy. Abstractly, we write

$$A \triangleq \{\, x \in B \mid P(x) \,\}$$

to state that A consists of elements (of B) for which $P(x)$ holds. Thus, saying that $x \in A$ is equivalent to saying that $x \in B$ and furthermore $P(x)$ holds.

Powerset $\mathcal{P}(A) \triangleq \{X \subseteq \mathcal{U} \mid X \subseteq A\}$

Intersection $A \cap B \triangleq \{\, x \in \mathcal{U} \mid x \in A \wedge x \in B \,\}$

Union $A \cup B \triangleq \{\, x \in \mathcal{U} \mid x \in A \vee x \in B \,\}$

Complement $A^{\complement} \triangleq \{\, x \in \mathcal{U} \mid x \notin A\}$

Set difference $A \setminus B \triangleq \{\, x \in A \mid x \notin B \,\}$

Big union For all families of sets $\mathcal{F} \subseteq \mathcal{P}(\mathcal{U})$

$$\bigcup \mathcal{F} \triangleq \{\, x \in \mathcal{U} \mid \exists A \in \mathcal{F}.\ x \in A\}$$

Big intersection For all families of sets $\mathcal{F} \subseteq \mathcal{P}(\mathcal{U})$

$$\bigcap \mathcal{F} \triangleq \{\, x \in \mathcal{U} \mid \forall A \in \mathcal{F}.\ x \in A\}$$

Ordered pair $(a, b) \triangleq \{\, a, \{\, a, b \,\} \,\}$

Cartesian product $A \times B \triangleq \{\, (a, b) \mid a \in A \wedge b \in B \,\}$

Finitary product

$$\prod_{i=1}^{n} A_i \triangleq A_1 \times \cdots \times A_n$$

Set exponent $A^n \triangleq \prod_{i=1}^{n} A$

Tagging $\{\ell\} \times A \triangleq \{(\ell, a) \mid a \in A\}$

Disjoint union $A \uplus B \triangleq (\{1\} \times A) \cup (\{2\} \times B)$

Finitary disjoint union

$$\biguplus_{i=1}^{n} A_i \triangleq A_1 \uplus \cdots \uplus A_n$$

Set multiplication $n \cdot A \triangleq \biguplus_{i=1}^{n} A$

Theorems

Algebraic and order-theoretic properties Let A be a carrier set. A can also taken to be the universe $\mathcal{U}$ so these apply to 'arbitrary sets' as well.

- $(\mathcal{P}(A), \subseteq)$ forms a partial order under subset inclusion.
- $(\mathcal{P}(A), \cup, \cap, \emptyset, A, (\cdot)^\complement)$ forms a Boolean algebra. This means that all the algebraic laws in the first part of the table in Section 1.1 apply to set theory when we replace the join of logic (disjunction) with the join of set theory (union), etc. The big union $\bigcup \mathcal{F}$ and intersection $\bigcap \mathcal{F}$ are the generalised joins and meets of the subset $\mathcal{F} \subseteq \mathcal{P}(A)$. The characteristic properties of big unions and intersections follow directly from the definitions of joins and meets: for all families of sets $\mathcal{F}$, we have

  $$\forall U.\ \bigcup \mathcal{F} \subseteq U \iff (\forall X \in \mathcal{F}.\ X \subseteq U) \qquad \forall L.\ L \subseteq \bigcap \mathcal{F} \iff (\forall X \in \mathcal{F}.\ L \subseteq X)$$

- $(\mathcal{P}(A), \uplus, \times, \emptyset, [1])$ forms a commutative semiring up to isomorphism: for example, $A \times [1]$ is not *equal* to A (as one consists of tuples $(a, 0)$ while the other is simply elements a), but there is a bijective correspondence between them.

Cardinality Let A, B be finite sets.

$$\#\emptyset = 0 \quad \#[n] = n \quad \#\mathcal{P}(A) = 2^{\#A} \quad \#(A \times B) = \#A \cdot \#B \quad \#(A \uplus B) = \#A + \#B$$

Relations, matrices and graphs

Definitions

Binary relation A *relation* R between sets A and B (denoted $R : A \nrightarrow B$) is a subset $R \subseteq A \times B$. The set of all relations between A and B is denoted $\mathrm{Rel}(A, B)$. An element $(a, b) \in R$ is usually written $a\,R\,b$ (as an operator) or $R(a, b)$ (as a predicate). A relation $R \in \mathrm{Rel}(A)$ *on* a set A is $R : A \nrightarrow A$.

Identity relation and relational composition The identity relation id_A is $\{(a, a) \mid a \in A\}$, so $\mathrm{id}_A(a, a') \iff a = a'$.

Relational composition *Composition* of relations $R : A \nrightarrow B$ and $S : B \nrightarrow C$ is a new relation $S \circ R : A \nrightarrow C$ defined as $\{(a, c) \mid \exists b \in B.\ a\,R\,b \wedge b\,S\,c\}$.

