TECHNISCHE UNIVERSITÄT DRESDEN

Institute of Theoretical Computer Science Chair for Automata Theory

CONTEXT REASONING FOR ROLE-BASED MODELS

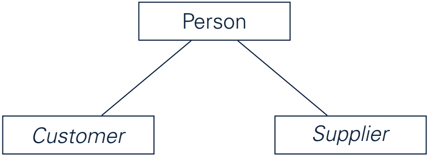
Stephan Böhme

Dresden, 27.10.2017

Motivation

What is a Role?

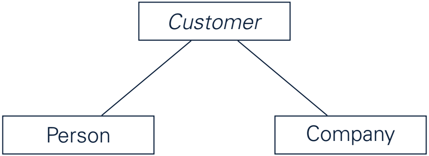
- Modelling concept from OOP introduced by Bachman in 1973
- Classification of roles with 26 *Features* including identity, behaviour, relationships, players, . . . and about Contexts and Constraints (Kühn, Leuthäuser, et al. 2014; Steimann 2000)



Motivation

What is a Role?

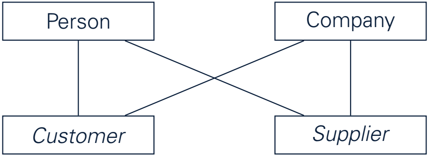
- Modelling concept from OOP introduced by Bachman in 1973
- Classification of roles with 26 *Features* including identity, behaviour, relationships, players, . . . and about Contexts and Constraints (Kühn, Leuthäuser, et al. 2014; Steimann 2000)



Motivation

What is a Role?

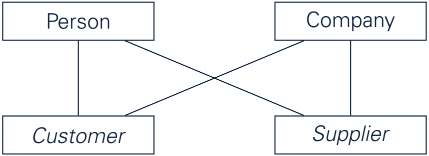
- Modelling concept from OOP introduced by Bachman in 1973
- Classification of roles with 26 *Features* including identity, behaviour, relationships, players, . . . and about Contexts and Constraints (Kühn, Leuthäuser, et al. 2014; Steimann 2000)



Motivation

What is a Role?

- Modelling concept from OOP introduced by Bachman in 1973
- Classification of roles with 26 *Features* including identity, behaviour, relationships, players, . . . and about Contexts and Constraints (Kühn, Leuthäuser, et al. 2014; Steimann 2000)



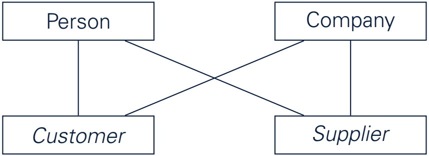
Requirements for modern Software Systems:

- Adaptability
- High expressiveness
- Longevity

Motivation

What is a Role?

- Modelling concept from OOP introduced by Bachman in 1973
- Classification of roles with 26 *Features* including identity, behaviour, relationships, players, . . . and about Contexts and Constraints (Kühn, Leuthäuser, et al. 2014; Steimann 2000)



Requirements for modern Software Systems:

- Adaptability → Roles allow for dynamic changes of the system.
- High expressiveness → Roles increase separation of concerns.
- Longevity → Roles enable updating running applications.

Role-Based Systems

- Software systems that use the notion of roles
- Focus on: Compartment Role Object Model (CROM), (Kühn, Leuthäuser, et al. 2014)
  - Well-defined semantical foundation (Kühn, Böhme, et al. 2015)

Role-Based Systems

- Software systems that use the notion of roles
- Focus on: Compartment Role Object Model (CROM), (Kühn, Leuthäuser, et al. 2014)
  - Well-defined semantical foundation (Kühn, Böhme, et al. 2015)

Key properties of roles:

- Roles depend on the context.
- Contexts, ‘players’ and roles themselves have each their own identity.
- Roles change over time.

Role-Based Systems

- Software systems that use the notion of roles
- Focus on: Compartment Role Object Model (CROM), (Kühn, Leuthäuser, et al. 2014)
  - Well-defined semantical foundation (Kühn, Böhme, et al. 2015)

Key properties of roles:

- Roles depend on the context.
- Contexts, 'players' and roles themselves have each their own identity.
- Roles change over time.

Problems:

- Large systems/models hard to comprehend
- Modelling errors stay undetected

Role-Based Systems

- Software systems that use the notion of roles
- Focus on: Compartment Role Object Model (CROM), (Kühn, Leuthäuser, et al. 2014)
  - Well-defined semantical foundation (Kühn, Böhme, et al. 2015)

Key properties of roles:

- Roles depend on the context.
- Contexts, 'players' and roles themselves have each their own identity.
- Roles change over time.

Problems:

- Large systems/models hard to comprehend
- Modelling errors stay undetected $\rightarrow$ Logic-based formalisms

Role-Based Systems

- Software systems that use the notion of roles
- Focus on: Compartment Role Object Model (CROM), (Kühn, Leuthäuser, et al. 2014)
  - Well-defined semantical foundation (Kühn, Böhme, et al. 2015)

Key properties of roles:

- Roles depend on the context.
- Contexts, 'players' and roles themselves have each their own identity.
- Roles change over time.

Problems:

- Large systems/models hard to comprehend
- Modelling errors stay undetected

↳ Logic-based formalisms

Requirements on logical formalism

- Decidable reasoning tasks
- Express contexts, 'players' and roles as formal objects
- Model contexts and ternary relation of role-playing
- Ability to handle *rigid*, i.e. context-independent, knowledge

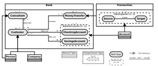
Workflow of Automated Analysis of Role-Based Models

Banking application

Workflow of Automated Analysis of Role-Based Models

Banking application

Graphical editor



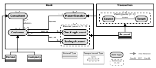
$$
\begin{aligned}
NT &:= \{\text{Person}, \text{Company}, \text{Account}\}\\
RT &:= \{\text{Customer}, \text{Consultant}, \text{CA}, \text{SA}, \text{Source}, \text{Target},\\
&\quad\ \text{MoneyTransfer}\}\\
CT &= \{\text{Bank}, \text{Transaction}\}\\
RST &= \{\text{own\_ca}, \text{own\_sa}, \text{advises}, \text{trans}\}\\
\mathit{fills} &:= \{(\text{Person}, \text{Consultant}), (\text{Person}, \text{Customer}),\\
&\quad\ (\text{Company}, \text{Customer}), (\text{Account}, \text{Source}),\\
&\quad\ (\text{Account}, \text{Target}), (\text{Account}, \text{CA}),\\
&\quad\ (\text{Account}, \text{SA}), (\text{Transaction}, \text{MoneyTransfer})\}\\
\mathit{parts} &:= \{\text{Bank} \rightarrow \{\text{Consultant}, \text{Customer}, \text{CA}, \text{SA},\\
&\quad\ \text{MoneyTransfer}\},\\
&\quad\ \text{Transaction} \rightarrow \{\text{Source}, \text{Target}\}\}\\
\mathit{rel} &:= \{\text{own\_ca} \rightarrow (\text{Customer}, \text{CA}),\\
&\quad\ \text{own\_sa} \rightarrow (\text{Customer}, \text{SA}),\\
&\quad\ \text{advises} \rightarrow (\text{Consultant}, \text{Customer}),\\
&\quad\ \text{trans} \rightarrow (\text{Source}, \text{Target})\}
\end{aligned}
$$

Graphical notation of a CROM and its formal representation

Workflow of Automated Analysis of Role-Based Models

Banking application

Graphical editor



$$
\begin{aligned}
NT &:= \{\text{Person}, \text{Company}, \text{Account}\}\\
RT &:= \{\text{Customer}, \text{Consultant}, \text{CA}, \text{SA}, \text{Source}, \text{Target},\\
&\qquad \text{MoneyTransfer}\}\\
CT &= \{\text{Bank}, \text{Transaction}\}\\
RST &= \{\text{own\_ca}, \text{own\_sa}, \text{advises}, \text{trans}\}\\
\mathit{fills} &:= \{(\text{Person}, \text{Consultant}), (\text{Person}, \text{Customer}),\\
&\qquad (\text{Company}, \text{Customer}), (\text{Account}, \text{Source}),\\
&\qquad (\text{Account}, \text{Target}), (\text{Account}, \text{CA}),\\
&\qquad (\text{Account}, \text{SA}), (\text{Transaction}, \text{MoneyTransfer})\}\\
\mathit{parts} &:= \{\text{Bank} \to \{\text{Consultant}, \text{Customer}, \text{CA}, \text{SA},\\
&\qquad \text{MoneyTransfer}\},\\
&\qquad \text{Transaction} \to \{\text{Source}, \text{Target}\}\}\\
\mathit{rel} &:= \{\text{own\_ca} \to (\text{Customer}, \text{CA}),\\
&\qquad \text{own\_sa} \to (\text{Customer}, \text{SA}),\\
&\qquad \text{advises} \to (\text{Consultant}, \text{Customer}),\\
&\qquad \text{trans} \to (\text{Source}, \text{Target})\}
\end{aligned}
$$

