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

$\curvearrowright$ tools to assist and self-assess certain topics

The Sequent Calculus

- proof calculus similar to natural deduction
- sequent: $\Gamma \Longrightarrow \Delta$
- rule: e.g. $\dfrac{\Gamma \Longrightarrow \varphi, \Delta \qquad \Gamma \Longrightarrow \psi, \Delta}{\Gamma \Longrightarrow \varphi \wedge \psi, \Delta}$

proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \Rightarrow A, B}\,(\text{Ax}) \quad \dfrac{\dfrac{}{B, C \Rightarrow A, B}\,(\text{Ax})}{B{\wedge}C \Rightarrow A, B}\,(\wedge_L)}{A{\vee}(B{\wedge}C) \Rightarrow A, B}\,(\vee_L)}{A{\vee}(B{\wedge}C) \Rightarrow A{\vee}B}\,(\vee_R) \quad \dfrac{\dfrac{\dfrac{}{A \Rightarrow A, C}\,(\text{Ax}) \quad \dfrac{\dfrac{}{B, C \Rightarrow A, C}\,(\text{Ax})}{B{\wedge}C \Rightarrow A, C}\,(\wedge_L)}{A{\vee}(B{\wedge}C) \Rightarrow A, C}\,(\vee_L)}{A{\vee}(B{\wedge}C) \Rightarrow A{\vee}C}\,(\vee_R)}{A{\vee}(B{\wedge}C) \Rightarrow (A{\vee}B){\wedge}(A{\vee}C)}\,(\wedge_R)$$

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

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

Steps in Learning a Proof Calculus

We distinguish between:

constructing a correct proof vs. finding the right proof

constructing a correct proof

$$\dfrac{}{\forall x \forall y.\, E(x,y) \rightarrow x = f(y) \Longrightarrow \forall x \forall y \forall z.\, E(x,z) \wedge E(y,z) \rightarrow x = y}\ 3 \times (\forall_{\mathrm{R}})$$

Steps in Learning a Proof Calculus

We distinguish between:

constructing a correct proof vs. finding the right proof

constructing a correct proof

$$\dfrac{\overline{\forall x \forall y.\ E(x,y) \to x = f(y) \Longrightarrow E(a,c) \land E(b,c) \to a = b}}{\forall x \forall y.\ E(x,y) \to x = f(y) \Longrightarrow \forall x \forall y \forall z.\ E(x,z) \land E(y,z) \to x = y}\ 3 \times (\forall_{\mathtt{R}})$$

Steps in Learning a Proof Calculus

We distinguish between:

constructing a correct proof vs. finding the right proof

constructing a correct proof

$$\dfrac{\dfrac{}{\forall x \forall y.\ E(x,y) \rightarrow x = f(y) \Longrightarrow E(a,c) \land E(b,c) \rightarrow a = b}\ (\rightarrow_{\mathtt{R}})}{\forall x \forall y.\ E(x,y) \rightarrow x = f(y) \Longrightarrow \forall x \forall y \forall z.\ E(x,z) \land E(y,z) \rightarrow x = y}\ 3 \times (\forall_{\mathtt{R}})$$

Steps in Learning a Proof Calculus

We distinguish between:

constructing a correct proof vs. finding the right proof

constructing a correct proof

$$\dfrac{\dfrac{\overline{\forall x \forall y.\ E(x,y) \rightarrow x = f(y), E(a,c) \land E(b,c) \Longrightarrow a = b}}{\forall x \forall y.\ E(x,y) \rightarrow x = f(y) \Longrightarrow E(a,c) \land E(b,c) \rightarrow a = b}\ (\rightarrow_{\mathtt{R}})}{\forall x \forall y.\ E(x,y) \rightarrow x = f(y) \Longrightarrow \forall x \forall y \forall z.\ E(x,z) \land E(y,z) \rightarrow x = y}\ 3 \times (\forall_{\mathtt{R}})$$