Iterated composition For a relation $R\colon A \nrightarrow A$ on a set A, the n-fold composition $R^{\circ n}\colon A \nrightarrow A$ is

$$R^{\circ n} \triangleq \underbrace{R \circ \cdots \circ R}_{n \text{ times}}$$

The iterated composition $R^{\circ *}\colon A \nrightarrow A$ is then

$$R^{\circ *} \triangleq \bigcup \{ R^{\circ n}\colon A \nrightarrow A \mid n \in \mathbb{N} \} = \bigcup_{n \in \mathbb{N}} R^{\circ n}$$

Closure Given a relation $R \in \mathrm{Rel}(A)$ and property P (such as reflexivity), the *closure of R under property P* is a relation $\widehat{R}_P \in \mathrm{Rel}(A)$ which is the smallest superset of R which satisfies P:

1. $R \subseteq \widehat{R}_P$
2. $\widehat{R}_P$ satisfies P
3. For any other $T \in \mathrm{Rel}(A)$ which is a superset of R and satisfies P, $\widehat{R}_P \subseteq T$.

More generally, we call an arbitrary set A *closed* under some operation if applying the operation to elements of A yields an element of A. The *closure* of a set under an operation is the smallest superset of A which is closed under the operation. For instance, natural numbers are closed under addition but not under subtraction; the closure of natural numbers under subtraction is the set of integers.

Matrices For positive integers m and n, an $(m \times n)$-*matrix* M over a semiring $(S, \oplus, \mathbf{1}, \otimes, \mathbf{0})$ is given by entries $M_{i,j} \in S$ for all $0 \leq i < m$ and $0 \leq j < n$. The set of all $(m \times n)$-matrices is denoted $\mathrm{Mat}(m, n)$. An $(n \times n)$-matrix is called a *square matrix of order n*, and the set of all such matrices is denoted $\mathrm{SqMat}(n)$.

Identity matrix The *identity* $(n \times n)$-matrix $I - n$ has entries

$$(I_n)_{i,j} \triangleq \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Matrix multiplication The *multiplication* of an $(\ell \times m)$-matrix L with an $(m \times n)$-matrix M is the $(\ell \times n)$-matrix $M \cdot L$ with entries

$$(M \cdot L)_{i,j} \triangleq (M_{0,j} \otimes L_{i,0}) \oplus \cdots \oplus (M_{m-1,j} \otimes L_{i,m-1}) = \bigoplus_{k=0}^{m-1} M_{k,j} \otimes L_{i,k}$$

Null matrix The *null* $(m \times n)$-matrix $Z_{m,n}$ has entries

$$(Z_{m,n})_{i,j} \triangleq 0$$

Matrix addition The *addition* of two $(m \times n)$-matrices M and L is the matrix $M + L$ with entries

$$(M + L)_{i,j} \triangleq M_{i,j} \oplus L_{i,j}$$

Matrix from relation For a relation $R\colon [m] \nrightarrow [n]$ we have a $(m \times n)$-matrix $\mathrm{mat}(R)$ over the commutative semiring of Booleans given by

$$\mathrm{mat}(R)_{i,j} \triangleq (i, j) \in R$$

Relation from matrix For an $(m \times n)$-matrix M over the semiring of Booleans we have a relation

$\mathrm{rel}(M)\colon [m] \nrightarrow [n]$ given by

$$(i,j) \in \mathrm{rel}(M) \iff M_{i,j}$$

Graphs A *directed graph* (A,R) consists of a set A and a relation $R\colon A \nrightarrow A$

Path For a directed graph (A,R) and $s,t \in A$, a *path* of length $n \in \mathbb{N}$ in R with *source* s and *target* t is a tuple $(a_0,\ldots,a_n) \in A^{n+1}$ such that $a_0 = s$, $a_n = t$ and $a_i\, R\, a_{i+1}$ for all $0 \le i < n$.

Adjacency matrix The *adjacency matrix* of a finite directed graph $([n],R)$ for $n \in \mathbb{Z}^+$ is the $(n \times n)$-matrix $\mathrm{mat}(R)$. There is an edge between nodes labelled i and j iff $(i,j) \in \mathrm{mat}(R)$.

Theorems

Algebraic properties Let A, B, C be sets and m,n be natural numbers.

- The set of relations $\mathrm{Rel}(A)$ on A forms a monoid $(\mathrm{Rel}(A), \circ, \mathrm{id}_A)$.
- More generally, sets and relations between them form a category **Rel** with the associative relational composition $S \circ R\colon A \nrightarrow C$ for $R\colon A \nrightarrow B$ and $S\colon B \nrightarrow C$, and unital identity relation id_A for all sets A.
- Square matrices of order n over a semiring form a monoid $(\mathrm{SqMat}(n), \otimes, I_n)$.
- More generally, matrices form a (slightly unusual) category **Mat** whose objects are natural numbers, and the arrows between naturals m and n are $(m \times n)$-matrices. The categorical composition of an $(\ell \times m)$-matrix M and an $(m \times n)$-matrix L is the $(\ell \times n)$-matrix $M \cdot L$, where $\cdot$ is the associative matrix multiplication. The identity arrow for every object $n \in \mathbb{N}$ is the identity matrix I_n.
- $(\mathrm{SqMat}(n), \otimes, I_n, \oplus, Z_{n,n})$ forms a semiring.
- More generally, matrix multiplication is associative and has the identity matrix as a neutral element; matrix addition is commutative and associative and has the null matrix as a neutral element; for every $(\ell \times m)$-matrix L, L' and $(m \times n)$-matrix M, M', we have the annihilation laws

  $$M \cdot Z_{\ell,m} = Z_{\ell,n} \qquad Z_{m,n} \cdot L = Z_{\ell,n}$$

  and distributivity laws

  $$M \cdot (L + L') = (M \cdot L) + (M \cdot L') \qquad (M + M') \cdot L = (M \cdot L) + (M' \cdot L)$$

Relations and matrices The functions rel and mat are bijective inverses: for every matrix M over the semiring of Booleans, $\mathrm{mat}(\mathrm{rel}(M)) = M$ and for every relation R, $\mathrm{rel}(\mathrm{mat}(R)) = R$.

