Glossary

accepted by The same concept as 'accepts' but considered in the reverse direction, e.g. "the language accepted by a finite automaton". L04:6, *see* accepts

accepting state A distinguished state of a finite state automaton (a DFA or NFA). When a string that is input to the automaton ends, that string is in the language "accepted" by the automaton if and only if the automaton is left in an accepting state. There can be multiple accepting states. L01:15, L01:17, L02:2, L04:6, N04:5

accepts An overloaded term that means either (i) that a specific finite automaton or Turing machine accepts an input string (by leaving the machine in an accepting state), (ii) that an automaton accepts precisely the strings in a given language, or (iii) that a class of automata accepts precisely the set of languages in a given class of language. The term *recognises* is used as a synonym of *accepts.* N04:5

Σ^* The set of all strings over an alphabet Σ. L02:3, *see* alphabet

alphabet The set of characters that can appear in the the strings comprising a formal language. L01:15, L01:17, L02:2, L02:6, L04:6, N04:1

anti-halting machine An (impossible) Turing machine that is constructed from a halting-machine in the proof that the halting problem is undecidable. L15:6, *see* halting problem & halting machine

augmenting path An augmenting path for a matching M is a path that starts and finishes at an unmatched vertex and whose edges alternate between not belonging to M and belonging to M. L23:9

bipartite graph A graph whose vertices can be partitioned into two sets A and B such that every edge has one endpoint in A and the other in B. L23:2

***c*-approximation algorithm** An algorithm for an optimisation problem that produces a feasible solution (i.e., an object of the desired type) whose objective value is at most k times the optimum. L24:13

Church-Turing thesis L11:3, N14:1

clause A formula in propositional logic that is a disjunction of literals, i.e. Boolean variables and negated Boolean variables. L18:5, *see* literal

closure property A property of a set and an operation defined on that set that states that applying the operation to members of the set results in another member of the set. An example in automata theory is that the language accepted by an NFA is closed under the operation of reversing strings. L04:13, N05:2, N08:7

CNF conjunctive normal form.

coarser If $\sim$ and $\equiv$ are equivalence relations on the same set, $\sim$ is *coarser* than $\equiv$ if $x \equiv y$ implies $x \sim y$. L06:6, *see* equivalence relation

computes A Turing machine M computes a function from some subset of Σ^* to some other subset of Σ^* if, when the tape is initialised to w in the domain of f, running $\mathbf{M}$ leaves the tape in final state $w\#f(w)$ (or possibly just $f(w)$) and halts and accepts. L12:2, N10:1, *see* Turing machine

conjunctive normal form A formula in propositional logic is in *conjunctive normal form* if it is a conjunction of clauses. L18:5

context-free grammar A grammar in which productions have the form `non-terminal` $\longrightarrow$ `any word in terminals and non-terminals`. L09:8, N08:4

context-free language The class of languages that can be accepted by a pushdown automaton or a context-free grammar. L09:9, *see* pushdown automaton & context-free grammar

Cook-Levin theorem A theorem shown that the Sat problem is **NP**-complete. L19:2, *see* satisfiability

decidable language Another term for a recursive language. L14:3, *see* recursive

decision problem A computational problem that can be posed as a yes-no question based on the given input values [Wikipedia.

decision procedure An algorithm that correctly answers instances of a decision problem. L14:3, *see* decision problem

deterministic finite state automaton A finite-state machine that accepts or rejects a given string of symbols by following the unique sequence of transition between states that are triggered by each element of the string in turn. L01:17, L02:2, N04:5

DFA Deterministic finite state automaton. L01:17

DFA minimisation The problem of transforming a given deterministic finite automaton (DFA) into an equivalent DFA that has a minimum number of states [Wikipedia]. L07:9, L07:10, N06:4:4, *see* Moore's algorithm & Hopcroft's algorithm

distinguishing extension Given a language L over alphabet Σ and two words $u, v \in \Sigma^*$, this is a term for any word w such that exactly one of uw and vw is in L. L06:10, L07:4, *see* suffix equivalence

enumerates A Turing machine *enumerates* a language L if it writes (all and only) the elements of L on an output tape (eventually). L14:5, N12:2

enumerates L by length A Turing machine enumerates a language L by length if it enumerates L and in the output all the words of length 0 precede those of length 1 which precede those of length 2 and so on. L14:7, N12:3

