Automated Theorem Proving

Lecture 1: Motivation and Preliminaries

Prof. Dr. Jasmin Blanchette

based on slides by Dr. Uwe Waldmann

Winter Term 2024/25

What Is Automated Theorem Proving?

This course is primarily about *automated theorem proving* and more generally about *automated reasoning* (also called *automated deduction*):

Logical reasoning using a computer program,

with little or no user interaction,

using general methods, rather than approaches that work only for one specific problem.

Two examples:

Solving a sudoku.

Reasoning with equations.

Introductory Example 1: Sudoku

	1	2	3	4	5	6	7	8	9
1								1	
2	4								
3		2							
4					5		4		7
5			8				3		
6			1		9				
7	3			4			2		
8		5		1					
9				8		6			

Goal:

Fill the empty fields with digits 1,. . . ,9, so that each digit occurs exactly once in each row, column, and 3×3 box.

Introductory Example 1: Sudoku

	1	2	3	4	5	6	7	8	9
1								1	
2	4								
3		2							
4					5		4		7
5			8				3		
6			1		9				
7	3			4			2		
8		5		1					
9				8		6			

Idea:

Use boolean variables $P^d_{i,j}$ with $d, i, j \in \{1, \ldots, 9\}$ to encode the problem:

$P^d_{i,j} = \mathit{true}$ iff the value of square i, j is d.

Introductory Example 1: Sudoku

	1	2	3	4	5	6	7	8	9
1								1	
2	4								
3		2							
4					5		4		7
5			8				3		
6			1		9				
7	3			4			2		
8		5		1					
9				8		6			

Idea:

Use boolean variables $P^d_{i,j}$ with $d, i, j \in \{1, \ldots, 9\}$ to encode the problem:

$P^d_{i,j} = \mathit{true}$ iff the value of square i, j is d.

For example:

$P^8_{5,3} = \mathit{true}.$

$P^7_{5,3} = \mathit{false}.$

Coding Sudoku in Boolean Logic

- Concrete values result in formulas $P^d_{i,j}$
- For every square (i, j) we generate $P^1_{i,j} \vee \ldots \vee P^9_{i,j}$
- For every square (i, j) and pair of values $d < d'$ we generate $\neg P^d_{i,j} \vee \neg P^{d'}_{i,j}$
- For every value d and row i we generate $P^d_{i,1} \vee \ldots \vee P^d_{i,9}$
  (Analogously for columns and 3×3 boxes)
- For every value d, row i, and pair of columns $j < j'$
  we generate $\neg P^d_{i,j} \vee \neg P^d_{i,j'}$
  (Analogously for columns and 3×3 boxes)

Coding Sudoku in Boolean Logic

Every assignment of boolean values to the variables $P_{i,j}^{d}$
so that all formulas become true
corresponds to a Sudoku solution (and vice versa).

Coding Sudoku in Boolean Logic

Now use a SAT solver to check whether there is an assignment to the variables $P^d_{i,j}$ so that all formulas become true:

Niklas Eén, Niklas Sörensson:
MiniSat (`http://minisat.se/`)

Beware:

The satisfiability problem is NP-complete.

Every known algorithm to solve it has an exponential time worst-case behavior (or worse).

Coding Sudoku in Boolean Logic

MiniSat solves the problem in a few milliseconds.

How? See part 2 of this lecture or Johannsen's SAT Solving lecture.

Does that contradict NP-completeness? No.

NP-completeness implies that there are really hard problem instances,

it does not imply that all practically interesting problem instances are hard (for a well-written SAT solver).

SAT Solvers in Practice

Some real-life applications of modern SAT solvers:

- hardware verification (model checking)
- with extensions:
  - software verification, hybrid system verification, . . .
- checking software package dependencies
- solving combinatory problems
- “The Largest Math Proof Ever” (Marijn Heule)
- . . .

Introductory Example 2: Equations

Task:

Prove: $$\frac{a}{a+1} = 1 + \frac{-1}{a+1}.$$

Introductory Example 2: Equations

$$\frac{a}{a+1}$$

$$1+\frac{-1}{a+1}$$

Introductory Example 2: Equations

$$\frac{a}{a+1} = \frac{a+0}{a+1}$$

$$1 + \frac{-1}{a+1}$$

$$x + 0 = x \qquad (1)$$

Introductory Example 2: Equations

$$\frac{a}{a+1} = \frac{a+0}{a+1}$$

$$= \frac{a+(1+(-1))}{a+1}$$

$$1 + \frac{-1}{a+1}$$

$$x + 0 = x \qquad (1)$$

$$x + (-x) = 0 \qquad (2)$$

Introductory Example 2: Equations

$$\frac{a}{a+1} = \frac{a+0}{a+1}$$

$$= \frac{a+(1+(-1))}{a+1}$$

$$= \frac{(a+1)+(-1)}{a+1}$$

$$1 + \frac{-1}{a+1}$$

$$x + 0 = x \quad (1)$$

$$x + (-x) = 0 \quad (2)$$

$$x + (y + z) = (x + y) + z \quad (3)$$

Introductory Example 2: Equations

$$\frac{a}{a+1} = \frac{a+0}{a+1}$$

$$= \frac{a+(1+(-1))}{a+1}$$

$$= \frac{(a+1)+(-1)}{a+1}$$

$$= \frac{a+1}{a+1} + \frac{-1}{a+1}$$

$$1 + \frac{-1}{a+1}$$

$$x + 0 = x \quad (1)$$

$$x + (-x) = 0 \quad (2)$$

$$x + (y+z) = (x+y)+z \quad (3)$$

$$\frac{x}{z} + \frac{y}{z} = \frac{x+y}{z} \quad (4)$$

Introductory Example 2: Equations

$$\frac{a}{a+1} = \frac{a+0}{a+1}$$

$$= \frac{a+(1+(-1))}{a+1}$$

$$= \frac{(a+1)+(-1)}{a+1}$$

$$= \frac{a+1}{a+1} + \frac{-1}{a+1}$$

$$= 1 + \frac{-1}{a+1}$$

$$x + 0 = x \quad (1)$$

$$x + (-x) = 0 \quad (2)$$

$$x + (y+z) = (x+y) + z \quad (3)$$

$$\frac{x}{z} + \frac{y}{z} = \frac{x+y}{z} \quad (4)$$

$$\frac{x}{x} = 1 \quad (5)$$

Introductory Example 2: Equations

How could we write a program that takes a set of equations and two terms and tests whether the terms can be connected via a chain of equalities?

It is easy to write a program that applies formulas *correctly*.

However, correct $\neq$ useful.