More generally, this shows that the categories **Rel** and **Mat** are *isomorphic*, and the inverse transformations $\mathrm{rel}\colon \mathbf{Mat} \to \mathbf{Rel}$ and $\mathrm{mat}\colon \mathbf{Rel} \to \mathbf{Mat}$ are structure-preserving mappings of categories (called *functors*), which means that they map composition and identities in one category into composition and identities in the other:

$$\mathrm{rel}(M \cdot L) = \mathrm{rel}(M) \circ \mathrm{rel}(L) \qquad \mathrm{mat}(S \circ R) = \mathrm{mat}(R) \cdot \mathrm{mat}(S)$$
$$\mathrm{rel}(I_n) = \mathrm{id}_{[n]} \qquad \mathrm{mat}\big(\mathrm{id}_{[n]}\big) = I_n$$

Paths and adjacency matrices Given a finite directed graph $([n],R)$ with adjacency matrix M, the adjacency matrix $M^* = \mathrm{mat}(R^{\circ *})$ can be computed as M_n by repeated matrix addition and

multiplication:

$$M_0 = I_n \qquad M_{k+1} = I_n + (M \cdot M_k)$$

The graph has a path of length k between nodes i and j if and only if $(M_k)_{i,j}$ holds. Thus, a node j is inaccessible from i iff $(M^*)_{i,j}$ does not hold.

Closures Given a relation $R : A \nrightarrow A$ and a property P (such as reflexivity), the closure of R under P can be defined as $\widehat{R}_P = \bigcap \mathcal{F}_{R,P}$ where the family $\mathcal{F}_{R,P} \subseteq \mathcal{P}(A \times A)$ is defined as:

$$\mathcal{F}_{R,P} \triangleq \{\, Q : A \nrightarrow A \mid R \subseteq Q \ \wedge\ Q \text{ satisfies } P \,\}$$

Intuitively, every relation of the family $\mathcal{F}_{R,P}$ satisfies the first two conditions in the definition of a closure above (so it is the set of upper bounds for the closure $\widehat{R}_P$), and the smallest such relation is the least element of this family of sets, i.e. the meet defined to be the intersection.

Functions

Definitions

Functionality A relation $R : A \nrightarrow B$ is *functional* if it relates an element $a \in A$ with at most one element $b \in B$: $\forall a \in A.\ \forall b_1, b_2 \in B.\ a R\, b_1 \wedge a R\, b_2 \implies b_1 = b_2$.

Totality A relation $R : A \nrightarrow B$ is *total* if it relates every element $a \in A$ with some element of $b \in B$: $\forall a \in A.\ \exists b \in B.\ a R\, b$.

Partial function A *partial function* $f : A \rightharpoonup B \in \mathrm{PFun}(A, B)$ is a functional relation. The *domain of definition* $D_f \triangleq \{\, a \in A \mid \exists b \in B.\ (a, b) \in f \,\}$ is the set of elements $a \in A$ which are mapped to some $b \in B$. If an $a \in A$ is also in D_f, we say that the partial function f is *defined at* a (denoted $f(a)\!\downarrow$) with value $f(a) = b$ (where b is unique because of functionality). If $a \notin D_f$, f is *undefined at* a, denoted $f(a)\!\uparrow$.

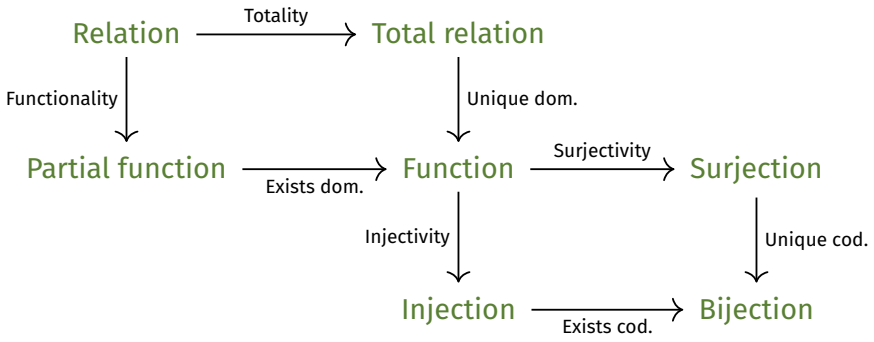
Total function A *total function* (or just *function* or *map*) $f : A \to B \in \mathrm{Fun}(A, B)$ is a total functional relation, or a total partial function. The totality and functionality (existence and uniqueness of mapping, respectively) combines into the unique existence of mapping for functions: $\forall a \in A.\ \exists! b \in B.\ f(a) = b$. A partial function $f : A \rightharpoonup B$ is total if its domain of definition coincides with its source: $D_f = A$.

