Reasoning with Exceptions in Contextualized Knowledge Repositories

Loris Bozzato

Data and Knowledge Management Research Unit,
Fondazione Bruno Kessler - Trento, Italy

bozzato[at]fbk.eu

eKNOW 2020 Tutorial

12th Int. Conf. on Information, Process, and Knowledge Management
November 21-25, 2020 – Valencia, Spain

Joint work with...

DKM and PDI units @ Fondazione Bruno Kessler:

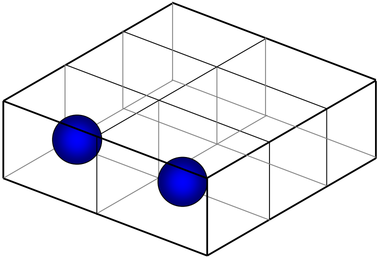
- Luciano Serafini
- Martin Homola
- Mathew Joseph
- Francesco Corcoglioniti
- Chiara Ghidini
- Andrei Tamilin
- Gaetano Calabrese

Institut für Informationssysteme @ TU Wien:

- Thomas Eiter

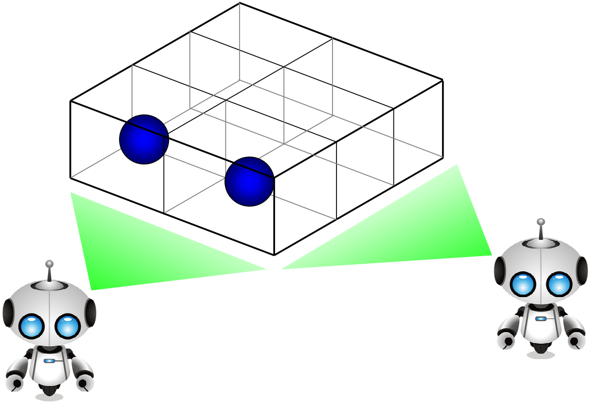
Reasoning in context

Classic example: Magic Box [Ghidini and Giunchiglia, 2001]



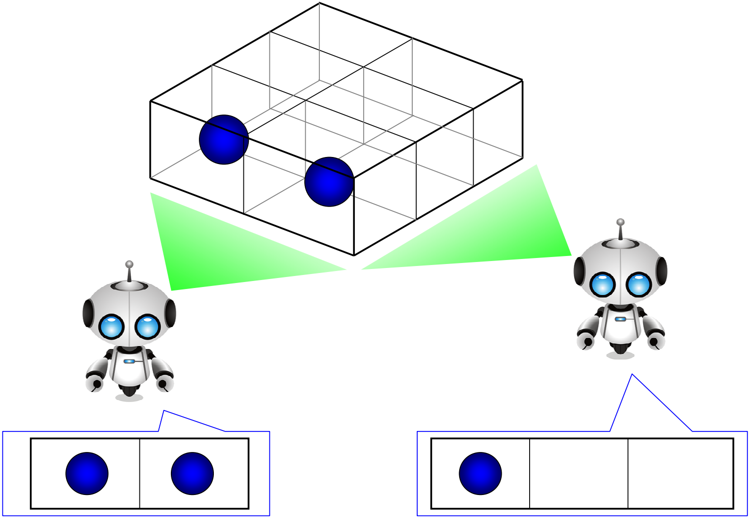
Reasoning in context

Classic example: Magic Box [Ghidini and Giunchiglia, 2001]



Reasoning in context

Classic example: Magic Box [Ghidini and Giunchiglia, 2001]



AI theory of context

Contextual AI theory principia: [McCarthy, 1993]

- Every formula is asserted in a context
- Context are first class logical objects
  (formulas can predicate about contexts)
- Knowledge propagates across contexts

AI theory of context

Contextual AI theory principia: [McCarthy, 1993]

- Every formula is asserted in a context
  "In FIFA World Cup 2006, the Winner is Italy."
  "In FIFA World Cup 2010, the Winner is Spain."
- Context are first class logical objects
  (formulas can predicate about contexts)
- Knowledge propagates across contexts

AI theory of context

Contextual AI theory principia: [McCarthy, 1993]

- Every formula is asserted in a context
  "In FIFA World Cup 2006, the Winner is Italy."
  "In FIFA World Cup 2010, the Winner is Spain."
- Context are first class logical objects
  (formulas can predicate about contexts)
  "Context FifaWC10 is about FifaWorldCup in year 2010"
  "Context Football9810 is about Football in years 1998-2010"
  "Football9810 is more general than FifaWC10"
- Knowledge propagates across contexts

AI theory of context

Contextual AI theory principia: [McCarthy, 1993]

- Every formula is asserted in a context
  "In FIFA World Cup 2006, the Winner is Italy."
  "In FIFA World Cup 2010, the Winner is Spain."
- Context are first class logical objects
  (formulas can predicate about contexts)
  "Context FifaWC10 is about FifaWorldCup in year 2010"
  "Context Football9810 is about Football in years 1998-2010"
  "Football9810 is more general than FifaWC10"
- Knowledge propagates across contexts
  "Every Winner in FifaWC06 is a QualifiedTeam in FifaWC10"

Theory of contexts: Context as a Box

Idea [Benerecetti et al., 2000]

- A context is a logical theory...
- ...associated to a region in a contextual space

Theory of contexts: Context as a Box

Idea [Benerecetti et al., 2000]

- A context is a logical theory...
- ...associated to a region in a contextual space

$$\mathbf{C} = \boxed{\begin{array}{l} \mathrm{HostTeam} \sqsubseteq \mathrm{QualifiedTeam} \\ \ldots \\ \mathrm{Winner}(\mathrm{team\_spain}) \\ \mathrm{RunnerUp}(\mathrm{team\_holland}) \\ \ldots \\ \mathrm{playsFor}(\mathrm{buffon}, \mathrm{team\_italy}) \\ \mathrm{playsFor}(\mathrm{cannavaro}, \mathrm{team\_italy}) \\ \ldots \end{array}}$$

Theory of contexts: Context as a Box

Idea [Benerecetti et al., 2000]

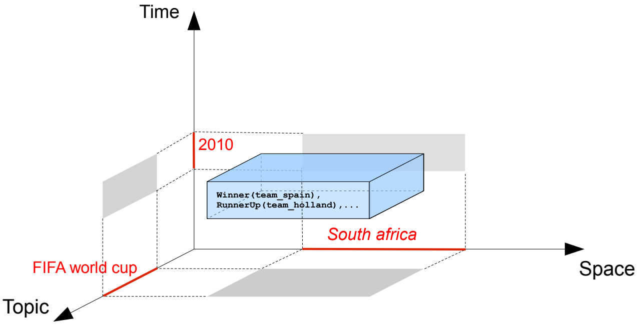
- A context is a logical theory...
- ...associated to a region in a contextual space

$$\text{time}(\mathbf{C}, 2010),\ \text{location}(\mathbf{C}, \text{South\_Africa}),\ \text{topic}(\mathbf{C}, \text{FIFA\_WC})$$

$$\mathbf{C} = \boxed{\begin{array}{l} \text{HostTeam} \sqsubseteq \text{QualifiedTeam} \\ \dots \\ \text{Winner}(\text{team\_spain}) \\ \text{RunnerUp}(\text{team\_holland}) \\ \dots \\ \text{playsFor}(\text{buffon}, \text{team\_italy}) \\ \text{playsFor}(\text{cannavaro}, \text{team\_italy}) \\ \dots \end{array}}$$

Idea [Benerecetti et al., 2000]

- A context is a logical theory...
- ...associated to a region in a contextual space



Motivation: contexts and SW data

Need for context in Semantic Web

- Most of Semantic Web data holds in specific contextual space (time, location, topic...)
- No explicit support for reasoning with context sensitive knowledge in Semantic Web languages

➔ Current practice:
Contextual information often “handcrafted” in implementation

Example: current context implementation

Freebase: context representation for events

`<fb:base.x2016fifaeurocupfrance.`
`euro_cup_team.qualified_as>`

represents:

Example: current context implementation

Freebase: context representation for events

<fb:base.x2016fifaeurocupfrance.euro_cup_team.qualified_as>

represents:

- a context dependent relation: **euro_cup_team.qualified_as**

Example: current context implementation

Freebase: context representation for events

`<fb:base.x2016fifaeurocupfrance.euro_cup_team.qualified_as>`

represents:

- a context dependent relation: `euro_cup_team.qualified_as`
- in the context identified by:
  - Time: `2016`

Example: current context implementation

Freebase: context representation for events

<fb:base.x2016fifaeurocupfrance.euro_cup_team.qualified_as>

represents:

- a context dependent relation: **euro_cup_team.qualified_as**
- in the context identified by:
  - Time: **2016**
  - Topic: **fifaeurocup**

Example: current context implementation

Freebase: context representation for events

`<fb:base.x2016fifaeurocupfrance.euro_cup_team.qualified_as>`

represents:

- a context dependent relation: `euro_cup_team.qualified_as`
- in the context identified by:
  - Time: `2016`
  - Topic: `fifaeurocup`
  - Location: `france`

Example: current context implementation

Freebase: context representation for events

`<fb:base.x2016fifaeurocupfrance.euro_cup_team.qualified_as>`

represents:

- a context dependent relation: `euro_cup_team.qualified_as`
- in the context identified by:
  - Time: `2016`
  - Topic: `fifaeurocup`
  - Location: `france`

- Context information encoded in the link is implicit knowledge!
- No way to uniformly retrieve and reason over such information

Example: current context implementation

Freebase: context representation for events

`<fb:base.x2016fifaeurocupfrance.euro_cup_team.qualified_as>`

represents:

- a context dependent relation: `euro_cup_team.qualified_as`
- in the context identified by:
  - Time: `2016`
  - Topic: `fifaeurocup`
  - Location: `france`

- Context information encoded in the link is implicit knowledge!
- No way to uniformly retrieve and reason over such information

➔ Context representation for Semantic Web data needs a well-defined theory of contexts

CKR: Contextualized Knowledge Repository

Contextualized Knowledge Repository (CKR)

- DL based framework for representation and reasoning with contextual knowledge in the Semantic Web
- Contextual theory: based on formal AI theories of context [McCarthy, 1993, Lenat, 1998, Ghidini and Giunchiglia, 2001]

Other DL contextual frameworks:
[Bao et al., 2010, Klarman and Gutiérrez-Basulto, 2011, Straccia et al., 2010].

Contextual modelling needs

From study on typical use of context in Semantic Web data:

Requirements

- Statement contextualization: associate context to facts
- Symbols locality: local meaning for symbols
- Cross-context TBox statements: knowledge relations across contexts
- Complex contextualization: more than one contextual values to facts
- Modularity: separation of knowledge in independent modules
- Unified reasoning and query: inference and query use context structure
- ...

