



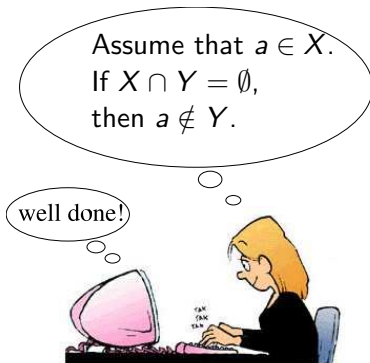
Verification of Human-level Proof Steps in Mathematics Education

Dominik Dietrich
Joint work with Mark Buckley

CADGME – Session ARME – 20th-23rd June 2007

Tutorial Dialog for Mathematics: help with solving maths problems/learning maths.

- ▶ Mathematical domain reasoning
- ▶ Dynamic generation of solutions
- ▶ Unrestricted input (human-level proof steps)
- ▶ Want to use theorem proving technology



- 1 Problems for the Verification
- 2 Choosing the Deduction System
- 3 Verification Algorithm
- 4 Evaluation & Summary

exercise given:

$$(R \circ S)^{-1} = S^{-1} \circ R^{-1}$$

(R, S binary relations)

Example 1

let $(x, y) \in (R \circ S)^{-1}$

correct

hence $(y, x) \in (S \circ R)$

incorrect

hence $(y, x) \in (R \circ S)$

correct

⋮

Example 2

we consider the subgoals

$$(R \circ S)^{-1} \subset S^{-1} \circ R^{-1}$$

$$\text{and } (R \circ S)^{-1} \supset S^{-1} \circ R^{-1}$$

Example 3

first, we consider the subgoal

$$(R \circ S)^{-1} \subset S^{-1} \circ R^{-1}$$

Example 1: to show: $(S \circ R)^{-1} = S^{-1} \circ R^{-1}$

let $(x, y) \in (R \circ S)^{-1}$



- ▶ Student decomposes the problem in \subset and \supset by $=$, giving two “directions”
- ▶ Starts with the “first direction”
 $(R \circ S)^{-1} \subset R \circ S$
- ▶ As $A \subset B \Leftrightarrow \forall x. x \in A \Rightarrow x \in B$ the assumption is a reasonable start



Example 2: $(S \circ R)^{-1} = S^{-1} \circ R^{-1}$

We consider the subgoals

$$(R \circ S)^{-1} \subset S^{-1} \circ R^{-1}$$

$$\text{and } (R \circ S)^{-1} \supset S^{-1} \circ R^{-1}$$

- ▶ What is the subgoal the student is working on?

Example 1: to show: $(S \circ R)^{-1} = S^{-1} \circ R^{-1}$

let $(x, y) \in (R \circ S)^{-1}$

- ▶ What is the assertion to be proved?

Example 3: to show: $(S \circ R)^{-1} = S^{-1} \circ R^{-1}$

First, we consider the subgoal $(R \circ S)^{-1} \subset S^{-1} \circ R^{-1}$

- ▶ Need all subgoals to be able to verify:
 $(R \circ S)^{-1} \subset S^{-1} \circ R^{-1} \not\Rightarrow (S \circ R)^{-1} = S^{-1} \circ R^{-1}$

Requirements

- ▶ Structure of reconstructed proof should correspond to the structure of the student's proof
- ▶ Need white box integration
 - ▶ to analyse unstated subgoals
 - ▶ to check whether assumptions are reasonable
- ▶ Not sufficient to find exactly one proof, but a set of proofs for further analysis
- ▶ System needs to be able to express alternatives

Machine oriented calculi, e.g. resolution

- ▶ Resulting proofs are different from human proofs
 - ▶ difficult to extract structure of the proof
 - ▶ difficult to inspect proof goals/subgoals
- ▶ Standard implementations
 - ▶ only return exactly one proof
 - ▶ do not support alternatives

```

3 [] setequal(x,y) — -subset(x,y) — -subset(y,x).
5 [] subset(x,y) — -member($f1(x,y),y).
22 [] -setequal(intersection($c2,$c1),intersection($c2,$c1)).
23 [factor,3.2.3] setequal(x,x) — -subset(x,x).
27 [] subset(x,y) — member($f1(x,y),x).
29 [hyper,27,23] member($f1(x,x),x) — setequal(x,x).
32 [hyper,29,22]
member($f1(intersection($c2,$c1),intersection($c2,$c1)),intersection($c2,$c1)).
41 [hyper,32,5] subset(intersection($c2,$c1),intersection($c2,$c1)).
53 [hyper,41,23] setequal(intersection($c2,$c1),intersection($c2,$c1)).
54 [binary,53.1,22.1] $F.

