will already fail for reasonably small n, if implemented in practice. For example, for the well-established 9×9-Sudoku puzzles the algorithm will in the worst case need about $9^{81} \approx 2 \cdot 10^{77}$ rule applications to figure out whether a given Sudoku has a solution. This way, assuming a fast computer that can perform 1 Million rule applications per second it will take longer to solve a single Sudoku than the currently estimated age of the universe. Nevertheless, human beings typically solve a 9×9-Sudoku in some minutes. So what is wrong here? First of all, as I already said, the algorithm presented in Section 1.1 is completely naive. This algorithm will definitely not solve 9×9-Sudokus in reasonable time. It can be turned into an algorithm that will work nicely in practice, see Exercise (1.2). Nevertheless, problems such as Sudokus are difficult to solve, in general. Testing whether a particular assignment is a solution can be done efficiently, in case of Sudokus in time $O(n^2)$. For the purpose of this book, I say a problem can be *efficiently solved* if there is an algorithm solving the problem and a polynomial $p(n)$ so that the execution time on inputs of size n is $O(p(n))$. Although it is efficient for Sudokus to validate whether an assignment is a solution, there are exponentially many possible assignments to check in order to figure out whether there exists a solution. So if we are allowed to make guesses, then Sudokus can be solved efficiently. This property describes the class of NP (Nondeterministic Polynomial) problems in general that I will introduce now.

A *decision problem* is a subset $L \subseteq \Sigma^*$ for some fixed finite alphabet Σ. The function $\mathrm{chr}(L, x)$ denotes the *characteristic function* for some decision problem L and is defined by $\mathrm{chr}(L, u) = 1$ if $u \in L$ and $\mathrm{chr}(L, u) = 0$ otherwise. A decision problem is solvable in polynomial-time iff its characteristic function can be computed in polynomial-time. The class P denotes all polynomial-time decision problems.

Definition 1.3.2 (NP)**.** A decision problem L is in NP iff there is a predicate $Q(x, y)$ and a polynomial $p(n)$ so that for all $u \in \Sigma^*$ we have (i) $u \in L$ iff there is an $v \in \Sigma^*$ with $|v| \leq p(|u|)$ and $Q(u, v)$ holds, and (ii) the predicate Q is in P.

A decision problem L is *polynomial time reducible* to a decision problem L' if there is a function $g \in \mathrm{P}$ so that for all $u \in \Sigma^*$ we have $u \in L$ iff $g(u) \in L'$. For example, if L is reducible to L' and $L' \in \mathrm{P}$ then $L \in \mathrm{P}$. A decision problem is NP-*hard* if every problem in NP is polynomial time reducible to it. A decision problem is NP-*complete* if it is NP-hard and in NP. Actually, the first NP-complete problem [7] has been propositional satisfiability (SAT). Chapter 2 is completely devoted to solving SAT.

1.3.4 Word Grammars

When Gödel presented his undecidability proof on the basis of arithmetic, many people still believed that the construction is so artificial that such problems will never arise in practice. This didn't change with Turing's invention of the Turing machine and the undecidable halting problem of such a machine. However, then

Post presented his correspondence problem in 1946 [18] it became obvious that undecidability is not an artificial concept.

Definition 1.3.3 (Finite Word)**.** Given a nonempty alphabet Σ the set Σ^* of *finite words* over Σ is defined by

1. the empty word $\epsilon \in \Sigma^*$
2. for each letter $a \in \Sigma$ also $a \in \Sigma^*$
3. if $u, v \in \Sigma^*$ so $uv \in \Sigma^*$ where uv denotes the concatenation of u and v.

Definition 1.3.4 (Length of a Finite Word)**.** The length $|u|$ of a word $u \in \Sigma^*$ is defined by

1. $|\epsilon| := 0$,
2. $|a| := 1$ for any $a \in \Sigma$ and
3. $|uv| := |u| + |v|$ for any $u, v \in \Sigma^*$.

Definition 1.3.5 (Word Embedding)**.** Given two words u, v, then u is *embedded* in v written $u \sqsubseteq v$ if for $u = a_1 \dots a_n$ there are words $v_0, \dots, v_n$ such that $v = v_0 a_1 v_1 a_2 \dots a_n v_n$.

Reformulating the above definition, a word u is embedded in v if u can be obtained from v by erasing letters. For example, *higman* is embedded in *highmountain*.

Definition 1.3.6 (PCP)**.** Given two finite lists of words $(u_1, \dots, u_n)$ and $(v_1, \dots, v_n)$ the *Post Correspondence Problem* (PCP) is to find a finite index list $(i_1, \dots, i_k)$, $1 \leq i_j \leq n$, so that $u_{i_1} u_{i_2} \dots u_{i_k} = v_{i_1} v_{i_2} \dots v_{i_k}$.

Take for example the two lists (a, b, bb) and (ab, ab, b) over alphabet $\Sigma = \{a, b\}$. Then the index list $(1, 3)$ is a solution to the PCP with common word abb.