Contextual modelling needs

From study on typical use of context in Semantic Web data:

Requirements

- Statement contextualization: associate context to facts
- Symbols locality: local meaning for symbols
- Cross-context TBox statements: knowledge relations across contexts
- Complex contextualization: more than one contextual values to facts
- Modularity: separation of knowledge in independent modules
- Unified reasoning and query: inference and query use context structure
- …

➔ Definition of "contextual primitives" of CKR

(e.g. cross-context statements ➔ *eval* operator,
complex contextualization ➔ c.classes and modules …)

CKR objectives and plan

CKR objectives

A general formalism and tool for the representation and reasoning with contextual knowledge in the Semantic Web.

- Theory: based on formal theories of context from AI
- Implementation: built over state of the art tools
- Evaluation: for performance and ease of modeling

CKR objectives and plan

CKR objectives

A general formalism and tool for the representation and reasoning with contextual knowledge in the Semantic Web.

- Theory: based on formal theories of context from AI
- Implementation: built over state of the art tools
- Evaluation: for performance and ease of modeling

Plan

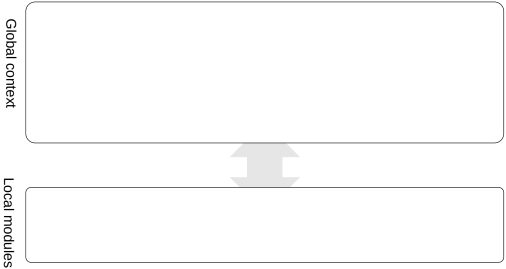
1. Tailor a logic of context in AI for Semantic Web needs
2. Provide an axiomatization of this new logic
3. Define reasoning services
4. Implement the theory on a platform
5. Evaluate by representation adequacy and performance

Overview

1. CKR model
2. Reasoning
3. Implementation on RDF
4. Defeasible axioms
5. Contextual hierarchies

Overview

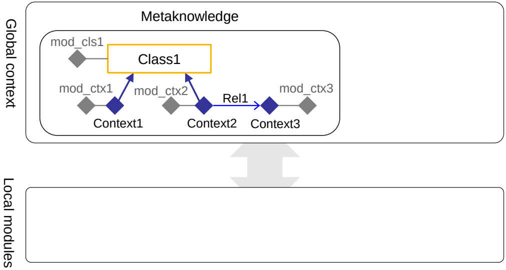
CKR structure



Global context

(Local) contexts

CKR structure

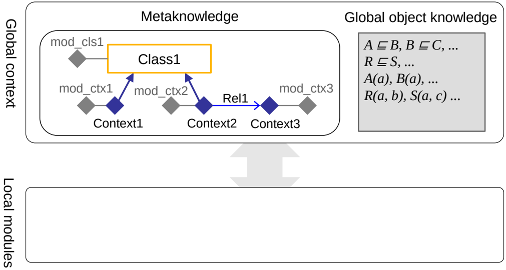


Global context

- Metaknowledge:
  structure of contexts, context classes, relations, modules and attributes

(Local) contexts

CKR structure

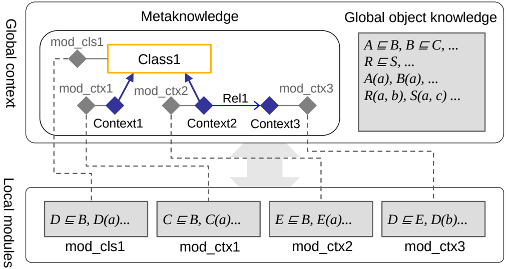


Global context

- Metaknowledge:
  structure of contexts, context classes, relations, modules and attributes
- Global object knowledge:
  knowledge shared by all contexts

(Local) contexts

CKR structure



Global context

- **Metaknowledge:** structure of contexts, context classes, relations, modules and attributes
- **Global object knowledge:** knowledge shared by all contexts

(Local) contexts

- **Object knowledge with references:** local knowledge with references to value of predicates in other contexts
- Knowledge distributed across different **modules K_m**

$\mathcal{SROIQ}$-RL

Basic modeling language: description logic $\mathcal{SROIQ}$-RL,

- $\mathcal{SROIQ}$-RL is a restriction of $\mathcal{SROIQ}$
- It corresponds to the syntax of the OWL-RL profile of OWL-2

$\mathcal{SROIQ}$-RL

$$C := A \mid \{a\} \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \exists R.C_1 \mid \exists R.\{a\} \mid \exists R.\top$$

$$D := A \mid D_1 \sqcap D_2 \mid \neg C_1 \mid \forall R.D_1 \mid \exists R.\{a\} \mid \leqslant [0,1] R.C_1 \mid \leqslant [0,1] R.\top$$

TBox axioms: $C \sqsubseteq D$ ABox axioms: $D(a), R(a,b)$

Example

- CulturalEvent $\sqsubseteq$ Event, SportsEvent $\sqsubseteq$ Event
- Event $\sqsubseteq$ $\exists$mod.{m_event}
- VolleyA1Competition(A1_2012-13), SportiveTourist(volley_fan_01)

Metalanguage $\mathcal{L}_\Gamma$

Metavocabulary Γ: Contexts structure objects

- **N**: context names (match1, volley_season2013)
- **M**: module names (m_match1, m_event)
  with role **mod** : **N** $\times$ **M**
- **C**: context classes (Event, VolleyMatch)
  with **Ctx** $\in$ **C**: class of all contexts
- **R**: contextual relations (hasSubEvent, covers)
- **A**: contextual attributes (time, location, topic)
- D_A attribute values of A $\in$ **A** (2013, trento, sport)

Metalanguage $\mathcal{L}_\Gamma$: DL language over Γ

Object language $\mathcal{L}_\Sigma$

Object vocabulary Σ: domain vocabulary

Eval expression

For X a concept or role expression in Σ, C a concept expression in Γ

$$eval(X, \mathsf{C})$$

"The interpretation of X in all the contexts of type C"

Idea: "imports" meaning of X from all contexts in C

Object language $\mathcal{L}_\Sigma$

Object vocabulary Σ: domain vocabulary

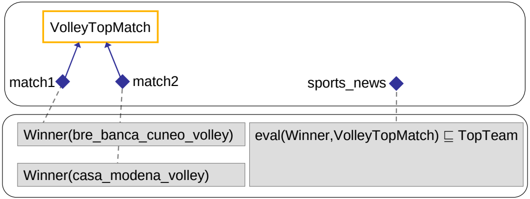
Eval expression

For X a concept or role expression in Σ, C a concept expression in Γ

$$eval(X, \mathsf{C})$$

"The interpretation of X in all the contexts of type C"

Idea: "imports" meaning of X from all contexts in C



Object language with references $\mathcal{L}^e_\Sigma$: $\mathcal{L}_\Sigma$ with eval expressions

Object language $\mathcal{L}_\Sigma$

Object vocabulary Σ: domain vocabulary

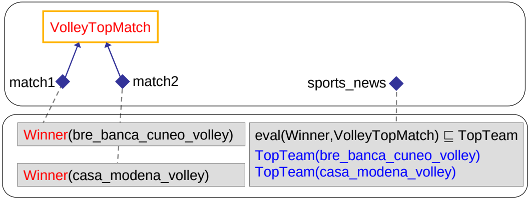
Eval expression

For X a concept or role expression in Σ, C a concept expression in Γ

$$eval(X, \mathsf{C})$$

"The interpretation of X in all the contexts of type C"

Idea: "imports" meaning of X from all contexts in C



Object language with references $\mathcal{L}^e_\Sigma$: $\mathcal{L}_\Sigma$ with eval expressions

Contextualized Knowledge Repository

Contextualized Knowledge Repository (CKR):

$$\mathfrak{K} = \langle \mathfrak{G}, \{\mathrm{K_m}\}_{\mathsf{m} \in \mathbf{M}} \rangle$$

- $\mathfrak{G}$ contains
  - metaknowledge axioms in $\mathcal{L}_\Gamma$
  - global object axioms in $\mathcal{L}_\Sigma$
- for every module name $\mathsf{m} \in \mathbf{M}$,
  $\mathrm{K_m}$ contains object axioms with references in $\mathcal{L}^e_\Sigma$

Tourism example: introduction

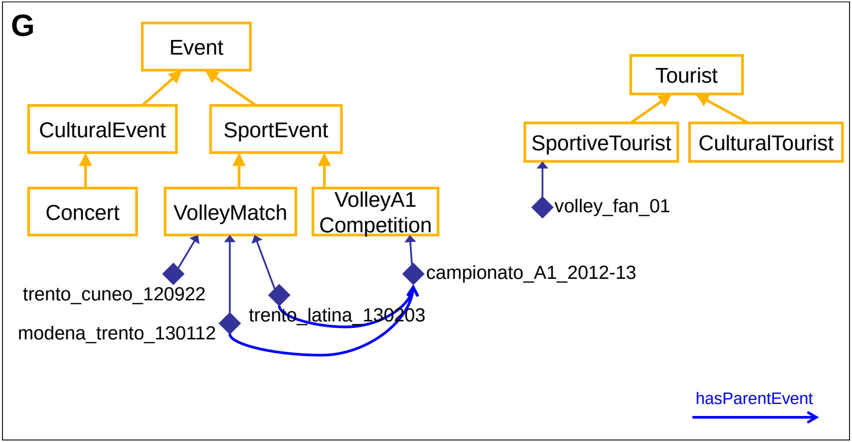
Tourism example:

- Idea: Tourism recommendation for events in Trentino
- Structure of contexts represents events and tourists information

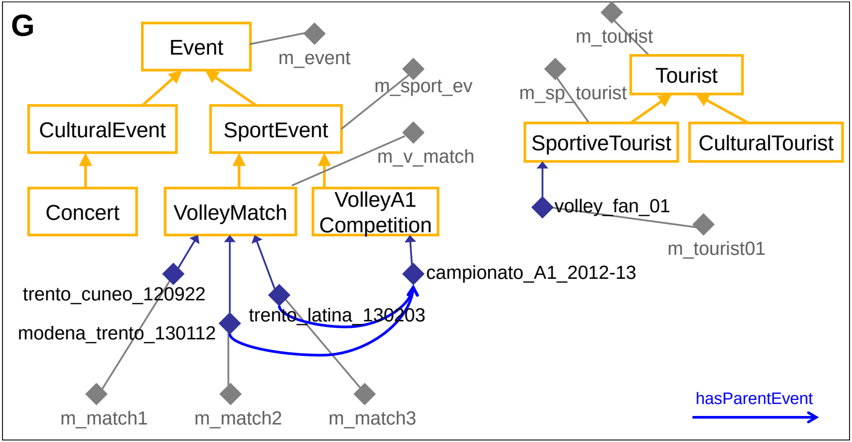
➔ Task: find interesting events on the base of tourists' preferences

We model this domain in a CKR $\mathfrak{K}_{tour} = \langle \mathfrak{G}, \{\mathrm{K_m}\}_{\mathsf{m} \in \mathbf{M}} \rangle$

