Generation of Proof Obligations to Ensure the Soundness of Taclets

Philipp Rümmer

ph_r@gmx.net

University of Karlsruhe

Institute for Logic, Complexity and Deduction Systems

D-76128 Karlsruhe, Germany

Overview

- Introduction to taclets

Overview

- Introduction to taclets
- Concept to ensure the soundness of taclets

Overview

- Introduction to taclets
- Concept to ensure the soundness of taclets
- Examples

Introduction to Taclets

- Method to define rules of sequent calculi

Introduction to Taclets

- Method to define rules of sequent calculi
- Introduced by Elmar Habermalz in: *Ein dynamisches automatisierbares interaktives Kalkül für schematische theoriespezifische Regeln* (PhD thesis)

Introduction to Taclets

- Method to define rules of sequent calculi
- Introduced by Elmar Habermalz in: *Ein dynamisches automatisierbares interaktives Kalkül für schematische theoriespezifische Regeln* (PhD thesis)
- In KeY: Majority of rules defined through taclets

Introduction to Taclets (2)

- Taclet in concrete syntax:

$$\texttt{if}\ (\mathit{ifseq})\ \texttt{find}\ (f)$$

$$\texttt{replacewith}\ (rw_1)\ \texttt{add}\ (add_1);$$

$$\vdots$$

$$\texttt{replacewith}\ (rw_k)\ \texttt{add}\ (add_k)$$

Introduction to Taclets (2)

- Taclet in concrete syntax:

$$\texttt{if}\ (\mathit{ifseq})\ \texttt{find}\ (f)$$
$$\texttt{replacewith}\ (rw_1)\ \texttt{add}\ (add_1);$$
$$\vdots$$
$$\texttt{replacewith}\ (rw_k)\ \texttt{add}\ (add_k)$$

- For instance: Modus ponens

$$\texttt{if}\ (\phi \vdash\ )\ \texttt{find}\ (\ \vdash\ \phi \rightarrow \psi)$$
$$\texttt{replacewith}\ (\ \vdash\ \psi)$$

Schema Variables

- For first-order logic:

Schema Variables

- For first-order logic:
  - Terms

Schema Variables

- For first-order logic:
  - Terms
  - Formulas

Schema Variables

- For first-order logic:
  - Terms
  - Formulas
  - Object variables

Schema Variables

- For first-order logic:
  - Terms
  - Formulas
  - Object variables
- For JavaCardDL in addition:

Schema Variables

- For first-order logic:
  - Terms
  - Formulas
  - Object variables
- For JavaCardDL in addition:
  - Program variables

Schema Variables

- For first-order logic:
  - Terms
  - Formulas
  - Object variables
- For JavaCardDL in addition:
  - Program variables
  - Java statements

Schema Variables

- For first-order logic:
  - Terms
  - Formulas
  - Object variables
- For JavaCardDL in addition:
  - Program variables
  - Java statements
  - Java expressions

Soundness of Calculi

- Sequent $\Gamma \vdash \Delta$ is called valid iff

$$\bigwedge \Gamma \rightarrow \bigvee \Delta$$

is valid

Soundness of Calculi

- Sequent $\Gamma \ \vdash \ \Delta$ is called valid iff

$$\bigwedge \Gamma \rightarrow \bigvee \Delta$$

  is valid
- A calculus is sound iff only valid sequents can be derived

Soundness of Rules

- Sufficient criterion for soundness of calculus:
  Rule applications preserve validity

Soundness of Rules

- Sufficient criterion for soundness of calculus: Rule applications preserve validity
-
$$\frac{P_1 \quad P_2 \quad \cdots \quad P_k}{Q}$$
$$P_1, P_2, \ldots, P_k \text{ valid} \Longrightarrow Q \text{ valid}$$

Concept to Ensure Soundness of Taclets

Method introduced by Elmar Habermalz:

Concept to Ensure Soundness of Taclets

Method introduced by Elmar Habermalz:

Taclet t

Concept to Ensure Soundness of Taclets

Method introduced by Elmar Habermalz:

Taclet t

$\downarrow$

Meaning formula $M(t)$

Concept to Ensure Soundness of Taclets

Method introduced by Elmar Habermalz:

$$\text{Taclet } t$$
$$\downarrow$$
$$\text{Meaning formula } M(t)$$
$$\downarrow$$
$$\text{Proof obligation } M_{\mathsf{Sk}}(t)$$

