

# Automated Reasoning in the Sequent Calculus Trainer

A. Ehle, N. Hundeshagen, and M. Lange

Fachbereich Elektrotechnik/Informatik, Universität Kassel,  
Kassel, Germany

THEdu 2017, Göteborg

# Logic at Kassel University

---

## Situation:

- ▶ second year mandatory course for bachelor students
- ▶ approx. 100 students
- ▶ content: standard topics from Prop. and FO-logic

## Inverted Classroom Model:

- ▶ aim: improving learning outcomes
- ▶ learning as a self-organized activity
- ↪ tools to assist and self-assess certain topics

# Logic at Kassel University

---

## Situation:

- ▶ second year mandatory course for bachelor students
- ▶ approx. 100 students
- ▶ content: standard topics from Prop. and FO-logic

## Inverted Classroom Model:

- ▶ aim: improving learning outcomes
- ▶ learning as a self-organized activity
- ↪ tools to assist and self-assess certain topics

# The Sequent Calculus

► proof calculus similar to natural deduction

► **sequent:**  $\Gamma \Rightarrow \Delta$

► **rule:** e.g. 
$$\frac{\Gamma \Rightarrow \varphi, \Delta \quad \Gamma \Rightarrow \psi, \Delta}{\Gamma \Rightarrow \varphi \wedge \psi, \Delta}$$

proof:

$$\begin{array}{c}
 \frac{}{A \Rightarrow A, B} \text{ (Ax)} \quad \frac{\frac{}{B, C \Rightarrow A, B} \text{ (Ax)}}{B \wedge C \Rightarrow A, B} \text{ (}\wedge\text{)} \quad \frac{}{A \Rightarrow A, C} \text{ (Ax)} \quad \frac{\frac{}{B, C \Rightarrow A, C} \text{ (Ax)}}{B \wedge C \Rightarrow A, C} \text{ (}\wedge\text{)} \\
 \hline
 \frac{}{A \vee (B \wedge C) \Rightarrow A, B} \text{ (}\vee\text{)} \quad \frac{}{A \vee (B \wedge C) \Rightarrow A, C} \text{ (}\vee\text{)} \\
 \hline
 \frac{A \vee (B \wedge C) \Rightarrow A \vee B \quad A \vee (B \wedge C) \Rightarrow A \vee C}{A \vee (B \wedge C) \Rightarrow (A \vee B) \wedge (A \vee C)} \text{ (}\wedge\text{)}
 \end{array}$$

# Didactical Perspective

# Steps in Learning a Proof Calculus

---

We distinguish between:

constructing a correct proof vs. finding the right proof

constructing a correct proof

# Steps in Learning a Proof Calculus

---

We distinguish between:

constructing a correct proof vs. finding the right proof

constructing a correct proof

$$\overline{\forall x \forall y. E(x, y) \rightarrow x = f(y)} \implies \forall x \forall y \forall z. E(x, z) \wedge E(y, z) \rightarrow x = y$$

# Steps in Learning a Proof Calculus

---

We distinguish between:

constructing a correct proof vs. finding the right proof

constructing a correct proof

$$\frac{}{\forall x \forall y. E(x, y) \rightarrow x = f(y) \implies \forall x \forall y \forall z. E(x, z) \wedge E(y, z) \rightarrow x = y} 3 \times (\forall_R)$$



# Steps in Learning a Proof Calculus

---

We distinguish between:

constructing a correct proof vs. finding the right proof

constructing a correct proof

$$\frac{\forall x \forall y. E(x, y) \rightarrow x = f(y) \implies E(a, c) \wedge E(b, c) \rightarrow a = b}{\forall x \forall y. E(x, y) \rightarrow x = f(y) \implies \forall x \forall y \forall z. E(x, z) \wedge E(y, z) \rightarrow x = y} 3 \times (\forall_R)$$