Introductory Example 2: Equations

$$\frac{a}{a+1}$$

$$x + 0 = x \tag{1}$$

$$x + (-x) = 0 \tag{2}$$

$$x + (y + z) = (x + y) + z \tag{3}$$

$$\frac{x}{z} + \frac{y}{z} = \frac{x + y}{z} \tag{4}$$

$$\frac{x}{x} = 1 \tag{5}$$

Introductory Example 2: Equations

$$\frac{a}{a+1} \longrightarrow \frac{a+0}{a+1}$$

$$x + 0 = x \tag{1}$$

$$x + (-x) = 0 \tag{2}$$

$$x + (y + z) = (x + y) + z \tag{3}$$

$$\frac{x}{z} + \frac{y}{z} = \frac{x+y}{z} \tag{4}$$

$$\frac{x}{x} = 1 \tag{5}$$

Introductory Example 2: Equations

$$\frac{a}{a+1} \longrightarrow \frac{a+0}{a+1}$$

$$\frac{a}{a+1} \searrow \frac{a}{a+1} + 0$$

$$x + 0 = x \qquad (1)$$

$$x + (-x) = 0 \qquad (2)$$

$$x + (y + z) = (x + y) + z \qquad (3)$$

$$\frac{x}{z} + \frac{y}{z} = \frac{x+y}{z} \qquad (4)$$

$$\frac{x}{x} = 1 \qquad (5)$$

Introductory Example 2: Equations

$$\frac{a}{a+1} \rightarrow \frac{a+0}{a+1}$$

$$\frac{a}{a+1} \rightarrow \frac{a}{a+1} + 0$$

$$\frac{a}{a+1} \rightarrow \frac{a}{a+(1+0)}$$

$$x + 0 = x \tag{1}$$

$$x + (-x) = 0 \tag{2}$$

$$x + (y + z) = (x + y) + z \tag{3}$$

$$\frac{x}{z} + \frac{y}{z} = \frac{x+y}{z} \tag{4}$$

$$\frac{x}{x} = 1 \tag{5}$$

Introductory Example 2: Equations

$$\frac{a}{a+1} \longrightarrow \frac{a+0}{a+1}$$

$$\frac{a}{a+1} \longrightarrow \frac{a}{a+1} + 0$$

$$\frac{a}{a+1} \longrightarrow \frac{a}{a+(1+0)}$$

$$\frac{a}{a+1} \longrightarrow \frac{a}{a+\frac{a+2}{a+2}}$$

$$x + 0 = x \tag{1}$$

$$x + (-x) = 0 \tag{2}$$

$$x + (y + z) = (x + y) + z \tag{3}$$

$$\frac{x}{z} + \frac{y}{z} = \frac{x+y}{z} \tag{4}$$

$$\frac{x}{x} = 1 \tag{5}$$

Introductory Example 2: Equations

$$\frac{a}{a+1} \longrightarrow \frac{a+0}{a+1}$$

$$\frac{a}{a+1} \longrightarrow \frac{a}{a+1} + 0$$

$$\frac{a}{a+1} \longrightarrow \frac{a}{a+(1+0)}$$

$$\frac{a}{a+1} \longrightarrow \frac{a}{a+\frac{a+2}{a+2}}$$

$$\vdots$$

$$x + 0 = x \tag{1}$$

$$x + (-x) = 0 \tag{2}$$

$$x + (y + z) = (x + y) + z \tag{3}$$

$$\frac{x}{z} + \frac{y}{z} = \frac{x+y}{z} \tag{4}$$

$$\frac{x}{x} = 1 \tag{5}$$

Introductory Example 2: Equations

$$1 + \frac{-1}{a+1}$$

$$x + 0 = x \tag{1}$$

$$x + (-x) = 0 \tag{2}$$

$$x + (y + z) = (x + y) + z \tag{3}$$

$$\frac{x}{z} + \frac{y}{z} = \frac{x+y}{z} \tag{4}$$

$$\frac{x}{x} = 1 \tag{5}$$

Introductory Example 2: Equations

$$1 + \frac{-1}{a+1} \longrightarrow \frac{a+1}{a+1} + \frac{-1}{a+1}$$

$$x + 0 = x \quad (1)$$

$$x + (-x) = 0 \quad (2)$$

$$x + (y + z) = (x + y) + z \quad (3)$$

$$\frac{x}{z} + \frac{y}{z} = \frac{x+y}{z} \quad (4)$$

$$\frac{x}{x} = 1 \quad (5)$$

Introductory Example 2: Equations

$$1 + \frac{-1}{a+1} \longrightarrow \frac{a+1}{a+1} + \frac{-1}{a+1}$$

$$\frac{a}{a} + \frac{-1}{a+1}$$

$$x + 0 = x \quad (1)$$

$$x + (-x) = 0 \quad (2)$$

$$x + (y + z) = (x + y) + z \quad (3)$$

$$\frac{x}{z} + \frac{y}{z} = \frac{x+y}{z} \quad (4)$$

$$\frac{x}{x} = 1 \quad (5)$$

Introductory Example 2: Equations

$$1 + \frac{-1}{a+1} \longrightarrow \frac{a+1}{a+1} + \frac{-1}{a+1}$$

$$\frac{a}{a} + \frac{-1}{a+1}$$

$$1 + \frac{-1}{a + \frac{a}{a}}$$

$$x + 0 = x \quad (1)$$

$$x + (-x) = 0 \quad (2)$$

$$x + (y + z) = (x + y) + z \quad (3)$$

$$\frac{x}{z} + \frac{y}{z} = \frac{x+y}{z} \quad (4)$$

$$\frac{x}{x} = 1 \quad (5)$$

Introductory Example 2: Equations

$$1 + \frac{-1}{a+1} \longrightarrow \frac{a+1}{a+1} + \frac{-1}{a+1}$$

$$\frac{a}{a} + \frac{-1}{a+1}$$

$$1 + \frac{-1}{a + \frac{a}{a}}$$

$$1 + \frac{-1+0}{a+1}$$

$$x + 0 = x \qquad (1)$$

$$x + (-x) = 0 \qquad (2)$$

$$x + (y + z) = (x + y) + z \qquad (3)$$

$$\frac{x}{z} + \frac{y}{z} = \frac{x+y}{z} \qquad (4)$$

$$\frac{x}{x} = 1 \qquad (5)$$

Introductory Example 2: Equations

$$1 + \frac{-1}{a+1} \longrightarrow \frac{a+1}{a+1} + \frac{-1}{a+1}$$

$$\longrightarrow \frac{a}{a} + \frac{-1}{a+1}$$

$$\longrightarrow 1 + \frac{-1}{a + \frac{a}{a}}$$

$$\longrightarrow 1 + \frac{-1+0}{a+1}$$

$$\vdots$$

$$x + 0 = x \tag{1}$$

$$x + (-x) = 0 \tag{2}$$

$$x + (y + z) = (x + y) + z \tag{3}$$

$$\frac{x}{z} + \frac{y}{z} = \frac{x+y}{z} \tag{4}$$

$$\frac{x}{x} = 1 \tag{5}$$

Introductory Example 2: Equations

Unrestricted application of equations leads to

- infinitely many equality chains,
- infinitely long equality chains.

$\Rightarrow$ The chance to reach the desired goal is very small.

In fact, the general problem is only semidecidable,
but not decidable.

Introductory Example 2: Equations

A better approach:

Apply equations in such a way that terms become "simpler."

Start from both sides:

-
-

Introductory Example 2: Equations

A better approach:

Apply equations in such a way that terms become “simpler.”

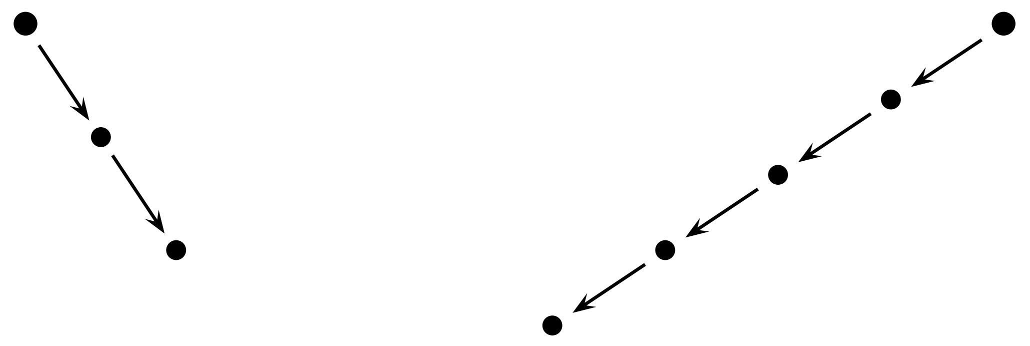
Start from both sides:

Introductory Example 2: Equations

A better approach:

Apply equations in such a way that terms become "simpler."