Theorem 1.3.7 (Post 1942)**.** PCP is undecidable.

Lemma 1.3.8 (Higman's Lemma 1952)**.** For any infinite sequence of words $u_1, u_2, \dots$ over a finite alphabet there are two words u_k, u_{k+l} such that $u_k \sqsubseteq u_{k+l}$.

Proof. By contradiction. Assume an infinite sequence $u_1, u_2, \dots$ such that for any two words u_k, u_{k+l} they are not embedded, i.e., $u_k \not\sqsubseteq u_{k+l}$. Furthermore, I assume that the sequence is minimal at any word with respect to length, i.e., considering any u_k, there is no infinite sequence with the above property that shares the words up to u_{k-1} and then continues with a word of smaller length than u_k. Next, the alphabet is finite, so there must be a letter, say a that occurs infinitely often as the first letter of the words of the sequence. The words starting with a form an infinite subsequence $au'_{k_1}, au'_{k_2}, \dots$ where $u_{k_i} = au'_{k_i}$.

This infinite subsequence itself has the non-embedding property, because it is a subsequence of the originial sequence. Now consider the infinite sequence $u_1, u_2, \ldots, u_{k_1-1}, u'_{k_1}, u'_{k_2}, \ldots$. Also this sequence has the non-embedding property: if some $u_i \sqsubseteq u'_{k_j}$ then $u_i \sqsubseteq a u'_{k_j}$ contradicting that the starting sequence is non-embedding. But then the constructed sequence contradicts the minimality assumption with respect to length, finishing the proof. □

Definition 1.3.9 (Context-Free Grammar)**.** A context-free grammar $G = (N, T, P, S)$ consists of:

1. a set of non-terminal symbols N
2. a set of terminal symbols T
3. a set P of rules $A \Rightarrow w$ where $A \in N$ and $w \in (N \cup T)^*$
4. a start symbol S where $S \in N$

For rules $A \Rightarrow w_1$, $A \Rightarrow w_2$ we write $A \Rightarrow w_1 \mid w_2$.

Given a context free grammar G and two words $u, v \in (N \cup T)^*$ I write $u \Rightarrow v$ if $u = u_1 A u_2$ and $v = u_1 w u_2$ and there is a rule $A \Rightarrow w$ in G. The *language* generated by G is $L(G) = \{w \in T^* \mid S \Rightarrow^* w\}$, where $\Rightarrow^*$ is the reflexive and transitive closure of $\Rightarrow$.

A context free grammar G is in *Chomsky Normal Form* [6] if all rules are if the form $A \Rightarrow B_1 B_2$ with $B_i \in N$ or $A \Rightarrow w$ with $w \in T^*$. It is said to be in *Greibach Normal Form* [12] if all rules are of the form $A \Rightarrow a w$ with $a \in T$ and $w \in N^*$.

1.4 Orderings

An ordering R is a binary relation on some set M. Depending on particular properties such as

(reflexivity)	$\forall x \in M\ R(x, x)$
(irreflexivity)	$\forall x \in M\ \neg R(x, x)$
(antisymmetry)	$\forall x, y \in M\ (R(x, y) \wedge R(y, x) \rightarrow x = y)$
(transitivity)	$\forall x, y, z \in M\ (R(x, y) \wedge R(y, z) \rightarrow R(x, z))$
(totality)	$\forall x, y \in M\ (R(x, y) \vee R(y, x))$

there are different types of orderings. The relation $=$ is the identity relation on M. The quantifier $\forall$ reads "for all", and the boolean connectives $\wedge$, $\vee$, and $\rightarrow$ read "and", "or", and "implies", respectively. For example, the above formula stating reflexivity $\forall x \in M\ R(x, x)$ is a shorthand for "for all $x \in M$ the relation $R(x, x)$ holds".

Actually, the definition of the above properties is informal in the sense that I rely on the meaning of certain symbols such as $\in$ or $\rightarrow$. While the former is assumed to be known from school math, the latter is "explained" above. So, strictly speaking this book is neither self contained, nor overall formal. For the concrete logics developed in subsequent chapters, I will formally define $\rightarrow$ but here, where it is used to state properties needed to eventually define the notion of an ordering, it remains informal. Although it is possible to develop the overall content of this book in a completely formal style, such an approach is typically impossible to read and comprehend. Since this book is about teaching a general framework to eventually generate automated reasoning procedures this would not be the right way to go. In particular, being informal starts already with the use of natural language. In order to support this "mixed" style, examples and exercises deepen the understanding and rule out potential misconceptions.

Now, based on the above defined properties of a relation, the usual notions with respect to orderings are stated below.

Definition 1.4.1 (Orderings)**.** A *partial ordering* $\succeq$ (or simply ordering) on a set M, denoted $(M, \succeq)$, is a reflexive, antisymmetric, and transitive binary relation on M. It is a *total ordering* if it also satisfies the totality property. A *strict ordering* $\succ$ is a transitive and irreflexive binary relation on M. A strict ordering is *well-founded*, if there is no infinite descending chain $m_0 \succ m_1 \succ m_2 \succ \ldots$ where $m_i \in M$.