ϵ The string with length zero, known as the empty string. L02:3

equivalence class The maximal subset of a set in which all elements are related to each other by some *equivalence relation.* Any equivalence relation $\sim$ on a set partitions it into disjoint equivalence classes. L06:7, *see* equivalence relation

equivalence relation A binary relation on a set that is reflexive, symmetric and transitive. It is commonly used to partition a set into *equivalence classes.* L06:4, *see* equivalence class

final state An alternative term for an accepting state of a finite automata. L01:17, L02:2

finer than If $\sim$ and $\equiv$ are equivalence relations on the same set, $\equiv$ is finer than $\sim$ if $x \equiv y$ implies $x \sim y$. The term *refines* is used as a synonym for "finer than". L06:6, *see* equivalence relation

finite language A language containing a finite number of words L04:15

fixed parameter tractable A problem is fixed-parameter tractable if it can be solved in polynomial time if one parameter is 'fixed' to a constant value. L24:3

generates A term used instead of 'accepts' in the context of formal grammars. A grammar is said to generate a string or all strings in a language, and a class of grammar is said to generate a class of language. *see* accepts

genie A type of supernatural creature from Muslim mythology that does whatever the person who controls it asks. This concept can be used as metaphor when thinking about non-deterministic computing. L04:10

grammar A set of "production rules" that defines a language. The strings in the language are those that can be generated from a specified start symbol by repeated non-deterministic application of rules. *see* start symbol, terminal symbol, non-terminal symbol & production rule

Hall's theorem A theorem stating that a bipartite graph G with parts A and B has a matching that saturates A if and only if for every subset X of A, the set of neighbours of X is at least as large as X. L23:5, L23:6, L23:7, L23:8

halting language The language HALT, consisting of all (and only) the strings $R(M)\#w$, where M is a Turing machine, $R(M)$ is a representation of M and M halts on w. L14:10, L15:2, N12:5, *see* representation

halting machine A Turing machine that implements a decision procedure for the halting language. Its existence is assumed in the proof that the halting problem is undecidable, leading to a contradiction. L15:4, *see* halting language

halting problem The problem of determining whether the halting language is recursive, i.e. can a Turing machine decide, given as input a representation of any Turing machine M and an input w for that machine, whether M will halt when run on input w. L15:2

Hopcroft's algorithm An algorithm for DFA minimisation L07:14, N06:6, *see* DFA minimisation

Hopcroft-Karp algorithm An algorithm that computes a maximum matching for a bipartite graph. L23:10, *see* maximum matching

hypergraph A set of vertices and *hyperedges* where a hyperedge is a subset of the vertices (and not restricted to only be a pair). L23:12

initial state Another term used for the start symbol of a deterministic finite state automaton. N04:5, *see* start symbol

k-SAT A version of the satisfiability problem (SAT) in which the input formula contains exactly k distinct literals in each clause. L20:3, *see* satisfiability

k-uniform If each hyperedge of a hypergraph contains k elements then the hypergraph is called k-uniform. L23:12, *see* hypergraph

language A set of strings (also known as words) that are formed from concatenating symbols, letters or tokens from an alphabet. A language can be defined by a finite automaton or by a formal grammar. L01:15, L02:2, L02:7, N04:2, N04:5, *see* string, word & accepting state

literal In logic, a literal is a atomic formula or its negation. In propositional logic, the atoms are variables, so the literals are variables and negated variables. L18:5

matching A set of edges in a bipartite graph in which two edges share a common endpoint. In a hypergraph, a matching is a disjoint set of hyperedges. L23:2, *see* bipartite graph

maximum matching A matching whose size is as large as it possibly can be for a given bipartite graph. L23:2, *see* matching