Start from both sides:

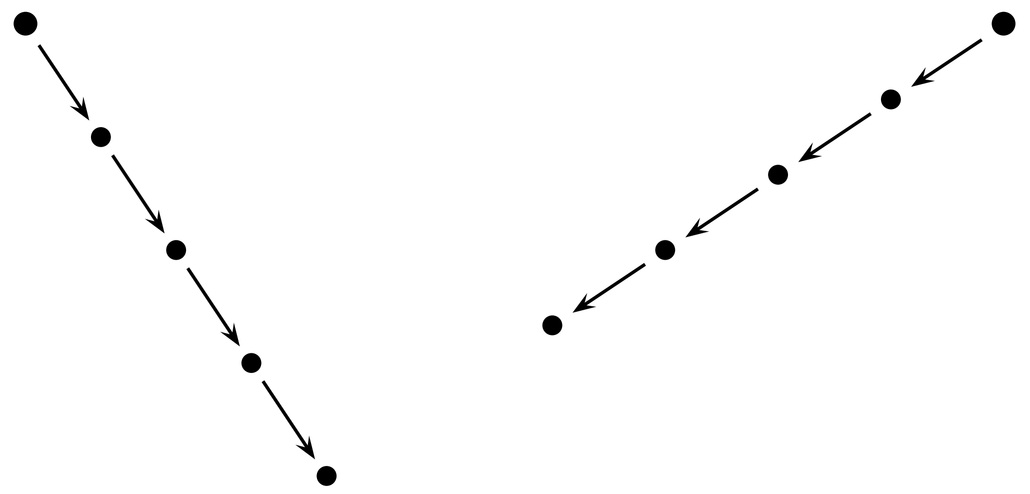


Introductory Example 2: Equations

A better approach:

Apply equations in such a way that terms become “simpler.”

Start from both sides:

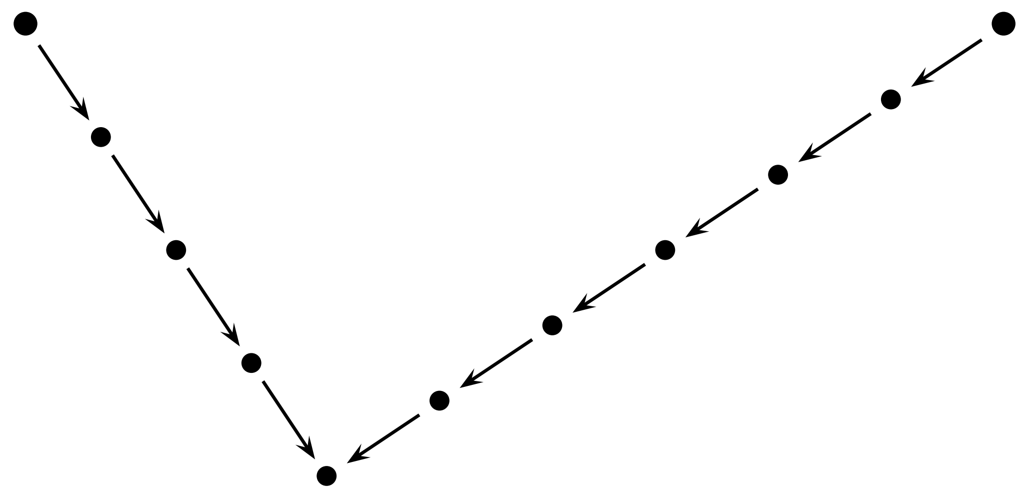


Introductory Example 2: Equations

A better approach:

Apply equations in such a way that terms become “simpler.”

Start from both sides:



The terms are equal if both derivations meet.

Introductory Example 2: Equations

$$x + 0 = x \tag{1}$$

$$x + (-x) = 0 \tag{2}$$

$$x + (y + z) = (x + y) + z \tag{3}$$

$$\frac{x}{z} + \frac{y}{z} = \frac{x + y}{z} \tag{4}$$

$$\frac{x}{x} = 1 \tag{5}$$

Introductory Example 2: Equations

Orient equations.

$$x + 0 \to x \tag{1}$$

$$x + (-x) \to 0 \tag{2}$$

$$x + (y + z) \to (x + y) + z \tag{3}$$

$$\frac{x}{z} + \frac{y}{z} \to \frac{x + y}{z} \tag{4}$$

$$\frac{x}{x} \to 1 \tag{5}$$

Introductory Example 2: Equations

Orient equations.

Advantage:
Now there are only finitely many and finitely long derivations.

$$x + 0 \to x \tag{1}$$

$$x + (-x) \to 0 \tag{2}$$

$$x + (y + z) \to (x + y) + z \tag{3}$$

$$\frac{x}{z} + \frac{y}{z} \to \frac{x + y}{z} \tag{4}$$

$$\frac{x}{x} \to 1 \tag{5}$$

Introductory Example 2: Equations

Orient equations.

But:

Now none of the equations is applicable to one of the terms

$$\frac{a}{a+1}, \quad 1 + \frac{-1}{a+1}$$

$$x + 0 \to x \tag{1}$$

$$x + (-x) \to 0 \tag{2}$$

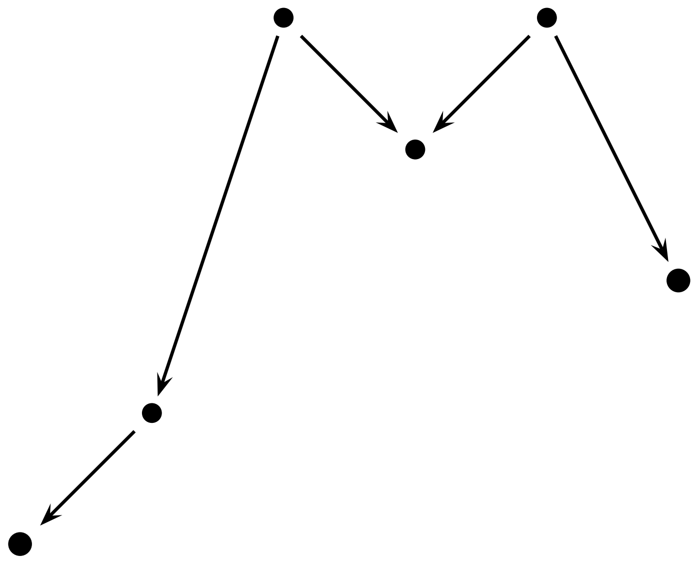
$$x + (y + z) \to (x + y) + z \tag{3}$$

$$\frac{x}{z} + \frac{y}{z} \to \frac{x+y}{z} \tag{4}$$

$$\frac{x}{x} \to 1 \tag{5}$$

Introductory Example 2: Equations

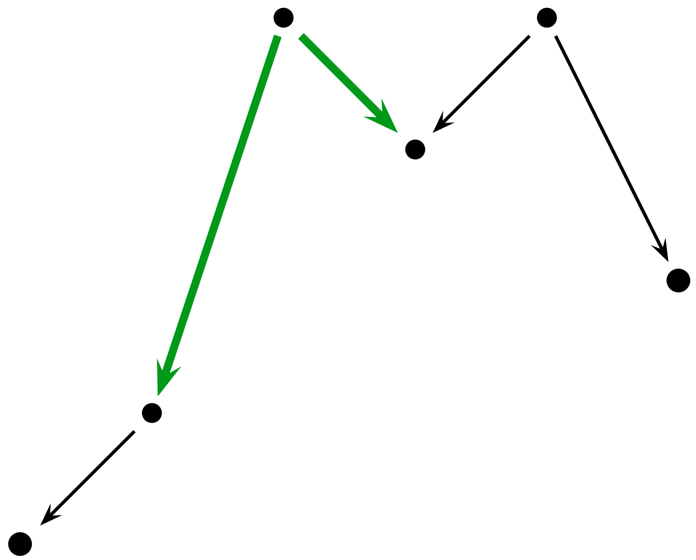
The chain of equalities that we considered at the beginning looks roughly like this:



Introductory Example 2: Equations

Idea:

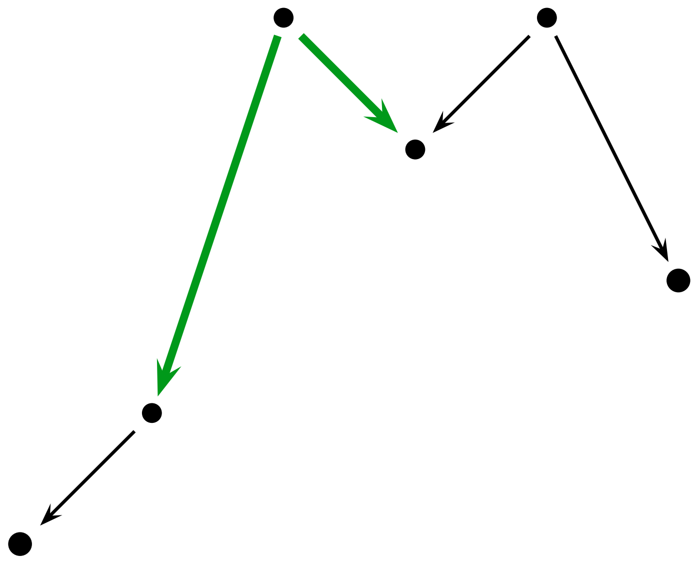
Derive new equations that enable shortcuts.



Introductory Example 2: Equations

Idea:

Derive new equations that enable shortcuts.



From

$$x + (-x) \to 0 \qquad (2)$$

$$x + (y + z) \to (x + y) + z \qquad (3)$$

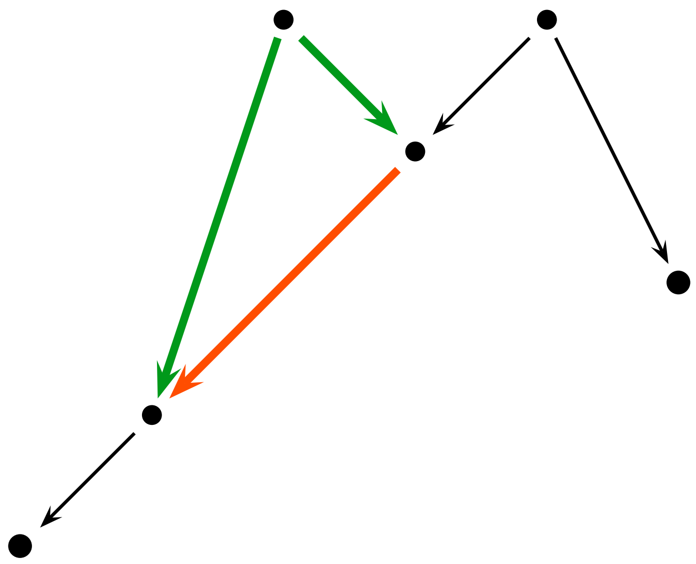
we derive

$$(x + y) + (-y) \to x + 0 \qquad (6)$$

Introductory Example 2: Equations

Idea:

Derive new equations that enable shortcuts.



From

$$x + (-x) \to 0 \qquad (2)$$

$$x + (y + z) \to (x + y) + z \qquad (3)$$

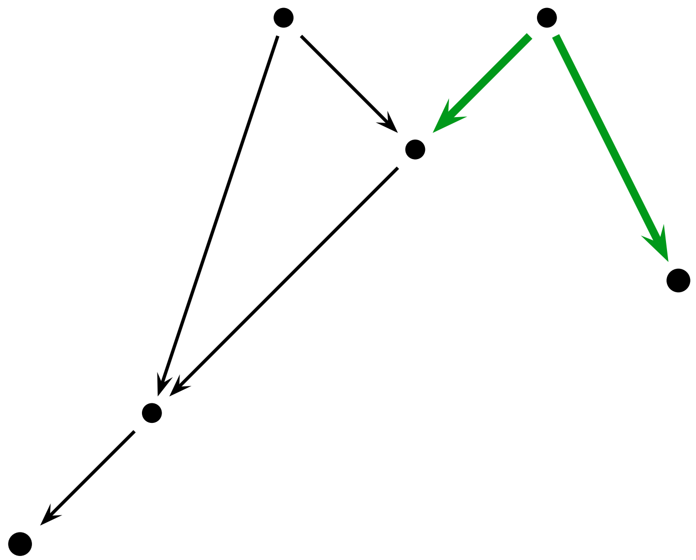
we derive

$$(x + y) + (-y) \to x + 0 \qquad (6)$$

Introductory Example 2: Equations

Idea:

Derive new equations that enable shortcuts.



From

$$\frac{x}{z} + \frac{y}{z} \to \frac{x+y}{z} \qquad (4)$$

$$\frac{x}{x} \to 1 \qquad (5)$$

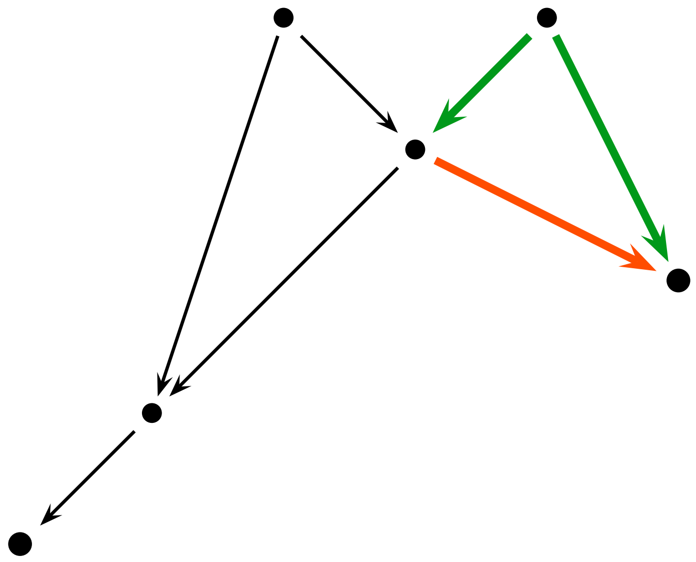
we derive

$$\frac{x+y}{x} \to 1 + \frac{y}{x} \qquad (7)$$

Introductory Example 2: Equations

Idea:

Derive new equations that enable shortcuts.