Graphical notation of a CROM and its formal representation

Automated mapping

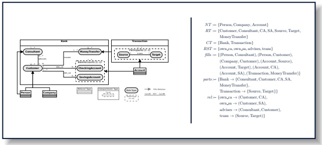
$$
\begin{aligned}
&\mathit{Bank} \sqsubseteq [\![(\geqslant_1 \mathit{counts}.\mathit{BankAccounts})(\mathit{counter})]\!]\\
&\quad \top \sqsubseteq [\![\exists \mathit{plays}.\mathit{BankAccounts} \equiv\\
&\qquad\qquad \geqslant_1 \mathit{plays}.(\mathit{CheckingAccount} \sqcup \mathit{SavingsAccount})\\
&\qquad\quad \sqcap \leqslant_1 \mathit{plays}.(\mathit{CheckingAccount} \sqcup \mathit{SavingsAccount})]\!]\\
&\quad \top \sqsubseteq [\![\mathit{SavingsAccount} \sqsubseteq \geqslant_1 \mathit{own\_sa}^-.\top]\!]\\
&\quad \top \sqsubseteq [\![\mathit{CheckingAccount} \sqsubseteq \geqslant_1 \mathit{own\_ca}^-.\top]\!]\\
&\quad \top \sqsubseteq [\![\exists \mathit{own\_sa}.\top \sqsubseteq \mathit{Customer}]\!]\\
&\quad \top \sqsubseteq [\![\exists \mathit{own\_ca}.\top \sqsubseteq \mathit{Customer}]\!]
\end{aligned}
$$

Contextualized DL ontology

Workflow of Automated Analysis of Role-Based Models

Banking application

Graphical editor



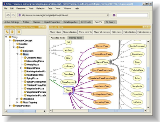
Graphical notation of a CROM and its formal representation

Automated mapping

$$
\begin{aligned}
&\mathit{Bank} \sqsubseteq \llbracket (\geqslant_1 \mathit{counts.BankAccounts})(\mathit{counter}) \rrbracket \\
&\quad \top \sqsubseteq \llbracket \exists \mathit{plays.BankAccounts} \equiv \\
&\qquad\qquad \geqslant_1 \mathit{plays.}(\mathit{CheckingAccount} \sqcup \mathit{SavingsAccount}) \\
&\qquad\quad \sqcap \leqslant_1 \mathit{plays.}(\mathit{CheckingAccount} \sqcup \mathit{SavingsAccount}) \rrbracket \\
&\quad \top \sqsubseteq \llbracket \mathit{SavingsAccount} \sqsubseteq \geqslant_1 \mathit{own\_sa}^-.\top \rrbracket \\
&\quad \top \sqsubseteq \llbracket \mathit{CheckingAccount} \sqsubseteq \geqslant_1 \mathit{own\_ca}^-.\top \rrbracket \\
&\quad \top \sqsubseteq \llbracket \exists \mathit{own\_sa}.\top \sqsubseteq \mathit{Customer} \rrbracket \\
&\quad \top \sqsubseteq \llbracket \exists \mathit{own\_ca}.\top \sqsubseteq \mathit{Customer} \rrbracket
\end{aligned}
$$

Contextualized DL ontology

Reasoning

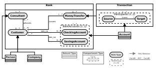


JConHT Reasoner

Workflow of Automated Analysis of Role-Based Models

Banking application

Graphical editor



$$NT := \{\mathrm{Person}, \mathrm{Company}, \mathrm{Account}\}$$
$$RT := \{\mathrm{Customer}, \mathrm{Consultant}, \mathrm{CA}, \mathrm{SA}, \mathrm{Source}, \mathrm{Target}, \mathrm{MoneyTransfer}\}$$
$$CT = \{\mathrm{Bank}, \mathrm{Transaction}\}$$
$$RST = \{\mathrm{own\_ca}, \mathrm{own\_sa}, \mathrm{advises}, \mathrm{trans}\}$$
$$fills := \{(\mathrm{Person}, \mathrm{Consultant}), (\mathrm{Person}, \mathrm{Customer}), (\mathrm{Company}, \mathrm{Customer}), (\mathrm{Account}, \mathrm{Source}), (\mathrm{Account}, \mathrm{Target}), (\mathrm{Account}, \mathrm{CA}), (\mathrm{Account}, \mathrm{SA}), (\mathrm{Transaction}, \mathrm{MoneyTransfer})\}$$
$$parts := \{\mathrm{Bank} \to \{\mathrm{Consultant}, \mathrm{Customer}, \mathrm{CA}, \mathrm{SA}, \mathrm{MoneyTransfer}\}, \mathrm{Transaction} \to \{\mathrm{Source}, \mathrm{Target}\}\}$$
$$rel := \{\mathrm{own\_ca} \to (\mathrm{Customer}, \mathrm{CA}), \mathrm{own\_sa} \to (\mathrm{Customer}, \mathrm{SA}), \mathrm{advises} \to (\mathrm{Consultant}, \mathrm{Customer}), \mathrm{trans} \to (\mathrm{Source}, \mathrm{Target})\}$$

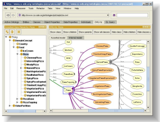
Graphical notation of a CROM and its formal representation

Automated mapping

$$
\begin{aligned}
&\mathit{Bank} \sqsubseteq [\![(\geqslant_1 \mathit{counts}.\mathit{BankAccounts})(\mathit{counter})]\!]\\
&\top \sqsubseteq [\![\exists \mathit{plays}.\mathit{BankAccounts} \equiv\\
&\qquad \geqslant_1 \mathit{plays}.(\mathit{CheckingAccount} \sqcup \mathit{SavingsAccount})\\
&\qquad \sqcap \leqslant_1 \mathit{plays}.(\mathit{CheckingAccount} \sqcup \mathit{SavingsAccount})]\!]\\
&\top \sqsubseteq [\![\mathit{SavingsAccount} \sqsubseteq \geqslant_1 \mathit{own\_sa}^-.\top]\!]\\
&\top \sqsubseteq [\![\mathit{CheckingAccount} \sqsubseteq \geqslant_1 \mathit{own\_ca}^-.\top]\!]\\
&\top \sqsubseteq [\![\exists \mathit{own\_sa}.\top \sqsubseteq \mathit{Customer}]\!]\\
&\top \sqsubseteq [\![\exists \mathit{own\_ca}.\top \sqsubseteq \mathit{Customer}]\!]
\end{aligned}
$$

Contextualized DL ontology

Reasoning



JConHT Reasoner

Inconsistencies?

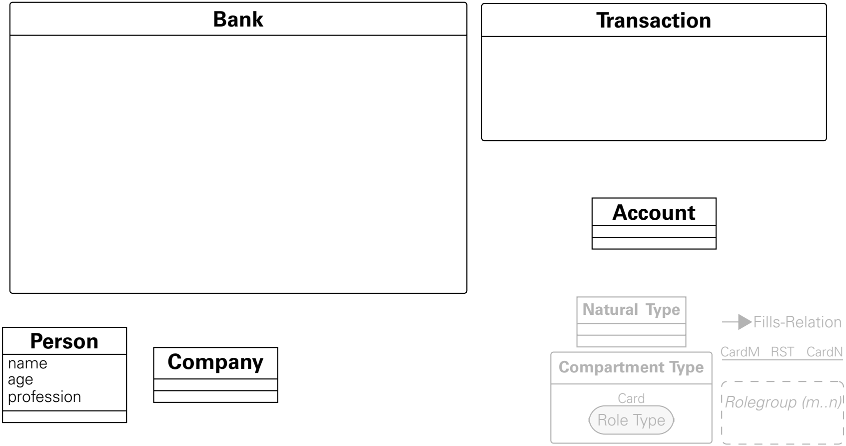
Can specific role be played?

Partial knowledge satisfiable?

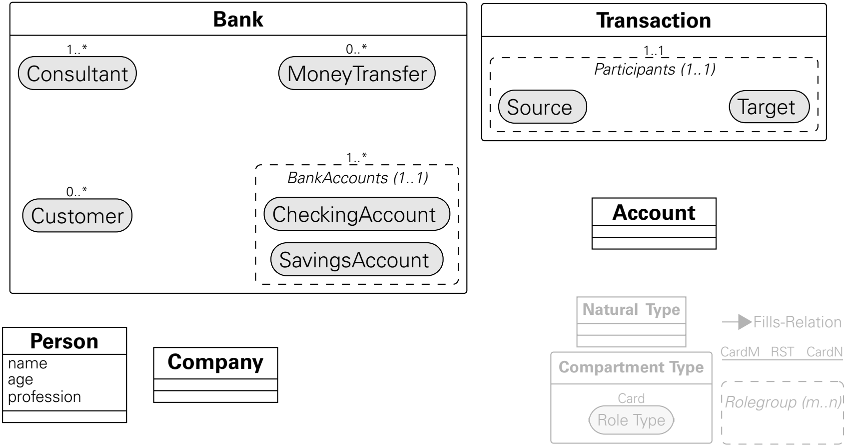
Are even stronger constraints entailed?