# Steps in Learning a Proof Calculus

---

We distinguish between:

constructing a correct proof vs. finding the right proof

constructing a correct proof

$$\frac{\frac{}{\forall x \forall y. E(x, y) \rightarrow x = f(y) \implies E(a, c) \wedge E(b, c) \rightarrow a = b} (\rightarrow_R)}{\forall x \forall y. E(x, y) \rightarrow x = f(y) \implies \forall x \forall y \forall z. E(x, z) \wedge E(y, z) \rightarrow x = y} 3 \times (\forall_R)$$

# Steps in Learning a Proof Calculus

---

We distinguish between:

constructing a correct proof vs. finding the right proof

constructing a correct proof

$$\frac{\frac{\frac{}{\forall x \forall y. E(x, y) \rightarrow x = f(y), E(a, c) \wedge E(b, c) \Longrightarrow a = b}}{\forall x \forall y. E(x, y) \rightarrow x = f(y) \Longrightarrow E(a, c) \wedge E(b, c) \rightarrow a = b} (\rightarrow_R)}{\forall x \forall y. E(x, y) \rightarrow x = f(y) \Longrightarrow \forall x \forall y \forall z. E(x, z) \wedge E(y, z) \rightarrow x = y} 3 \times (\forall_R)$$

# Steps in Learning a Proof Calculus

---

We distinguish between:

constructing a correct proof vs. finding the right proof

constructing a correct proof

$$\frac{\frac{\frac{}{\forall x \forall y. E(x, y) \rightarrow x = f(y), E(a, c) \wedge E(b, c) \Longrightarrow a = b} (\wedge_L)}{\forall x \forall y. E(x, y) \rightarrow x = f(y) \Longrightarrow E(a, c) \wedge E(b, c) \rightarrow a = b} (\rightarrow_R)}{\forall x \forall y. E(x, y) \rightarrow x = f(y) \Longrightarrow \forall x \forall y \forall z. E(x, z) \wedge E(y, z) \rightarrow x = y} 3 \times (\forall_R)$$

# Steps in Learning a Proof Calculus

---

We distinguish between:

constructing a correct proof vs. finding the right proof

constructing a correct proof

$$\begin{array}{c}
 \vdots \\
 \hline
 \forall x \forall y. E(x, y) \rightarrow x = f(y), E(a, c), E(b, c) \Longrightarrow a = b \\
 \hline
 \forall x \forall y. E(x, y) \rightarrow x = f(y), E(a, c) \wedge E(b, c) \Longrightarrow a = b \quad (\wedge_L) \\
 \hline
 \forall x \forall y. E(x, y) \rightarrow x = f(y) \Longrightarrow E(a, c) \wedge E(b, c) \rightarrow a = b \quad (\rightarrow_R) \\
 \hline
 \forall x \forall y. E(x, y) \rightarrow x = f(y) \Longrightarrow \forall x \forall y \forall z. E(x, z) \wedge E(y, z) \rightarrow x = y \quad 3 \times (\forall_R)
 \end{array}$$

# Steps in Learning a Proof Calculus

---

We distinguish between:

constructing a correct proof vs. finding the right proof

constructing a correct proof

$$\begin{array}{c}
 \vdots \\
 \hline
 \forall x \forall y. E(x, y) \rightarrow x = f(y), E(a, c), E(b, c) \Longrightarrow a = b \\
 \hline
 \forall x \forall y. E(x, y) \rightarrow x = f(y), E(a, c) \wedge E(b, c) \Longrightarrow a = b \quad (\wedge_L) \\
 \hline
 \forall x \forall y. E(x, y) \rightarrow x = f(y) \Longrightarrow E(a, c) \wedge E(b, c) \rightarrow a = b \quad (\rightarrow_R) \\
 \hline
 \forall x \forall y. E(x, y) \rightarrow x = f(y) \Longrightarrow \forall x \forall y \forall z. E(x, z) \wedge E(y, z) \rightarrow x = y \quad 3 \times (\forall_R)
 \end{array}$$