Surjection A *surjective function* or *surjection* $f : A \twoheadrightarrow B$ covers its whole codomain, i.e. maps every $a \in A$ to *at least* one $b \in B$: $\forall b \in B.\ \exists a \in A.\ f(a) = b$.

Injection An *injective function* or *injection* $f : A \rightarrowtail B$ embeds the domain into the codomain, i.e. maps every $a \in A$ to *at most* one $b \in B$: $\forall a_1, a_2 \in A.\ (f(a_1) = f(a_2)) \implies a_1 = a_2$.

Bijection A *bijective function* or *bijection* is a function that is both an injection and a surjection, i.e. maps every $a \in A$ to *exactly* one $b \in B$: $\forall b \in B.\ \exists! a \in A.\ f(a) = b$. Equivalently, a function $f : A \to B$ is a bijection if there exists a necessarily unique function $f^{-1} : B \to A$ that is both:

- a *retraction* (left inverse) for f: $g \circ f = \mathrm{id}_A$
- a *section* (right inverse) for f: $f \circ g = \mathrm{id}_B$



Theorems

Algebraic properties Sets and partial functions, functions, injections, surjections and bijections between them all form categories, i.e. they all have respective neutral identity maps and are closed under associative composition.

Cardinality Let A, B be finite sets.

$$\#\mathrm{PFun}(A,B) = (\#B+1)^{\#A} \quad \#\mathrm{Fun}(A,B) = \#B^{\#A} \quad \#\mathrm{Bij}(A,B) = \begin{cases} 0 & \text{if } \#A \neq \#B \\ n! & \text{if } \#A = \#B = n \end{cases}$$

Sections and retractions Every injection between sets is a section, and every surjection is a retraction. The latter statement is equivalent to the *axiom of choice*, also stated as the Cartesian product of a non-empty collection of sets is non-empty.

Isomorphisms and equivalence relations

Definitions

Isomorphism Two sets A, B are *isomorphic* (denoted $A \cong B$) if they are in a *bijective correspondence*, i.e. there exists a bijection $f : A \xrightarrow{\sim} B$ between them. A finite set A of cardinality $n \in \mathbb{N}$ is isomorphic to $[n]$. Two finite sets A and B are isomorphic if they have the same number of elements: $A \cong B \iff \#A = \#B$.

Characteristic function Given a set A and subset $S \subseteq A$, the *characteristic function* $\chi_S : A \to \mathbb{B}$ (where $\mathbb{B} \cong [2]$ is the set of Boolean truth values) associated with S is defined as: $\chi_S(a) = a \in S$.

Equivalence relation A relation $E \in \mathrm{Rel}(A)$ is an *equivalence relation* if it is:

1. reflexive: $\forall x \in A.\ x\ E\ x$
2. symmetric: $\forall x, y \in A.\ x\ E\ y \iff y\ E\ x$
3. transitive: $\forall x, y, z \in A.\ x\ E\ y \land y\ E\ z \implies x\ E\ z$

The set of all equivalence relations on A is denoted $\mathrm{EqRel}(A)$.

Partition A *partition* P of a set A is a set of non-empty subsets (i.e. $P \subseteq \mathcal{P}(A) \setminus \emptyset$) satisfying:

1. Covering: $\bigcup P = A$
2. Pairwise disjointness: $\forall B_1, B_2 \in P.\ B_1 \neq B_2 \implies B_1 \cap B_2 = \emptyset$

The set of all partitions of A is denoted $\mathrm{Part}(A)$.

Equivalence class Given a set A and equivalence relation $E \in \mathrm{EqRel}(A)$, the *equivalence class* of an element $a \in A$ is the set of elements it is related to by E:

$$[a]_E \triangleq \{\, x \in A \mid x \mathrel{E} a \,\}$$

Quotient set Given a set A and equivalence relation $E \in \mathrm{EqRel}(A)$, the *quotient of A by E* is the set of equivalence classes of A under E:

$$A/E \triangleq \{\, B \subseteq A \mid \exists a \in A.\ B = [a]_E \,\} = \{\, [a]_E \subseteq A \mid a \in A \}$$

Relational image Let $R : A \nrightarrow B$ be a relation.

- The *direct image* of a subset $X \subseteq A$ under R is the set $\overrightarrow{R}(X) \subseteq B$, defined as
$$\overrightarrow{R}(X) \triangleq \{\, b \in B \mid \exists x \in X.\ x \mathrel{R} b \,\}$$
- The *inverse image* of a subset $Y \subseteq B$ under R is the set $\overleftarrow{R}(Y) \subseteq A$, defined as
$$\overleftarrow{R}(X) \triangleq \{\, a \in A \mid \forall y \in B.\ a \mathrel{R} y \implies y \in Y \,\}$$

Functional image Let $f : A \to B$ be a function.