Running Example – Banking Application

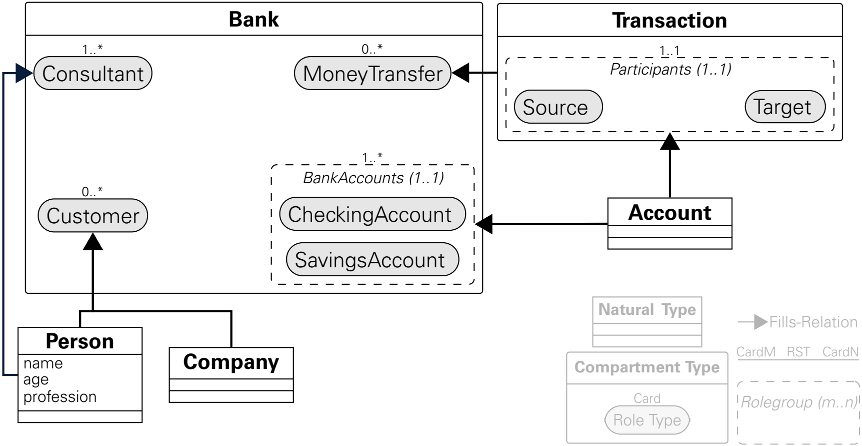
Running Example – Banking Application



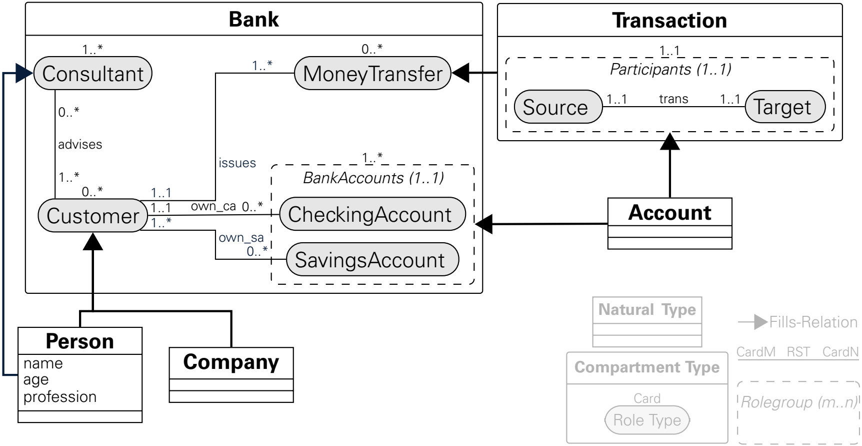
Running Example – Banking Application



Running Example – Banking Application



Running Example – Banking Application



Workflow of Automated Analysis of Role-Based Models

Banking application

Graphical editor

Graphical notation of a CROM and its formal representation

Automated mapping

$$
\begin{aligned}
&Bank \sqsubseteq [\![(\geqslant_1 counts.BankAccounts)(counter)]\!]\\
&\quad \top \sqsubseteq [\![\exists plays.BankAccounts \equiv\\
&\qquad\qquad \geqslant_1 plays.(CheckingAccount \sqcup SavingsAccount)\\
&\qquad\qquad \sqcap \leqslant_1 plays.(CheckingAccount \sqcup SavingsAccount)]\!]\\
&\quad \top \sqsubseteq \Big[\!\!\Big[SavingsAccount \sqsubseteq \geqslant_1 own\_sa^{-}.\top\Big]\!\!\Big]\\
&\quad \top \sqsubseteq \Big[\!\!\Big[CheckingAccount \sqsubseteq \geqslant_1 own\_ca^{-}.\top\Big]\!\!\Big]\\
&\quad \top \sqsubseteq [\![\exists own\_sa.\top \sqsubseteq Customer]\!]\\
&\quad \top \sqsubseteq [\![\exists own\_ca.\top \sqsubseteq Customer]\!]
\end{aligned}
$$

Contextualized DL ontology

Reasoning

JConHT Reasoner

Inconsistencies?

Can specific role be played?

Partial knowledge satisfiable?

Are even stronger constraints entailed?

Syntax and Semantics of the DL $\mathcal{ALC}$

Every consultant advises customers who own an checking account.
$\textsc{Consultant} \sqsubseteq \exists \mathit{advises}.(\textsc{Customer} \sqcap \exists \mathit{own\_ca}.\textsc{CheckingAccount})$

Peter is a consultant. $\textsc{Consultant}(\mathit{Peter})$

Syntax and Semantics of the DL $\mathcal{ALC}$

Every consultant advises customers who own an checking account.
$\textsc{Consultant} \sqsubseteq \exists \text{advises}.(\textsc{Customer} \sqcap \exists \text{own\_ca}.\textsc{CheckingAccount})$

Peter is a consultant. $\textsc{Consultant}(\mathit{Peter})$

N_C	... concept names	$\textsc{Consultant}$, $\textsc{Customer}$, $\textsc{CheckingAccount}$
N_R	... DL role names	advises, own_ca
N_I	... individual names	*Peter*

Syntax and Semantics of the DL $\mathcal{ALC}$

Every consultant advises customers who own an checking account.
$\textsc{Consultant} \sqsubseteq \exists\,\text{advises}.(\textsc{Customer} \sqcap \exists\,\text{own\_ca}.\textsc{CheckingAccount})$

Peter is a consultant. $\textsc{Consultant}(\mathit{Peter})$

$\mathsf{N_C}$ ... concept names	CONSULTANT, CUSTOMER, CHECKINGACCOUNT
$\mathsf{N_R}$ ... DL role names	advises, own_ca
$\mathsf{N_I}$ ... individual names	*Peter*

concept constructors:	$C_1 \sqcap C_2$, $C_1 \sqcup C_2$, $\neg C_1$, $\exists r.C$, $\forall r.C$
set of $\mathcal{ALC}$ concepts:	smallest set that is closed under $\mathsf{N_C}$ and the concept constructors of $\mathcal{ALC}$
General concept inclusion (GCI):	$C \sqsubseteq D$
assertion:	$C(a)$, $r(a, b)$
$\mathcal{ALC}$-axiom:	a GCI or an assertion

Syntax and Semantics of the DL $\mathcal{ALC}$

Every consultant advises customers who own an checking account.
$\textsc{Consultant} \sqsubseteq \exists \mathsf{advises}.(\textsc{Customer} \sqcap \exists \mathsf{own\_ca}.\textsc{CheckingAccount})$

Peter is a consultant. $\textsc{Consultant}(\mathit{Peter})$

A DL interpretation $\mathcal{I}$ has a domain $\Delta^\mathcal{I}$ and maps

- concept names A to sets $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$,
- DL role names r to binary relations $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$, and
- individual names a to elements $a^\mathcal{I} \in \Delta^\mathcal{I}$.

Syntax and Semantics of the DL $\mathcal{ALC}$

Every consultant advises customers who own an checking account.
$\text{CONSULTANT} \sqsubseteq \exists \text{advises}.(\text{CUSTOMER} \sqcap \exists \text{own\_ca}.\text{CHECKINGACCOUNT})$

Peter is a consultant. $\text{CONSULTANT}(\mathit{Peter})$

A DL interpretation $\mathcal{I}$ has a domain $\Delta^{\mathcal{I}}$ and maps

- concept names A to sets $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$,
- DL role names r to binary relations $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and
- individual names a to elements $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.

The semantics of the constructors is defined as

- $(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$,
- $(\neg C)^{\mathcal{I}} := \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$, and
- $(\exists r.C)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \exists e.(d, e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$

Syntax and Semantics of the DL $\mathcal{ALC}$

Every consultant advises customers who own an checking account.
$\textsc{Consultant} \sqsubseteq \exists \mathsf{advises}.(\textsc{Customer} \sqcap \exists \mathsf{own\_ca}.\textsc{CheckingAccount})$

Peter is a consultant. $\textsc{Consultant}(\mathit{Peter})$

A DL interpretation $\mathcal{I}$ has a domain $\Delta^{\mathcal{I}}$ and maps

- concept names A to sets $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$,
- DL role names r to binary relations $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and
- individual names a to elements $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.

The semantics of the constructors is defined as

- $(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$,
- $(\neg C)^{\mathcal{I}} := \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$, and
- $(\exists r.C)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \exists e.(d, e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$

Interpretation $\mathcal{I}$ is a model of

- the GCI $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, and
- the assertion $C(a)$ $(r(a, b))$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$ $((a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}})$.

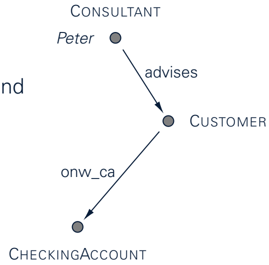
Syntax and Semantics of the DL $\mathcal{ALC}$

Every consultant advises customers who own an checking account.
$\textsc{Consultant} \sqsubseteq \exists \text{advises}.(\textsc{Customer} \sqcap \exists \text{own\_ca}.\textsc{CheckingAccount})$

Peter is a consultant. $\textsc{Consultant}(Peter)$

A DL interpretation $\mathcal{I}$ has a domain $\Delta^{\mathcal{I}}$ and maps

- concept names A to sets $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$,
- DL role names r to binary relations $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and
- individual names a to elements $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.



The semantics of the constructors is defined as

- $(C \sqcap D)^{\mathcal{I}} := C^{\mathcal{I}} \cap D^{\mathcal{I}}$,
- $(\neg C)^{\mathcal{I}} := \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$, and
- $(\exists r.C)^{\mathcal{I}} := \{d \in \Delta^{\mathcal{I}} \mid \exists e.(d, e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$

Interpretation $\mathcal{I}$ is a model of

- the GCI $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, and
- the assertion $C(a)$ ($r(a, b)$) iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$ ($(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$).