- students already have major problems in achieving the first goal

# A Tool for “Constructing Correct Proofs”

---

## The Sequent Calculus Trainer Version 1 ... (Ehle, H., Lange 2015)

- ▶ a verifier of proof trees, not an assistant
- ▶ clear and extensive feedback system

## Experiences:

- ▶ comparison of written exam results shows significant increase in number of students who construct correct proofs
- ▶ not a very “scientific” study
- ▶ however, effect too significant to be caused solely by other reasons
- ▶ tool seems to replace right amount of pen and paper work

# A Tool for “Constructing Correct Proofs”

---

## The Sequent Calculus Trainer Version 1 ... (Ehle, H., Lange 2015)

- ▶ a verifier of proof trees, not an assistant
- ▶ clear and extensive feedback system

## Experiences:

- ▶ comparison of written exam results shows significant increase in number of students who construct correct proofs
- ▶ not a very “scientific” study
- ▶ however, effect too significant to be caused solely by other reasons
- ▶ tool seems to replace right amount of pen and paper work



# A Tool for “Constructing Correct Proofs”

---

## The Sequent Calculus Trainer Version 1 ... (Ehle, H., Lange 2015)

- ▶ a verifier of proof trees, not an assistant
- ▶ clear and extensive feedback system

## Experiences:

- ▶ comparison of written exam results shows significant increase in number of students who construct correct proofs
- ▶ not a very “scientific” study
- ▶ however, effect too significant to be caused solely by other reasons
- ▶ tool seems to replace right amount of pen and paper work

# A Tool for “Constructing Correct Proofs”

---

## The Sequent Calculus Trainer Version 1 ... (Ehle, H., Lange 2015)

- ▶ a verifier of proof trees, not an assistant
- ▶ clear and extensive feedback system

## Experiences:

- ▶ comparison of written exam results shows significant increase in number of students who construct correct proofs
- ▶ not a very “scientific” study
- ▶ however, effect too significant to be caused solely by other reasons
- ▶ tool seems to replace right amount of pen and paper work

# A Tool for “Constructing Correct Proofs”

---

## The Sequent Calculus Trainer Version 1 ... (Ehle, H., Lange 2015)

- ▶ a verifier of proof trees, not an assistant
- ▶ clear and extensive feedback system

## Experiences:

- ▶ comparison of written exam results shows significant increase in number of students who construct correct proofs
- ▶ not a very “scientific” study
- ▶ however, effect too significant to be caused solely by other reasons
- ▶ tool seems to replace right amount of pen and paper work

# A Tool for “Finding the Right Proof”

---

Recall:

$$\frac{}{\forall x \forall y. E(x, y) \rightarrow x = f(y), E(a, c), E(b, c) \implies a = b} 2 \times (\forall_L)$$

# A Tool for “Finding the Right Proof”

---

Recall:

$$\frac{E(b, c) \rightarrow b = f(c), E(a, c), E(b, c) \implies a = b}{\forall x \forall y. E(x, y) \rightarrow x = f(y), E(a, c), E(b, c) \implies a = b} 2 \times (\forall_L)$$

# A Tool for “Finding the Right Proof”

---

Recall:

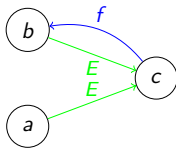
$$\frac{E(b, c) \rightarrow b = f(c), E(a, c), E(b, c) \implies a = b}{\forall x \forall y. E(x, y) \rightarrow x = f(y), E(a, c), E(b, c) \implies a = b} 2 \times (\forall_L)$$

# A Tool for “Finding the Right Proof”

Recall:

$$\frac{E(b, c) \rightarrow b = f(c), E(a, c), E(b, c) \implies a = b}{\forall x \forall y. E(x, y) \rightarrow x = f(y), E(a, c), E(b, c) \implies a = b} 2 \times (\forall_L)$$

counter model:

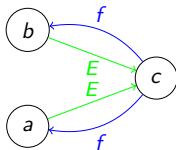


# A Tool for “Finding the Right Proof”

Recall:

$$\frac{E(b, c) \rightarrow b = f(c), E(a, c), E(b, c) \implies a = b}{\forall x \forall y. E(x, y) \rightarrow x = f(y), E(a, c), E(b, c) \implies a = b} 2 \times (\forall_L)$$

counter model:

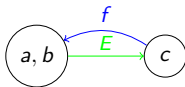




# A Tool for “Finding the Right Proof”

Recall:

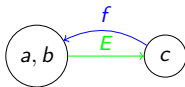
$$\frac{E(b, c) \rightarrow b = f(c), E(a, c), E(b, c) \implies a = b}{\forall x \forall y. E(x, y) \rightarrow x = f(y), E(a, c), E(b, c) \implies a = b} 2 \times (\forall_L)$$



# A Tool for “Finding the Right Proof”

Recall:

$$\frac{E(b, c) \rightarrow b = f(c), E(a, c), E(b, c) \implies a = b}{\forall x \forall y. E(x, y) \rightarrow x = f(y), E(a, c), E(b, c) \implies a = b} 2 \times (\forall_L)$$



## Specifications for Version 2

- ▶ stepping out of purely syntactical reasoning
- ▶ direct students' focus to the underlying structure

# The Sequent Calculus Trainer with Automated Reasoning

# Main View

Sequent Calculus Trainer

File Edit Help

Propositional Logic First-Order Logic

Rule set

- $\text{Ax}$   $\text{Reflexivity} - R$
- $\neg I$   $\neg E$
- $\neg L$   $\neg R$
- $\vee L$   $\vee R$
- $\wedge L$   $\wedge R$
- $\rightarrow L$   $\rightarrow R$
- $\leftrightarrow L$   $\leftrightarrow R$
- $\exists L$   $\exists R$
- $\forall L$   $\forall R$
- Substitution - L Substitution - R
- Contraction - L Contraction - R
- Reflexivity - L Weakening
- Delete subgoal

$$\frac{\frac{\frac{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow F(c)}{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow F(c) \wedge F(v(c))} (\wedge_R) \quad \frac{\frac{\forall x \exists y x=v(y), F(v(c)) \Rightarrow F(v(c))}{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow F(v(c))} (\forall_L) \quad \text{Ax}}{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow \forall x .F(x) \wedge F(v(x))} (\forall_R)$$

Zoom

# Advice via Traffic Light System

Sequent Calculus Trainer

File Edit Help

Propositional Logic First-Order Logic

Rule set

$$\frac{\frac{\frac{\exists y \ c=v(y), F(v(c)) \Rightarrow F(c)}{\forall x \ \exists y \ x=v(y), F(v(c)) \Rightarrow F(c)}^{(\forall_L)} \quad \frac{\forall x \ \exists y \ x=v(y), F(v(c)) \Rightarrow F(v(c))}{\forall x \ \exists y \ x=v(y), \forall x \ F(v(x)) \Rightarrow F(c)}^{(\forall_L)} \quad \frac{\forall x \ \exists y \ x=v(y), F(v(c)) \Rightarrow F(v(c))}{\forall x \ \exists y \ x=v(y), \forall x \ F(v(x)) \Rightarrow F(v(c))}^{(Ax)} \quad \frac{\forall x \ \exists y \ x=v(y), \forall x \ F(v(x)) \Rightarrow F(c) \wedge F(v(c))}{\forall x \ \exists y \ x=v(y), \forall x \ F(v(x)) \Rightarrow \forall x . F(x) \wedge F(v(x))}^{(\forall_R)} \quad \frac{\forall x \ \exists y \ x=v(y), \forall x \ F(v(x)) \Rightarrow \forall x . F(x) \wedge F(v(x))}{\forall x \ \exists y \ x=v(y) \wedge \forall x \ F(v(x)) \Rightarrow \forall x . F(x) \wedge F(v(x))}^{(\wedge_L)}$$