- The *direct image* of a subset $X \subseteq A$ under f is the set $\overrightarrow{f}(X) \subseteq B$, defined as
$$\overrightarrow{f}(X) \triangleq \{\, b \in B \mid \exists x \in X.\ f(x) = b \,\} = \{\, f(x) \in B \mid x \in X \,\}$$
- The *inverse image* of a subset $Y \subseteq B$ under f is the set $\overleftarrow{f}(Y) \subseteq A$, defined as
$$\overleftarrow{f}(X) \triangleq \{\, a \in A \mid f(a) \in Y \,\}$$

Theorems

Partitions and equivalence classes For every set A, the the sets of equivalence relations and partitions on A are isomorphic: $\mathrm{EqRel}(A) \cong \mathrm{Part}(A)$. The bijection is established by mapping an equivalence relation $E \in \mathrm{EqRel}(A)$ to the quotient set A/E, and it inverse maps a partition $P \in \mathrm{Part}(A)$ to the relation $\equiv_P$ defined as $x \equiv_P y \iff \exists B \in P.\ x \in B \land y \in B$.

Calculus of bijections Let A, B, C, X, Y be sets.

- Isomorphism is an equivalence relation:
$$A \cong A \qquad A \cong B \implies B \cong A \qquad (A \cong B \ \land\ B \cong C) \implies A \cong C$$
- Isomorphism is a *congruence relation*: if $A \cong X$ and $B \cong Y$, then
$$\mathcal{P}(A) \cong \mathcal{P}(X) \qquad A \times B \cong X \times Y \qquad A \uplus B \cong X \uplus Y \qquad \mathrm{Rel}(A,B) \cong \mathrm{Rel}(X,Y)$$
$$\mathrm{PFun}(A,B) \cong \mathrm{PFun}(X,Y) \qquad \mathrm{Fun}(A,B) \cong \mathrm{Fun}(X,Y) \qquad \mathrm{Bij}(A,B) \cong \mathrm{Bij}(X,Y)$$
- $(\mathcal{P}(A), \uplus, \times, \emptyset, [1])$ forms a commutative semiring up to isomorphism:
  - $(A \uplus B) \uplus C \cong A \uplus (B \uplus C)$ and $(A \times B) \times C \cong A \times (B \times C)$: $\uplus$ and $\times$ are associative.
  - $A \uplus B \cong B \uplus A$ and $A \times B \cong B \times A$: $\uplus$ and $\times$ are commutative.
  - $A \cong A \uplus [0]$ and $A \cong A \times [1]$: $[0]$ is the unit for $\uplus$, and $[1]$ is the unit for $\times$.
  - $(A \uplus B) \times C \cong (A \times C) \uplus (B \times C)$: $\times$ distributes over $\uplus$.
- Properties of functions (cf. Section 1.1):

- $(A \Rightarrow [1]) \cong [1]$: unique constantly 0 function from A into the singleton set.
- $([0] \Rightarrow A) \cong [1]$: unique empty function from the empty set to A.
- $([1] \Rightarrow A) \cong A$: elements a of A are isomorphic to functions $0 \mapsto a$.
- $(A \Rightarrow [0]) \cong [0]$: if A is nonempty, there are no functions from A into the empty set (not even the empty function, since there is no $b \in [0]$ for every $a \in A$).
- $(A \Rightarrow (B \times C)) \cong (A \Rightarrow B) \times (A \Rightarrow C)$: a function returning a pair can be defined as a pair of functions.
- $((A \uplus B) \Rightarrow C) \cong (A \Rightarrow C) \times (B \Rightarrow C)$: a function taking a disjoint union can be defined as a pair of functions (cf. case analysis).
- $((A \times B) \Rightarrow C) \cong (A \Rightarrow (B \Rightarrow C))$: a function taking a pair can be defined as a function returning a function (cf. currying).
- $\mathrm{PFun}(A,B) \cong (A \Rightarrow (B \uplus [1]))$: a partial function can be defined as a function with an extra unique value in its codomain (cf. `option` type).
- $\mathcal{P}(A) \cong (A \Rightarrow [2])$: a subset S of A can be defined via its characteristic function χ_S.

Finite set isomorphisms For all $m, n \in \mathbb{N}$,

$$\mathcal{P}([n]) \cong [2^n] \qquad [m] \times [n] \cong [m \cdot n] \qquad [m] \uplus [n] \cong [m+n]$$
$$\mathrm{PFun}([m],[n]) \cong [(n+1)^m] \qquad \mathrm{Fun}([m],[n]) \cong [n^m] \qquad \mathrm{Bij}([n],[n]) \cong [n!]$$

Images of surjections and injections Let $s : A \twoheadrightarrow B$ be a surjection, and $i : A \rightarrowtail B$ be an injection. The direct image of A under s is the full codomain, and the direct image of A under i is isomorphic to the domain: $\vec{s}(A) = B$ and $\vec{i}(A) \cong A$.

Indexing and enumerability

Definitions

Set-indexed constructions Given a set I, the *family of sets indexed by* I is the indexed family $\{A_i\}_{i \in I}$ consisting of a set A_i for every $i \in I$. That is, we have a mapping $i \mapsto A_i$ from I to $\{A_i\}_{i \in I}$. Various set operations can be indexed by a set I: we recover the usual finitary versions with $I = \mathbb{N}$ and $\{A_n\}_{n \in \mathbb{N}} = \{A_0, A_1, \dots\}$, and binary versions with $I = [2]$ and $\{A_i\}_{i \in [2]} = \{A_0, A_1\}$.