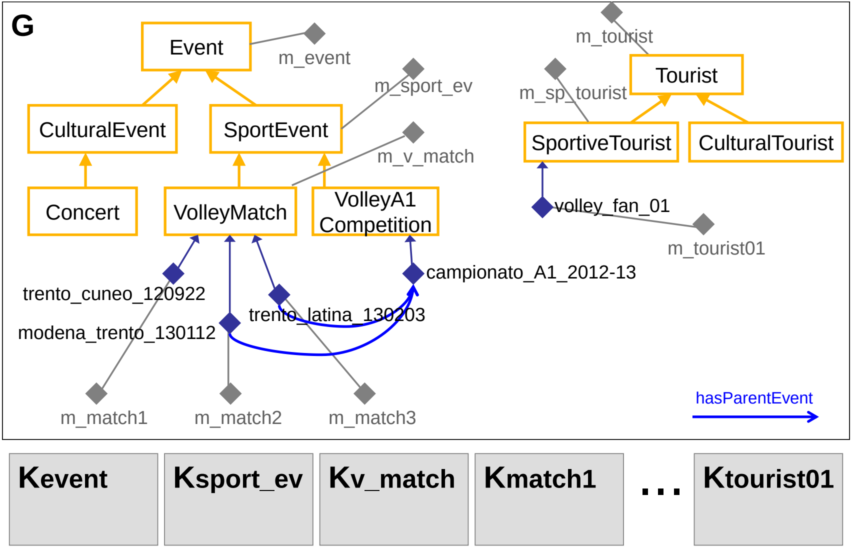
Tourism example: CKR structure



Tourism example: CKR structure



Tourism example: CKR structure



Tourism example: some modules contents

In $\mathrm{K}_{\text{v\_match}}$: $HomeTeam \sqsubseteq Team$ $\quad HostTeam \sqsubseteq Team$
$Winner \sqsubseteq Team$ $\quad Loser \sqsubseteq Team$

In $\mathrm{K}_{\text{match2}}$: $HomeTeam(casa\_modena\_volley)$ $\quad HostTeam(itas\_trentino\_volley)$
$Winner(casa\_modena\_volley)$ $\quad Loser(itas\_trentino\_volley)$

Tourism example: some modules contents

In $\mathrm{K}_{\text{v\_match}}$: $HomeTeam \sqsubseteq Team$ $\quad HostTeam \sqsubseteq Team$
$Winner \sqsubseteq Team$ $\quad Loser \sqsubseteq Team$

In $\mathrm{K}_{\text{match2}}$: $HomeTeam(casa\_modena\_volley)$ $\quad HostTeam(itas\_trentino\_volley)$
$Winner(casa\_modena\_volley)$ $\quad Loser(itas\_trentino\_volley)$

...

In $\mathrm{K}_{\text{sport\_ev}}$: "Winners of major volley matches are top teams"

$$eval(Winner, \mathsf{VolleyMatch} \sqcap \exists \mathsf{hasParentEvent.VolleyA1Competition}) \sqsubseteq TopTeam$$

In $\mathrm{K}_{\text{sp\_tourist}}$: "Top teams are preferred teams"

$$eval(TopTeam, \mathsf{SportEvent}) \sqsubseteq PreferredTeam$$

CKR interpretation

Idea

CKR interpretations are two layered interpretations

CKR interpretation $\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$

- $\mathcal{M}$ is a DL interpretation over $\Gamma \cup \Sigma$
- For every $x \in \mathsf{Ctx}^{\mathcal{M}}$, $\mathcal{I}(x)$ is a DL interpretation over Σ
  - $\Delta^{\mathcal{I}(x)} = \Delta^{\mathcal{M}}$
  - for $a \in \mathrm{NI}_{\Sigma}$, $a^{\mathcal{I}(x)} = a^{\mathcal{M}}$

Interpretation of *eval*: $\mathit{eval}(X, \mathsf{C})^{\mathcal{I}(x)} = \bigcup_{\mathsf{e} \in \mathsf{C}^{\mathcal{M}}} X^{\mathcal{I}(\mathsf{e})}$

CKR model

CKR model $\mathfrak{I} \models \mathfrak{K}$

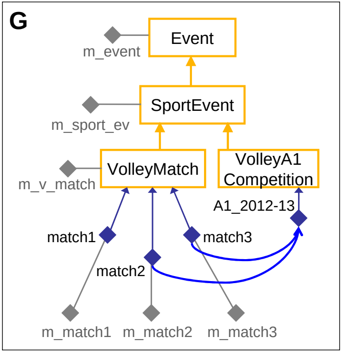
$\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ is a CKR model of $\mathfrak{K}$ if:

- for $\alpha \in \mathcal{L}_\Sigma \cup \mathcal{L}_\Gamma$ in $\mathfrak{G}$, $\mathcal{M} \models \alpha$
- for $\langle x, y \rangle \in \mathsf{mod}^{\mathcal{M}}$ with $y = \mathsf{m}^{\mathcal{M}}$, $\mathcal{I}(x) \models \mathrm{K}_\mathsf{m}$
- for $\alpha \in \mathfrak{G} \cap \mathcal{L}_\Sigma$ and $x \in \mathsf{Ctx}^{\mathcal{M}}$, $\mathcal{I}(x) \models \alpha$

Tourism example: semantics

Suppose we have $\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ s.t. $\mathfrak{I} \models \mathfrak{K}_{tour}$.

For each match matchN, its KB is:
$K(\text{matchN}^{\mathcal{M}}) = K_{event} \cup K_{sport\_ev} \cup K_{v\_match} \cup K_{matchN}$



K_{match1}	Winner(bre_banca_cuneo_volley) ...
K_{match2}	Winner(casa_modena_volley) ...
K_{match3}	Winner(itas_trentino_volley) ...

...

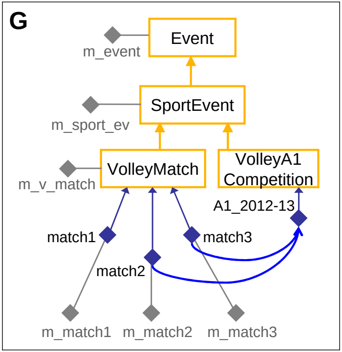
Suppose we have $\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ s.t. $\mathfrak{I} \models \mathfrak{K}_{tour}$.

For each match matchN, its KB is:

$\mathrm{K}(\mathrm{matchN}^{\mathcal{M}}) = \mathrm{K}_{\mathrm{event}} \cup \mathrm{K}_{\mathrm{sport\_ev}} \cup \mathrm{K}_{\mathrm{v\_match}} \cup \mathrm{K}_{\mathrm{matchN}}$

$\mathrm{VolleyMatch} \sqcap \exists \mathrm{hasParentEvent.VolleyA1Competition} = \mathrm{TopMatch}$

$eval(Winner, \mathrm{TopMatch}) \sqsubseteq TopTeam \in \mathrm{K}_{\mathrm{sport\_ev}}$



$\mathrm{K}_{\mathbf{match1}}$	Winner(bre_banca_cuneo_volley) ...
$\mathrm{K}_{\mathbf{match2}}$	Winner(casa_modena_volley) ...
$\mathrm{K}_{\mathbf{match3}}$	Winner(itas_trentino_volley) ...

...

Tourism example: semantics

Suppose we have $\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ s.t. $\mathfrak{I} \models \mathfrak{K}_{tour}$.

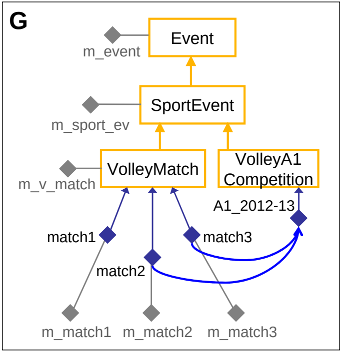
For each match matchN, its KB is:

$\mathrm{K}(\mathrm{matchN}^{\mathcal{M}}) = \mathrm{K}_{\mathrm{event}} \cup \mathrm{K}_{\mathrm{sport\_ev}} \cup \mathrm{K}_{\mathrm{v\_match}} \cup \mathrm{K}_{\mathrm{matchN}}$

$\mathrm{VolleyMatch} \sqcap \exists \mathrm{hasParentEvent.VolleyA1Competition} = \mathrm{TopMatch}$

$eval(Winner, \mathrm{TopMatch}) \sqsubseteq TopTeam \in \mathrm{K}_{\mathrm{sport\_ev}}$

$eval(Winner, \mathrm{TopMatch})^{\mathcal{I}(\mathrm{matchN})} \subseteq TopTeam^{\mathcal{I}(\mathrm{matchN})}$



$\mathrm{K}_{\mathrm{match1}}$	Winner(bre_banca_cuneo_volley) ...
$\mathrm{K}_{\mathrm{match2}}$	Winner(casa_modena_volley) ...
$\mathrm{K}_{\mathrm{match3}}$	Winner(itas_trentino_volley) ...

...

Tourism example: semantics

Suppose we have $\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ s.t. $\mathfrak{I} \models \mathfrak{K}_{tour}$.

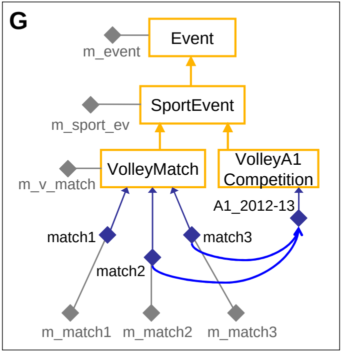
For each match matchN, its KB is:

$K(\text{matchN}^{\mathcal{M}}) = K_{\text{event}} \cup K_{\text{sport\_ev}} \cup K_{\text{v\_match}} \cup K_{\text{matchN}}$

$\text{VolleyMatch} \sqcap \exists \text{hasParentEvent.VolleyA1Competition} = \text{TopMatch}$

$eval(Winner, \text{TopMatch}) \sqsubseteq TopTeam \in K_{\text{sport\_ev}}$

$$\bigcup_{e \in \text{TopMatch}^{\mathcal{M}}} Winner^{\mathcal{I}(e)} \subseteq TopTeam^{\mathcal{I}(\text{matchN})}$$



K_{match1}	Winner(bre_banca_cuneo_volley) ...
K_{match2}	Winner(casa_modena_volley) ...
K_{match3}	Winner(itas_trentino_volley) ...

...

Tourism example: semantics

Suppose we have $\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ s.t. $\mathfrak{I} \models \mathfrak{K}_{tour}$.