Zoom

Rule set  
 Ax Reflexivity - R  
 H - L H - R  
 ~ - L ~ - R  
 v - L v - R  
 ^ - L ^ - R  
 => - L => - R  
 => + L => + R  
 @ - L @ - R  
 v + L v + R  
 Substitution - L Substitution - R  
 Contraction - L Contraction - R  
 Reflexivity - L Weakening  
 Delete subterm

# Advice via Traffic Light System

Sequent Calculus Trainer

File Edit Help

Propositional Logic First-Order Logic

Rule set

- $\forall$  - L
- $\forall$  - R
- $\exists$  - L
- $\exists$  - R
- $\rightarrow$  - L
- $\rightarrow$  - R
- $\rightarrow$  + L
- $\rightarrow$  + R
- $\leftrightarrow$  - L
- $\leftrightarrow$  - R
- $\leftrightarrow$  + L
- $\leftrightarrow$  + R
- $\wedge$  - L
- $\wedge$  - R
- $\wedge$  + L
- $\wedge$  + R
- $\vee$  - L
- $\vee$  - R
- $\vee$  + L
- $\vee$  + R
- Substitution - L
- Substitution - R
- Contraction - L
- Contraction - R
- Reflexivity - L
- Weakening
- Delete subtree

$$\frac{\frac{\forall x \exists y x=v(y), F(v(c)) \Rightarrow F(c)}{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow F(c)} (\forall_L) \quad \frac{\forall x \exists y x=v(y), F(v(c)) \Rightarrow F(v(c))}{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow F(v(c))} (\forall_R)}{\frac{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow F(c) \wedge F(v(c))}{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow \forall x .F(x) \wedge F(v(x))} (\forall_R)} (\wedge_L)$$

Zoom

# Dialog System

The screenshot shows the Sequent Calculus Trainer (SCT) interface. The main window displays a logical derivation in sequent calculus. The derivation is as follows:

$$\begin{array}{c}
 \frac{\frac{\frac{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow F(c)}{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow F(c) \wedge F(v(c))} (\wedge_R)}{\frac{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow F(c) \wedge F(v(c))}{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow \forall x .F(x) \wedge F(v(x))} (\forall_R)} \\
 \frac{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow \forall x .F(x) \wedge F(v(x))}{\forall x \exists y x=v(y) \wedge \forall x F(v(x)) \Rightarrow \forall x .F(x) \wedge F(v(x))} (\wedge_L)
 \end{array}$$

The first sequent,  $\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow F(c)$ , is highlighted in green. An information tooltip above it says: "Click on a sequent where you need help".

On the right side, there is a "Rule set" panel with various logical rules:

- Ass** (Assumption)
- Reflexivity - R**
- R - L** (Right contraction)
- tt - R** (True on the right)
- ¬ - L** (Negation on the left)
- ¬ - R** (Negation on the right)
- ∨ - L** (Disjunction on the left)
- ∨ - R** (Disjunction on the right)
- ∧ - L** (Conjunction on the left)
- ∧ - R** (Conjunction on the right)
- ⇒ - L** (Implication on the left)
- ⇒ - R** (Implication on the right)
- ⇔ - L** (Biconditional on the left)
- ⇔ - R** (Biconditional on the right)
- ∃ - L** (Existential on the left)
- ∃ - R** (Existential on the right)
- ∀ - L** (Universal on the left)
- ∀ - R** (Universal on the right)
- Substitution - L**
- Substitution - R**
- Contraction - L**
- Contraction - R**
- Reflexivity - L**
- Weakening**
- Delete subtree**

The bottom of the window includes a "Zoom" slider.

