Automated Reasoning I

Christoph Weidenbach

Max Planck Institute for Informatics

November 5, 2020
Outline

Preliminaries

Propositional Logic
Automated Reasoning

Given a specification of a system, develop technology

logics,
calculi,
algorithms,
implementations,

to automatically execute the specification and to automatically prove properties of the specification.
Concept

- **Slides**: Definitions, Lemmas, Theorems, . . .
- **Blackboard**: Examples, Proofs, . . .
- **Speech**: Motivate, Explain, . . .
- **Script**: Slides, partially Blackboard . . .
- **Exams**: able to calculate $\rightarrow$ pass
  understand $\rightarrow$ (very) good grade
Orderings

1.4.1 Definition (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 (partial) 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 $m_i \in M$. 
1.4.3 Definition (Minimal and Smallest Elements)

Given a strict ordering \((M, \succ)\), an element \(m \in M\) is called \textit{minimal}, if there is no element \(m' \in M\) so that \(m \succ m'\).

An element \(m \in M\) is called \textit{smallest}, if \(m' \succ m\) for all \(m' \in M\) different from \(m\).
Multisets

Given a set $M$, a multiset $S$ 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 $S$. I use the standard set notations $\in$, $\subset$, $\subseteq$, $\cup$, $\cap$ with the analogous meaning for multisets, for example $(S_1 \cup S_2)(m) = S_1(m) + S_2(m)$.

A multiset $S$ over a set $M$ is finite if $\{ m \in M \mid S(m) > 0 \}$ is finite. For the purpose of this lecture I only consider finite multisets.
1.4.5 Definition (Lexicographic and Multiset 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 strict ordering. The *multiset extension* \(\succ_{\text{mul}}\) to multisets over \(M\) is defined by \(S_1 \succ_{\text{mul}} S_2\) iff \(S_1 \neq S_2\) and \(\forall m \in M [S_2(m) > S_1(m) \rightarrow \exists m' \in M (m' \succ m \wedge S_1(m') > S_2(m'))]\).
1.4.7 Proposition (Properties of $\succ_{\text{lex}}$, $\succ_{\text{mul}}$)

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

Please recall that multisets are finite.
**Theorem (Noetherian Induction)**

Let \((M, \succ)\) be a well-founded ordering, and let \(Q\) be a predicate over elements of \(M\). If for all \(m \in M\) the implication

\[
\text{if } Q(m'), \text{ for all } m' \in M \text{ so that } m \succ m', \quad \text{(induction hypothesis)}
\]

then \(Q(m)\). \quad \text{(induction step)}

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

1.6.1 Definition (Rewrite System)

A *rewrite system* is a pair \((M, \rightarrow)\), where \(M\) is a non-empty set and \(\rightarrow \subseteq M \times M\) is a binary relation on \(M\).

\[

to^0 = \{(a, a) | a \in M\} \\
to^{i+1} = to^i \circ to \\
to^+ = \bigcup_{i>0} to^i \\
to^* = \bigcup_{i \geq 0} to^i = to^+ \cup to^0 \\
to = to \cup to^0 \\
to^{-1} = \leftarrow = \{(b, c) | c \rightarrow b\} \\
\leftrightarrow = to \cup \leftarrow \\
\leftrightarrow^+ = (\leftrightarrow)^+ \\
\leftrightarrow^* = (\leftrightarrow)^*
\]

*Identity*  
*i + 1-fold composition*  
*Transitive closure*  
*Reflexive transitive closure*  
*Reflexive closure*  
*Inverse*  
*Symmetric closure*  
*Transitive symmetric closure*  
*Reflex. trans. symmetric closure*
1.6.2 Definition (Reducible)

Let \((M, \rightarrow)\) be a rewrite system. An element \(a \in M\) is reducible, if there is a \(b \in M\) such that \(a \rightarrow b\).

An element \(a \in M\) is in normal form (irreducible), if it is not reducible.

An element \(c \in M\) is a normal form of \(b\), if \(b \rightarrow^* c\) and \(c\) is in normal form, denoted by \(c = b\downarrow\).

Two elements \(b\) and \(c\) are joinable, if there is an \(a\) so that \(b \rightarrow^* a \leftarrow^* c\), denoted by \(b \downarrow c\).
1.6.3 Definition (Properties of $\rightarrow$)

A relation $\rightarrow$ is called

- **Church-Rosser** if $b \leftrightarrow^* c$ implies $b \downarrow c$
- **confluent** if $b \leftarrow^* a \rightarrow^* c$ implies $b \downarrow c$
- **locally confluent** if $b \leftarrow a \rightarrow c$ implies $b \downarrow c$
- **terminating** if there is no infinite descending chain $b_0 \rightarrow b_1 \rightarrow b_2 \ldots$
- **normalizing** if every $b \in A$ has a normal form
- **convergent** if it is confluent and terminating
1.6.4 Lemma (Termination vs. Normalization)
If $\rightarrow$ is terminating, then it is normalizing.

1.6.5 Theorem (Church-Rosser vs. Confluence)
The following properties are equivalent for any $(M, \rightarrow)$:
(i) $\rightarrow$ has the Church-Rosser property.
(ii) $\rightarrow$ is confluent.