Concept to Ensure Soundness of Taclets

Method introduced by Elmar Habermalz:

$$\text{Taclet } t$$
$$\downarrow$$
$$\text{Meaning formula } M(t)$$
$$\downarrow$$
$$\text{Proof obligation } M_{\mathsf{Sk}}(t)$$

$$M_{\mathsf{Sk}}(t) \text{ valid} \iff t \text{ sound}$$

Example of Meaning Formula

- Taclet t_1 exchanging quantifiers:

$$\texttt{find} \ (\vdash \forall \boldsymbol{y}.\forall \boldsymbol{x}.\boldsymbol{\phi})$$

$$\texttt{replacewith} \ (\vdash \forall \boldsymbol{x}.\forall \boldsymbol{y}.\boldsymbol{\phi})$$

Example of Meaning Formula

- Taclet t_1 exchanging quantifiers:

  $\texttt{find} \ (\ \vdash\ \forall \boldsymbol{y}.\forall \boldsymbol{x}.\boldsymbol{\phi})$

  $\texttt{replacewith} \ (\ \vdash\ \forall \boldsymbol{x}.\forall \boldsymbol{y}.\boldsymbol{\phi})$

- $M(t_1) = \forall \boldsymbol{x}.\forall \boldsymbol{y}.\boldsymbol{\phi} \rightarrow \forall \boldsymbol{y}.\forall \boldsymbol{x}.\boldsymbol{\phi}$

Example of Meaning Formula (2)

- Taclet t_2 splitting an if-statement:

$$\texttt{find}(\langle \mathsf{l} : \ \textbf{if} \ (\mathsf{x{=}{=}0}) \ \#\mathsf{s} \ \textbf{else} \ \#\mathsf{t}\rangle\phi)$$

$$\texttt{replacewith}(\langle \mathsf{l} : \ \#\mathsf{s} \ \rangle\phi) \quad \texttt{add}(\mathsf{x} \doteq 0 \ \vdash \ );$$

$$\texttt{replacewith}(\langle \mathsf{l} : \ \#\mathsf{t} \ \rangle\phi) \quad \texttt{add}(\ \vdash \ \mathsf{x} \doteq 0)$$

Example of Meaning Formula (2)

$\texttt{find}(\langle \mathsf{l} : \textbf{if}\ (\mathsf{x{=}{=}0})\ \#\mathsf{s}\ \textbf{else}\ \#\mathsf{t} \rangle \phi)$

$\texttt{replacewith}(\langle \mathsf{l} : \#\mathsf{s} \rangle \phi) \quad \texttt{add}(\mathsf{x} \doteq 0 \vdash\ );$

$\texttt{replacewith}(\langle \mathsf{l} : \#\mathsf{t} \rangle \phi) \quad \texttt{add}(\ \vdash \mathsf{x} \doteq 0)$

Example of Meaning Formula (2)

$\texttt{find}(\langle \mathsf{l} : \textbf{if } (\mathsf{x{=}{=}0})\ \#\mathsf{s}\ \textbf{else}\ \#\mathsf{t}\rangle\phi)$

$\quad\texttt{replacewith}(\langle \mathsf{l} : \#\mathsf{s}\ \rangle\phi)\quad \texttt{add}(\mathsf{x} \doteq 0 \vdash\ );$

$\quad\texttt{replacewith}(\langle \mathsf{l} : \#\mathsf{t}\ \rangle\phi)\quad \texttt{add}(\ \vdash\ \mathsf{x} \doteq 0)$

$$
\begin{aligned}
M(t_2) = {} & \\
& \big(\mathsf{x} \doteq 0 \land \\
& \quad (\langle \mathsf{l} : \textbf{if } (\mathsf{x{=}{=}0})\ \#\mathsf{s}\ \textbf{else}\ \#\mathsf{t}\rangle\phi \leftrightarrow \langle \mathsf{l} : \#\mathsf{s}\ \rangle\phi)\big) \\
\lor & \big(\neg(\mathsf{x} \doteq 0) \land \\
& \quad (\langle \mathsf{l} : \textbf{if } (\mathsf{x{=}{=}0})\ \#\mathsf{s}\ \textbf{else}\ \#\mathsf{t}\rangle\phi \leftrightarrow \langle \mathsf{l} : \#\mathsf{t}\ \rangle\phi)\big)
\end{aligned}
$$

Taclet Proof Obligations

- Schema variables of meaning formulas are replaced with skolem symbols

Taclet Proof Obligations

- Schema variables of meaning formulas are replaced with skolem symbols

  SVs for terms $\rightarrow$ function symbols

Taclet Proof Obligations

- Schema variables of meaning formulas are replaced with skolem symbols

SVs for terms	$\longrightarrow$	function symbols
SVs for formulas	$\longrightarrow$	predicate symbols

Taclet Proof Obligations

- Schema variables of meaning formulas are replaced with skolem symbols

SVs for terms	$\longrightarrow$	function symbols
SVs for formulas	$\longrightarrow$	predicate symbols
SVs for logical variables	$\longrightarrow$	logical variables