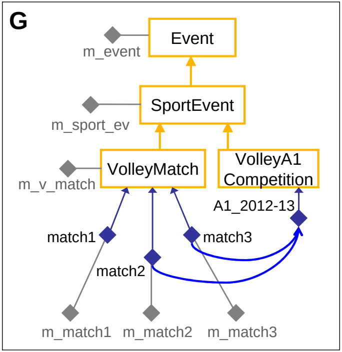
For each match matchN, its KB is:

$\mathrm{K}(\mathsf{matchN}^{\mathcal{M}}) = \mathrm{K}_{\mathsf{event}} \cup \mathrm{K}_{\mathsf{sport\_ev}} \cup \mathrm{K}_{\mathsf{v\_match}} \cup \mathrm{K}_{\mathsf{matchN}}$

$\mathsf{VolleyMatch} \sqcap \exists \mathsf{hasParentEvent}.\mathsf{VolleyA1Competition} = \mathsf{TopMatch}$

$eval(Winner, \mathsf{TopMatch}) \sqsubseteq TopTeam \in \mathrm{K}_{\mathsf{sport\_ev}}$

$$\bigcup_{\mathsf{e} \in \{\mathsf{match\_2}, \mathsf{match\_3}\}} Winner^{\mathcal{I}(\mathsf{e})} \subseteq TopTeam^{\mathcal{I}(\mathsf{matchN})}$$



$\mathrm{K}_{\mathbf{match1}}$	Winner(bre_banca_cuneo_volley) ...
$\mathrm{K}_{\mathbf{match2}}$	Winner(casa_modena_volley) ...
$\mathrm{K}_{\mathbf{match3}}$	Winner(itas_trentino_volley) ...

...

Tourism example: semantics

Suppose we have $\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ s.t. $\mathfrak{I} \models \mathfrak{K}_{tour}$.

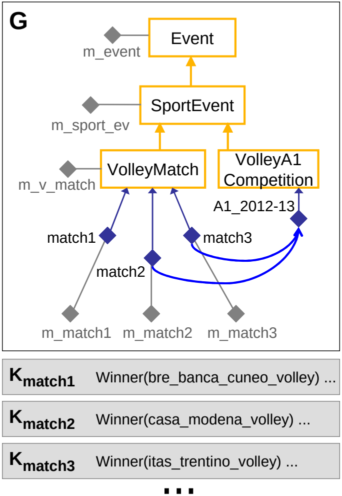
For each match matchN, its KB is:

$\mathrm{K}(\mathrm{matchN}^{\mathcal{M}}) = \mathrm{K}_{\mathrm{event}} \cup \mathrm{K}_{\mathrm{sport\_ev}} \cup \mathrm{K}_{\mathrm{v\_match}} \cup \mathrm{K}_{\mathrm{matchN}}$

$\mathrm{VolleyMatch} \sqcap \exists \mathrm{hasParentEvent.VolleyA1Competition} = \mathrm{TopMatch}$

$eval(Winner, \mathrm{TopMatch}) \sqsubseteq TopTeam \in \mathrm{K}_{\mathrm{sport\_ev}}$

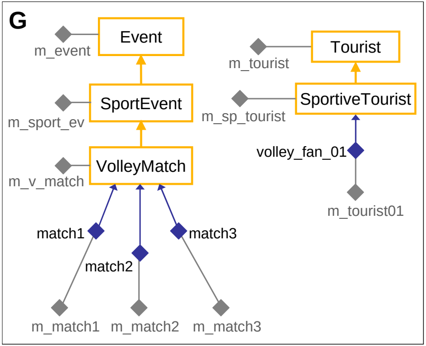
$\{itas\_trentino, casa\_modena\} \subseteq TopTeam^{\mathcal{I}(\mathrm{matchN})}$



Tourism example: semantics

Suppose we have $\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ s.t. $\mathfrak{I} \models \mathfrak{K}_{tour}$.

For the context of volley_fan:
$K(\text{volley\_fan}^{\mathcal{M}}) = K_{\text{tourist}} \cup K_{\text{sp\_tourist}} \cup K_{\text{tourist01}}$



K$_{\text{match1}}$	Winner(bre_banca_cuneo_volley) ...
K$_{\text{match2}}$	Winner(casa_modena_volley) ...
K$_{\text{match3}}$	Winner(itas_trentino_volley) ...

...

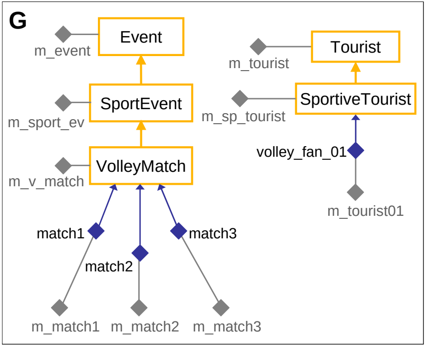
Tourism example: semantics

Suppose we have $\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ s.t. $\mathfrak{I} \models \mathfrak{K}_{tour}$.

For the context of volley_fan:

$\mathrm{K}(\text{volley\_fan}^{\mathcal{M}}) = \mathrm{K}_{\text{tourist}} \cup \mathrm{K}_{\text{sp\_tourist}} \cup \mathrm{K}_{\text{tourist01}}$

$eval(TopTeam, \text{SportEvent}) \sqsubseteq PreferredTeam$
$\in \mathrm{K}_{\text{sp\_tourist}}$



$\mathrm{K}_{\mathbf{match1}}$	Winner(bre_banca_cuneo_volley) ...
$\mathrm{K}_{\mathbf{match2}}$	Winner(casa_modena_volley) ...
$\mathrm{K}_{\mathbf{match3}}$	Winner(itas_trentino_volley) ...

...

Tourism example: semantics

Suppose we have $\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ s.t. $\mathfrak{I} \models \mathfrak{K}_{tour}$.

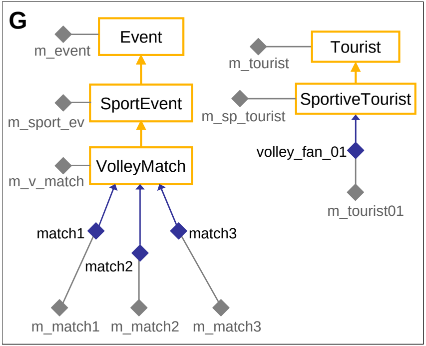
For the context of volley_fan:

$\mathrm{K}(\text{volley\_fan}^{\mathcal{M}}) = \mathrm{K}_{\text{tourist}} \cup \mathrm{K}_{\text{sp\_tourist}} \cup \mathrm{K}_{\text{tourist01}}$

$eval(TopTeam, \text{SportEvent}) \sqsubseteq PreferredTeam \in \mathrm{K}_{\text{sp\_tourist}}$

$$eval(TopTeam, \text{SportEvent})^{\mathcal{I}(\text{volley\_fan})}$$

$$\subseteq PreferredTeam^{\mathcal{I}(\text{volley\_fan})}$$



$\mathbf{K}_{\mathbf{match1}}$	Winner(bre_banca_cuneo_volley) ...
$\mathbf{K}_{\mathbf{match2}}$	Winner(casa_modena_volley) ...
$\mathbf{K}_{\mathbf{match3}}$	Winner(itas_trentino_volley) ...

...

Tourism example: semantics

Suppose we have $\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ s.t. $\mathfrak{I} \models \mathfrak{K}_{tour}$.

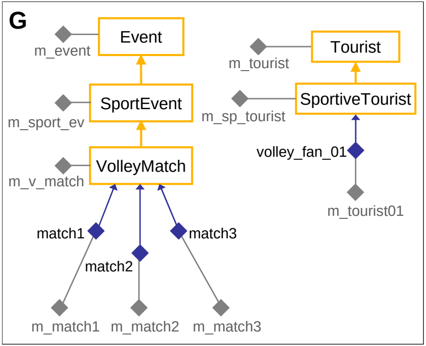
For the context of volley_fan:

$\mathrm{K}(\text{volley\_fan}^{\mathcal{M}}) = \mathrm{K}_{\text{tourist}} \cup \mathrm{K}_{\text{sp\_tourist}} \cup \mathrm{K}_{\text{tourist01}}$

$eval(TopTeam, \text{SportEvent}) \sqsubseteq PreferredTeam \in \mathrm{K}_{\text{sp\_tourist}}$

$$\bigcup_{\text{e} \in \text{SportEvent}^{\mathcal{M}}} TopTeam^{\mathcal{I}(\text{e})}$$

$$\subseteq PreferredTeam^{\mathcal{I}(\text{volley\_fan})}$$



K$_{\textbf{match1}}$	Winner(bre_banca_cuneo_volley) ...
K$_{\textbf{match2}}$	Winner(casa_modena_volley) ...
K$_{\textbf{match3}}$	Winner(itas_trentino_volley) ...

...

Tourism example: semantics

Suppose we have $\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ s.t. $\mathfrak{I} \models \mathfrak{K}_{tour}$.

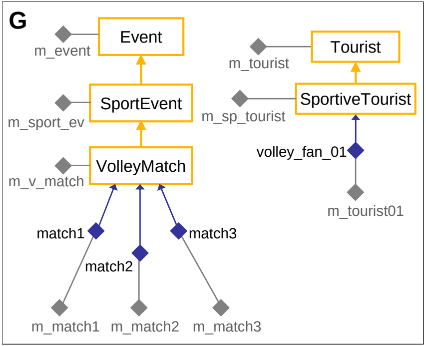
For the context of volley_fan:

$K(\text{volley\_fan}^{\mathcal{M}}) = K_{\text{tourist}} \cup K_{\text{sp\_tourist}} \cup K_{\text{tourist01}}$

$eval(TopTeam, \text{SportEvent}) \sqsubseteq PreferredTeam \in K_{\text{sp\_tourist}}$

$$\bigcup_{e \in \{\text{match\_1}, \text{match\_2}, \text{match\_3}\}} TopTeam^{\mathcal{I}(e)}$$

$$\subseteq PreferredTeam^{\mathcal{I}(\text{volley\_fan})}$$



K_{match1}	Winner(bre_banca_cuneo_volley) ...
K_{match2}	Winner(casa_modena_volley) ...
K_{match3}	Winner(itas_trentino_volley) ...

...

Tourism example: semantics

Suppose we have $\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ s.t. $\mathfrak{I} \models \mathfrak{K}_{tour}$.

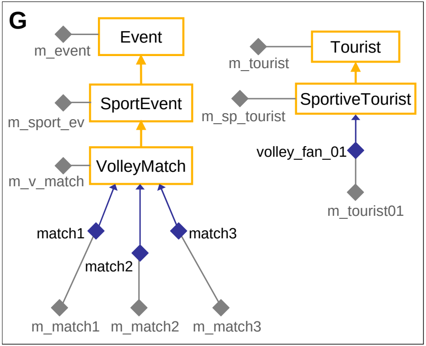
For the context of volley_fan:

$\mathrm{K}(\text{volley\_fan}^{\mathcal{M}}) = \mathrm{K}_{\text{tourist}} \cup \mathrm{K}_{\text{sp\_tourist}} \cup \mathrm{K}_{\text{tourist01}}$

$eval(TopTeam, \text{SportEvent}) \sqsubseteq PreferredTeam \in \mathrm{K}_{\text{sp\_tourist}}$

$$\{itas\_trentino, casa\_modena\}$$

$$\subseteq PreferredTeam^{\mathcal{I}(\text{volley\_fan})}$$



$\mathbf{K}_{\mathbf{match1}}$	Winner(bre_banca_cuneo_volley) ...
$\mathbf{K}_{\mathbf{match2}}$	Winner(casa_modena_volley) ...
$\mathbf{K}_{\mathbf{match3}}$	Winner(itas_trentino_volley) ...