Moore's algorithm An algorithm for DFA minimisation L07:11, N06:4, *see* DFA minimisation

multi-tape Turing machine A Turing machine with multiple tapes, each with its own independent read-write head. Transitions depend on the current state and the sequence of symbols currently read from the tapes. L13:5, N11:2, *see* Turing machine

multi-track Turing machine A Turing machine with multiple tapes and a single read-write head that reads and writes on all tapes simultaneously before moving in the same direction on all tapes. L13:3, N11:1, *see* Turing machine

Myhill-Nerode theorem A theorem showing that a language is regular if and only if its suffix-equivalence relation has only finitely many equivalence classes. L06:11, L07:5, N06:1, *see* suffix equivalence

NDO Non-determinism by oracle.

NDT Non-determinism by transition.

NFA Non-deterministic finite state automaton.

non-determinism by oracle A mechanism to create a non-deterministic Turing machine. Two tapes are used. The input is written on the main tape and a black box (or genie) then, in a single step, writes something on an oracle tape that the TM can use as a guide in its verification that the input word is in the language of the TM. L13:8, N11:3, *see* Turing machine & oracle tape

non-determinism by transition A mechanism to create a non-deterministic Turing machine whereby multiple transitions are allowed for a given state/symbol pair. L13:8, N11:3, *see* Turing machine

non-deterministic finite state automaton A type of finite-state machine in which the transition function is not required to be a function, i.e.any state may have multiple alternative transitions defined for the same symbol. L04:6

non-terminal symbol A symbol that can be replaced by applying a production rule in a formal grammar. Non-terminal symbols do not appear in any of the strings generated by the grammar. Also known as a *syntactic variable*. L02:6, *see* grammar & production rule

NP The class of decision problems that can be resolved by a non-deterministic Turing machine whose running time is bounded by a polynomial in the input size. L16:7, N14:7

NP-complete A decision problem is **NP**-complete if it is **NP**-hard and is also a member of the complexity class **NP**. Thus, **NP**-complete problems are harder than **NP** problems in general. L17:6, *see* **NP**-hard

NP-hard A computational problem H is called **NP**-hard if, for every problem L which can be solved in non-deterministic polynomial-time, there is a polynomial-time reduction from L to H [Wikipedia]. Thus, solving H in deterministic polynomial time would mean that $\mathbf{P} = \mathbf{NP}$. L17:6

oracle tape In the context of a Turing machine (TM), an oracle tape is an extra tape that is a mechanism to introduce non-determinism. In the version discussed in COSC341, when the output is written on the main tape, a black box (or genie) then writes a solution to a specific decision problem or function call on the oracle tape in a single step. The machine accepts all strings for which it halts and accepts given *some* word the genie can write on the oracle tape. L13:8, N11:3, *see* Turing machine

P The class of decision problems that can be resolved by a deterministic Turing machine whose running time is bounded by a polynomial in the input size. L16:5, N14:6

partition A partition of a set is a grouping of its elements into non-empty subsets, in such a way that every element is included in exactly one subset [Wikipedia]. L06:7

PDA Pushdown automaton.

perfect matching A matching in which every vertex of the bibartite graph is the endpoint of some edge in the matching. In a hypergraph, a perfect matching is one where the union of hyperedges is the set of all vertices. L23:2

polynomial-time reduction An algorithm for converting instances of one decision problem to another that preserves positive and negative instances and has polynomial time complexity. L17:4

product automaton A finite-state automaton constructed P from two existing automata (A and B, say). P's set of states is the cross product of the states of A with the states of B. The pair of the individual automatas' start states is the start state of P and there is a transition $(U, X) \xrightarrow{a} (V, Y)$ in P if $U \xrightarrow{a} V$ in A and $X \xrightarrow{a} Y$ in B. To accept the union of the languages of A and B, the accepting states of P are the states (U, X) where U is an accepting state of A or X is an accepting state of B. Replace "or" with "and" to accept the intersection of $L(A)$ and $L(B)$. S02:2

production rule A production rule "specifies a replacement of a particular string (its left-hand side) with another (its right-hand side)". "A rule can be applied to each string that contains its left-hand side and produces a string in which an occurrence of that left-hand side has been replaced with its right-hand side" [Quotes from Wikipedia (Formal Grammar)]. L02:6