Taclet Proof Obligations

- Schema variables of meaning formulas are replaced with skolem symbols

SVs for terms	$\longrightarrow$	function symbols
SVs for formulas	$\longrightarrow$	predicate symbols
SVs for logical variables	$\longrightarrow$	logical variables
SVs for program var.	$\longrightarrow$	program variables

Taclet Proof Obligations

- Schema variables of meaning formulas are replaced with skolem symbols

SVs for terms	$\longrightarrow$	function symbols
SVs for formulas	$\longrightarrow$	predicate symbols
SVs for logical variables	$\longrightarrow$	logical variables
SVs for program var.	$\longrightarrow$	program variables
SVs for statements	$\longrightarrow$	atomic programs

Taclet Proof Obligations

- Schema variables of meaning formulas are replaced with skolem symbols

SVs for terms	$\longrightarrow$	function symbols
SVs for formulas	$\longrightarrow$	predicate symbols
SVs for logical variables	$\longrightarrow$	logical variables
SVs for program var.	$\longrightarrow$	program variables
SVs for statements	$\longrightarrow$	atomic programs
SVs for expressions	$\longrightarrow$	“atomic expr.”

Example of Proof Obligation

- Meaning formula of t_1 is

$$M(t_1) = \forall \boldsymbol{x}.\forall \boldsymbol{y}.\boldsymbol{\phi} \rightarrow \forall \boldsymbol{y}.\forall \boldsymbol{x}.\boldsymbol{\phi}$$

Example of Proof Obligation

- Meaning formula of t_1 is

$$M(t_1) = \forall \boldsymbol{x}.\forall \boldsymbol{y}.\phi \rightarrow \forall \boldsymbol{y}.\forall \boldsymbol{x}.\phi$$

- Taclet proof obligation:

$$M_{\mathsf{Sk}}(t_1) = \forall u.\forall v.p_{\mathsf{Sk}}(u, v) \rightarrow \forall v.\forall u.p_{\mathsf{Sk}}(u, v)$$

Example of Proof Obligation (2)

- Meaning formula of t_2 is

$$\Big(\mathsf{x} \doteq 0 \wedge$$
$$\big(\langle \mathsf{l}:\ \textbf{if}\ \ (\mathsf{x}{=}{=}0)\ \#\mathsf{s}\ \textbf{else}\ \#\mathsf{t}\rangle\phi \leftrightarrow \langle \mathsf{l}:\ \#\mathsf{s}\ \rangle\phi\big)\Big)$$
$$\vee\Big(\neg(\mathsf{x} \doteq 0) \wedge$$
$$\big(\langle \mathsf{l}:\ \textbf{if}\ \ (\mathsf{x}{=}{=}0)\ \#\mathsf{s}\ \textbf{else}\ \#\mathsf{t}\rangle\phi \leftrightarrow \langle \mathsf{l}:\ \#\mathsf{t}\ \rangle\phi\big)\Big)$$

Example of Proof Obligation (2)

$$\big(\mathsf{x} \doteq 0 \wedge$$

$$(\langle \mathsf{l} : \ \textbf{if} \ (\mathsf{x}{=}{=}0) \ \#\mathsf{s} \ \textbf{else} \ \#\mathsf{t} \rangle \phi \leftrightarrow \langle \mathsf{l} : \ \#\mathsf{s} \ \rangle \phi)\big)$$

$$\vee \big(\neg(\mathsf{x} \doteq 0) \wedge$$

$$(\langle \mathsf{l} : \ \textbf{if} \ (\mathsf{x}{=}{=}0) \ \#\mathsf{s} \ \textbf{else} \ \#\mathsf{t} \rangle \phi \leftrightarrow \langle \mathsf{l} : \ \#\mathsf{t} \ \rangle \phi)\big)$$

Example of Proof Obligation (2)

$$\Big(\mathsf{x} \doteq 0 \land$$
$$\big(\langle \mathsf{l} : \ \textbf{if} \ (\mathsf{x}{=}{=}0) \ \#\mathsf{s} \ \textbf{else} \ \#\mathsf{t} \rangle \phi \leftrightarrow \langle \mathsf{l} : \ \#\mathsf{s} \ \rangle \phi \big)\Big)$$
$$\lor \Big(\neg(\mathsf{x} \doteq 0) \land$$
$$\big(\langle \mathsf{l} : \ \textbf{if} \ (\mathsf{x}{=}{=}0) \ \#\mathsf{s} \ \textbf{else} \ \#\mathsf{t} \rangle \phi \leftrightarrow \langle \mathsf{l} : \ \#\mathsf{t} \ \rangle \phi \big)\Big)$$