From

$$\frac{x}{z} + \frac{y}{z} \to \frac{x+y}{z} \qquad (4)$$

$$\frac{x}{x} \to 1 \qquad (5)$$

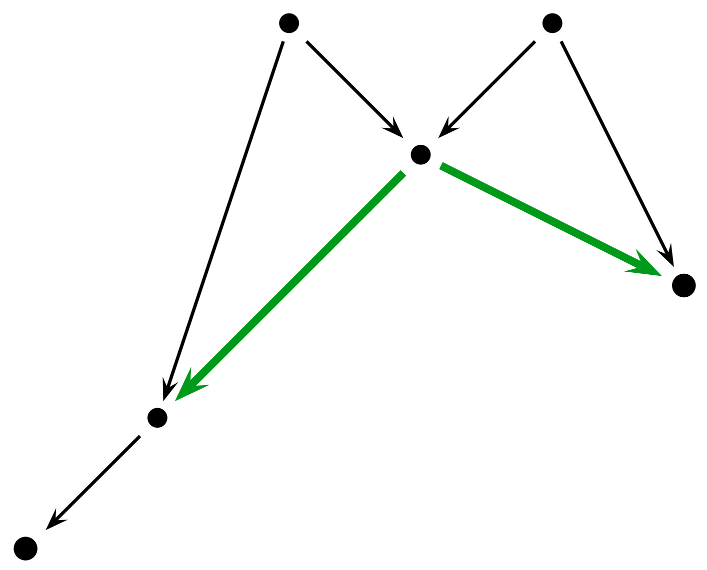
we derive

$$\frac{x+y}{x} \to 1 + \frac{y}{x} \qquad (7)$$

Introductory Example 2: Equations

Idea:

Derive new equations that enable shortcuts.



From

$$(x + y) + (-y) \to x + 0 \qquad (6)$$

$$\frac{x + y}{x} \to 1 + \frac{y}{x} \qquad (7)$$

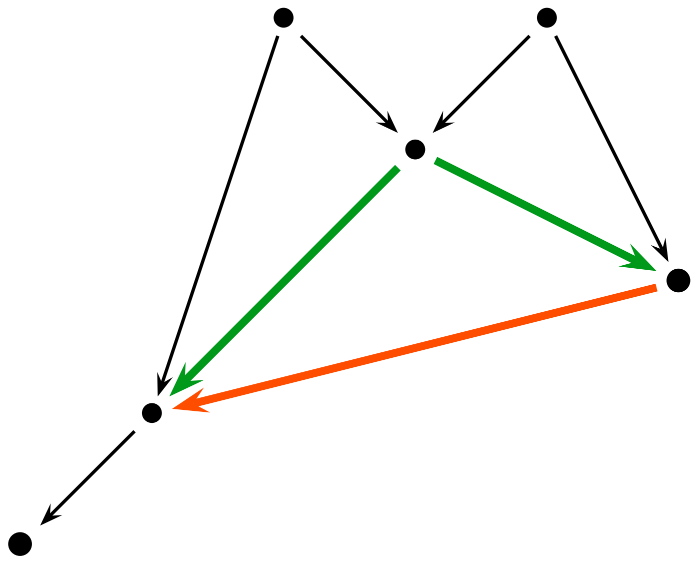
we derive

$$1 + \frac{-y}{x + y} \to \frac{x + 0}{x + y} \qquad (8)$$

Introductory Example 2: Equations

Idea:

Derive new equations that enable shortcuts.



From

$$(x + y) + (-y) \rightarrow x + 0 \qquad (6)$$

$$\frac{x + y}{x} \rightarrow 1 + \frac{y}{x} \qquad (7)$$

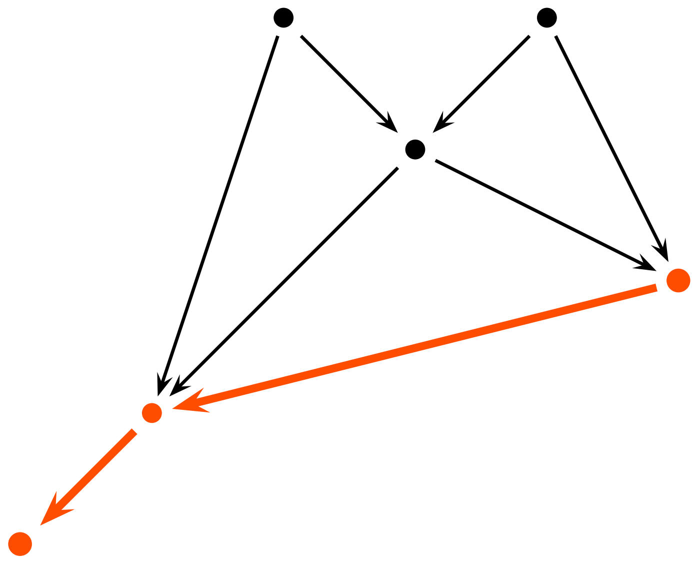
we derive

$$1 + \frac{-y}{x + y} \rightarrow \frac{x + 0}{x + y} \qquad (8)$$

Introductory Example 2: Equations

Idea:

Derive new equations that enable shortcuts.



Using these equations we can get a chain of equalities of the desired form.

Introductory Example 2: Equations

In fact, it is not necessary to know some equational proof for the problem in advance.

We can derive these shortcut equations just by looking at the existing equation set.

How? See parts 4 and 5 of this lecture.

Result

The Waldmeister prover solves the problem within milliseconds.

So it works, but it looks like a lot of effort for a problem that one can solve with a little bit of high-school mathematics.

Reason: Pupils learn not only axioms, but also recipes to work efficiently with these axioms.

Result

It makes a huge difference whether we work with well-known axioms

$$x + 0 = x$$

$$x + (-x) = 0$$

or with "new" unknown ones

$$\begin{aligned} &\forall Agent\ \forall Message\ \forall Key. \\ &\quad knows(Agent, crypt(Message, Key)) \\ &\quad \wedge\ knows(Agent, Key) \\ &\quad \rightarrow knows(Agent, Message). \end{aligned}$$

Result

This difference is also important for automated reasoning:

- For axioms that are well-known and frequently used, we can develop optimal specialized methods.
  $\Rightarrow$ computer algebra
  $\Rightarrow$ Waldmann's Automated Reasoning II lecture at Saarland University

- For new axioms, we have to develop methods that do something reasonable for arbitrary formulas.
  $\Rightarrow$ this lecture