Steps in Learning a Proof Calculus

We distinguish between:

constructing a correct proof vs. finding the right proof

constructing a correct proof

$$\dfrac{\dfrac{\dfrac{}{\forall x \forall y.\ E(x,y) \to x = f(y), E(a,c) \land E(b,c) \Longrightarrow a = b}(\land_L)}{\forall x \forall y.\ E(x,y) \to x = f(y) \Longrightarrow E(a,c) \land E(b,c) \to a = b}(\to_R)}{\forall x \forall y.\ E(x,y) \to x = f(y) \Longrightarrow \forall x \forall y \forall z.\ E(x,z) \land E(y,z) \to x = y}\ 3 \times (\forall_R)$$

Steps in Learning a Proof Calculus

We distinguish between:

constructing a correct proof vs. finding the right proof

constructing a correct proof

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\forall x \forall y.\ E(x,y) \to x = f(y), E(a,c), E(b,c) \Longrightarrow a = b}}{\forall x \forall y.\ E(x,y) \to x = f(y), E(a,c) \land E(b,c) \Longrightarrow a = b}(\land_{\mathtt{L}})}{\forall x \forall y.\ E(x,y) \to x = f(y) \Longrightarrow E(a,c) \land E(b,c) \to a = b}(\to_{\mathtt{R}})}{\forall x \forall y.\ E(x,y) \to x = f(y) \Longrightarrow \forall x \forall y \forall z.\ E(x,z) \land E(y,z) \to x = y}\, 3 \times (\forall_{\mathtt{R}})$$

Steps in Learning a Proof Calculus

We distinguish between:

constructing a correct proof vs. finding the right proof

constructing a correct proof

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\forall x \forall y.\ E(x,y) \to x = f(y), E(a,c), E(b,c) \Longrightarrow a = b}}{\forall x \forall y.\ E(x,y) \to x = f(y), E(a,c) \wedge E(b,c) \Longrightarrow a = b}\,(\wedge_{\mathtt{L}})}{\forall x \forall y.\ E(x,y) \to x = f(y) \Longrightarrow E(a,c) \wedge E(b,c) \to a = b}\,(\to_{\mathtt{R}})}{\forall x \forall y.\ E(x,y) \to x = f(y) \Longrightarrow \forall x \forall y \forall z.\ E(x,z) \wedge E(y,z) \to x = y}\,3 \times (\forall_{\mathtt{R}})$$

- 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:

$$\dfrac{}{\forall x \forall y.\ E(x,y) \to x = f(y), E(a,c), E(b,c) \Longrightarrow a = b}\ 2 \times (\forall_{\mathrm{L}})$$

A Tool for "Finding the Right Proof"

Recall:

$$\frac{E(b,c) \rightarrow b = f(c), E(a,c), E(b,c) \Longrightarrow a = b}{\forall x \forall y.\ E(x,y) \rightarrow x = f(y), E(a,c), E(b,c) \Longrightarrow a = b}\ 2 \times (\forall_{\mathrm{L}})$$

A Tool for "Finding the Right Proof"

Recall:

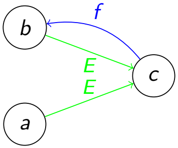
$$\dfrac{E(b,c) \to b = f(c), E(a,c), E(b,c) \Longrightarrow a = b}{\forall x \forall y.\ E(x,y) \to x = f(y), E(a,c), E(b,c) \Longrightarrow a = b}\ 2 \times (\forall_{\mathrm{L}})$$

A Tool for "Finding the Right Proof"

Recall:

$$\frac{E(b,c) \to b = f(c), E(a,c), E(b,c) \Longrightarrow a = b}{\forall x \forall y.\ E(x,y) \to x = f(y), E(a,c), E(b,c) \Longrightarrow a = b}\ 2 \times (\forall_{\mathrm{L}})$$

counter model:

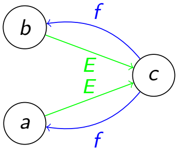


A Tool for “Finding the Right Proof”

Recall:

$$\dfrac{E(b,c) \to b = f(c), E(a,c), E(b,c) \Longrightarrow a = b}{\forall x \forall y.\ E(x,y) \to x = f(y), E(a,c), E(b,c) \Longrightarrow a = b}\ 2 \times (\forall_{\mathrm{L}})$$

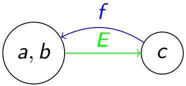
counter model:



A Tool for “Finding the Right Proof”

Recall:

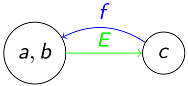
$$\dfrac{E(b,c) \rightarrow b = f(c), E(a,c), E(b,c) \Longrightarrow a = b}{\forall x \forall y.\ E(x,y) \rightarrow x = f(y), E(a,c), E(b,c) \Longrightarrow a = b}\ 2 \times (\forall_{\mathrm{L}})$$



A Tool for "Finding the Right Proof"

Recall:

$$\frac{E(b,c) \to b = f(c), E(a,c), E(b,c) \Longrightarrow a = b}{\forall x \forall y.\ E(x,y) \to x = f(y), E(a,c), E(b,c) \Longrightarrow a = b}\ 2 \times (\forall_{\mathrm{L}})$$

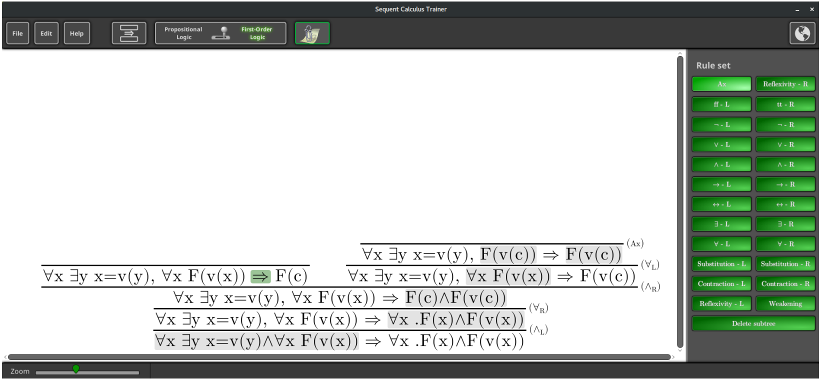


Specifications for Version 2

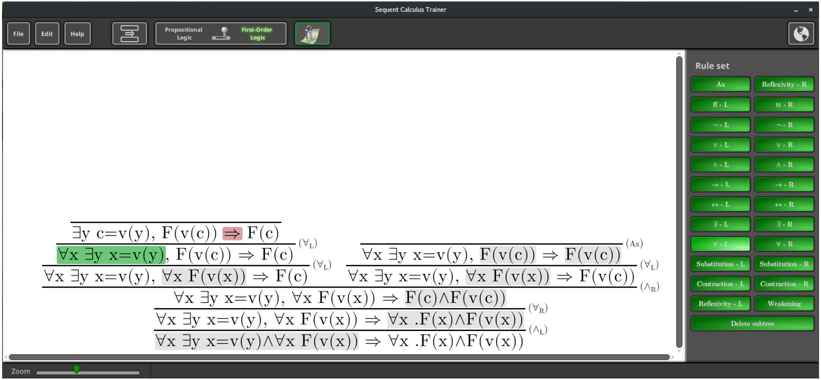
- stepping out of purely syntactical reasoning
- direct students' focus to the underlying structure