Pumping lemma for context-free languages Describes a requirement for a language to be context-free: all sufficiently long strings in the language must have a pair of substrings that can be repeated arbitrarily many times. This is usually used to prove that a certain language is not context-free. L09:14, N08:4:4

Pumping lemma for regular languages Describes a requirement for a language to be regular: all sufficiently long strings in the language must have a substring that can be repeated arbitrarily many times. This is usually used to prove that a certain language is not regular. L09:12, N06:3:3

pushdown automaton A finite state machine augmented with a stack of unbounded capacity. Transitions depend on the symbol on the top of the stack as well as the state and current input symbol, and as well as causing a state change, they can push or pop from the stack L09:3, N08:1

queue machine An automaton similar to a pushdown automaton but using an unbounded queue rather than a stack. An alternative term is *pullup automaton.* L09:19, N09:1

recognises An alternative word that is sometimes used instead of *accepts.* N04:5, *see* accepts

recursive A language L is *recursive* if there is some TM that halts on all inputs and accepts only (and exactly) the words in L. L14:3, N12:1

recursively enumerable A language L is *recursively enumerable* if there is some Turing machine that accepts it. L14:2, N12:1

refines If $\sim$ and $\equiv$ are equivalence relations on the same set, $\equiv$ refines $\sim$ if $x \equiv y$ implies $x \sim y$. This is a synonym for "finer than". L06:6, *see* equivalence relation

reflexive A binary relation $\sim$ on a set S is *reflexive* if $a \sim a$ for all $a \in S$. L06:4, *see* equivalence relation

regular expression A notation for defining a regular language. L04:15, *see* regular language

regular language The class of languages over an alphabet Σ that contains the languages $\{\}$, $\{\epsilon\}$ and $\{a\}$ (for all $a \in \Sigma$), is closed under the operations of adding all concatenations of strings within a language, concatenating strings from one language with strings from another language, and set union, and which contains no other languages. L04:15, N05:1, *see* non-deterministic finite state automaton

rejects The opposite of 'accepts' in the context of an automaton processing an input string. If the automaton does not end in an accepting state, the automaton is said to reject the string. N04:5

representation A symbolic description of a Turing machine that completely defines it. L14:9, N12:3

Rice's theorem A theorem that generalises the proof that the halting problem is undecidable. It states that for any subset X of the recursively-enumerable languages that is neither empty nor the set of all languages then the problem "Does $L(M)$ belong to X?" is undecidable. L15:13

right-regular grammar A grammar in which every production rules has one of the following forms: $A \to a$, $A \to aB$ or $A \to \epsilon$. L02:6, *see* grammar, non-terminal symbol & production rule

Russell's paradox A paradox in set theory showing that a contradiction results if you make the assumption that for any well-defined property there must be a set of all and only the objects that have that property. The paradox arises from considering the set of all sets that are not members of themselves. L15:3

satisfiability The decision problem of determining whether an input formula in conjunctive normal form over a set of Boolean variables is satisfiable. L18:5, L18:8, *see* satisfiable

satisfiable A formula in propositional logic is *satisfiable* if it evaluates to *true* for *some* truth assignment. L18:5

satisfying assignment A truth assignment for which a given formula in propositional logic evaluates to *true*. L18:5, *see* truth assignment

saturates A matching *saturates* a set of vertices of a bipartite graph if every vertex in the set is the endpoint of an edge in the matching. L23:2

start state The state that a specific finite state automaton (a DFA or NFA) starts in. L01:17, L02:2, L04:6

start symbol The initial non-terminal symbol representing any expression the grammar can generate. The generation process starts by non-deterministically choosing a production rule that has this symbol on its left-hand side. Also called the initial state. L02:6