```

- ▶ Based on a fixed set of **introduction** and **elimination** rules
- ▶ Structure is similar to structure of a human proof
- ▶ However, much more detailed

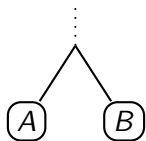
$$\frac{\frac{\frac{A := (z, y) \in r^{-1} \wedge (x, z) \in s^{-1}}{(y, z) \in r^{-1}} \wedge E}{(y, z) \in r} \text{Def.}^{-1}}{(y, z) \in r \wedge (x, z) \in s^{-1}} \wedge I}{(x, z) \in s^{-1}} \wedge E$$

Natural Deduction

- ▶ Operators are **dynamically** generated from theory (i.e. axioms)
- ▶ **Deep inference** mechanism: \Rightarrow short proofs
- ▶ What about alternatives?

$$\frac{(z, y) \in r^{-1} \wedge (x, z) \in s^{-1}}{(y, z) \in r \wedge (x, z) \in s^{-1}} \text{ Def}$$

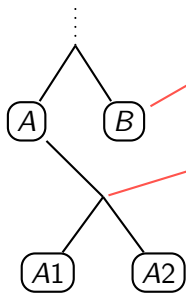
Ω MEGA^{Core}



The Proof Data Structure (PDS):

Tasks $\varphi_1, \dots, \varphi_n \vdash \psi_1, \psi_2, \dots, \psi_m$

e.g. $\vdash (R \circ S)^{-1} = S^{-1} \circ R^{-1}$

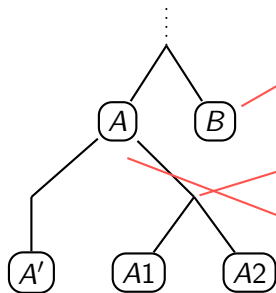


The Proof Data Structure (PDS):

Tasks $\varphi_1, \dots, \varphi_n \vdash \psi_1, \psi_2, \dots, \psi_m$

e.g. $\vdash (R \circ S)^{-1} = S^{-1} \circ R^{-1}$

Reduction: Each operator reduces a task to a set of subtasks



The Proof Data Structure (PDS):

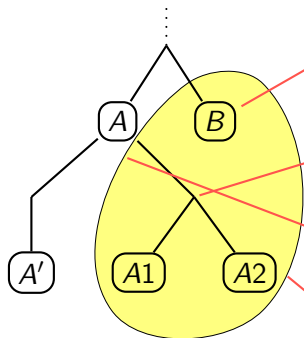
Tasks $\varphi_1, \dots, \varphi_n \vdash \psi_1, \psi_2, \dots, \psi_m$

e.g. $\vdash (R \circ S)^{-1} = S^{-1} \circ R^{-1}$

Reduction: Each operator reduces a task to a set of subtasks

Alternatives: can be represented within one datastructures

Omega's Proof Data Structure



The Proof Data Structure (PDS):

Tasks $\varphi_1, \dots, \varphi_n \vdash \psi_1, \psi_2, \dots, \psi_m$

e.g. $\vdash (R \circ S)^{-1} = S^{-1} \circ R^{-1}$

Reduction: Each operator reduces a task to a set of subtasks

Alternatives: can be represented within one datastructures

Agenda $\langle \{A1, A2, B\}, \sigma \rangle$ to manage goals and substitutions for proof alternative

- ▶ PDS holds all proof states the student might be in, called **student proof states**
- ▶ Initially there is a unique proof state, for our example $\langle \{ \vdash (R \circ S)^{-1} = S^{-1} \circ R^{-1} \}, \emptyset \rangle$
- ▶ In addition: a set of **proof operators** \mathcal{OP} the student is supposed to know

Algorithm

- ▶ Verification/Completion (for each agenda):
 - ▶ Expand student proof state
 - ▶ Filter consistent successor tasks
 - ▶ Complete Tasks to agendas
 - ▶ Prune away unused nodes
 - ▶ Check for identical agendas
-
- ▶ Step is analysed to be **correct** if at least one consistent successor task can be found for one agenda
 - ▶ Otherwise **incorrect**

Hence φ

- ▶ Only allow transformations of assumptions
- ▶ Look for an assumption \approx input φ

Let φ

- ▶ Look for an assumption \approx input φ
- ▶ Treat free variables as eigenvariables

Subgoal(s) $\Gamma \vdash \Delta$

- ▶ Look for a goal \approx goal Δ of input, and which has assumptions Γ of input in context
- ▶ Only use transformations modifying the goals

Conjecture Ψ

- ▶ Create a new proof tree with $\vdash \Psi$ as initial task
- ▶ Try to prove it
- ▶ Extend OP

