Words

The length $|u|$ of a word $u \in \Sigma^*$ is defined by

(i) $|\epsilon| := 0$,

(ii) $|a| := 1$ for any $a \in \Sigma$ and

(iii) $|uv| := |u| + |v|$ for any $u, v \in \Sigma^*$.

1.2 Computer Science Prerequisites

A little bit on computational complexity theory.

Big O

Let $f(n)$ and $g(n)$ be functions from the naturals into the non-negative reals. Then

$$O(f(n)) = \{g(n) \mid \exists\, c > 0\, \exists\, n_0 \in \mathbb{N}^+\, \forall\, n \geq n_0\; g(n) \leq c \cdot f(n)\}$$

We use $\forall$, reads "for all", and $\exists$, reads "exists", on the object and meta level.

Decision Problem

A decision problem is a subset $L \subseteq \Sigma^*$ for some fixed finite alphabet Σ. The function $\text{chr}(L, x)$ denotes the characteristic function for some decision problem L and is defined by $\text{chr}(L, u) = 1$ if $u \in L$ and $\text{chr}(L, u) = 0$ 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.

NP

A decision problem L is in **NP** iff there is a predicate $Q(x, y)$ and a polynomial $p(n)$ such 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**.

Reducible,NP-Hard, NP-Complete

A decision problem L is polynomial-time reducible to a decision problem L' if there is a function $g \in \mathbf{P}$ such 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 $L' \in \mathbf{P}$ then $L \in \mathbf{P}$.

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**.

1.3 Ordering

Termination of rewrite systems and proof theory is strongly related to the concept of (well-founded) orderings.

An ordering R is a binary relation on some set M.

Ordering

Relevant properties of orderings are: Depending on particular properties such as

$$\begin{array}{rl}
\text{(reflexivity)} & \forall x \in M \, R(x,x) \\
\text{(irreflexivity)} & \forall x \in M \, \neg R(x,x) \\
\text{(antisymmetry)} & \forall x,y \in M \, (R(x,y) \wedge R(y,x) \rightarrow x = y) \\
\text{(transitivity)} & \forall x,y,z \in M \, (R(x,y) \wedge R(y,z) \rightarrow R(x,z)) \\
\text{(totality)} & \forall x,y \in M \, (R(x,y) \vee R(y,x))
\end{array}$$

where $=$ is the identity relation on M. The boolean connectives $\wedge$, $\vee$, and $\rightarrow$ read "and", "or", and "implies", respectively.

Partial Ordering

A strict partial ordering $\succ$ on a set M is a transitive and irreflexive binary relation on M.

An $a \in M$ is called minimal, if there is no b in M such that $a \succ b$.

An $a \in M$ is called smallest, if $b \succ a$ for all $b \in M$ different from a.

Notation:
$\prec$ for the inverse relation $\succ^{-1}$
$\succeq$ for the reflexive closure $(\succ \cup =)$ of $\succ$

Well-Foundedness

A strict partial ordering $\succ$ on M is called well-founded (Noetherian), if there is no infinite descending chain $a_0 \succ a_1 \succ a_2 \succ \ldots$ with $a_i \in M$.

Well-Foundedness and Termination

Let $\rightarrow$, $>$ be binary relations on the same set.

Lemma 1.1:
If $>$ is a well-founded partial ordering and $\rightarrow \subseteq >$,
then $\rightarrow$ is terminating.

Lemma 1.2:
If $\rightarrow$ is a terminating binary relation over A,
then $\rightarrow^{+}$ is a well-founded partial ordering.

Well-Founded Orderings: Examples

Natural numbers. $(\mathbb{N}, >)$

Lexicographic orderings. Let $(M_1, \succ_1), (M_2, \succ_2)$ be well-founded orderings. Then let their lexicographic combination

$$\succ = (\succ_1, \succ_2)_{lex}$$

on $M_1 \times M_2$ be defined as

$$(a_1, a_2) \succ (b_1, b_2) \quad :\Leftrightarrow$$

$$a_1 \succ_1 b_1 \text{ or } (a_1 = b_1 \text{ and } a_2 \succ_2 b_2)$$

(analogously for more than two orderings)

This again yields a well-founded ordering (proof below).

Well-Founded Orderings: Examples

Length-based ordering on words. For alphabets Σ with a well-founded ordering $>_\Sigma$, the relation $\succ$ defined as

$$w \succ w' \quad :\Leftrightarrow$$

$$|w| > |w'| \text{ or } (|w| = |w'| \text{ and } w >_{\Sigma,lex} w')$$

is a well-founded ordering on Σ^* (Exercise).

Counterexamples:

$(\mathbb{Z}, >)$

$(\mathbb{N}, <)$

the lexicographic ordering on Σ^*

Basic Properties of Well-Founded Orderings

Lemma 1.3:
$(M, \succ)$ is well-founded if and only if every $\emptyset \subset M' \subseteq M$ has a minimal element.

Lemma 1.4:
$(M_1, \succ_1)$ and $(M_2, \succ_2)$ are well-founded if and only if
$(M_1 \times M_2,\ \succ)$ with $\succ = (\succ_1, \succ_2)_{lex}$ is well-founded.

Monotone Mappings

Let $(M_1, >_1)$ and $(M_2, >_2)$ be strict partial orderings.
A mapping $\varphi : M_1 \to M_2$ is called monotone,
if $a >_1 b$ implies $\varphi(a) >_2 \varphi(b)$ for all $a, b \in M_1$.