state A component of a finite state automaton that is used as a simple form of memory. It records that the string of input symbols so far has some property that distinguishes it from other states. L01:15, L01:17, L02:2, L04:6, N04:5

state equivalence An equivalence relation $\sim_{\text{state}}$ on strings over the alphabet Σ of a deterministic finite state automaton A. $w \sim_{\text{state}} v$ if and only if the the reached in $\mathbf{A}$ by processing v is the same as that reached by processing w. L06:8, L07:3, *see* equivalence relation

string A string of length n over alphabet Σ is a sequence $\sigma_0\sigma_1 \ldots \sigma_{n-1}$ where each $\sigma_i \in \Sigma$. L02:3, N04:1, *see* word

subset construction algorithm An algorithm for converting a non-deterministic finite state automaton (DFA) to a deterministic finite state automaton (DFA), where the DFA's states are the subsets of the set of states of the NFA. L05:11

suffix equivalence An equivalence relation $\sim_{\text{suffix}}$ on strings over the alphabet Σ of a deterministic finite state automaton A. $u \sim_{\text{suffix}} v$ if and only if there is no word w such that one of uw and vw is accepted by $\mathbf{A}$ and the other is not. Suffix equivalence can also be viewed as a relation on a words in a language without reference to a DFA. L06:9, L06:10, L07:4, *see* equivalence relation

suffix language The suffix language of a word w modulo a language L (with alphabet Σ) is the set $\{y \in \Sigma^* : wy \in L\}$, i.e.the set of words that can be appended to w to produce a word in L. L06:10, L07:4, N06:1, *see* suffix equivalence

symmetric A binary relation $\sim$ on a set S is *symmetric* if for all x and y in S, if $x \sim y$ then $y \sim x$. L06:4, *see* equivalence relation

terminal symbol A symbol that can appear within the strings generated by a formal grammar. It cannot be replaced by the application of any production rule. *see* grammar & production rule

time complexity In the context of a Turing machine, this is the maximum number of transitions required for the operation of M on any input string of length n. It is written as $tc(n)$. L16:2, N14:1

transition A change of state within a finite state automaton that is triggered by inputting a symbol. L01:15, L04:6

transition function A component of a finite state automaton (a DFA or NFA) that defines, for a given state and a symbol, the new state (for a DFA) or states (for an NFA) reached when that symbol is input. L01:17, L02:2, N04:5

transitive A binary relation $\sim$ on a set S is *transitive* if for all x, y and z in S, if $x \sim y$ and $y \sim z$ then $z \sim z$. L06:4, *see* equivalence relation

truth assignment A map that assigns a value of *true* or *false* to each member of a set of Boolean variables.

Turing machine a mathematical model of computation describing an abstract machine that manipulates symbols on a strip of tape according to a table of rules. Despite the model's simplicity, it is capable of implementing any computer algorithm [Wikipedia]. L11:4, L11:7, N09:2:2

Turing reducible L is Turing-reducible to K, denoted $L \xrightarrow{TR} K$, if there is a TM, M that always halts and on input w leaves some other word $r(w)$ on the tape in such a way that $w \in L$ if and only if $r(w) \in K$. L15:11

universal Turing machine A Turing machine that takes as input a description of another Turing machine followed by an input for that machine, and can then simulate it running with that input. L14:9, N12:4

verifier A term referring to a non-deterministic Turing machine that solves a decision problem. L16:7

vertex cover A set of vertices in a graph such that every edge has at least one endpoint in the set. L24:5

word Another name used for a string in formal language theory. N04:1, *see* string

ϵ-closure The ϵ-closure of a state S in a non-deterministic finite state automaton is the set of all states that can be reached from S using only ϵ-transitions. L05:11, *see* ϵ-transition

ϵ-transition A transition in a non-deterministic finite state automaton that is triggered by the empty string ϵ. L04:6, *see* ϵ & non-deterministic finite state automaton