Intuition of Contextualized DL $\mathcal{L}_M [\![\mathcal{L}_O]\!]$

- Two-dimensional, two-sorted description logic
- $\mathcal{L}_M$ to describe knowledge *about* contexts (meta level)
- $\mathcal{L}_O$ to describe knowledge *within* contexts (object level)
- Contexts $\hat{=}$ possible worlds
- Concepts, axioms of object logic are usual $\mathcal{L}_O$ concepts, axioms
- Object axioms used as meta concepts
  - $\underbrace{[\![C \sqsubseteq D]\!]}_{\text{meta concept}}$ describes set of worlds where $\underbrace{C \sqsubseteq D}_{\text{object axiom}}$ holds

Intuition of Contextualized DL $\mathcal{L}_M [\![\mathcal{L}_O]\!]$

- Two-dimensional, two-sorted description logic
- $\mathcal{L}_M$ to describe knowledge *about* contexts (meta level)
- $\mathcal{L}_O$ to describe knowledge *within* contexts (object level)
- Contexts $\hat{=}$ possible worlds
- Concepts, axioms of object logic are usual $\mathcal{L}_O$ concepts, axioms
- Object axioms used as meta concepts
  - $\underbrace{[\![C \sqsubseteq D]\!]}_{\text{meta concept}}$ describes set of worlds where $\underbrace{C \sqsubseteq D}_{\text{object axiom}}$ holds

$$\textsc{Bank} \sqsubseteq [\![\textsc{SavingsAccount} \sqsubseteq \geqslant_1 \mathit{own\_sa}^- .\top]\!]$$

$$\textsc{Bank} \sqsubseteq [\![\exists\, \mathit{own\_sa}.\top \sqsubseteq \textsc{Customer}]\!]$$

$$\neg[\![(\textsc{Transaction}' \sqcap \exists\, \mathit{plays}.\top) \sqsubseteq \bot]\!] \sqsubseteq \exists\, \mathit{nested}.\textsc{Transaction}$$

Syntax and Semantics of $\mathcal{L}_M [\![\mathcal{L}_O]\!]$

Syntax and Semantics of $\mathcal{L}_M$ $[\![\mathcal{L}_O]\!]$

$$\mathsf{O} = (\mathsf{O}_\mathsf{C}, \mathsf{O}_\mathsf{R}, \mathsf{O}_\mathsf{I}) \qquad \mathsf{M} = (\mathsf{M}_\mathsf{C}, \mathsf{M}_\mathsf{R}, \mathsf{M}_\mathsf{I})$$

$$\mathsf{O}_{\mathsf{Crig}} \subseteq \mathsf{O}_\mathsf{C}$$

$$\mathsf{O}_{\mathsf{Rrig}} \subseteq \mathsf{O}_\mathsf{R}$$

object concept name $A \in \mathsf{O}_\mathsf{C}$

object concept C (using constructors of $\mathcal{L}_O$)

object axioms $C \sqsubseteq D$

$C(a)$

Syntax and Semantics of $\mathcal{L}_M \; [\![\mathcal{L}_O]\!]$

$$\mathsf{O} = (\mathsf{O_C}, \mathsf{O_R}, \mathsf{O_I}) \qquad \mathsf{M} = (\mathsf{M_C}, \mathsf{M_R}, \mathsf{M_I})$$

$$\mathsf{O_{Crig}} \subseteq \mathsf{O_C}$$

$$\mathsf{O_{Rrig}} \subseteq \mathsf{O_R}$$

object concept name $A \in \mathsf{O_C}$

object concept C (using constructors of $\mathcal{L}_O$)

meta concepts $[\![C \sqsubseteq D]\!]$, $[\![C(a)]\!]$

Syntax and Semantics of $\mathcal{L}_M$ $[\![\mathcal{L}_O]\!]$

$\mathsf{O} = (\mathsf{O}_\mathsf{C}, \mathsf{O}_\mathsf{R}, \mathsf{O}_\mathsf{I})$

$\mathsf{O}_{\mathsf{Crig}} \subseteq \mathsf{O}_\mathsf{C}$

$\mathsf{O}_{\mathsf{Rrig}} \subseteq \mathsf{O}_\mathsf{R}$

$\mathsf{M} = (\mathsf{M}_\mathsf{C}, \mathsf{M}_\mathsf{R}, \mathsf{M}_\mathsf{I})$

object concept name	$A \in \mathsf{O}_\mathsf{C}$
object concept	C (using constructors of $\mathcal{L}_O$)
meta concepts	$[\![C \sqsubseteq D]\!]$ $[\![C(a)]\!]$
meta concept name	$B \in \mathsf{M}_\mathsf{C}$
meta concept	E (using constructors of $\mathcal{L}_M$)
meta axioms	$E \sqsubseteq F$ $E(c)$

Syntax and Semantics of $\mathcal{L}_M [\![\mathcal{L}_O]\!]$

$$O = (O_C, O_R, O_I) \qquad\qquad M = (M_C, M_R, M_I)$$

$$O_{Crig} \subseteq O_C$$
$$O_{Rrig} \subseteq O_R$$

object concept name	$A \in O_C$
object concept	C (using constructors of $\mathcal{L}_O$)
meta concepts	$[\![C \sqsubseteq D]\!]$ $[\![C(a)]\!]$
meta concept name	$B \in M_C$
meta concept	E (using constructors of $\mathcal{L}_M$)
meta axioms	$E \sqsubseteq F$ $E(c)$

$\mathcal{L}_M [\![\mathcal{L}_O]\!]$ ontology $\mathcal{B}$ ...conjunction of m-axioms

Syntax and Semantics of $\mathcal{L}_M [\![\mathcal{L}_O]\!]$

Nested interpretation $\mathcal{J} = (\mathbb{C}, \cdot^{\mathcal{J}}, \Delta^{\mathcal{J}}, (\cdot^{\mathcal{I}_c})_{c \in \mathbb{C}})$

- $(\mathbb{C}, \cdot^{\mathcal{J}})$ DL interpretation on meta level
- $(\Delta, \cdot^{\mathcal{I}_c})$ DL interpretation on object level for each possible world
- $x^{\mathcal{I}_c} = x^{\mathcal{I}_d}$ for all $c, d \in \mathbb{C}$, $x \in \mathsf{O}_{\mathsf{Crig}} \cup \mathsf{O}_{\mathsf{Rrig}} \cup \mathsf{O}_{\mathsf{I}}$

object concept name	$A^{\mathcal{I}_c} \subseteq \Delta$
object concept	C
meta concepts	$[\![C \sqsubseteq D]\!]$ $[\![C(a)]\!]$
meta concept name	$B^{\mathcal{J}} \subseteq \mathbb{C}$
meta concept	E
meta axioms	$E \sqsubseteq F$ $E(c)$

$\mathcal{L}_M [\![\mathcal{L}_O]\!]$ ontology $\mathcal{B}$

Syntax and Semantics of $\mathcal{L}_M [\![\mathcal{L}_O]\!]$

Nested interpretation $\mathcal{J} = (\mathbb{C}, \cdot^{\mathcal{J}}, \Delta^{\mathcal{J}}, (\cdot^{\mathcal{I}_c})_{c \in \mathbb{C}})$

- $(\mathbb{C}, \cdot^{\mathcal{J}})$ DL interpretation on meta level
- $(\Delta, \cdot^{\mathcal{I}_c})$ DL interpretation on object level for each possible world
- $x^{\mathcal{I}_c} = x^{\mathcal{I}_d}$ for all $c, d \in \mathbb{C}$, $x \in \mathsf{O}_{\mathsf{Crig}} \cup \mathsf{O}_{\mathsf{Rrig}} \cup \mathsf{O}_{\mathsf{I}}$

object concept name	$A^{\mathcal{I}_c} \subseteq \Delta$
object concept	$C^{\mathcal{I}_c} \subseteq \Delta$ (acc. to semantics of $\mathcal{L}_O$)
meta concepts	$[\![C \sqsubseteq D]\!]^{\mathcal{J}} := \{c \in \mathbb{C} \mid \mathcal{I}_c \models C \sqsubseteq D\}$ $[\![C(a)]\!]^{\mathcal{J}} := \{c \in \mathbb{C} \mid \mathcal{I}_c \models C(a)\}$
meta concept name	$B^{\mathcal{J}} \subseteq \mathbb{C}$
meta concept	$E^{\mathcal{J}} \subseteq \mathbb{C}$ (acc. to semantics of $\mathcal{L}_M$)
meta axioms	$\mathcal{J} \models E \sqsubseteq F$ $\mathcal{J} \models E(c)$
$\mathcal{L}_M [\![\mathcal{L}_O]\!]$ ontology	$\mathcal{J} \models \mathcal{B}$

$$\left.\begin{array}{l} \mathcal{J} \models E \sqsubseteq F \\ \mathcal{J} \models E(c) \\ \mathcal{J} \models \mathcal{B} \end{array}\right\}\ \text{standard entailment relation}$$

Example of Contextualized DL $\mathcal{L}_M$ $[\![\mathcal{L}_O]\!]$

$$\textsc{Bank} \sqsubseteq [\![\textsc{SavingsAccount} \sqsubseteq \geqslant_1 \mathit{own\_sa}^{-}.\top]\!]$$

$$\textsc{Bank} \sqsubseteq [\![\exists\, \mathit{own\_sa}.\top \sqsubseteq \textsc{Customer}]\!]$$

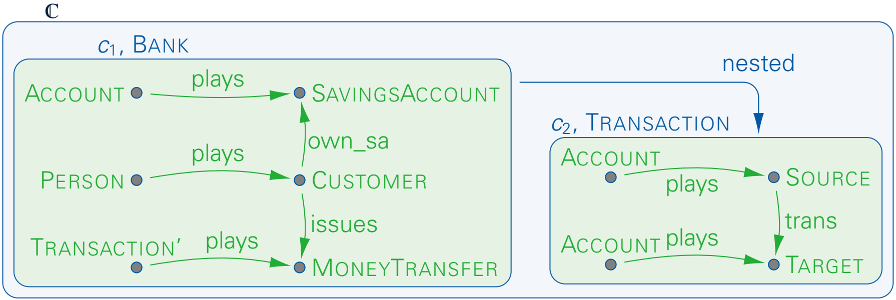
$$\neg[\![(\textsc{Transaction}' \sqcap \exists\, \mathit{plays}.\top) \sqsubseteq \bot]\!] \sqsubseteq \exists\, \mathit{nested}.\textsc{Transaction}$$