...

CKR model

Summary:

- Two-layered DL knowledge base
- General context structure (extending [Serafini and Homola, 2012])
- *eval* operator: knowledge propagation across contexts
- Model theoretic DL semantics

Overview

Overview

Reasoning tasks

Satisfiability

Instance query answering

Boolean conjunctive query answering

Reasoning tasks

Satisfiability

- Does a given CKR $\mathfrak{K}$ have some CKR model?

Instance query answering

Boolean conjunctive query answering

Reasoning tasks

Satisfiability

- Does a given CKR $\mathfrak{K}$ have some CKR model?

Instance query answering

- Given a CKR $\mathfrak{K}$, an assertion α, a context c of $\mathfrak{K}$
- Does $\mathfrak{K}$ entail α at c (denoted $\mathfrak{K} \models \mathrm{c} : \alpha$), i.e., does $\mathcal{I}(\mathrm{c}^{\mathcal{M}}) \models \alpha$ hold for every CKR model $\mathfrak{I}$ of $\mathfrak{K}$?

Boolean conjunctive query answering

Reasoning tasks

Satisfiability

- Does a given CKR $\mathfrak{K}$ have some CKR model?

Instance query answering

- Given a CKR $\mathfrak{K}$, an assertion α, a context c of $\mathfrak{K}$
- Does $\mathfrak{K}$ entail α at c (denoted $\mathfrak{K} \models \mathsf{c} : \alpha$), i.e., does $\mathcal{I}(\mathsf{c}^{\mathcal{M}}) \models \alpha$ hold for every CKR model $\mathfrak{I}$ of $\mathfrak{K}$?

Boolean conjunctive query answering

- Given a CKR $\mathfrak{K}$ and a formula $q = \exists \mathbf{y} \gamma(\mathbf{y})$, where $\gamma(\mathbf{y}) = \mathsf{c}_1 : \alpha_1, \ldots, \mathsf{c}_n : \alpha_n$, the c_i are contexts and the α_i atoms that may contain variables
- Does $\mathfrak{K}$ entail q (denoted $\mathfrak{K} \models q$), i.e., does for every CKR model $\mathfrak{I}$ of $\mathfrak{K}$, some variable assignment σ to $\mathbf{y}$ exists s.t. $\mathcal{I}(\mathsf{c}_i^{\mathcal{I}}), \sigma \models \alpha_i$ for every i?

Materialization calculus:

- Calculus for instance checking in OWL RL CKR
- Extension to the CKR structure of materialization calculus for OWL EL of [Krötzsch, 2010]
- Formalizes the operation of forward closure in implementation

Idea

Composed by 3 kinds of rule sets:

- Input rules I: translation of DL axioms to datalog atoms
- Deduction rules P: forward inference rules
- Output rules O: translation for DL proved ABox assertion

Translation rules

Input rules I

Deduction rules P

Output rules O

Translation rules

Input rules I

I_{rl}: $\mathcal{SROIQ}$-RL input rules

$c : A(a) \Rightarrow \{\mathtt{inst}(a, A, c)\}$ $\quad$ $c : A \sqsubseteq B \Rightarrow \{\mathtt{subClass}(A, B, c)\}$

Deduction rules P

Output rules O

Translation rules

Input rules I

I_{rl}: $\mathcal{SROIQ}$-RL input rules

$c : A(a) \Rightarrow \{\mathtt{inst}(a, A, c)\} \quad c : A \sqsubseteq B \Rightarrow \{\mathtt{subClass}(A, B, c)\}$

Deduction rules P

P_{rl}: $\mathcal{SROIQ}$-RL deduction rules

$\mathtt{subClass}(y, z, c), \mathtt{inst}(x, y, c) \rightarrow \mathtt{inst}(x, z, c)$

Output rules O

Translation rules

Input rules I

I_{rl}: $\mathcal{SROIQ}$-RL input rules

$c : A(a) \Rightarrow \{\mathtt{inst}(a, A, c)\}$ $\quad$ $c : A \sqsubseteq B \Rightarrow \{\mathtt{subClass}(A, B, c)\}$

I_{glob}: Global input rules

$\mathsf{c} \in \mathbf{N} \Rightarrow \{\mathtt{inst}(\mathsf{c}, \mathsf{Ctx}, \mathsf{gm})\}$ $\quad$ $\mathsf{C} \in \mathbf{C} \Rightarrow \{\mathtt{subClass}(\mathsf{C}, \mathsf{Ctx}, \mathsf{gm})\}$

Deduction rules P

P_{rl}: $\mathcal{SROIQ}$-RL deduction rules

$\mathtt{subClass}(y, z, c), \mathtt{inst}(x, y, c) \rightarrow \mathtt{inst}(x, z, c)$

Output rules O

Translation rules

Input rules I

I_{rl}: $\mathcal{SROIQ}$-RL input rules

$c : A(a) \Rightarrow \{\texttt{inst}(a, A, c)\}$ $\quad$ $c : A \sqsubseteq B \Rightarrow \{\texttt{subClass}(A, B, c)\}$

I_{glob}: Global input rules

$\mathsf{c} \in \mathbf{N} \Rightarrow \{\texttt{inst}(\mathsf{c}, \mathsf{Ctx}, \mathsf{gm})\}$ $\quad$ $\mathsf{C} \in \mathbf{C} \Rightarrow \{\texttt{subClass}(\mathsf{C}, \mathsf{Ctx}, \mathsf{gm})\}$

I_{loc}: Local input rules

$c : eval(A, \mathsf{C}) \sqsubseteq B \Rightarrow \{\texttt{subEval}(A, \mathsf{C}, B, c)\}$

Deduction rules P

P_{rl}: $\mathcal{SROIQ}$-RL deduction rules

$\texttt{subClass}(y, z, c), \texttt{inst}(x, y, c) \rightarrow \texttt{inst}(x, z, c)$

P_{loc}: Local deduction rules

$\texttt{subEval}(a, c_1, b, c), \texttt{inst}(c', c_1, \mathsf{gm}), \texttt{inst}(x, a, c') \rightarrow \texttt{inst}(x, b, c)$

Output rules O

Translation rules

Input rules I

I_{rl}: $\mathcal{SROIQ}$-RL input rules

$c : A(a) \Rightarrow \{\mathtt{inst}(a, A, c)\}$ $\quad$ $c : A \sqsubseteq B \Rightarrow \{\mathtt{subClass}(A, B, c)\}$

I_{glob}: Global input rules

$\mathsf{c} \in \mathbf{N} \Rightarrow \{\mathtt{inst}(\mathsf{c}, \mathsf{Ctx}, \mathsf{gm})\}$ $\quad$ $\mathsf{C} \in \mathbf{C} \Rightarrow \{\mathtt{subClass}(\mathsf{C}, \mathsf{Ctx}, \mathsf{gm})\}$

I_{loc}: Local input rules

$c : eval(A, \mathsf{C}) \sqsubseteq B \Rightarrow \{\mathtt{subEval}(A, \mathsf{C}, B, c)\}$

Deduction rules P

P_{rl}: $\mathcal{SROIQ}$-RL deduction rules

$\mathtt{subClass}(y, z, c), \mathtt{inst}(x, y, c) \rightarrow \mathtt{inst}(x, z, c)$

P_{loc}: Local deduction rules

$\mathtt{subEval}(a, c_1, b, c), \mathtt{inst}(c', c_1, \mathsf{gm}), \mathtt{inst}(x, a, c') \rightarrow \mathtt{inst}(x, b, c)$

Output rules O

$\{\mathtt{inst}(a, A, \mathsf{c})\} \Rightarrow \mathsf{c} : A(a)$ $\quad$ $\{\mathtt{triple}(a, R, b, \mathsf{c})\} \Rightarrow \mathsf{c} : R(a, b)$

Translation process

1. Global program $PG(\mathfrak{G})$: translation for global context

Translation process

1. Global program $PG(\mathfrak{G})$: translation for global context
2. Computation of local knowledge bases K_{c} for each context c in $\mathfrak{G}$

Translation process

1. Global program $PG(\mathfrak{G})$: translation for global context
2. Computation of local knowledge bases $\mathrm{K_c}$ for each context c in $\mathfrak{G}$
3. Local programs $PC(\mathrm{c})$: translation for local contexts

Translation process

1. Global program $PG(\mathfrak{G})$: translation for global context
2. Computation of local knowledge bases $\mathrm{K_c}$ for each context c in $\mathfrak{G}$
3. Local programs $PC(\mathrm{c})$: translation for local contexts
4. CKR program $PK(\mathfrak{K})$: union of global and local programs

Query answering by translation

- Consider CKR $\mathfrak{K}$ where the axioms are in a normal form
- Needed for universal encoding: e.g., $A_1 \sqcap A_2 \sqcap \cdots \sqcap A_n \sqsubseteq B$

Translation completeness

1. $\mathfrak{K} \models \mathrm{c} : \alpha$ iff $PK(\mathfrak{K}) \models O(\alpha, \mathrm{c})$ (axiom α in context c)
2. $\mathfrak{K} \models \exists \mathbf{y} \gamma(\mathbf{y})$ iff $PK(\mathfrak{K}) \models O(\exists \mathbf{y} \gamma(\mathbf{y}))$ (boolean conjunctive queries)

Reasoning

Summary:

- Instance checking procedure for CKRs in OWL RL
- Calculus based on a translation to datalog
- Formalizes forward closure in implementation

Overview

Overview

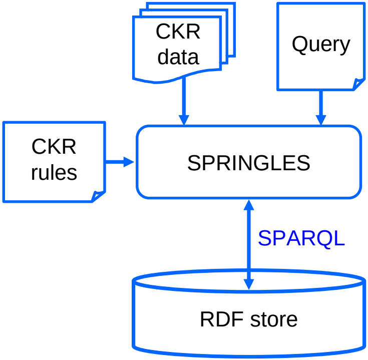
SPRINGLES: implementation on SPARQL

Semantic Web languages

- RDF: representation for data
- OWL: representation for schema
- SPARQL: query on RDF data

CKR implementation

- Contexts as OWL/RDF repositories
- Reasoning rules as SPARQL queries



SPRINGLES

CKR implementation on top of SPRINGLES:
SParql-based Rule Inference over Named Graphs Layer Extending Sesame

SPRINGLES features:

- transparent/on-demand closure materialization based on rules
- rules encoded as SPARQL queries on Named Graphs (NG)
- customizable rule evaluation strategy

Why SPRINGLES:

- no inference over NGs in RDF stores

Why SPARQL:

- exploits optimized query engines
- can scale to large KBs (cf. RETE)

SPRINGLES rules and evaluation strategy

SPRINGLES rule

Forward SPARQL-based rules of the form:

$$
\begin{array}{l}
:<\mathtt{rule-name}>\ \mathtt{a\ spr:Rule;}\\
\quad \mathtt{spr:head}\ """<\mathtt{graphpattern}>""";\\
\quad \mathtt{spr:body}\ """<\mathtt{sparqlquery}>""".
\end{array}
$$