Indexed unions

$$\bigcup_{i \in I} A_i \triangleq \bigcup \{A_i \mid i \in I\} = \{a \mid \exists i \in I.\ a \in A_i\}$$

Indexed intersections For $I \neq \emptyset$,

$$\bigcap_{i \in I} A_i \triangleq \left\{ a \in \bigcup_{i \in I} A_i \;\middle|\; \forall i \in I.\ a \in A_i \right\}$$

Indexed disjoint unions

$$\biguplus_{i \in I} A_i \triangleq \bigcup_{i \in I} \{i\} \times A_i$$

Indexed products

$$\textstyle\prod_{i \in I} A_i \triangleq \left\{ \alpha \in \left(I \Rightarrow \bigcup_{i \in I} A_i\right) \;\middle|\; \forall i \in I.\ \alpha(i) \in A_i \right\}$$

Finite sequences

$$A^* \triangleq \biguplus_{n \in \mathbb{N}} A^n$$

Indexed partial functions

$$\mathrm{PFun}_{\mathrm{fin}}(A, B) \triangleq \biguplus_{S \in \mathcal{P}_{\mathrm{fin}}(A)} (S \Rightarrow B)$$

Enumerability A set is *enumerable* when there exists a surjection $e : \mathbb{N} \twoheadrightarrow A$ called an *enumeration*.

Countability A set is *countable* if it is either empty or enumerable.

Theorems

Closure of countability Let A, B be countable sets.

- The set of natural numbers $\mathbb{N}$ is countable.
- If $S \subseteq A$, then S is countable.
- If $A \cong X$ then X is countable.
- $A \times B$ and $A \uplus B$ are countable.
- If $\{X_i\}_{i \in A}$ is a countable indexed family of countable sets, then $\biguplus_{i \in A} X_i$ is countable.

As corollaries of the above, we have:

- Every finite set is countable.
- The set of integers $\mathbb{Z} \cong \mathbb{N} \uplus \mathbb{N}^+$ is countable.
- The set of rational numbers $\mathbb{Q} \cong \mathbb{N} \times \mathbb{N}$ is countable.
- A^*, $\mathcal{P}_{\mathrm{fin}}(A)$ and $\mathrm{PFun}_{\mathrm{fin}}(A, B)$ are countable.

Cantor's diagonalisation argument For every set A, no surjection $A \twoheadrightarrow \mathcal{P}(A)$ exists. As a result, the set $\mathcal{P}(\mathbb{N})$ is *uncountable* – it is 'more infinite' than the already infinite set $\mathbb{N}$. This establishes that the cardinality of natural numbers (called *countable infinity* and denoted $\aleph_0$) is strictly smaller than the cardinality of $\mathcal{P}(\mathbb{N})$ (which can be denoted as $2^{\aleph_0}$ in cardinal arithmetic). Similarly, $|\mathcal{P}(\mathcal{P}(\mathbb{N}))| = 2^{(2^{\aleph_0})}$ is strictly greater still, and repeated application of the powerset operation lets us generate a countably infinite sequence of *uncountably infinite* sets, each strictly larger than the previous one.

Cardinality of real numbers Any closed interval $[a, b]$ of real numbers between $a, b \in \mathbb{R}$ is isomorphic to the real numbers. In particular, elements of the interval $[0, 1]$ can be described as binary sequence, corresponding to the bits after the 'binary point' in the binary expansion of a real in $[0, 1]$. The set of all binary sequences $[2]^\infty$ thus also isomorphic to the real numbers, and so is the set of functions from $\mathbb{N}$ to $[2]$ that enumerate the bits in a binary sequence. Finally, $(\mathbb{N} \Rightarrow [2])$ is the set of characteristic functions on naturals and is isomorphic to $\mathcal{P}(\mathbb{N})$.

$$\mathbb{R} \cong [0, 1] \cong [2]^\infty \cong (\mathbb{N} \Rightarrow [2]) \cong \mathcal{P}(\mathbb{N})$$

Given that the cardinalities of isomorphic sets are equal, this derivation shows that the cardinality of the real numbers (called the *continuum*, denoted $\mathfrak{c}$) is equal to $|\mathcal{P}(\mathbb{N})| = 2^{\aleph_0}$.

Continuum hypothesis There is no set of cardinality strictly between the cardinality of $\mathbb{N}$ and $\mathbb{R}$, that is, there is no cardinal number between $\aleph_0$ and $\mathfrak{c}$. Equivalently, if the smallest cardinal

number greater than $\aleph_0$ is denoted $\aleph_1$, the continuum hypothesis states that $\aleph_1 = \mathfrak{c} = 2^{\aleph_0}$. A generalisation of the hypothesis states that $\aleph_{k+1} = 2^{\aleph_k}$ for any $k \in \mathbb{N}$, so the infinite sequence of cardinalities generated by $\left(\#\mathcal{P}^k(\mathbb{N})\right)_{k\in\mathbb{N}}$ is exactly the sequence $(\aleph_k)_{k\in\mathbb{N}}$. Kurt Gödel and Paul Cohen showed that the continuum hypothesis is neither provable nor disprovable in Zermelo–Fraenkel set theory with the axiom of choice, so its validity is undecidable: no contradiction would arise if either the CH or its negation is added to ZFC.