1.6.6 Lemma (Newman’s Lemma)
Let $(M, \rightarrow)$ be a terminating rewrite system. Then the following properties are equivalent:
(i) $\rightarrow$ is confluent
(ii) $\rightarrow$ is locally confluent
LA Equations Rewrite System

$M$ is the set of all LA equations sets $N$ over $\mathbb{Q}$

$\models$ includes normalizing the equation

**Eliminate** \[ \{x \models s, x \models t\} \cup N \Rightarrow_{\text{LAE}} \{x \models s, x \models t, s \models t\} \cup N \]

provided $s \neq t$, and $s \models t \notin N$

**Fail** \[ \{q_1 \models q_2\} \cup N \Rightarrow_{\text{LAE}} \emptyset \]

provided $q_1, q_2 \in \mathbb{Q}$, $q_1 \neq q_2$
LAE Redundancy

Subsume \( \{ s \models t, s' \models t' \} \uplus N \Rightarrow_{\text{LAE}} \{ s \models t \} \cup N \)

provided \( s \models t \) and \( qs' \models qt' \) are identical for some \( q \in \mathbb{Q} \)
### Rewrite Systems on Logics: Calculi

<table>
<thead>
<tr>
<th></th>
<th>Validity</th>
<th>Satisfiability</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sound</strong></td>
<td>If the calculus derives a proof of validity for the formula, it is valid.</td>
<td>If the calculus derives satisfiability of the formula, it has a model.</td>
</tr>
<tr>
<td><strong>Complete</strong></td>
<td>If the formula is valid, a proof of validity is derivable by the calculus.</td>
<td>If the formula has a model, the calculus derives satisfiability.</td>
</tr>
<tr>
<td><strong>Strongly Complete</strong></td>
<td>For any validity proof of the formula, there is a derivation in the calculus producing this proof.</td>
<td>For any model of the formula, there is a derivation in the calculus producing this model.</td>
</tr>
</tbody>
</table>
2.1.1 Definition (Propositional Formula)

The set $\text{PROP}(\Sigma)$ of *propositional formulas* over a signature $\Sigma$, is inductively defined by:

<table>
<thead>
<tr>
<th>$\text{PROP}(\Sigma)$</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bot$</td>
<td>connective $\bot$ denotes “false”</td>
</tr>
<tr>
<td>$\top$</td>
<td>connective $\top$ denotes “true”</td>
</tr>
<tr>
<td>$P$</td>
<td>for any propositional variable $P \in \Sigma$</td>
</tr>
<tr>
<td>$(\neg \phi)$</td>
<td>connective $\neg$ denotes “negation”</td>
</tr>
<tr>
<td>$(\phi \land \psi)$</td>
<td>connective $\land$ denotes “conjunction”</td>
</tr>
<tr>
<td>$(\phi \lor \psi)$</td>
<td>connective $\lor$ denotes “disjunction”</td>
</tr>
<tr>
<td>$(\phi \rightarrow \psi)$</td>
<td>connective $\rightarrow$ denotes “implication”</td>
</tr>
<tr>
<td>$(\phi \leftrightarrow \psi)$</td>
<td>connective $\leftrightarrow$ denotes “equivalence”</td>
</tr>
</tbody>
</table>

where $\phi, \psi \in \text{PROP}(\Sigma)$. 
### Propositional Logic: Semantics

#### 2.2.1 Definition ((Partial) Valuation)

A \( \Sigma \)-valuation is a map

\[
\mathcal{A} : \Sigma \rightarrow \{0, 1\}.
\]

where \( \{0, 1\} \) is the set of *truth values*. A *partial \( \Sigma \)-valuation* is a map \( \mathcal{A} : \Sigma' \rightarrow \{0, 1\} \) where \( \Sigma' \subseteq \Sigma \).
2.2.2 Definition (Semantics)

A $\Sigma$-valuation $\mathcal{A}$ is inductively extended from propositional variables to propositional formulas $\phi, \psi \in \text{PROP}(\Sigma)$ by

- $\mathcal{A}(\bot) := 0$
- $\mathcal{A}(\top) := 1$
- $\mathcal{A}(\neg \phi) := 1 - \mathcal{A}(\phi)$
- $\mathcal{A}(\phi \land \psi) := \min(\{\mathcal{A}(\phi), \mathcal{A}(\psi)\})$
- $\mathcal{A}(\phi \lor \psi) := \max(\{\mathcal{A}(\phi), \mathcal{A}(\psi)\})$
- $\mathcal{A}(\phi \rightarrow \psi) := \max(\{1 - \mathcal{A}(\phi), \mathcal{A}(\psi)\})$
- $\mathcal{A}(\phi \leftrightarrow \psi) := \text{if } \mathcal{A}(\phi) = \mathcal{A}(\psi) \text{ then } 1 \text{ else } 0$
If $\mathcal{A}(\phi) = 1$ for some $\Sigma$-valuation $\mathcal{A}$ of a formula $\phi$ then $\phi$ is *satisfiable* and we write $\mathcal{A} \models \phi$. In this case $\mathcal{A}$ is a *model* of $\phi$.

If $\mathcal{A}(\phi) = 1$ for all $\Sigma$-valuations $\mathcal{A}$ of a formula $\phi$ then $\phi$ is *valid* and we write $\models \phi$.

If there is no $\Sigma$-valuation $\mathcal{A}$ for a formula $\phi$ where $\mathcal{A}(\phi) = 1$ we say $\phi$ is *unsatisfiable*.

A formula $\phi$ *entails* $\psi$, written $\phi \models \psi$, if for all $\Sigma$-valuations $\mathcal{A}$ whenever $\mathcal{A} \models \phi$ then $\mathcal{A} \models \psi$. 