Proof obligation of t_2:

$$\Big(\mathsf{x} \doteq 0 \land$$
$$\big(\langle \mathsf{l} : \ \textbf{if} \ (\mathsf{x}{=}{=}0) \ \beta_1 \ \textbf{else} \ \beta_2 \rangle p_{\mathsf{Sk}}(\mathsf{x}) \leftrightarrow \langle \mathsf{l} : \ \beta_1 \ \rangle p_{\mathsf{Sk}}(\mathsf{x}) \big)\Big)$$
$$\lor \Big(\neg(\mathsf{x} \doteq 0) \land$$
$$\big(\langle \mathsf{l} : \ \textbf{if} \ (\mathsf{x}{=}{=}0) \ \beta_1 \ \textbf{else} \ \beta_2 \rangle p_{\mathsf{Sk}}(\mathsf{x}) \leftrightarrow \langle \mathsf{l} : \ \beta_2 \ \rangle p_{\mathsf{Sk}}(\mathsf{x}) \big)\Big)$$

Example of Proof Obligation (2)

Proof obligation of t_2:

$$\big(\mathsf{x} \doteq 0 \wedge$$

$$(\langle \mathsf{l} : \ \textbf{if} \ (\mathsf{x}{=}{=}0)\ \beta_1 \ \textbf{else} \ \beta_2 \rangle p_{\mathsf{Sk}}(\mathsf{x}) \leftrightarrow \langle \mathsf{l} : \ \beta_1 \ \rangle p_{\mathsf{Sk}}(\mathsf{x}))\big)$$

$$\vee \big(\neg(\mathsf{x} \doteq 0) \wedge$$

$$(\langle \mathsf{l} : \ \textbf{if} \ (\mathsf{x}{=}{=}0)\ \beta_1 \ \textbf{else} \ \beta_2 \rangle p_{\mathsf{Sk}}(\mathsf{x}) \leftrightarrow \langle \mathsf{l} : \ \beta_2 \ \rangle p_{\mathsf{Sk}}(\mathsf{x}))\big)$$

Example of Proof Obligation (2)

Proof obligation of t_2:

$$\big(\mathsf{x} \doteq 0 \wedge$$

$$(\langle \mathsf{l}: \ \textbf{if} \ (\mathsf{x}{==}0) \ \beta_1 \ \textbf{else} \ \beta_2 \rangle p_{\mathsf{Sk}}(\mathsf{x}) \leftrightarrow \langle \mathsf{l}: \ \beta_1 \ \rangle p_{\mathsf{Sk}}(\mathsf{x}))\big)$$

$$\vee \big(\neg(\mathsf{x} \doteq 0) \wedge$$

$$(\langle \mathsf{l}: \ \textbf{if} \ (\mathsf{x}{==}0) \ \beta_1 \ \textbf{else} \ \beta_2 \rangle p_{\mathsf{Sk}}(\mathsf{x}) \leftrightarrow \langle \mathsf{l}: \ \beta_2 \ \rangle p_{\mathsf{Sk}}(\mathsf{x}))\big)$$

$$\beta_1 = s_{\mathsf{Sk}}(\mathsf{x}, \mathsf{t}_{\#\mathsf{s}}, \mathsf{d}_{\#\mathsf{s}}; \textbf{break } \mathsf{l}; \textbf{throw } \mathsf{t}_{\#\mathsf{s}});$$

$$\beta_2 = t_{\mathsf{Sk}}(\mathsf{x}, \mathsf{t}_{\#\mathsf{t}}, \mathsf{d}_{\#\mathsf{t}}; \textbf{break } \mathsf{l}; \textbf{throw } \mathsf{t}_{\#\mathsf{t}});$$

Summary

- The presented approach . . .

Summary

- The presented approach . . .
  - treats first-order taclets completely

Summary

- The presented approach . . .
  - treats first-order taclets completely
  - handles the most important kinds of JavaCardDL schema variables

Summary

- The presented approach . . .
  - treats first-order taclets completely
  - handles the most important kinds of JavaCardDL schema variables
  - is fully implemented

Summary

- The presented approach . . .
  - treats first-order taclets completely
  - handles the most important kinds of JavaCardDL schema variables
  - is fully implemented
- Future work:

Summary

- The presented approach . . .
  - treats first-order taclets completely
  - handles the most important kinds of JavaCardDL schema variables
  - is fully implemented
- Future work:
  - Support more JavaCardDL schema variables

Summary

- The presented approach . . .
  - treats first-order taclets completely
  - handles the most important kinds of JavaCardDL schema variables
  - is fully implemented
- Future work:
  - Support more JavaCardDL schema variables
  - Consider some special characteristics of KeY, e.g. untyped schema variables