Example

Show that $\vdash (R \circ S)^{-1} = S^{-1} \circ R^{-1} (=: T_0)$

let $(x, y) \in (R \circ S)^{-1}$

treat x, y as

eigenvariables

T_0

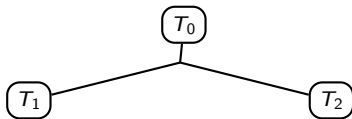
Example

Show that $\vdash (R \circ S)^{-1} = S^{-1} \circ R^{-1} (=: T_0)$

let $(x, y) \in (R \circ S)^{-1}$ $T_1: \vdash (R \circ S)^{-1} \subset S^{-1} \circ R^{-1}$

treat x, y as $T_2: \vdash S^{-1} \circ R^{-1} \subset (R \circ S)^{-1}$

eigenvariables



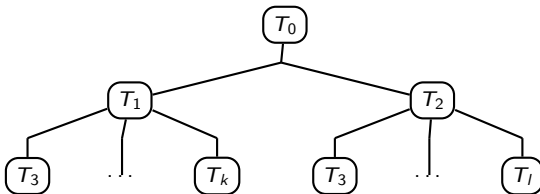
Example

Show that $\vdash (R \circ S)^{-1} = S^{-1} \circ R^{-1}$ ($=: T_0$)

let $(x, y) \in (R \circ S)^{-1}$ $T_3: (a, b) \in (R \circ S)^{-1} \vdash (a, b) \in S^{-1} \circ R^{-1}$

treat x, y as

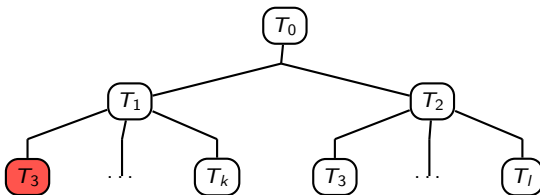
eigenvariables



Example

Show that $\vdash (R \circ S)^{-1} = S^{-1} \circ R^{-1}$ ($=: T_0$)

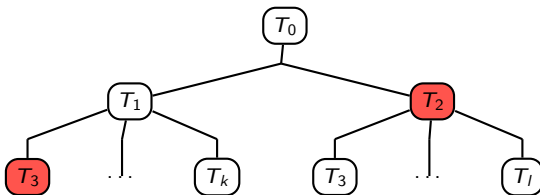
let $(x, y) \in (R \circ S)^{-1}$ T_3 : $(a, b) \in (R \circ S)^{-1}$ $\vdash (a, b) \in S^{-1} \circ R^{-1}$
 treat x, y as
 eigenvariables



Example

Show that $\vdash (R \circ S)^{-1} = S^{-1} \circ R^{-1} (=: T_0)$

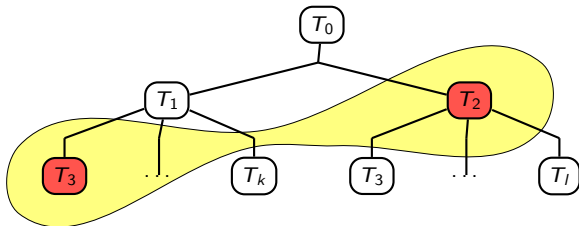
let $(x, y) \in (R \circ S)^{-1}$ $T_3: (a, b) \in (R \circ S)^{-1} \vdash (a, b) \in S^{-1} \circ R^{-1}$
 treat x, y as $T_2: \vdash S^{-1} \circ R^{-1} \subset (R \circ S)^{-1}$
 eigenvariables



Example

Show that $\vdash (R \circ S)^{-1} = S^{-1} \circ R^{-1} (=: T_0)$

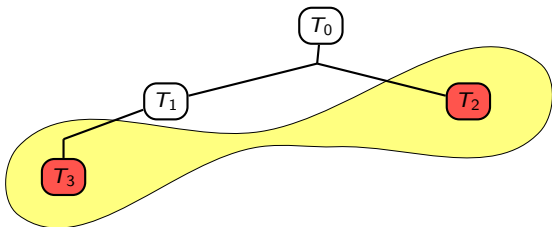
let $(x, y) \in (R \circ S)^{-1}$ $T_3: (a, b) \in (R \circ S)^{-1} \vdash (a, b) \in S^{-1} \circ R^{-1}$
 treat x, y as $T_2: \vdash S^{-1} \circ R^{-1} \subset (R \circ S)^{-1}$
 eigenvariables



Example

Show that $\vdash (R \circ S)^{-1} = S^{-1} \circ R^{-1}$ ($=: T_0$)

let $(x, y) \in (R \circ S)^{-1}$ T_3 : $(x, y) \in (R \circ S)^{-1} \vdash (x, y) \in S^{-1} \circ R^{-1}$
 treat x, y as
 eigenvariables



- ▶ 17 human-level proofs by real students

Results

correctly rejected:	28
correctly accepted:	113
wrongly accepted:	0
not verified:	3

Runtime

- ▶ Usually less than 1s if correct
- ▶ Wrong steps, conjecture can take longer (usually 10s)

Summary

- ▶ Want **dynamic solutions** and **unrestricted user input**
- ▶ Problems are **underspecification**, **ambiguity**, and **incomplete information**
- ▶ Need a **human oriented calculus**, **white box integration** (direct access to **proof state**)
- ▶ **BFS** algorithm tries to resolve ambiguity
 - ▶ propagates **alternatives** if ambiguity/underspecification cannot be resolved
 - ▶ BFS feasible due to **deep inference** and **pruning**
- ▶ Case study: can check 96%