# Dialog System

Sequent Calculus Trainer

File Edit Help

Propositional Logic First-Order Logic

Info

OK try to apply the Rule

for-all-left

to the formula

$\forall x \exists y x = v(y)$

Replace the bound variable symbol by the ground term c.

Rule set

Ass Reflexivity - R

$\Pi$  - L  $\Pi$  - R

$\neg$  - L  $\neg$  - R

$\vee$  - L  $\vee$  - R

$\wedge$  - L  $\wedge$  - R

$\rightarrow$  - L  $\rightarrow$  - R

$\leftrightarrow$  - L  $\leftrightarrow$  - R

$\exists$  - L  $\exists$  - R

$\forall$  - L  $\forall$  - R

Substitution - L Substitution - R

Contraction - L Contraction - R

Reflexivity - L Weakening

Delete subgoal

Zoom

$$\frac{\frac{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow F(c)}{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow F(c) \wedge F(v(c))} (\wedge_R) \quad \frac{\forall x \exists y x=v(y), F(v(c)) \Rightarrow F(v(c))}{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow F(v(c))} (\forall_L) \quad (Ax)}{\forall x \exists y x=v(y), \forall x F(v(x)) \Rightarrow \forall x .F(x) \wedge F(v(x))} (\wedge_L)$$



# Main Idea behind the Algorithm

---

**Problem:** Instantiation of existentially quantified variables

- note: brute-force semi-decision procedure for FO only works theoretically

instead, we use an **SMT solver** (Z3) and the following reduction:

$$\Gamma \implies \Delta \text{ is invalid iff } \bigwedge_{\psi \in \Gamma} \psi \wedge \bigwedge_{\psi' \in \Delta} \neg \psi' \text{ is satisfiable}$$

**Example:**

$$\forall x \forall y P(g(x, y)) \implies \exists x P(x)$$

# Main Idea behind the Algorithm

---

**Problem:** Instantiation of existentially quantified variables

- note: brute-force semi-decision procedure for FO only works theoretically

instead, we use an **SMT solver** (Z3) and the following reduction:

$$\Gamma \implies \Delta \text{ is invalid iff } \bigwedge_{\psi \in \Gamma} \psi \wedge \bigwedge_{\psi' \in \Delta} \neg \psi' \text{ is satisfiable}$$

Example:

$$\forall x \forall y P(g(x, y)) \implies \exists x P(x)$$

# Main Idea behind the Algorithm

---

**Problem:** Instantiation of existentially quantified variables

- note: brute-force semi-decision procedure for FO only works theoretically

instead, we use an **SMT solver** (Z3) and the following reduction:

$$\Gamma \implies \Delta \text{ is invalid iff } \bigwedge_{\psi \in \Gamma} \psi \wedge \bigwedge_{\psi' \in \Delta} \neg \psi' \text{ is satisfiable}$$

**Example:**

$$\forall x \forall y P(g(x, y)) \implies \exists x P(x)$$

# Main Idea behind the Algorithm

**Problem:** Instantiation of existentially quantified variables

- note: brute-force semi-decision procedure for FO only works theoretically

instead, we use an **SMT solver** (Z3) and the following reduction:

$$\Gamma \implies \Delta \text{ is invalid iff } \bigwedge_{\psi \in \Gamma} \psi \wedge \bigwedge_{\psi' \in \Delta} \neg \psi' \text{ is satisfiable}$$

**Example:**

1. introduce “next” topmost symbol of a groundterm

$$\begin{aligned} \forall x \forall y P(g(x, y)) &\implies \exists x P(x) \\ \forall x \forall y P(g(x, y)) &\implies P(\textcolor{red}{c}) \end{aligned}$$

# Main Idea behind the Algorithm

**Problem:** Instantiation of existentially quantified variables

- note: brute-force semi-decision procedure for FO only works theoretically

instead, we use an **SMT solver** (Z3) and the following reduction:

$$\Gamma \implies \Delta \text{ is invalid iff } \bigwedge_{\psi \in \Gamma} \psi \wedge \bigwedge_{\psi' \in \Delta} \neg \psi' \text{ is satisfiable}$$