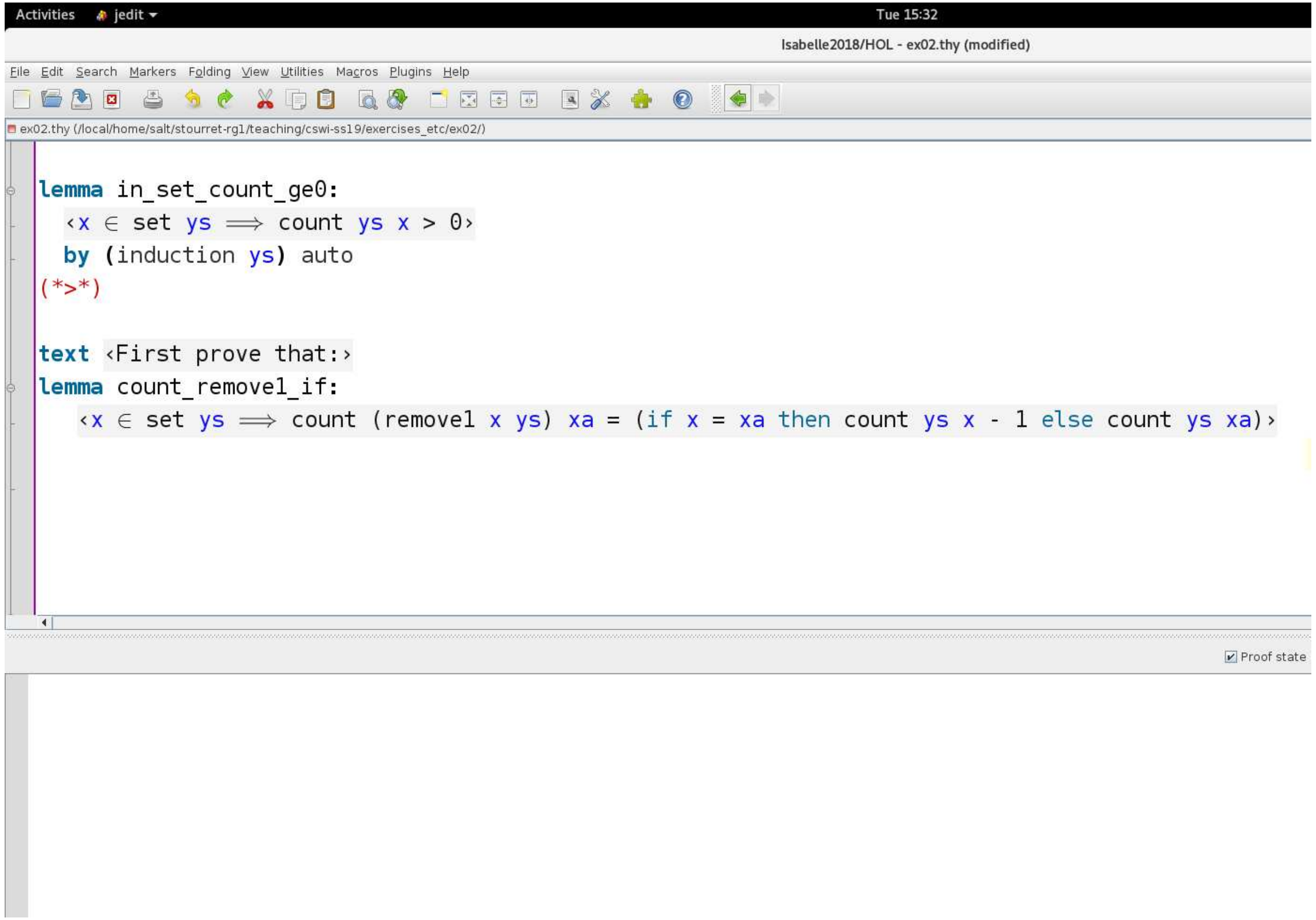
First-Order Provers in Practice

Real-life application:

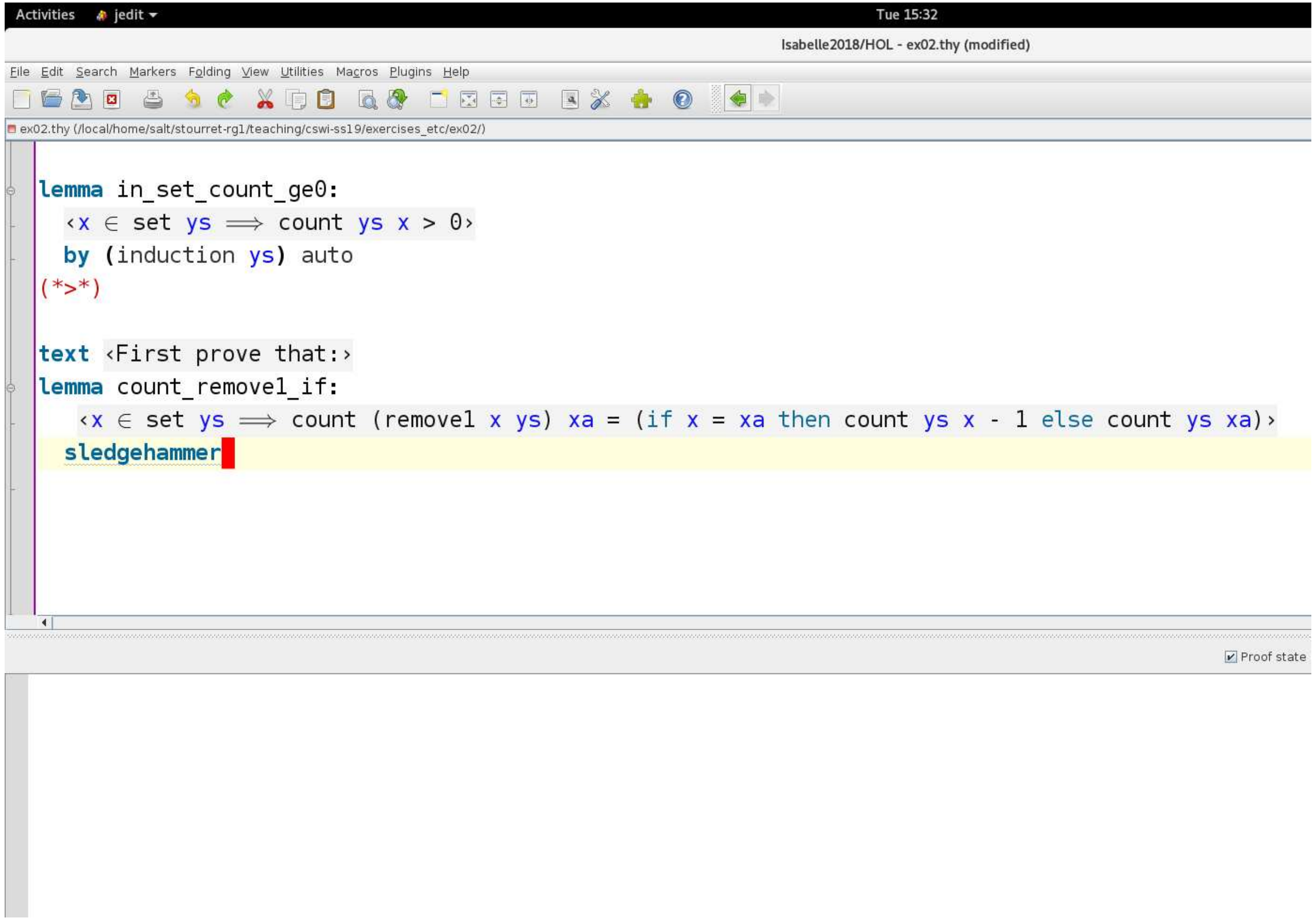
Use general-purpose provers to make interactive proof assistants more automatic:

Isabelle tool Sledgehammer.