Given a strict partial order $\succ$ on some set M, its respective partial order $\succeq$ is constructed by taking the transitive closure of $(\succ \cup =)$.

Example 1.4.2. The well-known relation $\leq$ on $\mathbb{N}$, where $k \leq l$ if there is a j so that $k + j = l$ for $k, l, j \in \mathbb{N}$, is a total ordering on the naturals. Its strict subrelation $<$ is well-founded on the naturals. However, $<$ is not well-founded on $\mathbb{Z}$.

Definition 1.4.3 (Minimal and Smallest Elements)**.** Given a strict ordering $(M, \succ)$, an element $m \in M$ is called *minimal*, if there is no element $m' \in M$ so that $m \succ m'$. An element $m \in M$ is called *smallest*, if $m' \succ m$ for all $m' \in M$ different from m.

Note the subtle difference between minimal and smallest. There may be several minimal elements in a set M but only one smallest element. Furthermore, in order for an element being smallest in M it needs to be comparable to all other elements from M.

Example 1.4.4. In $\mathbb{N}$ the number 0 is smallest and minimal with respect to $<$. For the set $M = \{q \in \mathbb{Q} \mid q \geq 5\}$ the ordering $<$ on M is total, has the minimal element 5 but is not well-founded.

If $<$ is the ancestor relation on the members of a human family, then $<$ typically will have several minimal elements, the currently youngest children of the family, but no smallest element, as long as there is a couple with more than one child. Furthermore, $<$ is not total, but well-founded.

Well-founded orderings can be combined to more complex well-founded orderings by lexicographic or multiset extensions.

Definition 1.4.5 (Lexicographic and Multi-Set Ordering Extensions)**.** Let $(M_1, \succ_1)$ and $(M_2, \succ_2)$ be two strict orderings. Their *lexicographic combination* $\succ_{\text{lex}} = (\succ_1, \succ_2)$ on $M_1 \times M_2$ is defined as $(m_1, m_2) \succ (m_1', m_2')$ iff $m_1 \succ_1 m_1'$ or $m_1 = m_1'$ and $m_2 \succ_2 m_2'$.

Let $(M, \succ)$ be a partial ordering. The multi-set extension $\succ_{\text{mul}}$ to multi-sets over M is defined by $S_1 \succ_{\text{mul}} S_2$ iff $S_1 \neq S_2$ and $\forall\, m \in M\, [S_2(m) > S_1(m) \rightarrow \exists\, m' \in M\, (m' \succ m \wedge S_1(m') > S_2(m'))]$.

The definition of the lexicographic ordering extensions can be exapanded to n-tuples in the obvious way. So it is also the basis for the standard lexicographic ordering on words as used, e.g., in dictionaries. In this case the M_i are alphabets, say a-z, where $a \prec b \prec \ldots \prec z$. Then according to the above definition *tiger* $\prec$ *tree*.

Example 1.4.6 (Multi Set Ordering)**.** Consider the multiset extension of $(\mathbb{N}, >)$. Then $\{2\} >_{\text{mul}} \{1,1,1\}$ because there is no element in $\{1,1,1\}$ that is larger than 2. As a border case, $\{2,1\} >_{\text{mul}} \{2\}$ because there is no element that has more occurrences in $\{2\}$ compared to $\{2,1\}$. The other way round, 1 has more occurrences in $\{2,1\}$ than in $\{2\}$ and there is no larger element to compensate for it, so $\{2\} \not>_{\text{mul}} \{2,1\}$.

Proposition 1.4.7 (Properties of Lexicographic and Multi-Set Ordering Extensions)**.** Let $(M, \succ)$, $(M_1, \succ_1)$, and $(M_2, \succ_2)$ be orderings. Then

1. $\succ_{\text{lex}}$ is an ordering on $M_1 \times M_2$.
2. if $(M_1, \succ_1)$ and $(M_2, \succ_2)$ are well-founded so is $\succ_{\text{lex}}$.
3. if $(M_1, \succ_1)$ and $(M_2, \succ_2)$ are total so is $\succ_{\text{lex}}$.
4. $\succ_{\text{mul}}$ is an ordering on multi-sets over M.
5. if $(M, \succ)$ is well-founded so is $\succ_{\text{mul}}$.
6. if $(M, \succ)$ is total so is $\succ_{\text{mul}}$.

The lexicographic ordering on words is not well-founded if words of arbitrary length are considered. Starting from the standard ordering on the alphabet, e.g., the following infinite descending sequence can be constructed: $b \succ ab \succ aab \succ \ldots$. It becomes well-founded if it is lexicographically combined with the length oordering, see Exercise **??**.

Lemma 1.4.8 (König's Lemma)**.** Every finitely branching tree with infinitely many nodes contains an infinite path.