The Sequent Calculus Trainer with Automated Reasoning

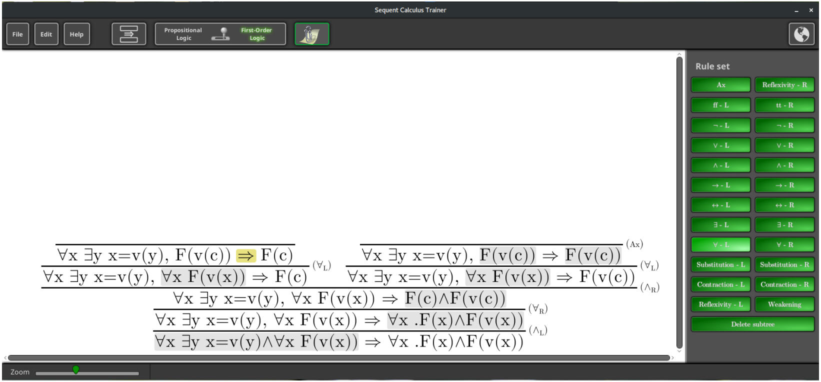
Main View


Sequent Calculus Trainer
File
Edit
Help
Propositional Logic
First-Order Logic
Rule set
Ax
Reflexivity - R
ff - L
tt - R
¬ - L
¬ - R
∨ - L
∨ - R
∧ - L
∧ - R
→ - L
→ - R
↔ - L
↔ - R
∃ - L
∃ - R
∀ - L
∀ - R
Substitution - L
Substitution - R
Contraction - L
Contraction - R
Reflexivity - L
Weakening
Delete subtree
∀x ∃y x=v(y), F(v(c)) ⇒ F(v(c)) (Ax)
∀x ∃y x=v(y), ∀x F(v(x)) ⇒ F(c)
∀x ∃y x=v(y), ∀x F(v(x)) ⇒ F(v(c)) (∀L)
∀x ∃y x=v(y), ∀x F(v(x)) ⇒ F(c)∧F(v(c)) (∧R)
∀x ∃y x=v(y), ∀x F(v(x)) ⇒ ∀x .F(x)∧F(v(x)) (∀R)
∀x ∃y x=v(y)∧∀x F(v(x)) ⇒ ∀x .F(x)∧F(v(x)) (∧L)
Zoom

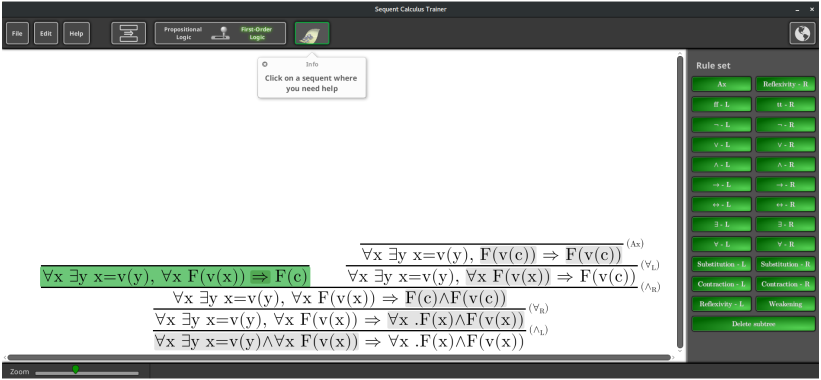
Advice via Traffic Light System



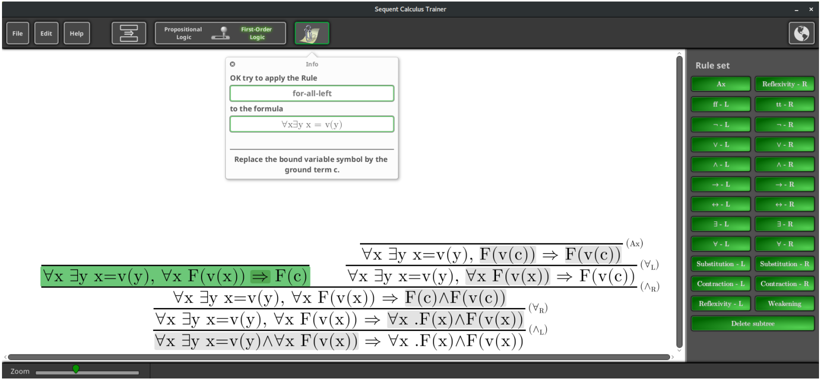
Advice via Traffic Light System



Dialog System



Dialog System



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:

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

Example:

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

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:

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

Example:

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

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:

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

Example:

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

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:

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

Example:

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

1. introduce "next" topmost symbol of a groundterm

$$\forall x \forall y\ P(g(x,y)) \Longrightarrow P(c)$$

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:

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

Example:

	$\forall x \forall y\ P(g(x,y)) \Longrightarrow \exists x\ P(x)$
1. introduce "next" topmost symbol of a groundterm	$\forall x \forall y\ P(g(x,y)) \Longrightarrow P(c)$
2. use SMT solver to verify	$\forall x \forall y\ P(g(x,y)) \wedge \neg P(c)$ is satisfiable

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:

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

Example:

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

1. introduce "next" topmost symbol of a groundterm

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

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:

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

Example:

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

1. introduce "next" topmost symbol of a groundterm: $\forall x \forall y\ P(g(x,y)) \Longrightarrow \exists x_1 \exists x_2 P(g(x_1, x_2))$
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!