Example of Contextualized DL $\mathcal{L}_M [\![\mathcal{L}_O]\!]$

$$\textsc{Bank} \sqsubseteq [\![\textsc{SavingsAccount} \sqsubseteq \geqslant_1 \mathit{own\_sa}^{-}.\top]\!]$$

$$\textsc{Bank} \sqsubseteq [\![\exists\, \mathit{own\_sa}.\top \sqsubseteq \textsc{Customer}]\!]$$

$$\neg[\![(\textsc{Transaction}' \sqcap \exists\, \mathit{plays}.\top) \sqsubseteq \bot]\!] \sqsubseteq \exists\, \mathit{nested}.\textsc{Transaction}$$



Complexity of Consistency Problem

	$\mathcal{L}_M$ \ $\mathcal{L}_O$	$\mathcal{EL}$	$\mathcal{ALC}-\mathcal{SHOQ}$	$\mathcal{SHOIQ}$
no rigid names	$\mathcal{EL}$	constant	EXP-complete	NEXP-complete
	$\mathcal{ALC}-\mathcal{SHOQ}$	EXP-complete	EXP-complete	NEXP-complete
	$\mathcal{SHOIQ}$	NEXP-complete	NEXP-complete	NEXP-complete
only rigid concepts	$\mathcal{EL}$	constant	NEXP-complete	NEXP-hard and in 2NEXP
	$\mathcal{ALC}-\mathcal{SHOQ}$	NEXP-complete	NEXP-complete	NEXP-hard and in 2NEXP
	$\mathcal{SHOIQ}$	NEXP-complete	NEXP-complete	NEXP-hard and in 2NEXP
with rigid roles	$\mathcal{EL}$	constant	2EXP-complete	2EXP-hard and in 2NEXP
	$\mathcal{ALC}-\mathcal{SHOQ}$	NEXP-complete	2EXP-complete	2EXP-hard and in 2NEXP
	$\mathcal{SHOIQ}$	NEXP-complete	2EXP-complete	2EXP-hard and in 2NEXP

Upper Bounds for $\mathcal{L}_M \, [\![\mathcal{L}_O]\!]$

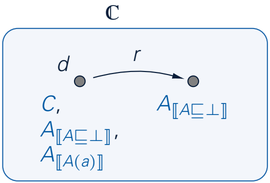
Idea: Split consistency problem into two separate decision problems

- *Outer consistency* with $\mathcal{X}$
- *Admissibility* of $\mathcal{X}$

Check whether meta level is consistent.
Check whether the induced o-axioms are consistent.

$$
\begin{aligned}
\mathcal{B}_{ex} = {} & C \sqsubseteq [\![A(a)]\!] \\
& \wedge (C \sqcap [\![A \sqsubseteq \bot]\!])(d) \\
& \wedge (\exists\, r.[\![A \sqsubseteq \bot]\!])(d)
\end{aligned}
$$

$$
\mathcal{X} = \underbrace{\{\varnothing, \{[\![A(a)]\!]\}, \{[\![A \sqsubseteq \bot]\!]\}\}}_{\{[\![A(a)]\!],[\![A \sqsubseteq \bot]\!]\} \notin \mathcal{X}}
$$



✓

Upper Bounds for $\mathcal{L}_M [\![\mathcal{L}_O]\!]$

Idea: Split consistency problem into two separate decision problems

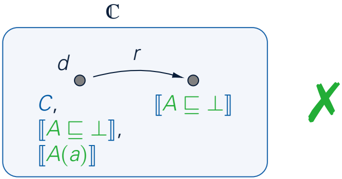
- *Outer consistency* with $\mathcal{X}$
- *Admissibility* of $\mathcal{X}$

Check whether meta level is consistent.

Check whether the induced o-axioms are consistent.

$$
\begin{aligned}
\mathcal{B}_{ex} = {} & C \sqsubseteq [\![A(a)]\!] \\
& \wedge (C \sqcap [\![A \sqsubseteq \bot]\!])(d) \\
& \wedge (\exists r.[\![A \sqsubseteq \bot]\!])(d)
\end{aligned}
$$

$$
\mathcal{X} = \underbrace{\{\varnothing, \{[\![A(a)]\!]\}, \{[\![A \sqsubseteq \bot]\!]\}\}}_{\{[\![A(a)]\!],[\![A \sqsubseteq \bot]\!]\} \notin \mathcal{X}}
$$



Upper Bounds for $\mathcal{L}_M [\![\mathcal{L}_O]\!]$

Idea: Split consistency problem into two separate decision problems

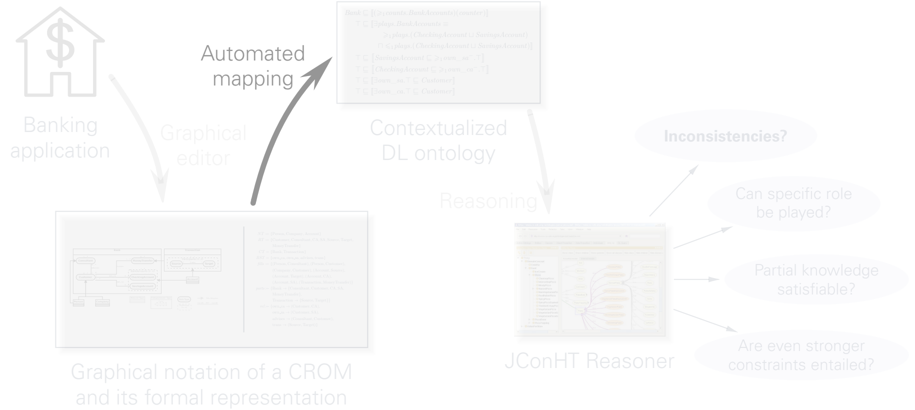
- *Outer consistency* with $\mathcal{X}$
- *Admissibility* of $\mathcal{X}$

Lemma

The $\mathcal{L}_M [\![\mathcal{L}_O]\!]$ ontology $\mathcal{B}$ is consistent iff there is a set $\mathcal{X}$ such that

1. *$\mathcal{X}$ is admissible, and*
2. *the outer abstraction of $\mathcal{B}$ is outer consistent w.r.t. $\mathcal{X}$.*

Workflow of Automated Analysis of Role-Based Models



Automated Mapping

Objective: Given a CROM $\mathcal{M}$, construct an ontology $\mathcal{O}_{\mathcal{M}}$ s.t.
$\mathcal{O}_{\mathcal{M}}$ consistent iff $\mathcal{M}$ satisfiable

Automated Mapping

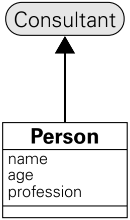
Objective: Given a CROM $\mathcal{M}$, construct an ontology $\mathcal{O}_\mathcal{M}$ s.t.
$\mathcal{O}_\mathcal{M}$ consistent iff $\mathcal{M}$ satisfiable

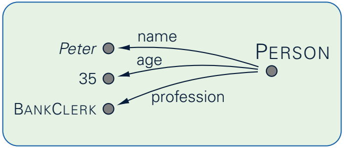
General idea:
- Compartment types $\rightsquigarrow$ meta concepts
- Natural types $\rightsquigarrow$ (rigid) object concepts
- Fields of natural types $\rightsquigarrow$ (rigid) object DL roles

Automated Mapping

Objective: Given a CROM $\mathcal{M}$, construct an ontology $\mathcal{O}_{\mathcal{M}}$ s.t. $\mathcal{O}_{\mathcal{M}}$ consistent iff $\mathcal{M}$ satisfiable

General idea:
- Compartment types $\leadsto$ meta concepts
- Natural types $\leadsto$ (rigid) object concepts
- Fields of natural types $\leadsto$ (rigid) object DL roles



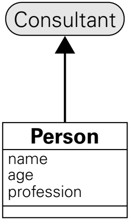


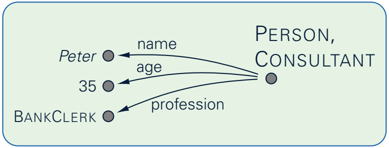
Automated Mapping

Objective: Given a CROM $\mathcal{M}$, construct an ontology $\mathcal{O}_\mathcal{M}$ s.t.
$\mathcal{O}_\mathcal{M}$ consistent iff $\mathcal{M}$ satisfiable

General idea:

- Compartment types $\rightsquigarrow$ meta concepts
- Natural types $\rightsquigarrow$ (rigid) object concepts
- Fields of natural types $\rightsquigarrow$ (rigid) object DL roles

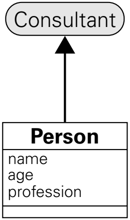


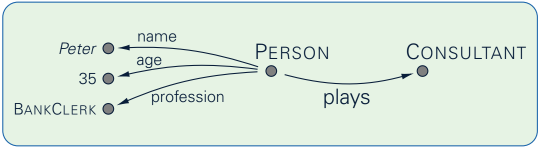


Automated Mapping

Objective: Given a CROM $\mathcal{M}$, construct an ontology $\mathcal{O}_\mathcal{M}$ s.t.
$\mathcal{O}_\mathcal{M}$ consistent iff $\mathcal{M}$ satisfiable

General idea:
- Compartment types $\rightsquigarrow$ meta concepts
- Natural types $\rightsquigarrow$ (rigid) object concepts
- Fields of natural types $\rightsquigarrow$ (rigid) object DL roles





Automated Mapping

Objective: Given a CROM $\mathcal{M}$, construct an ontology $\mathcal{O}_{\mathcal{M}}$ s.t.
$\mathcal{O}_{\mathcal{M}}$ consistent iff $\mathcal{M}$ satisfiable

General idea:

- Compartment types $\rightsquigarrow$ meta concepts
- Natural types $\rightsquigarrow$ (rigid) object concepts
- Fields of natural types $\rightsquigarrow$ (rigid) object DL roles
- Role Types $\rightsquigarrow$ (non-rigid) object concepts
- plays-relation $\rightsquigarrow$ (non-rigid) object DL role
- Relationship types $\rightsquigarrow$ (non-rigid) object DL roles
- Occurrence constraints $\rightsquigarrow$ object individual δ, DL role 'counting'

Automated Mapping

Objective: Given a CROM $\mathcal{M}$, construct an ontology $\mathcal{O}_{\mathcal{M}}$ s.t. $\mathcal{O}_{\mathcal{M}}$ consistent iff $\mathcal{M}$ satisfiable

General idea:

- Compartment types $\leadsto$ meta concepts
- Natural types $\leadsto$ (rigid) object concepts
- Fields of natural types $\leadsto$ (rigid) object DL roles
- Role Types $\leadsto$ (non-rigid) object concepts
- plays-relation $\leadsto$ (non-rigid) object DL role
- Relationship types $\leadsto$ (non-rigid) object DL roles
- Occurrence constraints $\leadsto$ object individual δ, DL role 'counting'

Constraints:

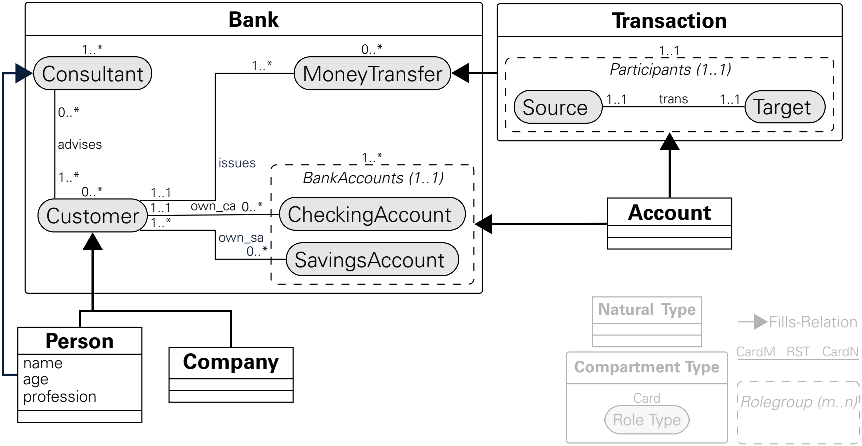
- without constraints $\leadsto$ $\mathcal{ALC}$ $[\![\mathcal{ALCIQ}]\!]$
- with occurrence constraints $\leadsto$ $\mathcal{ALC}$ $[\![\mathcal{ALCOIQ}]\!]$
- full $\leadsto$ $\mathcal{ALC}$ $[\![\mathcal{SHOIQ}]\!]$

- current version of CROM $\leadsto$ no rigid DL roles needed
- additional constraints based on fields of natural type $\leadsto$ rigid roles needed!

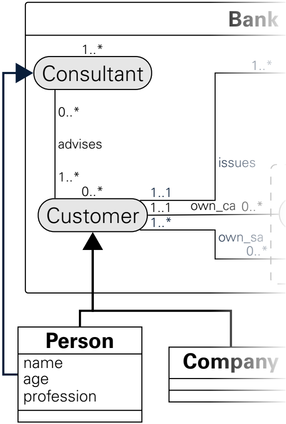
Complexity of Consistency Problem

<table>
<tr><th colspan="2">$\mathcal{L}_O$ \ $\mathcal{L}_M$</th><th>$\mathcal{EL}$</th><th>$\mathcal{ALC} - \mathcal{SHOQ}$</th><th>$\mathcal{SHOIQ}$</th></tr>
<tr><td rowspan="3">no rigid names</td><td>$\mathcal{EL}$</td><td>constant</td><td rowspan="2">ExP-complete</td><td rowspan="3">CROM ★ NExP-complete</td></tr>
<tr><td>$\mathcal{ALC} - \mathcal{SHOQ}$</td><td></td></tr>
<tr><td>$\mathcal{SHOIQ}$</td><td colspan="2"></td></tr>
<tr><td rowspan="3">only rigid concepts</td><td>$\mathcal{EL}$</td><td>constant</td><td rowspan="3">NExP-complete</td><td rowspan="3">NExP-hard and in 2NExP</td></tr>
<tr><td>$\mathcal{ALC} - \mathcal{SHOQ}$</td><td></td></tr>
<tr><td>$\mathcal{SHOIQ}$</td><td></td></tr>
<tr><td rowspan="3">with rigid roles</td><td>$\mathcal{EL}$</td><td>constant</td><td rowspan="3">2ExP-complete</td><td rowspan="3">2ExP-hard and in 2NExP</td></tr>
<tr><td>$\mathcal{ALC} - \mathcal{SHOQ}$</td><td rowspan="2">NExP-complete</td></tr>
<tr><td>$\mathcal{SHOIQ}$</td></tr>
</table>

Example of Mapping

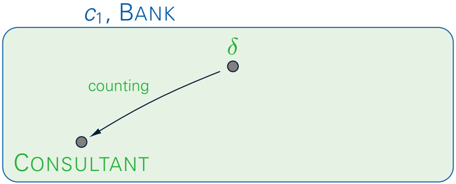


Example of Mapping

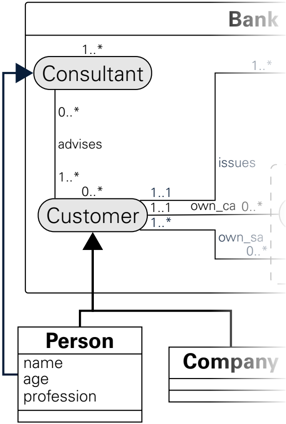


$$\top \sqsubseteq [\![\textsc{Consultant} \sqcup \textsc{Customer} \sqsubseteq =_1 \text{counting}^-.\{\delta\}]\!]$$

$$\textsc{Bank} \sqsubseteq [\![(\geqslant_1 \text{counting}.\textsc{Consultant})(\delta)]\!]$$



Example of Mapping



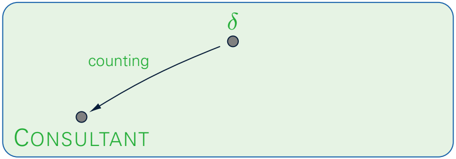
$$\top \sqsubseteq [\![\textsc{Consultant} \sqcup \textsc{Customer} \sqsubseteq =_1 \text{counting}^{-}.\{\delta\}]\!]$$

$$\textsc{Bank} \sqsubseteq [\![(\geqslant_1 \text{counting}.\textsc{Consultant})(\delta)]\!]$$

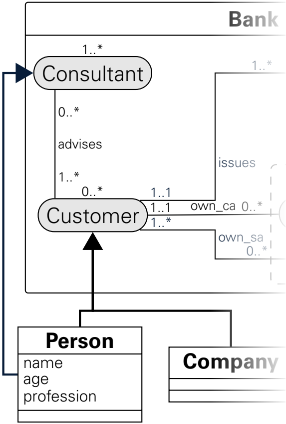
$$\textsc{Bank} \sqsubseteq [\![\top \sqsubseteq \forall.\mathit{advises}.\textsc{Customer}]\!]$$

$$\textsc{Bank} \sqsubseteq [\![\textsc{Consultant} \sqsubseteq \geqslant_1 \text{advises}.\top]\!]$$

c_1, BANK



Example of Mapping

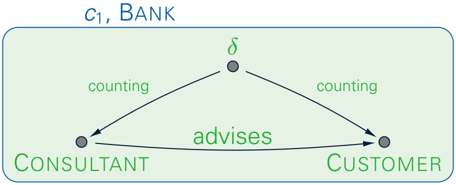


$$\top \sqsubseteq [\![\textsc{Consultant} \sqcup \textsc{Customer} \sqsubseteq {=_1}\text{counting}^-.\{\delta\}]\!]$$

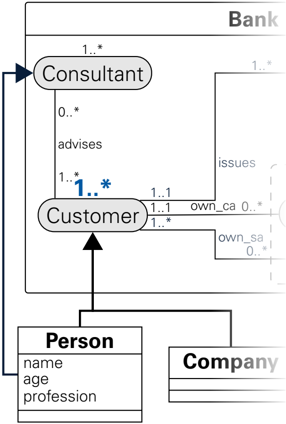
$$\textsc{Bank} \sqsubseteq [\![(\geqslant_1 \text{counting}.\textsc{Consultant})(\delta)]\!]$$

$$\textsc{Bank} \sqsubseteq [\![\top \sqsubseteq \forall.\mathit{advises}.\textsc{Customer}]\!]$$

$$\textsc{Bank} \sqsubseteq [\![\textsc{Consultant} \sqsubseteq \geqslant_1 \text{advises}.\top]\!]$$



Example of Mapping



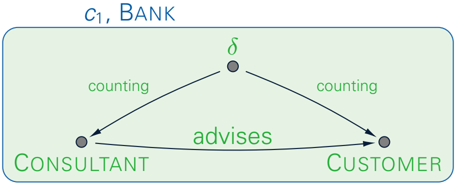
$\top \sqsubseteq [\![\textsc{Consultant} \sqcup \textsc{Customer} \sqsubseteq =_1 \text{counting}^-.\{\delta\}]\!]$

$\textsc{Bank} \sqsubseteq [\![(\geqslant_1 \text{counting}.\textsc{Consultant})(\delta)]\!]$