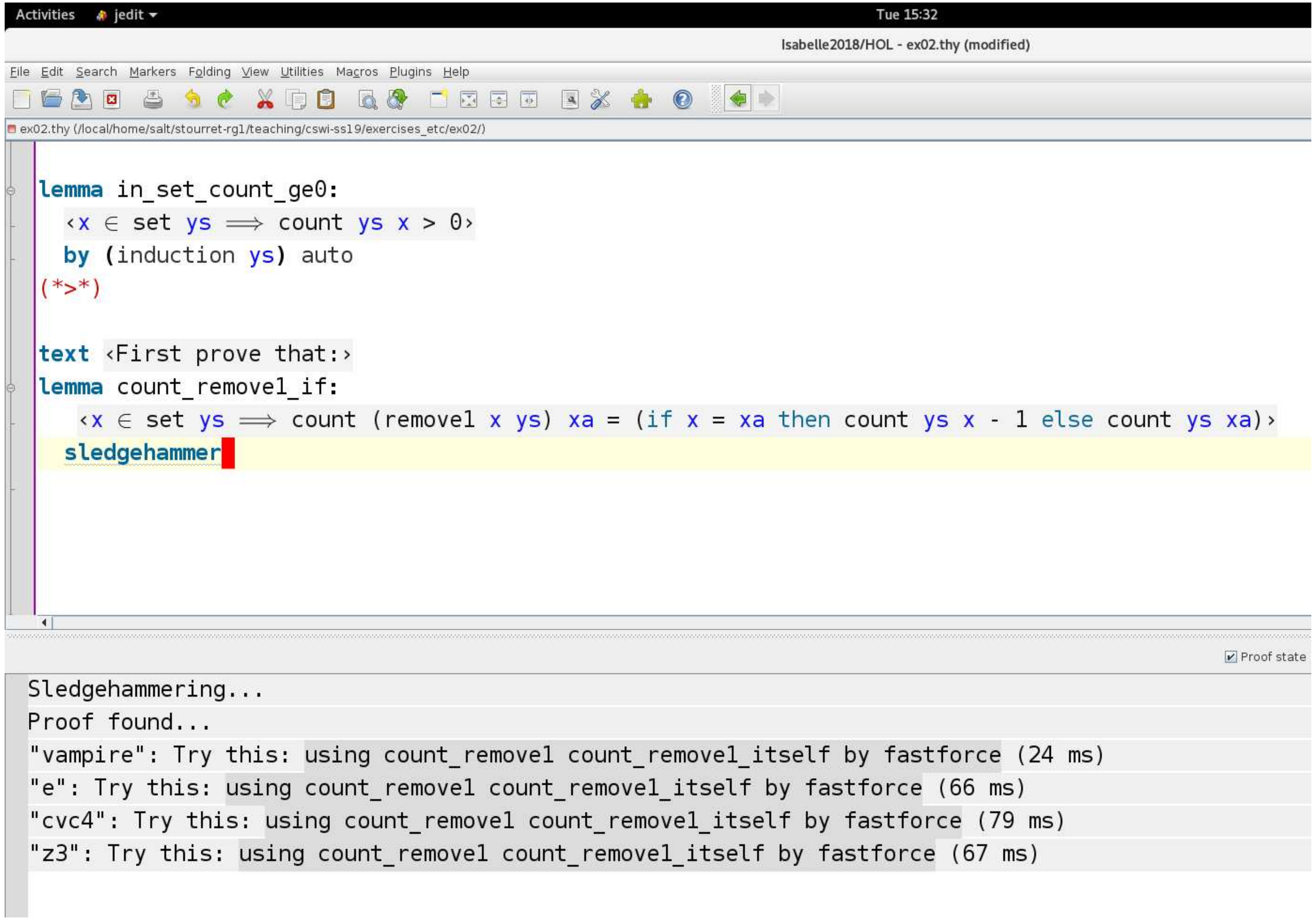
First-Order Provers in Practice



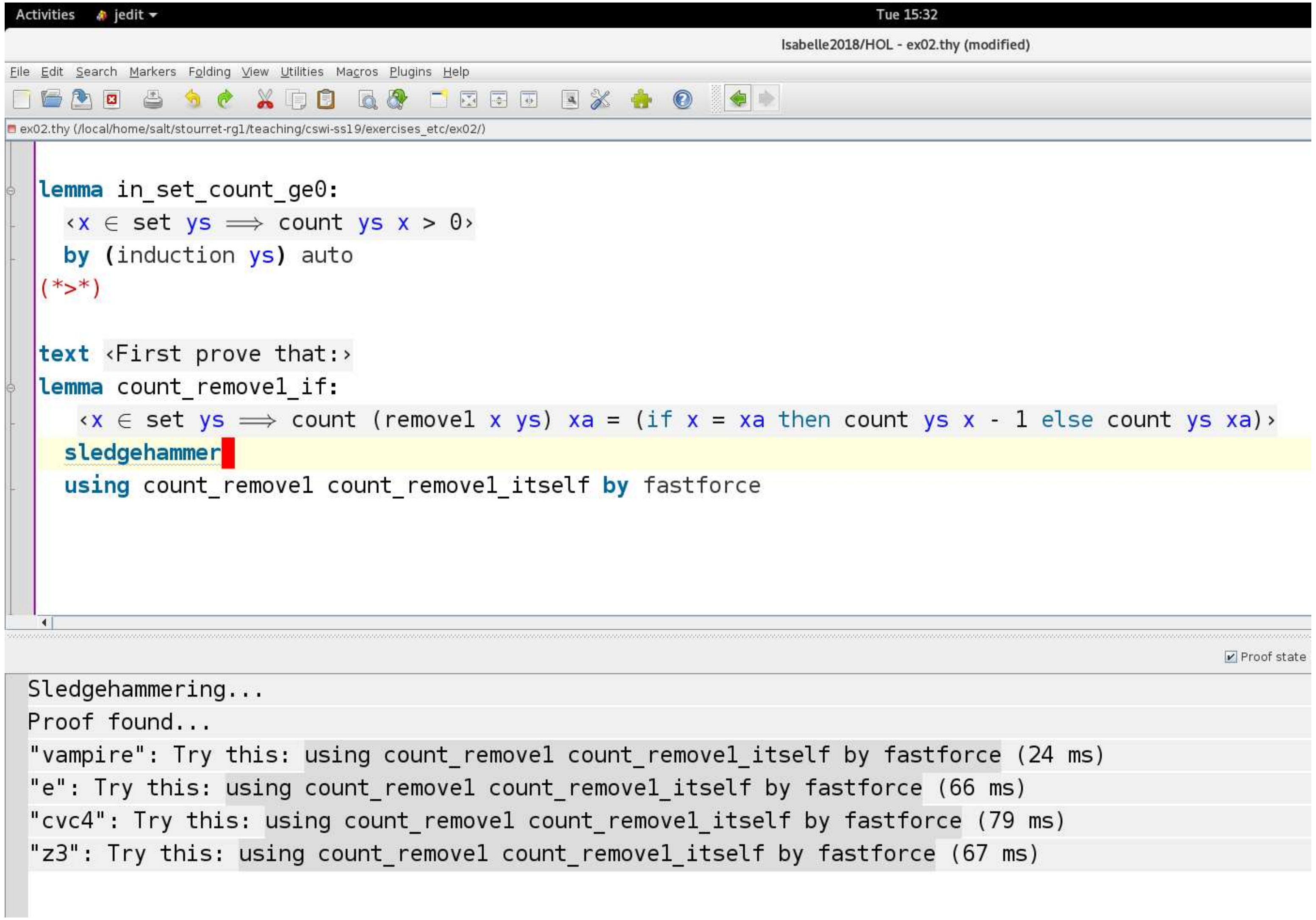
First-Order Provers in Practice



First-Order Provers in Practice



First-Order Provers in Practice



Topics of the Course

Preliminaries

abstract reduction systems

well-founded orderings

Propositional logic

syntax, semantics

calculi: DPLL procedure, OBDDs

Topics of the Course

First-order predicate logic

syntax, semantics, model theory, ...

calculi: resolution, tableaux

First-order predicate logic with equality

term rewriting systems

calculi: Knuth–Bendix completion, unfailing completion, superposition

Topics of the Course

Emphasis on:

logics and their properties,

proof systems for these logics and their properties:

soundness, completeness, implementation

Part 1: Preliminaries

Literature:

Franz Baader and Tobias Nipkow: *Term Rewriting and All That*, Cambridge Univ. Press, 1998, Chapter 2.

Before we start with the main subjects of the lecture, we repeat some prerequisites from mathematics and computer science and introduce some tools that we will need throughout the lecture.

1.1 Mathematical Prerequisites

$\mathbb{N} = \{0, 1, 2, \ldots\}$ is the set of natural numbers (including 0).