SPRINGLES evaluation strategy

Composition of SPRINGLES primitives:

- parallel rule evaluation
- sequence
- fixpoint
- repeat

CKR ruleset and evaluation strategy

Ruleset

Translation to SPRINGLES rules of materialization calculus rules:

```
:pel-c-subc a spr:Rule ;
  spr:head """ GRAPH ?mx { ?x rdf:type ?z } """ ;
  spr:body """ GRAPH ?m1 { ?y rdfs:subClassOf ?z }
               GRAPH ?m2 { ?x rdf:type ?y }
               GRAPH sys:dep { ?mx sys:derivedFrom ?m1,?m2 }
               FILTER NOT EXISTS {
                 GRAPH ?m0 { ?x rdf:type ?z }
                 GRAPH sys:dep { ?mx sys:derivedFrom ?m0 }
               } """ .
```

Evaluation strategy

- Associate inferred graph to `ckr:global`
- By fixpoint, compute OWL RL and global closure on `ckr:global`
- Compute modules associated to each context
- Create local graphs for contexts and for inference
- Evaluate local rules for OWL RL on context graphs

Implementation on RDF

Current CKR implementations:

- CKR prototype:
  1st implementation on Sesame/OWLIM [Tamilin et al., 2010]
- CKR on SPRINGLES: SPARQL-based forward rules on named graphs over Sesame [Bozzato and Serafini, 2013]
- CKRew: CKR datalog rewriter [Bozzato et al., 2018a]
- CKR on RDFpro:
  SPARQL rules for RDF processor [Schuetz et al., 2020]

Evaluation

Findings [Bozzato et al., 2013, Bozzato and Serafini, 2014]

- Modelling:
  - Language: CKR model reduce redundancy, easier references
  - Model: CKR uses less symbols than Flat modelling
  - Query: CKR performs better on context based queries
- Reasoning:
  - Scalability: influenced by expressivity and number of contexts
  - Propagation: CKR connections outperform flat replication

Overview

1. CKR model
2. Reasoning
3. Implementation on RDF
4. Defeasible axioms
5. Contextual hierarchies

Overview

Need for defeasibility in contexts

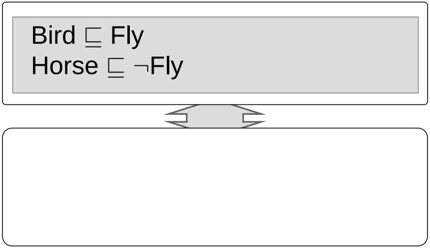
CKR structure: two layers

- Global context:
  Structure of contexts and object knowledge shared by all contexts
- (Local) contexts:
  Local object knowledge (with references)

Need for defeasibility in contexts

CKR structure: two layers

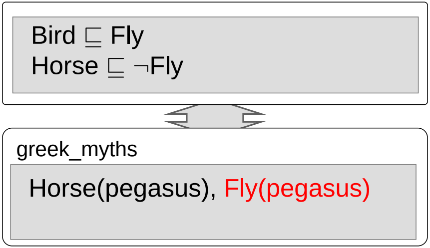
- Global context:
  Structure of contexts and object knowledge shared by all contexts
- (Local) contexts:
  Local object knowledge (with references)



Need for defeasibility in contexts

CKR structure: two layers

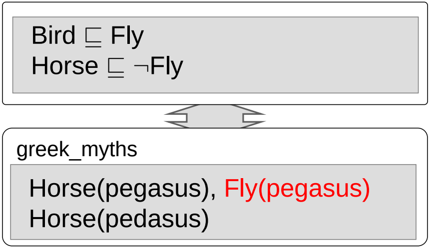
- Global context:
  Structure of contexts and object knowledge shared by all contexts
- (Local) contexts:
  Local object knowledge (with references)



Need for defeasibility in contexts

CKR structure: two layers

- Global context:
  Structure of contexts and object knowledge shared by all contexts
- (Local) contexts:
  Local object knowledge (with references)



Need for defeasibility in contexts

CKR structure: two layers

- Global context:
  Structure of contexts and object knowledge shared by all contexts
- (Local) contexts:
  Local object knowledge (with references)

$\mathrm{Bird} \sqsubseteq \mathrm{Fly}$
$\mathrm{Horse} \sqsubseteq \neg \mathrm{Fly}$

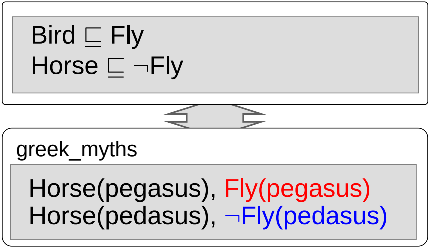
greek_myths

Horse(pegasus), Fly(pegasus)
Horse(pedasus), ¬Fly(pedasus)

Need for defeasibility in contexts

CKR structure: two layers

- Global context:
  Structure of contexts and object knowledge shared by all contexts
- (Local) contexts:
  Local object knowledge (with references)



➔ We want to specify that certain global axioms are defeasible:
they hold globally, but allow exceptional instances in local contexts

Proposal: CKR extension for defeasibility

CKR extension for defeasibility:
AI Journal (257):72-126, 2018 [Bozzato et al., 2018a]

- Syntax and semantics of an extension of CKR with defeasible axioms in global context
- Define reasoning problems:
  - extended CKR satisfiability
  - CKR axiom entailment $\mathfrak{K} \models \mathsf{c} : \alpha$
    $\alpha = \mathrm{Fly}(\mathrm{pegasus})$
  - CKR conjunctive query answering $\mathfrak{K} \models \exists \mathbf{y} \gamma(\mathbf{y})$
    $\gamma(\mathbf{y}) = \mathrm{greek\_myths} : \mathrm{Horse}(\mathrm{pegasus}), \mathrm{hasFeature}(\mathrm{pegasus}, \mathrm{y}), \mathrm{Wing}(\mathrm{y})$
- Characterize their computational cost (complexity)
- Extend datalog translation for OWL RL based CKR with rules for the translation of defeasible axioms
- Prototype implementation for CKR datalog rewriter

Notable aspects

Interesting points of our work:

- Expressive means for defeasibility on structured KBs in DL
  - defeasibility in contextual systems
  - non-monotonic reasoning in DLs
- Reason by cases: conflicts in overridings not ruled by “preference”
- Inheritance of properties: no “exceptional” elements
- Translation to datalog extends monotonic materialization calculus

Syntax: defeasible axioms

➔ We extend the type of axioms appearing in global object knowledge:

Defeasible axiom α of $\mathfrak{G}$: $\mathrm{D}(\alpha) \in \mathfrak{G}$ for $\alpha \in \mathcal{L}_\Sigma$

"α propagates to local contexts, but admits exceptional instances"

D(Cheap $\sqsubseteq$ Interesting)
Cheap(fbmatch), Cheap(market)

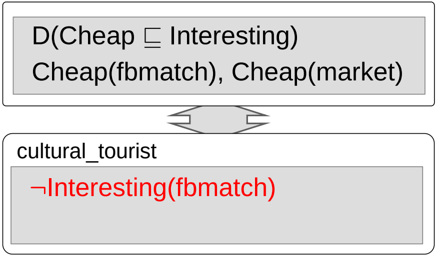
DL language $\mathcal{L}^{\mathrm{D}}_{\Sigma}$ $\mathcal{L}_\Sigma$ with defeasibile axioms

Syntax: defeasible axioms

➔ We extend the type of axioms appearing in global object knowledge:

Defeasible axiom α of $\mathfrak{G}$: $\mathrm{D}(\alpha) \in \mathfrak{G}$ for $\alpha \in \mathcal{L}_\Sigma$

"α propagates to local contexts, but admits exceptional instances"



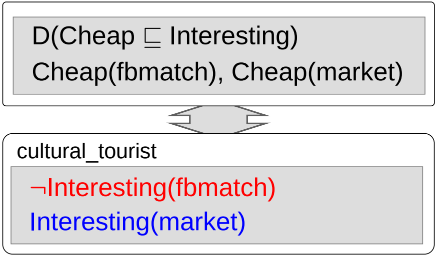
DL language $\mathcal{L}^{\mathrm{D}}_{\Sigma}$ $\mathcal{L}_\Sigma$ with defeasibile axioms

Syntax: defeasible axioms

➔ We extend the type of axioms appearing in global object knowledge:

Defeasible axiom α of $\mathfrak{G}$: $\mathrm{D}(\alpha) \in \mathfrak{G}$ for $\alpha \in \mathcal{L}_\Sigma$

"α propagates to local contexts, but admits exceptional instances"



DL language $\mathcal{L}^{\mathrm{D}}_\Sigma$ $\mathcal{L}_\Sigma$ with defeasibile axioms

Semantics: clashing assumptions

Idea

- Exception of axiom instances modelled as clashing assumptions $\langle \alpha, \mathbf{e} \rangle$
  "In context c, ignore instance $\mathbf{e}$ in evaluation of α"

Semantics: clashing assumptions

Idea

- Exception of axiom instances modelled as clashing assumptions $\langle \alpha, \mathbf{e} \rangle$
  "In context c, ignore instance $\mathbf{e}$ in evaluation of α" $\langle (\mathit{Cheap} \sqsubseteq \mathit{Interesting}), \mathit{fbmatch} \rangle$

Semantics: clashing assumptions

Idea

- Exception of axiom instances modelled as clashing assumptions $\langle \alpha, \mathbf{e} \rangle$
  "In context c, ignore instance $\mathbf{e}$ in evaluation of α" $\langle (Cheap \sqsubseteq Interesting), fbmatch \rangle$

- Clashing assumption $\langle \alpha, \mathbf{e} \rangle$:
  assumption that $\mathbf{e}$ is exceptional for α
- CAS-interpretation $\mathfrak{I}_{CAS} = \langle \mathcal{M}, \mathcal{I}, \chi \rangle$:
  $\chi(c)$: set of clashing assumptions of context c

Semantics: clashing assumptions

Idea

- Exception of axiom instances modelled as clashing assumptions $\langle \alpha, \mathbf{e} \rangle$
  "In context c, ignore instance $\mathbf{e}$ in evaluation of α" $\langle (Cheap \sqsubseteq Interesting), fbmatch \rangle$

- Clashing assumption $\langle \alpha, \mathbf{e} \rangle$:
  assumption that $\mathbf{e}$ is exceptional for α
- CAS-interpretation $\mathfrak{I}_{CAS} = \langle \mathcal{M}, \mathcal{I}, \chi \rangle$:
  $\chi(c)$: set of clashing assumptions of context c

CAS-model $\mathfrak{I}_{CAS} \models \mathfrak{K}$

$\mathfrak{I}_{CAS}$ is a CAS-model for $\mathfrak{K}$ if:

- $\mathcal{M} \models \alpha$, for every $\alpha \in \mathfrak{G}$ strict or defeasible