**Example:**

- |  |   |
|--|---|
|  | $\forall x \forall y P(g(x, y)) \implies \exists x P(x)$                |
| 1. introduce “next” topmost symbol of a groundterm | $\forall x \forall y P(g(x, y)) \implies P(c)$                          |
| 2. use <b>SMT solver to verify</b>                 | $\forall x \forall y P(g(x, y)) \wedge \neg P(c)$ is <b>satisfiable</b> |

# Main Idea behind the Algorithm

**Problem:** Instantiation of existentially quantified variables

- note: brute-force semi-decision procedure for FO only works theoretically

instead, we use an **SMT solver** (Z3) and the following reduction:

$$\Gamma \implies \Delta \text{ is invalid iff } \bigwedge_{\psi \in \Gamma} \psi \wedge \bigwedge_{\psi' \in \Delta} \neg \psi' \text{ is satisfiable}$$

**Example:**

$$\begin{array}{l} \forall x \forall y P(g(x, y)) \implies \exists x P(x) \\ \text{1. introduce "next" topmost} \quad \forall x \forall y P(g(x, y)) \implies \exists x_1 \exists x_2 P(g(x_1, x_2)) \\ \text{symbol of a groundterm} \end{array}$$

# Main Idea behind the Algorithm

**Problem:** Instantiation of existentially quantified variables

- note: brute-force semi-decision procedure for FO only works theoretically

instead, we use an **SMT solver** (Z3) and the following reduction:

$$\Gamma \implies \Delta \text{ is invalid iff } \bigwedge_{\psi \in \Gamma} \psi \wedge \bigwedge_{\psi' \in \Delta} \neg \psi' \text{ is satisfiable}$$

**Example:**

1. introduce “next” topmost symbol of a groundterm  

$$\forall x \forall y P(g(x, y)) \implies \exists x P(x)$$

$$\forall x \forall y P(g(x, y)) \implies \exists x_1 \exists x_2 P(g(x_1, x_2))$$
2. use **SMT solver to verify**  

$$\forall x \forall y P(g(x, y)) \wedge \neg \exists x_1 \exists x_2 P(g(x_1, x_2))$$

is **unsatisfiable**

## Further Techniques

---

- ▶ decision procedure for quantifier-free FO with equality based on results from term rewriting of equality groundterms
- ▶ further goal: short (human readable) proofs tackled by ordering rule applications and term instantiations
  - ▶ trying to introduce more “complicated” rules and terms last
  - ▶ for instance: 1-ary terms before 2-ary terms
- ▶ results heavily rely on used SMT solver
  - ▶ is governed by a timeout (indicated by yellow sequent)
  - ▶ good results on typical “didactic” examples
  - ▶ problems (Z3) with high nesting depth of terms



## Further Techniques

---

- ▶ decision procedure for quantifier-free FO with equality based on results from term rewriting of equality groundterms
- ▶ further goal: short (human readable) proofs tackled by ordering rule applications and term instantiations
  - ▶ trying to introduce more “complicated” rules and terms last
  - ▶ for instance: 1-ary terms before 2-ary terms
- ▶ results heavily rely on used SMT solver
  - ▶ is governed by a timeout (indicated by yellow sequent)
  - ▶ good results on typical “didactic” examples
  - ▶ problems (Z3) with high nesting depth of terms

## Further Techniques

---

- ▶ decision procedure for quantifier-free FO with equality based on results from term rewriting of equality groundterms
- ▶ further goal: short (human readable) proofs tackled by ordering rule applications and term instantiations
  - ▶ trying to introduce more “complicated” rules and terms last
  - ▶ for instance: 1-ary terms before 2-ary terms
- ▶ results heavily rely on used SMT solver
  - ▶ is governed by a timeout (indicated by yellow sequent)
  - ▶ good results on typical “didactic” examples
  - ▶ problems (Z3) with high nesting depth of terms