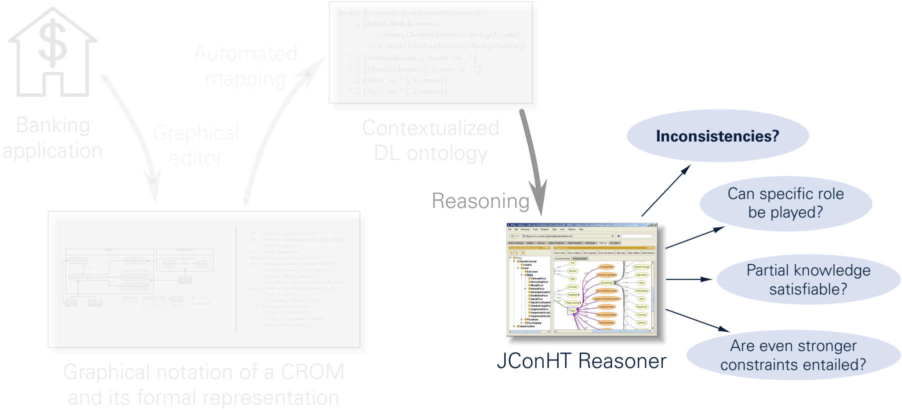
$\textsc{Bank} \sqsubseteq [\![\top \sqsubseteq \forall.\mathit{advises}.\textsc{Customer}]\!]$

$\textsc{Bank} \sqsubseteq [\![\textsc{Consultant} \sqsubseteq \geqslant_1 \text{advises}.\top]\!]$

$\mathit{Bank} \sqsubseteq [\![(\geqslant_1 \text{counting}.\textsc{Customer})(\delta)]\!]$



Workflow of Automated Analysis of Role-Based Models



JConHT – A $\mathcal{SHOIQ}$ $[\![\mathcal{SHOIQ}]\!]$ Reasoner

- **J**ava implemented **Con**textualized description logic reasoner based on **H**ermi**T**/**H**yper**T**ableau
- First reasoner capable of processing contextualized DL ontologies
- Utilize Lemma about separation of reasoning tasks
  1. Check consistency of meta level
  2. Check whether induced object axioms consistent
- Reuse existing, highly optimized reasoners
  - Model-construction based reasoner necessary
  - Implemented in Java
  - Good performance on DL Consistency

  ↳ HermiT as core reasoner

JConHT – A $\mathcal{SHOIQ}$ $[\![\mathcal{SHOIQ}]\!]$ Reasoner

- **J**ava implemented **Con**textualized description logic reasoner based on **H**ermi**T**/**H**yper**T**ableau
- First reasoner capable of processing contextualized DL ontologies
- Utilize Lemma about separation of reasoning tasks
  1. Check consistency of meta level
  2. Check whether induced object axioms consistent
- Reuse existing, highly optimized reasoners
  - Model-construction based reasoner necessary
  - Implemented in Java
  - Good performance on DL Consistency

↳ HermiT as core reasoner

JConHT – A $\mathcal{SHOIQ}[\![\mathcal{SHOIQ}]\!]$ Reasoner

- **J**ava implemented **Con**textualized description logic reasoner based on **H**ermi**T**/**H**yper**T**ableau
- First reasoner capable of processing contextualized DL ontologies
- Utilize Lemma about separation of reasoning tasks
    1. Check consistency of meta level
    2. Check whether induced object axioms consistent
- Reuse existing, highly optimized reasoners
    - Model-construction based reasoner necessary
    - Implemented in Java
    - Good performance on DL Consistency

  ↳ HermiT as core reasoner

JConHT – A $\mathcal{SHOIQ}$ $[\![\mathcal{SHOIQ}]\!]$ Reasoner

- **J**ava implemented **Con**textualized description logic reasoner based on **H**ermi**T**/**H**yper**T**ableau
- First reasoner capable of processing contextualized DL ontologies
- Utilize Lemma about separation of reasoning tasks
  1. Check consistency of meta level
  2. Check whether induced object axioms consistent
- Reuse existing, highly optimized reasoners
  - Model-construction based reasoner necessary
  - Implemented in Java
  - Good performance on DL Consistency

  ↳ HermiT as core reasoner

Algorithm for Checking Consistency

Input : $\mathcal{SHOIQ}$ [[$\mathcal{SHOIQ}$]]-ontology $\mathcal{O}$
Output: true if $\mathcal{O}$ is consistent, false otherwise
Preprocessing (results in $(\mathcal{C}, \mathcal{A})$):

1. Elimination of transitivity axioms, normalization, clausification
2. Repletion of DL-clauses

Let $(\mathcal{T}, \lambda)$ be any derivation for $(\mathcal{C}, \mathcal{A})$.
$\mathfrak{A} := \{\mathcal{A}' \mid$ there exists a leaf node in $(\mathcal{T}, \lambda)$ that is labelled with $\mathcal{A}'\}$

for $\mathcal{A}' \in \mathfrak{A}$ **do**
  if $\mathcal{A}'$ *is clash-free* **then**
    if $\mathcal{O}$ *contains rigid names* **then**
      if $\mathcal{K}_{\mathsf{rig}} := (\mathcal{O}_{\mathcal{A}'}, \mathcal{R}_{\mathsf{O}}{}')$ *is consistent* **then**
        return true
    else
      Let $\{c_1, \ldots, c_k\}$ be the individuals occurring in $\mathcal{A}'$
      if $\mathcal{K}_i := (\mathcal{O}_{c_i}, \mathcal{R}_{\mathsf{O}})$ *is consistent for all* $1 \leq i \leq k$ **then**
        return true

return *false*

Repletion of DL-Clauses

$$\begin{aligned} \mathcal{B}_{\mathrm{ex}} := {} & \neg C(s) \\ & \wedge [\![\neg B(a)]\!] \sqsubseteq C \\ & \wedge \neg C \sqsubseteq [\![B \sqsubseteq \bot]\!] \end{aligned}$$

Repletion of DL-Clauses

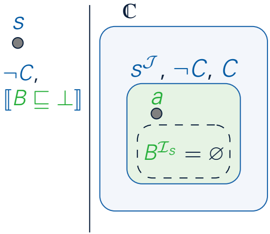
$$\begin{aligned}\mathcal{B}_{\text{ex}} &:= \neg C(s) \\ &\quad \wedge [\![\neg B(a)]\!] \sqsubseteq C \\ &\quad \wedge \neg C \sqsubseteq [\![B \sqsubseteq \bot]\!]\end{aligned}$$

$$\begin{aligned}\mathcal{A}_{\text{ex}} &:= \{\neg C(s)\} \\ \mathcal{C}_{\text{ex}} &:= \{[\![\neg B(a)]\!](x) \to C(x), \\ &\qquad \top \to C(x) \vee [\![B \sqsubseteq \bot]\!](x)\}\end{aligned}$$

Repletion of DL-Clauses

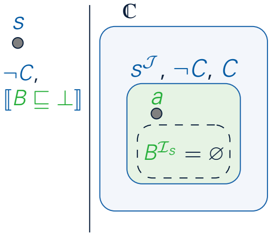
$$\begin{aligned}\mathcal{B}_{\mathrm{ex}} :=\ & \neg C(s)\\ & \wedge [\![\neg B(a)]\!] \sqsubseteq C\\ & \wedge \neg C \sqsubseteq [\![B \sqsubseteq \bot]\!]\end{aligned}$$

$$\begin{aligned}\mathcal{A}_{\mathrm{ex}} &:= \{\neg C(s)\}\\ \mathcal{C}_{\mathrm{ex}} &:= \{[\![\neg B(a)]\!](x) \to C(x),\\ &\qquad \top \to C(x) \vee [\![B \sqsubseteq \bot]\!](x)\}\end{aligned}$$



Repletion of DL-Clauses

$$\begin{aligned}\mathcal{B}_{\text{ex}} &:= \neg C(s)\\ &\wedge [\![\neg B(a)]\!] \sqsubseteq C\\ &\wedge \neg C \sqsubseteq [\![B \sqsubseteq \bot]\!]\end{aligned} \qquad \begin{aligned}\mathcal{A}_{\text{ex}} &:= \{\neg C(s)\}\\ \mathcal{C}_{\text{ex}} &:= \{[\![\neg B(a)]\!](x) \to C(x),\\ &\qquad \top \to C(x) \vee [\![B \sqsubseteq \bot]\!](x)\}\end{aligned}$$



Definition (Repletion)

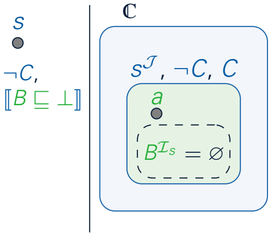
Let $\mathcal{C}$ be a set of DL-clauses. The **repletion** of $\mathcal{C}$ is obtained from $\mathcal{C}$ by adding the DL-clause $\top \to [\![\alpha]\!](x) \vee [\![\neg\alpha]\!](x)$ for each o-axiom $[\![\alpha]\!]$ occurring in $\mathcal{C}$.

