Confluence

Another important property for don’t care non-deterministic rule based definitions of algorithms is confluence.

It means that whenever several sequences of rules are applicable to a given states, the respective results can be rejoined by further rule applications to a common problem state.
Proposition 0.4 (Deduce and Conflict are Locally Confluent):
Given a state \( (N; D; \top) \) out of which two different states
\( (N; D_1; \top) \) and \( (N; D_2; \bot) \) can be generated by Deduce and
Conflict in one step, respectively, then the two states can be
rejoined to a state \( (N; D'; *) \) via further rule applications.
Result

It works.

But: It looks like a lot of effort for a problem that one can solve with a little bit of thinking.

Reason: Our approach is very general, it can actually be used to “potentially solve” any problem in computer science.
This difference is also important for automated reasoning:

- For problems that are well-known and frequently used, we can develop optimal specialized methods.
  ⇒ Algorithms & Data-structures

- For new/unknown/changing problems, we have to develop generic methods that do “something useful”.
  ⇒ this lecture: Logic + Calculus + Implementation

- Combining the two approaches
  ⇒ Automated Reasoning II (next semester): Logic modulo Theory + Calculus + Implementation
Topics of the Course

Preliminaries

- math repetition
- computer science repetition
- orderings
- induction (repetition)
- rewrite systems

Propositional logic

- logic: syntax, semantics
- calculi: superposition, CDCL
- implementation: 2-watched literal, clause learning
Topics of the Course

First-order predicate logic
  logic: syntax, semantics, model theory
  calculus: superposition
  implementation: sharing, indexing

First-order predicate logic with equality
  equational logic: unit equations
  calculus: term rewriting systems, Knuth-Bendix completion
  implementation: dependency pairs
  first-order logic with equality
  calculus: superposition
  implementation: rewriting
Literature

Is a big problem, actually you are the “guinea-pigs” for a new textbook.

Franz Baader and Tobias Nipkow: *Term rewriting and all that*, Cambridge Univ. Press, 1998. (Textbook on equational reasoning)


Part 1: Preliminaries

- math repetition
- computer science repetition
- orderings
- induction (repetition)
- rewrite systems
1.1 Mathematical Prerequisites

\[ \mathbb{N} = \{0, 1, 2, \ldots \} \] is the set of natural numbers

\[ \mathbb{N}^+ \] is the set of positive natural numbers without 0

\[ \mathbb{Z}, \mathbb{Q}, \mathbb{R} \] denote the integers, rational numbers and the real numbers, respectively.
Multisets

Given a set $M$, a multi-set $S$ over $M$ is a mapping $S : M \rightarrow \mathbb{N}$, where $S$ specifies the number of occurrences of elements $m$ of the base set $M$ within the multiset $S$.

We use the standard set notations $\in$, $\subset$, $\subseteq$, $\cup$, $\cap$ with the analogous meaning for multisets, e.g., $(S_1 \cup S_2)(m) = S_1(m) + S_2(m)$.

We also write multi-sets in a set like notation, e.g., the multi-set $S = \{1, 2, 2, 4\}$ denotes a multi-set over the set $\{1, 2, 3, 4\}$ where $S(1) = 1$, $S(2) = 2$, $S(3) = 0$, and $S(4) = 1$.

A multi-set $S$ over a set $M$ is finite if $\{m \in M \mid S(m) > 0\}$ is finite. In this lecture we only consider finite multi-sets.
Relations

An $n$-ary relation $R$ over some set $M$ is a subset of $M^n$: $R \subseteq M^n$.

For two $n$-ary relations $R, Q$ over some set $M$, their union ($\bigcup$) or intersection ($\cap$) is again an $n$-ary relation, where

$R \cup Q := \{(m_1, \ldots, m_n) \in M \mid (m_1, \ldots, m_n) \in R \text{ or } (m_1, \ldots, m_n) \in Q\}$

$R \cap Q := \{(m_1, \ldots, m_n) \in M \mid (m_1, \ldots, m_n) \in R \text{ and } (m_1, \ldots, m_n) \in Q\}$.

A relation $Q$ is a subrelation of a relation $R$ if $Q \subseteq R$. 
The **characteristic function** of a relation $R$ or sometimes called **predicate** indicates membership. In addition of writing $(m_1, \ldots, m_n) \in R$ we also write $R(m_1, \ldots, m_n)$. So the predicate $R(m_1, \ldots, m_n)$ holds or is true if in fact $(m_1, \ldots, m_n)$ belongs to the relation $R$. 
Words

Given a nonempty alphabet $\Sigma$ the set $\Sigma^*$ of finite words over $\Sigma$ is defined by

(i) the empty word $\epsilon \in \Sigma^*$

(ii) for each letter $a \in \Sigma$ also $a \in \Sigma^*$

(iii) if $u, v \in \Sigma^*$ so $uv \in \Sigma^*$ where $uv$ denotes the concatenation of $u$ and $v$. 
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 $u, v \in \Sigma^*.$