## Further Techniques

---

- ▶ decision procedure for quantifier-free FO with equality based on results from term rewriting of equality groundterms
- ▶ further goal: short (human readable) proofs tackled by ordering rule applications and term instantiations
  - ▶ trying to introduce more “complicated” rules and terms last
  - ▶ for instance: 1-ary terms before 2-ary terms
- ▶ results heavily rely on used SMT solver
  - ▶ is governed by a timeout (indicated by yellow sequent)
  - ▶ good results on typical “didactic” examples
  - ▶ problems (Z3) with high nesting depth of terms

## Further Techniques

---

- ▶ decision procedure for quantifier-free FO with equality based on results from term rewriting of equality groundterms
- ▶ further goal: short (human readable) proofs tackled by ordering rule applications and term instantiations
  - ▶ trying to introduce more “complicated” rules and terms last
  - ▶ for instance: 1-ary terms before 2-ary terms
- ▶ results heavily rely on used SMT solver
  - ▶ is governed by a timeout (indicated by yellow sequent)
  - ▶ good results on typical “didactic” examples
  - ▶ problems (Z3) with high nesting depth of terms

## Further Techniques

---

- ▶ decision procedure for quantifier-free FO with equality based on results from term rewriting of equality groundterms
- ▶ further goal: short (human readable) proofs tackled by ordering rule applications and term instantiations
  - ▶ trying to introduce more “complicated” rules and terms last
  - ▶ for instance: 1-ary terms before 2-ary terms
- ▶ results heavily rely on used SMT solver
  - ▶ is governed by a timeout (indicated by yellow sequent)
  - ▶ good results on typical “didactic” examples
  - ▶ problems (Z3) with high nesting depth of terms

## Further Techniques

---

- ▶ decision procedure for quantifier-free FO with equality based on results from term rewriting of equality groundterms
- ▶ further goal: short (human readable) proofs tackled by ordering rule applications and term instantiations
  - ▶ trying to introduce more “complicated” rules and terms last
  - ▶ for instance: 1-ary terms before 2-ary terms
- ▶ results heavily rely on used SMT solver
  - ▶ is governed by a timeout (indicated by yellow sequent)
  - ▶ good results on typical “didactic” examples
  - ▶ problems (Z3) with high nesting depth of terms

## Further Techniques

---

- ▶ decision procedure for quantifier-free FO with equality based on results from term rewriting of equality groundterms
- ▶ further goal: short (human readable) proofs tackled by ordering rule applications and term instantiations
  - ▶ trying to introduce more “complicated” rules and terms last
  - ▶ for instance: 1-ary terms before 2-ary terms
- ▶ results heavily rely on used SMT solver
  - ▶ is governed by a timeout (indicated by yellow sequent)
  - ▶ good results on typical “didactic” examples
  - ▶ problems (Z3) with high nesting depth of terms

# Conclusion

---

- ▶ introduced a (fairly simple) traffic light system to the Sequent Calculus Trainer
- ▶ trigger a thought process, which leads to “semantical” understanding
- ▶ unfortunately, no empirical data yet
- ▶ try it: <http://www.uni-kassel.de/eecs/fachgebiete/fmv/projects/sequent-calculus-trainer.html>

Future Work: Combining a model-checking tool with the Sequent Calculus Trainer, thus, enabling students to input counter models.



# Conclusion

---

- ▶ introduced a (fairly simple) traffic light system to the Sequent Calculus Trainer
- ▶ trigger a thought process, which leads to “semantical” understanding
- ▶ unfortunately, no empirical data yet
- ▶ try it: <http://www.uni-kassel.de/eecs/fachgebiete/fmv/projects/sequent-calculus-trainer.html>

**Future Work:** Combining a **model-checking tool** with the Sequent Calculus Trainer, thus, enabling students to input **counter models**.

Thank you!