Repletion of DL-Clauses

$$\begin{aligned}\mathcal{B}_{\text{ex}} := {} & \neg C(s) \\ & \wedge [\![\neg B(a)]\!] \sqsubseteq C \\ & \wedge \neg C \sqsubseteq [\![B \sqsubseteq \bot]\!]\end{aligned}$$

$$\mathcal{A}_{\text{ex}} := \{\neg C(s)\}$$

$$\begin{aligned}\mathcal{C}_{\text{ex}} := \{ & [\![\neg B(a)]\!](x) \to C(x), \\ & \top \to C(x) \vee [\![B \sqsubseteq \bot]\!](x), \\ & \top \to [\![B \sqsubseteq \bot]\!](x) \vee [\![\neg(B \sqsubseteq \bot)]\!](x), \\ & \top \to [\![B(a)]\!](x) \vee [\![\neg B(a)]\!](x)\}\end{aligned}$$



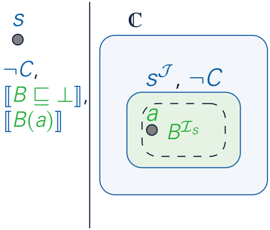
Definition (Repletion)

Let $\mathcal{C}$ be a set of DL-clauses. The **repletion** of $\mathcal{C}$ is obtained from $\mathcal{C}$ by adding the DL-clause $\top \to [\![\alpha]\!](x) \vee [\![\neg\alpha]\!](x)$ for each o-axiom $[\![\alpha]\!]$ occurring in $\mathcal{C}$.

Repletion of DL-Clauses

$$\begin{aligned}\mathcal{B}_{\text{ex}} := {} & \neg C(s) \\ & \wedge [\![\neg B(a)]\!] \sqsubseteq C \\ & \wedge \neg C \sqsubseteq [\![B \sqsubseteq \bot]\!]\end{aligned}$$

$$\begin{aligned}\mathcal{A}_{\text{ex}} &:= \{\neg C(s)\} \\ \mathcal{C}_{\text{ex}} &:= \{[\![\neg B(a)]\!](x) \rightarrow C(x), \\ &\qquad \top \rightarrow C(x) \vee [\![B \sqsubseteq \bot]\!](x), \\ &\qquad \top \rightarrow [\![B \sqsubseteq \bot]\!](x) \vee [\![\neg(B \sqsubseteq \bot)]\!](x), \\ &\qquad \top \rightarrow [\![B(a)]\!](x) \vee [\![\neg B(a)]\!](x)\}\end{aligned}$$



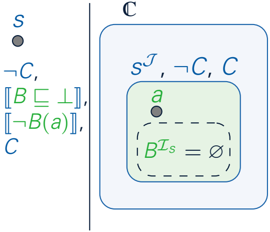
Definition (Repletion)

Let $\mathcal{C}$ be a set of DL-clauses. The **repletion** of $\mathcal{C}$ is obtained from $\mathcal{C}$ by adding the DL-clause $\top \rightarrow [\![\alpha]\!](x) \vee [\![\neg\alpha]\!](x)$ for each o-axiom $[\![\alpha]\!]$ occurring in $\mathcal{C}$.

Repletion of DL-Clauses

$$
\begin{aligned}
\mathcal{B}_{\mathrm{ex}} := {} & \neg C(s) \\
& \wedge [\![\neg B(a)]\!] \sqsubseteq C \\
& \wedge \neg C \sqsubseteq [\![B \sqsubseteq \bot]\!]
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{A}_{\mathrm{ex}} := {} & \{\neg C(s)\} \\
\mathcal{C}_{\mathrm{ex}} := {} & \{[\![\neg B(a)]\!](x) \to C(x), \\
& \quad \top \to C(x) \vee [\![B \sqsubseteq \bot]\!](x), \\
& \quad \top \to [\![B \sqsubseteq \bot]\!](x) \vee [\![\neg(B \sqsubseteq \bot)]\!](x), \\
& \quad \top \to [\![B(a)]\!](x) \vee [\![\neg B(a)]\!](x)\}
\end{aligned}
$$



Definition (Repletion)

Let $\mathcal{C}$ be a set of DL-clauses. The **repletion** of $\mathcal{C}$ is obtained from $\mathcal{C}$ by adding the DL-clause $\top \to [\![\alpha]\!](x) \vee [\![\neg\alpha]\!](x)$ for each o-axiom $[\![\alpha]\!]$ occurring in $\mathcal{C}$.

Thank you for your attention!

Any questions?

Performance Evaluation

- Pseudo-random domain models of increasing complexity:
  - based on parameter *n*
  - *n* natural types, *n* compartment types, for each compartment type *n* role types
- 3 different scenarios
  1. Number of (constrained) relationship types per compartment type
  2. Number of role groups per compartment type
  3. Whether or not compartments can play roles.
- Average execution time of JConHT
  - Translation from CROM models into ontologies neglected
  - For each configuration 100 executions
  - Average time needed to decide consistency

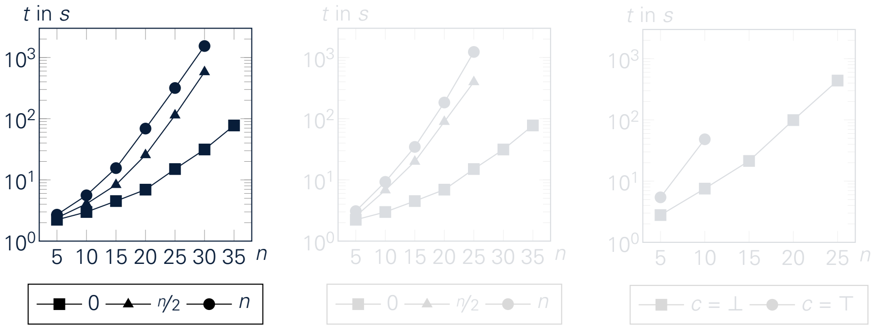
Performance Evaluation

- Pseudo-random domain models of increasing complexity:
  - based on parameter n
  - n natural types, n compartment types, for each compartment type n role types
- 3 different scenarios
  1. Number of (constrained) relationship types per compartment type
  2. Number of role groups per compartment type
  3. Whether or not compartments can play roles.
- Average execution time of JConHT
  - Translation from CROM models into ontologies neglected
  - For each configuration 100 executions
  - Average time needed to decide consistency

Performance Evaluation

- Pseudo-random domain models of increasing complexity:
  - based on parameter *n*
  - *n* natural types, *n* compartment types, for each compartment type *n* role types
- 3 different scenarios
  1. Number of (constrained) relationship types per compartment type
  2. Number of role groups per compartment type
  3. Whether or not compartments can play roles.
- Average execution time of JConHT
  - Translation from CROM models into ontologies neglected
  - For each configuration 100 executions
  - Average time needed to decide consistency

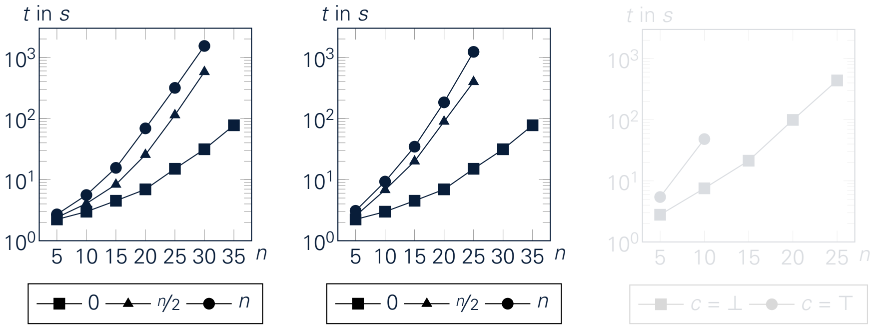
Performance Evaluation



(a) Variation of ♯ RSTs. (b) Variation of ♯ RGs. (c) Flat vs. nested CTs.

Average execution times of JConHT for benchmark ontologies.

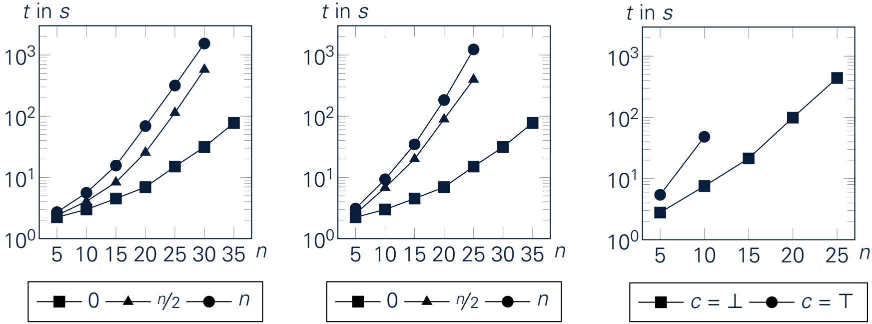
Performance Evaluation



(a) Variation of ♯ RSTs. (b) Variation of ♯ RGs. (c) Flat vs. nested CTs.

Average execution times of JConHT for benchmark ontologies.

Performance Evaluation



(a) Variation of ♯ RSTs. (b) Variation of ♯ RGs. (c) Flat vs. nested CTs.

Average execution times of JConHT for benchmark ontologies.

References I

Kühn, Thomas, Stephan Böhme, et al. (2015). "A combined formal model for relational context-dependent roles". In: *Proc. of the 8th ACM SIGPLAN Int. Conf. on Software Language Engineering (SLE 2015)*. (Pittsburgh, PA, USA). Ed. by Richard F. Paige, Davide Di Ruscio, and Markus Völter. ACM, pp. 113–124.

Kühn, Thomas, Max Leuthäuser, et al. (2014). "A Metamodel Family for Role-Based Modeling and Programming Languages". In: *Proc. of the 7th Int. Conf. on Software Language Engineering (SLE 2014)*. (Västerås, Sweden). Ed. by Benoît Combemale et al. Vol. 8706. Lecture Notes in Computer Science. Springer-Verlag, pp. 141–160.

Steimann, Friedrich (2000). "On the representation of roles in object-oriented and conceptual modelling". In: *Data & Knowledge Engineering* 35.1, pp. 83–106.