Semantics: clashing assumptions

Idea

- Exception of axiom instances modelled as clashing assumptions $\langle \alpha, \mathbf{e} \rangle$
  "In context c, ignore instance $\mathbf{e}$ in evaluation of α" $\langle (Cheap \sqsubseteq Interesting), fbmatch \rangle$

- Clashing assumption $\langle \alpha, \mathbf{e} \rangle$:
  assumption that $\mathbf{e}$ is exceptional for α
- CAS-interpretation $\mathfrak{I}_{CAS} = \langle \mathcal{M}, \mathcal{I}, \chi \rangle$:
  $\chi(c)$: set of clashing assumptions of context c

CAS-model $\mathfrak{I}_{CAS} \models \mathfrak{K}$

$\mathfrak{I}_{CAS}$ is a CAS-model for $\mathfrak{K}$ if:

- $\mathcal{M} \models \alpha$, for every $\alpha \in \mathfrak{G}$ strict or defeasible
- $\mathcal{I}(x) \models \mathrm{K}_{\mathrm{m}}$, if m is a module of context x
- $\mathcal{I}(x) \models \alpha$, for every $\alpha \in \mathfrak{G}$ strict

Semantics: clashing assumptions

Idea

- Exception of axiom instances modelled as clashing assumptions $\langle \alpha, \mathbf{e} \rangle$
  "In context c, ignore instance $\mathbf{e}$ in evaluation of α" $\langle (Cheap \sqsubseteq Interesting), fbmatch \rangle$

- Clashing assumption $\langle \alpha, \mathbf{e} \rangle$:
  assumption that $\mathbf{e}$ is exceptional for α
- CAS-interpretation $\mathfrak{I}_{CAS} = \langle \mathcal{M}, \mathcal{I}, \chi \rangle$:
  $\chi(c)$: set of clashing assumptions of context c

CAS-model $\mathfrak{I}_{CAS} \models \mathfrak{K}$

$\mathfrak{I}_{CAS}$ is a CAS-model for $\mathfrak{K}$ if:

- $\mathcal{M} \models \alpha$, for every $\alpha \in \mathfrak{G}$ strict or defeasible
- $\mathcal{I}(x) \models \mathrm{K_m}$, if m is a module of context x
- $\mathcal{I}(x) \models \alpha$, for every $\alpha \in \mathfrak{G}$ strict
- for every $\mathrm{D}(\alpha) \in \mathfrak{G}$, if $\langle \alpha, \mathbf{e} \rangle \notin \chi(x)$, then $\mathcal{I}(x) \models \alpha(\mathbf{e})$

Semantics: justification

Idea

- Assumptions must be justified by local assertions in a clashing set S
  "In context c, $\alpha(\mathbf{e}) \cup S$ is unsatisfiable"

Semantics: justification

Idea

- Assumptions must be justified by local assertions in a clashing set S
  "In context c, $\alpha(\mathbf{e}) \cup S$ is unsatisfiable" $\{Cheap(fbmatch), \neg Interesting(fbmatch)\}$

Semantics: justification

Idea

- Assumptions must be justified by local assertions in a clashing set S
  "In context c, $\alpha(\mathbf{e}) \cup S$ is unsatisfiable" $\{Cheap(fbmatch), \neg Interesting(fbmatch)\}$

Justification

$\mathfrak{I}_\chi = \langle \mathcal{M}, \mathcal{I}, \chi \rangle$ model of $\mathfrak{K}$ is justified, if for every context $x \in \mathsf{Ctx}^\mathcal{M}$ and clashing assumption $\langle \alpha, \mathbf{e} \rangle \in \chi(x)$,

1. some clashing set $S = S_{\langle \alpha, \mathbf{e} \rangle, x}$ exists such that $\mathcal{I}(x) \models S_{\langle \alpha, \mathbf{e} \rangle, x}$, and
2. for every model $\mathfrak{I}'_\chi = \langle \mathcal{M}', \mathcal{I}', \chi \rangle$ of $\mathfrak{K}$ that is NI-congruent with $\mathfrak{I}_\chi$ (i.e., $c^\mathcal{M} = c^{\mathcal{M}'}$ for every individual name c), $\mathcal{I}'(x) \models S_{\langle \alpha, \mathbf{e} \rangle, x}$

➔ Justified if, for every clashing assumption $\langle \alpha, \mathbf{e} \rangle$,
we have a factual evidence S of its local unsatisfiability

Moreover, this factual evidence is a logical consequence (provable)

Semantics: CKR model

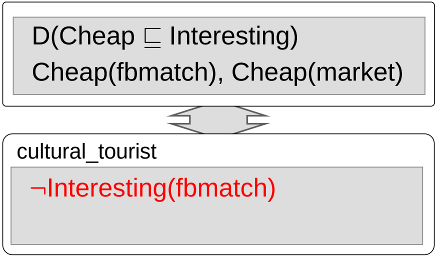
Idea

- CKR models are interpretation where all c. assumptions are justified

CKR model $\mathfrak{I} \models \mathfrak{K}$

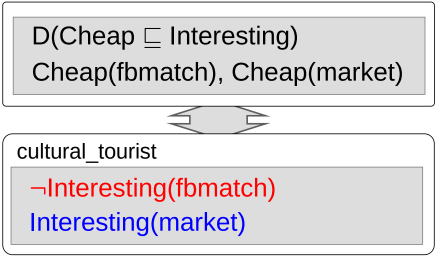
$\mathfrak{I} = \langle \mathcal{M}, \mathcal{I} \rangle$ is a CKR model of $\mathfrak{K}$,
if some $\mathfrak{I}_{CAS} = \langle \mathcal{M}, \mathcal{I}, \chi \rangle$ is a justified CAS-model of $\mathfrak{K}$

Examples



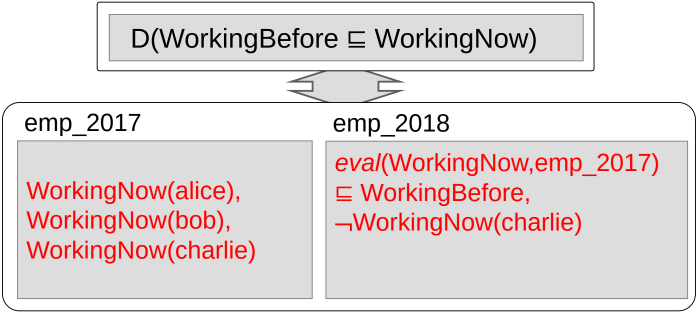
- CAS-model: $\mathfrak{I}_{CAS} = \langle \mathcal{M}, \mathcal{I}, \chi \rangle$
  with $\langle (Cheap \sqsubseteq Interesting), fbmatch \rangle \in \chi(cultural\_tourist)$
- Justification: $S = \{Cheap(fbmatch), \neg Interesting(fbmatch)\}$

Examples



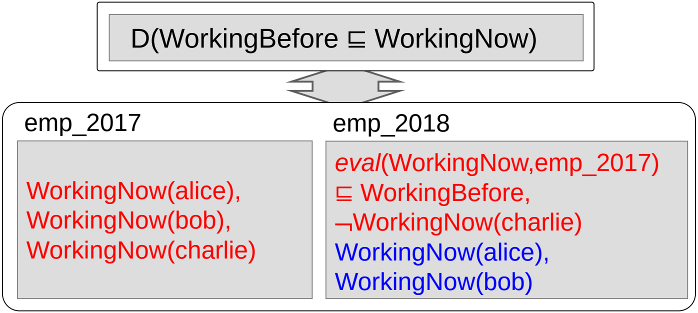
- CAS-model: $\mathfrak{I}_{CAS} = \langle \mathcal{M}, \mathcal{I}, \chi \rangle$
  with $\langle (Cheap \sqsubseteq Interesting), fbmatch \rangle \in \chi(cultural\_tourist)$
- Justification: $S = \{Cheap(fbmatch), \neg Interesting(fbmatch)\}$
- $\mathfrak{I}_{CAS} \not\models Interesting(fbmatch)$ but $\mathfrak{I}_{CAS} \models Interesting(market)$ and $\mathfrak{I}'_{CAS} \models Interesting(market)$ for each $\mathfrak{I}'_{CAS}$ NI-congruent with $\mathfrak{I}_{CAS}$

Examples



- CAS-model: $\mathfrak{I}_{CAS} = \langle \mathcal{M}, \mathcal{I}, \chi \rangle$
  with $\langle (WorkingBefore \sqsubseteq WorkingNow), charlie \rangle \in \chi(emp\_2018)$
- Justification: $S = \{WorkingBefore(charlie), \neg WorkingNow(charlie)\}$

Examples



- CAS-model: $\mathfrak{I}_{CAS} = \langle \mathcal{M}, \mathcal{I}, \chi \rangle$
  with $\langle (WorkingBefore \sqsubseteq WorkingNow), charlie \rangle \in \chi(emp\_2018)$
- Justification: $S = \{WorkingBefore(charlie), \neg WorkingNow(charlie)\}$
- $\mathfrak{I}_{CAS} \not\models WorkingNow(charlie)$ but
  $\mathfrak{I}_{CAS} \models WorkingNow(alice)$ and $\mathfrak{I}_{CAS} \models WorkingNow(bob)$

Reasoning tasks

1. CKR satisfiability (does $\mathfrak{K}$ have a CKR model)
2. CKR axiom entailment $\mathfrak{K} \models \mathrm{c} : \alpha$
3. CKR conjunctive query answering $\mathfrak{K} \models \exists \mathbf{y} \gamma(\mathbf{y})$

Main complexity results

- Deciding whether $\mathfrak{K}$ has some CKR-model is NP-complete
- Deciding $\mathfrak{K} \models \mathrm{c} : \alpha$ is coNP-complete
- Deciding $\mathfrak{K} \models \exists \mathbf{y} \gamma(\mathbf{y})$ is Π_2^p-complete

Extended CKR translation to datalog

Main Idea

- extend the materialization calculus for instance checking in [Bozzato and Serafini, 2013]
- add rules for overriding
- use a fixed set of rules and provide $\mathfrak{K}$ etc as data
- requires a normal form for $\mathfrak{K}$ + slight restrictions on $\mathrm{D}(\alpha)$

Program Structure

Composed by 3 kinds of rule sets:

- Input rules I: translation of DL axioms to Datalog atoms
- Deduction rules P: forward inference rules
- Output rules O: translation for DL proved ABox assertion

➔ I and P, contain "overriding rules" to treat defeasible propagation

Defeasibility rules

I_{D}: Defeasibility input rules (overriding conditions)

$\mathrm{D}(A \sqsubseteq B) \Rightarrow$

$\{\texttt{ovr}(\texttt{subClass}, x, A, B, c) \leftarrow \texttt{ninstd}(x, B, c), \texttt{instd}(x, A, c), \texttt{prec}(c, g).\}$

where $\texttt{ninstd}(x, B, c)$ represents $\neg\texttt{instd}(x, B, c)$

P_{D}: Defeasibility deduction rules (defeasible propagation)

$$\texttt{instd}(x, z, c) \leftarrow \texttt{subClass}(y, z, g), \texttt{instd}(x, y, c), \texttt{prec}(c, g), \texttt{not ovr}(\texttt{subClass}, x, y, z, c).$$