Lemma 1.5:
If φ is a monotone mapping from $(M_1, >_1)$ to $(M_2, >_2)$
and $(M_2, >_2)$ is well-founded, then $(M_1, >_1)$ is well-founded.

Multiset Orderings

Lemma 1.6 (König's Lemma):
Every finitely branching tree with infinitely many nodes contains an infinite path.

Multiset Orderings

Let $(M, \succ)$ be a strict partial ordering. The multiset extension of $\succ$ to multisets over M is defined by

$$
\begin{aligned}
&S_1 \succ_{\mathrm{mul}} S_2 \;\Leftrightarrow \\
&\quad S_1 \neq S_2 \text{ and} \\
&\qquad \forall m \in M\colon \big(S_2(m) > S_1(m) \\
&\qquad\qquad \Rightarrow\; \exists m' \in M\colon m' \succ m \text{ and } S_1(m') > S_2(m')\big)
\end{aligned}
$$

1.4 Induction

More or less all sets of objects in computer science or logic are defined *inductively*. Typically, this is done in a bottom-up way, where starting with some definite set, it is closed under a given set of operations.

Induction

Example 1.7 (Inductive Sets):

1. The set of all Sudoku problem states, consists of the set of start states $(N; \top; \top)$ for consistent assignments N plus all states that can be derived from the start states by the rules Deduce, Conflict, Backtrack, and Fail. This is a finite set.
2. The set $\mathbb{N}$ of the natural numbers, consists of 0 plus all numbers that can be computed from 0 by adding 1. This is an infinite set.
3. The set of all strings Σ^* over a finite alphabet Σ where all letters of Σ are contained in Σ^* and if u and v are words out of Σ^* so is the word uv. This is an infinite set.

Induction

All the previous examples have in common that there is an underlying well-founded ordering on the sets induced by the construction. The minimal elements for the Sudoku are the problem states $(N; \top; \top)$, for the natural numbers it is 0 and for the set of strings the empty word.

Now if we want to prove a property of an inductive set it is sufficient to prove it (i) for the minimal element(s) and (ii) assuming the property for an arbitrary set of elements, to prove that it holds for all elements that can be constructed "in one step" out those elements. This is the principle of *Noetherian Induction*.

Induction

Theorem 1.8 (Noetherian Induction):

Let $(M, \succ)$ be a well-founded ordering, let Q be a property of elements of M.

If for all $m \in M$ the implication

> if $Q(m')$ for all $m' \in M$ such that $m \succ m'$,[a]
>
> then $Q(m)$.[b]

is satisfied, then the property $Q(m)$ holds for all $m \in M$.

[a] induction hypothesis

[b] induction step

Induction

Theorem 1.9 (Properties Multi-Set Ordering):

(a) $\succ_{\mathrm{mul}}$ is a strict partial ordering.

(b) $\succ$ well-founded $\Rightarrow$ $\succ_{\mathrm{mul}}$ well-founded.

(c) $\succ$ total $\Rightarrow$ $\succ_{\mathrm{mul}}$ total.

1.5 Rewrite Systems

A rewrite system is a pair $(A, \to)$, where

A is a set,

$\to \subseteq A \times A$ is a binary relation on A.

The relation $\to$ is usually written in infix notation, i. e.,
$a \to b$ instead of $(a, b) \in \to$.

Rewrite Systems

Let $\rightarrow' \subseteq A \times B$ and $\rightarrow'' \subseteq B \times C$ be two binary relations. Then the binary relation $(\rightarrow' \circ \rightarrow'') \subseteq A \times C$ is defined by

$$a\,(\rightarrow' \circ \rightarrow'')\;c \quad \text{if and only if}$$

$$a \rightarrow' b \text{ and } b \rightarrow'' c \text{ for some } b \in B.$$

Rewrite Systems

$\to^0 \;=\; \{ (a, a) \mid a \in A \}$	identity
$\to^{i+1} \;=\; \to^i \circ \to$	$i + 1$-fold composition
$\to^+ \;=\; \bigcup_{i>0} \to^i$	transitive closure
$\to^* \;=\; \bigcup_{i\geq 0} \to^i \;=\; \to^+ \cup \to^0$	reflexive transitive closure
$\to^= \;=\; \to \cup \to^0$	reflexive closure
$\to^{-1} \;=\; \leftarrow \;=\; \{ (b, c) \mid c \to b \}$	inverse
$\leftrightarrow \;=\; \to \cup \leftarrow$	symmetric closure
$\leftrightarrow^+ \;=\; (\leftrightarrow)^+$	transitive symmetric closure
$\leftrightarrow^* \;=\; (\leftrightarrow)^*$	refl. trans. symmetric closure

Rewrite Systems

$b \in A$ is reducible, if there is a c such that $b \to c$.

b is in normal form (irreducible), if it is not reducible.

c is a normal form of b, if $b \to^* c$ and c is in normal form.
Notation: $c = b{\downarrow}$ (if the normal form of b is unique).

Rewrite Systems

A relation $\rightarrow$ is called

terminating, if there is no infinite descending chain $b_0 \rightarrow b_1 \rightarrow b_2 \rightarrow \ldots$.

normalizing, if every $b \in A$ has a normal form.

Rewrite Systems

Lemma 1.10:

If $\rightarrow$ is terminating, then it is normalizing.

Note: The reverse implication does not hold.