Regular languages and finite automata

Formal languages

Definitions

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

Strings A *string of length* $n \in \mathbb{N}$ over an alphabet Σ is an ordered n-tuple of symbols. Σ^* is the set of all finite strings over Σ.

Empty string The unique string of length 0 is the *empty string* denoted ε.

Concatenation Joining two strings u and v end-to-end creates the *concatenation* denoted by $u \cdot v$ or uv. This generalises to the concatenation of any finite number of strings.

Language Given an alphabet Σ, a subset of Σ^* is called a *formal language*.

Theorem

For an alphabet Σ, the structure $(\Sigma^*, \cdot, \varepsilon)$ forms a monoid.

Inductive definitions

Definitions

Axioms and rules We can inductively define a subset of a given set U with *axioms* and *rules*, written

$$\frac{}{a} \qquad\qquad \frac{h_1 \quad h_2 \quad \cdots \quad h_n}{c}$$

where a (axiom), $h_1, h_2, \ldots, h_n$ (hypotheses), and c (conclusion) are all elements of U.

Derivation Given a set of axioms and rules for inductively defining a subset of a set U, a *derivation* or *proof* that a particular element $u \in U$ is in the subset is a finite rooted tree with vertices labelled by elements of U such that:

- the root of the tree is u (the conclusion of the derivation)
- each vertex of the tree is the conclusion of a rule whose hypotheses are the children of the vertex
- each leaf of the tree is an axiom

A derivation for $u \in U$ is therefore a concrete proof that the axioms can be combined and translated into u via the rules.

Inductively defined subset Given a set of axioms and rules over U, the subset of U *inductively defined* by the axioms and rules consists of all and only the elements $u \in U$ for which there is a derivation with the conclusion u.

Theorem

Closure property Given a set U and a collection of axioms and rules, the inductively defined subset $I \subseteq U$ is the closure of the set of axioms under the application of rules. That is, I is closed under the rules (applying any rule to any elements of I will yield a conclusion already in I), and is the smallest such subset (for any other closed $S \subseteq U$, $I \subseteq S$). This property gives rise to the proof technique of rule induction.

Regular expressions

Definitions

Regular expression A symbolic description of a collection of strings of a language. Given an alphabet Σ, the concrete syntax of *regular expressions* on Σ is the subset of $U \triangleq (\Sigma \cup \{\epsilon, \emptyset, |, *, (,)\})^*$ inductively defined by the following axioms and rules:

$$\frac{}{a} \qquad \frac{}{\epsilon} \qquad \frac{}{\emptyset} \qquad \frac{r}{(r)} \qquad \frac{r \quad s}{r \,|\, s} \qquad \frac{r \quad s}{rs} \qquad \frac{r}{r^*}$$

Matching Each regular expression r over an alphabet Σ induces a language $L(r) \subseteq \Sigma^*$. The strings u in $L(r)$ are by definition the ones that *match* r, where:

- a for a symbol $a \in \Sigma$ is matched iff $u = a$
- ϵ is matched iff u is the empty string ε
- $\emptyset$ is not matched by any string
- $r \,|\, s$ is matched iff u either matches r or it matches s
- rs is matched iff u can be expressed as the concatenation $u = vw$ where v matches r and w matches s
- r^* is matched iff either $u = \varepsilon$ or $u = r$ or u can be expressed as the concatenation of two or more strings, each of which matches r

Finite automata

Definitions

Non-deterministic finite automaton (NFA) A 5-tuple $M = (Q, \Sigma, \Delta, s, F)$, where:

- Q is a finite set of *states*
- Σ is the finite alphabet of *input symbols*
- $\Delta \subseteq Q \times \Sigma \times Q$ is the *transition relation*
- $s \in Q$ is the *start state*
- $F \subseteq Q$ is the set of *accepting states*

We write $q \xrightarrow{a} q'$ for $(q, a, q') \in \Delta$.

Deterministic finite automaton (DFA) An NFA $M = (Q, \Sigma, \Delta, s, F)$ with the property that for each state $q \in Q$ and each input symbol $a \in \Sigma$, there is a unique state $q' \in Q$ satisfying $q \xrightarrow{a} q'$. That is, Δ can be expressed as a *next-state function* $\delta : Q \times \Sigma \to Q$ where $\forall q'.\ q \xrightarrow{a} q' \iff q' = \delta(q, a)$.

ε-transitions State transitions $q \xrightarrow{\varepsilon} q'$ that do not consume any input symbol.

NFA with ε-transitions (NFA$^\varepsilon$) A tuple $M = (Q, \Sigma, \Delta, s, F, T)$ where $(Q, \Sigma, \Delta, s, F)$ is an NFA and $T \subseteq Q \times Q$ is the ε-transition relation with elements $(q, q') \in T$ written as $q \xrightarrow{\varepsilon} q'$.

Language accepted by an NFA$^\varepsilon$ We write $q \overset{u}{\Rightarrow} q'$ to mean that there is a path in an NFA$^\varepsilon$ $M = (Q, \Sigma, \Delta, s, F, T)$ from state q to state q' whose non-ε labels form the string $u \in \Sigma^*$. A string $u \in \Sigma^*$ is *accepted* by M if there exists a state $t \in F$ such that $s \overset{u}{\Rightarrow} t$. A *language* $L(M)$ *accepted by* M is the set of all strings accepted by M.