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

$\emptyset$ is the empty set.

If M and M' are sets, then $M \cap M'$, $M \cup M'$, and $M \setminus M'$ denote the intersection, union, and set difference of M and M'.

The subset relation is denoted by $\subseteq$. The strict subset relation is denoted by $\subset$ (i.e., $M \subset M'$ if and only if $M \subseteq M'$ and $M \neq M'$).

Relations

Let M be a set, let $n \geq 2$.
We write M^n for the n-fold cartesian product $M \times \cdots \times M$.

To handle the cases $n \geq 2$, $n = 1$, and $n = 0$ simultaneously,
we also define $M^1 = M$ and $M^0 = \{()\}$.

(We do not distinguish between an element m of M and
a 1-tuple (m) of an element of M.)

Relations

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

We often use predicate notation for relations:

Instead of $(m_1, \ldots, m_n) \in R$ we write $R(m_1, \ldots, m_n)$,
and say that $R(m_1, \ldots, m_n)$ holds or is true.

For binary relations, we often use infix notation, so
$(m, m') \in\, < \;\Leftrightarrow\; <(m, m') \;\Leftrightarrow\; m < m'$.

Relations

Since relations are sets, we can use the usual set operations for them.

Example:

Let $R = \{(0,2),\ (1,2),\ (2,2),\ (3,2)\} \subseteq \mathbb{N} \times \mathbb{N}$.

$$
\begin{aligned}
\text{Then } R \cap < &= R \cap \{(n,m) \in \mathbb{N} \times \mathbb{N} \mid n < m\} \\
&= \{(0,2),\ (1,2)\}.
\end{aligned}
$$

A relation Q is a subrelation of a relation R if $Q \subseteq R$.

Words

Given a nonempty set (also called alphabet) Σ,
the set Σ^* of finite words over Σ is defined inductively by

(i) the empty word ε is in Σ^*,

(ii) if $u \in \Sigma^*$ and $a \in \Sigma$ then ua is in Σ^*.

The set of nonempty finite words Σ^+ is $\Sigma^* \setminus \{\varepsilon\}$.

The concatenation of two words $u, v \in \Sigma^*$ is denoted by uv.

Words

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

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

(ii) $|ua| := |u| + 1$ for any $u \in \Sigma^*$ and $a \in \Sigma$.

1.2 Abstract Reduction Systems

Throughout the lecture, we will have to work with reduction systems.

An abstract reduction system is a pair $(A, \rightarrow)$, where

A is a nonempty set,

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

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

Abstract Reduction Systems

Let $\to' \subseteq A \times A$ and $\to'' \subseteq A \times A$ be two binary relations. Then the composition of $\to'$ and $\to''$ is the binary relation $(\to' \circ \to'') \subseteq A \times A$ defined by

$$a \, (\to' \circ \to'') \, c \quad \text{if and only if}$$
$$\text{there exists some } b \in A \text{ such that } a \to' b \text{ and } b \to'' c.$$

Abstract Reduction Systems

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

Abstract Reduction Systems

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

b is in normal form (or 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.

Abstract Reduction Systems

A relation $\rightarrow$ is called

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

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

Abstract Reduction Systems

Lemma 1.2.1:
If $\rightarrow$ is terminating, then it is normalizing.

Note: The reverse implication does not hold (see exercise).

1.3 Orderings

Important properties of binary relations:

Let $M \neq \emptyset$. A binary relation $R \subseteq M \times M$ is called

reflexive if $R(x, x)$ for all $x \in M$,

irreflexive if $\neg R(x, x)$ for all $x \in M$,

antisymmetric if $R(x, y)$ and $R(y, x)$ imply $x = y$ for all $x, y \in M$,

transitive if $R(x, y)$ and $R(y, z)$ imply $R(x, z)$ for all $x, y, z \in M$,

total if $R(x, y)$ or $R(y, x)$ or $x = y$ for all $x, y \in M$.

Orderings

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

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

Orderings

Let $\succ$ be a strict partial ordering on M; let $M' \subseteq M$.

$a \in M'$ is called minimal in M' if there is no $b \in M'$ with $a \succ b$.

$a \in M'$ is called smallest in M' if $b \succ a$ for all $b \in M' \setminus \{a\}$.

Analogously:

$a \in M'$ is called maximal in M' if there is no $b \in M'$ with $a \prec b$.

$a \in M'$ is called largest in M' if $b \prec a$ for all $b \in M' \setminus \{a\}$.

Orderings

Notation:

$M^{\prec x} = \{y \in M \mid y \prec x\}$,

$M^{\preceq x} = \{y \in M \mid y \preceq x\}$.

A subset $M' \subseteq M$ is called downward-closed if $x \in M'$ and $x \succ y$ implies $y \in M'$.

Well-Foundedness

Termination of reduction systems is strongly related to the concept of well-founded orderings.

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

Well-Foundedness and Termination

Lemma 1.3.1:
If $\succ$ is a well-founded partial ordering and $\rightarrow\ \subseteq\ \succ$,
then $\rightarrow$ is terminating.

Lemma 1.3.2:
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. Define their lexicographic combination

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

on $M_1 \times M_2$ by

$$(a_1, a_2) \succ (b_1, b_2) \quad :\Leftrightarrow \quad a_1 \succ_1 b_1 \text{ or } (a_1 = b_1 \text{ and } a_2 \succ_2 b_2)$$

(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

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

is a well-founded ordering on the set Σ^* of finite words over the alphabet Σ.

Nonexamples:

$(\mathbb{Z}, >)$

$(\mathbb{N}, <)$

the lexicographic ordering on Σ^*

Basic Properties of Well-Founded Orderings

Lemma 1.3.3:
$(M, \succ)$ is well-founded if and only if every nonempty $M' \subseteq M$ has a minimal element.

Lemma 1.3.4:
$(M_1, \succ_1)$ and $(M_2, \succ_2)$ are well-founded if and only if
$(M_1 \times M_2,\ \succ)$ with $\succ = (\succ_1, \succ_2)_{\mathrm{lex}}$ is well-founded.

Monotone Mappings

Let $(M, \succ)$ and $(M', \succ')$ be strict partial orderings.
A mapping $\varphi : M \to M'$ is called monotone
if $a \succ b$ implies $\varphi(a) \succ' \varphi(b)$ for all $a, b \in M$.

Lemma 1.3.5:
If φ is a monotone mapping from $(M, \succ)$ to $(M', \succ')$
and $(M', \succ')$ is well-founded, then $(M, \succ)$ is well-founded.

Well-Founded Induction

Theorem 1.3.6 (Well-Founded (or 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

Well-Founded Recursion

Let M and S be sets, let $N \subseteq M$, and let $f : M \to S$ be a function. Then the restriction of f to N, denoted by $f\big|_N$, is a function from N to S with $f\big|_N(x) = f(x)$ for all $x \in N$.

Theorem 1.3.7 (Well-Founded (or Noetherian) Recursion):
Let $(M, \succ)$ be a well-founded ordering, let S be a set. Let ϕ be a binary function that takes two arguments x and g and maps them to an element of S, where $x \in M$ and g is a function from $M^{\prec x}$ to S.

Then there exists exactly one function $f : M \to S$ such that for all $x \in M$

$$f(x) = \phi(x, f\big|_{M^{\prec x}})$$

Well-Founded Recursion

The well-founded recursion scheme generalizes terminating recursive programs.

Note that functions defined by well-founded recursion need *not* be computable, in particular since for many well-founded orderings the sets $M^{\prec x}$ may be infinite.