Theorem

Subset construction For each NFA$^\varepsilon$ $M = (Q, \Sigma, \Delta, s, F, T)$ there is a DFA $PM = (\mathcal{P}(Q), \Sigma, \delta, s', F')$ with $L(PM) = L(M)$ defined as follows:

- a state of PM is a set of states of M
- the input alphabet is the same Σ as for M
- the next-state function $\delta : \mathcal{P}(Q) \times \Sigma \to \mathcal{P}(Q)$ defined as:

$$\delta(S, a) \triangleq \left\{ q' \in Q \;\middle|\; \exists q \in S.\ q \overset{a}{\Rightarrow} q' \text{ in } M \right\}$$

- the start state is $s' \triangleq \left\{ q' \in Q \;\middle|\; s \overset{\varepsilon}{\Rightarrow} q' \right\}$
- the set of accepting states is $F' \triangleq \{ S \subseteq Q \mid S \cap F \neq \emptyset \}$

Since every DFA is an NFA$^\varepsilon$ and the *subset construction* above creates a DFA from any NFA$^\varepsilon$, the class of languages accepted by DFAs and NFA$^\varepsilon$s is the same – nondeterminism does not add extra power.

Regular languages

Regular language A language is *regular* iff it is equal to $L(M)$ for some DFA M.

Theorems

Kleene's Theorem The class of regular languages equals the class of languages accepted by a regular expression. That is:

1. *For every regex r, $L(r)$ is a regular language.* We establish this by constructing an NFA$^\varepsilon$ for any regular expression r by structural recursion on r.
2. *Every regular language is of the form $L(r)$ for some regex r.* We establish this by inductively defining a regex $r^S_{q,q'}$ for an NFA $M = (Q, \Sigma, \Delta, s, F)$ satisfying:

   $$L\left(r^S_{q,q'}\right) \triangleq \left\{ u \in \Sigma^* \;\middle|\; q \xrightarrow{u}{}^* q' \text{ in } M \text{ with all intermediate states of the transitions in } S \right\}$$

   and constructing the regex $r \triangleq r^Q_{s,t_1} | \cdots | r^Q_{s,t_k}$ for all accepting states $t_i \in F$.

Closure properties Regular languages are closed under the following operations:

Union $L(r) \cup L(s) = L(r \mid s)$

Concatenation $L(r)L(s) = \{ vw \in \Sigma^* \mid v \in L(r) \wedge w \in L(s) \} = L(rs)$

Kleene star $L(r)^* = \{ \varepsilon \} \cup \{ vw \in \Sigma^* \mid v \in L(r) \wedge w \in L(r)^* \} = L(r^*)$

Complementation If L is a regular language over Σ accepted by a DFA $M = (Q, \Sigma, \delta, s, F)$, then its complement $L^{\complement} \triangleq \{ u \in \Sigma^* \mid u \notin L \}$ is also regular, since it is accepted by $Not(M) \triangleq (Q, \Sigma, \delta, s, Q \setminus F)$. The corresponding regular expression is denoted $\sim r$, matched by u iff u does not match r.

Intersection If L_1 and L_2 are regular languages over Σ, then their intersection $L_1 \cap L_2 \triangleq \{ u \in \Sigma^* \mid u \in L_1 \wedge u \in L_2 \}$ is also regular. This can be established either by constructing the DFA $And(M_1, M_2)$ directly, or observing that by the de Morgan law $L_1 \cap L_2 = \left(L_1^{\complement} \cup L_2^{\complement}\right)^{\complement}$ and regular languages are closed under union and complementation. The corresponding regular expression is denoted $r \,\&\, s$, matched by u iff u matches both r and s.

Decidability of matching Determining whether a string u matches a regular expression r is decidable by converting r into an NFA$^{\varepsilon}$ M using Kleene's Theorem, converting M into a DFA PM via the subset construction, and checking whether PM accepts u.

Decidability of equivalence Determining whether two regular expressions r and s accept the same set of languages is decidable, since $L(r) = L(s)$ if and only if $L((\sim r) \,\&\, s) = \emptyset$, and checking whether the DFA corresponding to the regex $(\sim r) \,\&\, s$ accepts any strings can be done in a finite number of queries.

Pumping Lemma For every regular language L, there is a number $\ell \in \mathbb{Z}^+$ satisfying the *pumping lemma property*: all $w \in L$ with $|w| \geq \ell$ can be expressed as a concatenation of three strings $w = u_1 v u_2$ which satisfy (1) $v \neq \varepsilon$, (2) $|u_1 v| \leq \ell$, and (3) for all $n \in \mathbb{N}$, $u_1 v^n u_2 \in L$.

To prove that a language L is *not* regular, we can establish the negation of the Pumping Lemma: for each $\ell \geq 1$, there is some $w \in L$ with $|w| \geq \ell$ such that no matter how w is split into three strings $w = u_1 v u_2$, with $v \neq \varepsilon$ and $|u_1 v| \leq \ell$, there is some $n \in \mathbb{N}$ for which $u_1 v^n u_2 \notin L$.