Defeasibility rules

I_D: Defeasibility input rules (overriding conditions)

$\mathrm{D}(A \sqsubseteq B) \Rightarrow$
$\{\texttt{ovr}(\texttt{subClass}, x, A, B, c) \leftarrow \texttt{ninstd}(x, B, c), \texttt{instd}(x, A, c), \texttt{prec}(c, g).\}$
where $\texttt{ninstd}(x, B, c)$ represents $\neg\texttt{instd}(x, B, c)$

P_D: Defeasibility deduction rules (defeasible propagation)

$$\texttt{instd}(x, z, c) \leftarrow \texttt{subClass}(y, z, g), \texttt{instd}(x, y, c), \texttt{prec}(c, g), \texttt{not ovr}(\texttt{subClass}, x, y, z, c).$$

$\mathrm{D}(\mathit{Cheap} \sqsubseteq \mathit{Interesting}) \Rightarrow$
$\{\texttt{ovr}(\texttt{subClass}, x, \mathit{Cheap}, \mathit{Interesting}, c) \leftarrow \texttt{ninstd}(x, \mathit{Interesting}, c), \texttt{instd}(x, \mathit{Cheap}, c), \texttt{prec}(c, g).\}$

Defeasibility rules

I_{D}: Defeasibility input rules (overriding conditions)

$\mathrm{D}(A \sqsubseteq B) \Rightarrow$
$\{\texttt{ovr}(\texttt{subClass}, x, A, B, c) \leftarrow \texttt{ninstd}(x, B, c), \texttt{instd}(x, A, c), \texttt{prec}(c, g).\}$
where $\texttt{ninstd}(x, B, c)$ represents $\neg\texttt{instd}(x, B, c)$

P_{D}: Defeasibility deduction rules (defeasible propagation)

$\texttt{instd}(x, z, c) \leftarrow \texttt{subClass}(y, z, g), \texttt{instd}(x, y, c), \texttt{prec}(c, g),$
$\qquad \texttt{not ovr}(\texttt{subClass}, x, y, z, c).$

$\mathrm{D}(\mathit{Cheap} \sqsubseteq \mathit{Interesting}) \Rightarrow$
$\{\texttt{ovr}(\texttt{subClass}, x, \mathit{Cheap}, \mathit{Interesting}, c) \leftarrow \texttt{ninstd}(x, \mathit{Interesting}, c),$
$\qquad \texttt{instd}(x, \mathit{Cheap}, c), \texttt{prec}(c, g).\}$

➔ $PK(\mathfrak{K}) \models \texttt{ovr}(\texttt{subClass}, \mathit{fbmatch}, \mathit{Cheap}, \mathit{Interesting}, c)$ but
$PK(\mathfrak{K}) \not\models \texttt{ovr}(\texttt{subClass}, \mathit{market}, \mathit{Cheap}, \mathit{Interesting}, c)$ thus
$PK(\mathfrak{K}) \models \texttt{instd}(\mathit{market}, \mathit{Interesting}, c)$

Defeasibility rules: negative literals

Disjunctive information

Negative rule for $A \sqcap B \sqsubseteq C$:

$\texttt{ninstd}(x, y_1, c) \lor \texttt{ninstd}(x, y_2, c) \leftarrow \texttt{subConj}(y_1, y_2, z, c), \texttt{ninstd}(x, z, c).$

- needed for completeness of justifications
- in practice, may generate large number of models
- is in general not sufficient to derive all negative consequences

Defeasibility rules: negative literals

Disjunctive information

Negative rule for $A \sqcap B \sqsubseteq C$:

$\texttt{ninstd}(x, y_1, c) \vee \texttt{ninstd}(x, y_2, c) \leftarrow \texttt{subConj}(y_1, y_2, z, c), \texttt{ninstd}(x, z, c).$

- needed for completeness of justifications
- in practice, may generate large number of models
- is in general not sufficient to derive all negative consequences

Solution: contradiction testing

$$\mathfrak{K} \models \mathsf{c} : \neg p(\mathbf{e}) \text{ iff } \mathfrak{K}' = \mathfrak{K} \cup \{\mathsf{c} : p(\mathbf{e})\} \text{ is unsatisfiable}$$

- use $\texttt{nlit}(p, \mathbf{e})$ to represent negative literals
- use $\texttt{unsat}(\texttt{nlit}(p, \mathbf{e}))$ for unsatisfiability with $p(\mathbf{e})$
- use $\texttt{test}(\texttt{nlit}(p, \mathbf{e}))$ and $\texttt{test\_fails}(\texttt{nlit}(p, \mathbf{e}))$ for test environment for $\texttt{nlit}(p, \mathbf{e})$ and for test failure, resp.

Defeasibility rules: contradiction tests

Contradiction testing: example rules

- Instantiate the test. E.g., for atomic inclusions:
  $\texttt{test}(\texttt{nlit}(x,z,c)) \leftarrow$
  $\texttt{def\_subclass}(y,z), \texttt{instd}(x,y,c,\text{main}), \texttt{prec}(c,g).$
- Exclude overriding, if the test fails.
  E.g., for the `subClass` overriding,
  $\leftarrow \texttt{test\_fails}(\texttt{nlit}(x,z,c)), \texttt{ovr}(\texttt{subClass},x,y,z,c).$
- Determine if test fails
  i.e., no clashes (= instances `unsat`) are found:
  $\texttt{test\_fails}(\texttt{nlit}(x,z,c)) \leftarrow$
  $\texttt{instd}(x,z,c,\texttt{nlit}(x,z,c)), \texttt{not unsat}(\texttt{nlit}(x,z,c)).$
- Generate test environment for each negative literal:
  e.g., for assertions
  $\texttt{instd}(x_1,y_1,c,t) \leftarrow \texttt{instd}(x_1,y_1,c,\text{main}), \texttt{test}(t).$
  $\texttt{instd}(x,z,c,\texttt{nlit}(x,z,c)) \leftarrow \texttt{test}(\texttt{nlit}(x,z,c)).$

Translation process

1. Global program $PG(\mathfrak{G})$: translation for global context

Translation process

1. Global program $PG(\mathfrak{G})$: translation for global context
2. Computation of local knowledge bases K_c for each context c in $\mathfrak{G}$

Translation process

1. Global program $PG(\mathfrak{G})$: translation for global context
2. Computation of local knowledge bases K_{c} for each context c in $\mathfrak{G}$
3. Local programs $PC(\mathsf{c})$: translation for local contexts

Translation process

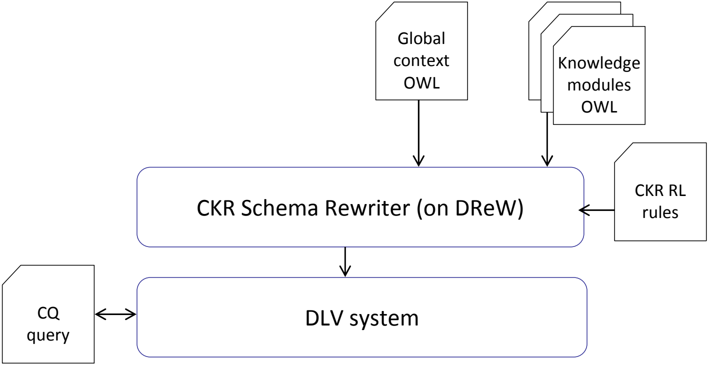
1. Global program $PG(\mathfrak{G})$: translation for global context
2. Computation of local knowledge bases K_{c} for each context c in $\mathfrak{G}$
3. Local programs $PC(\mathsf{c})$: translation for local contexts
4. CKR program $PK(\mathfrak{K})$: union of global and local programs

Translation Correctness

For a $\mathfrak{K}$ in normal form

1. $\mathfrak{K}$ entails $\mathsf{c} : \alpha$ iff $PK(\mathfrak{K}) \models O(\alpha, \mathsf{c})$ (axiom α in context c)
2. $\mathfrak{K} \models \exists \mathbf{y} \gamma(\mathbf{y})$ iff $PK(\mathfrak{K}) \models O(\exists \mathbf{y} \gamma(\mathbf{y}))$ (Boolean conjunctive queries)

CKRew: CKR datalog rewriter

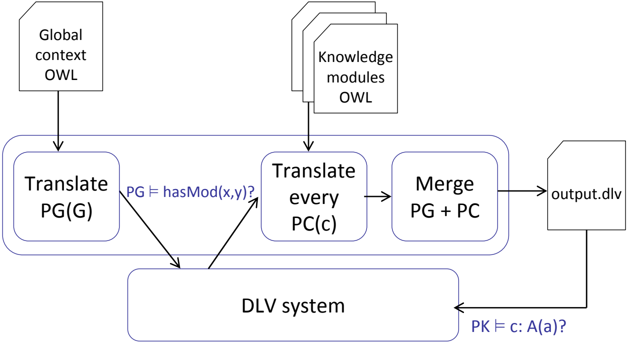


Prototype implementation:

- Extends basic translation of OWL RL ontologies to 2 layer CKR structure
- Input: OWL files for global context and knowledge modules
- Output: datalog translation for CKR program

CKRew translation process

Translation process implementation:



Prototype and examples available at: `http://ckrew.fbk.eu/`

Other approaches

- Normality in DLs: cf. [Britz and Varzinczak, 2016]
  no complex contextual structure with contextual reasoning inside modules

Other approaches

- Normality in DLs: cf. [Britz and Varzinczak, 2016]
  no complex contextual structure with contextual reasoning inside modules
- Typicality in DL: $\mathcal{ALC} + \mathbf{T}_{min}$ [Giordano et al., 2013]
  - defeasible membership similar to typical instances of C (**T**C)
  - model-based, our approach is syntax-sensitive

Other approaches

- Normality in DLs: cf. [Britz and Varzinczak, 2016]
  no complex contextual structure with contextual reasoning inside modules
- Typicality in DL: $\mathcal{ALC} + \mathbf{T}_{min}$ [Giordano et al., 2013]
  - defeasible membership similar to typical instances of C (**T**C)
  - model-based, our approach is syntax-sensitive
- Normality via Circumscription: [Bonatti et al., 2006]
  similar notion of abnormality, model based minimization

Other approaches

- Normality in DLs: cf. [Britz and Varzinczak, 2016]
  no complex contextual structure with contextual reasoning inside modules
- Typicality in DL: $\mathcal{ALC} + \mathbf{T}_{min}$ [Giordano et al., 2013]
  - defeasible membership similar to typical instances of C ($\mathbf{T}C$)
  - model-based, our approach is syntax-sensitive
- Normality via Circumscription: [Bonatti et al., 2006]
  similar notion of abnormality, model based minimization
- Nonmonotonic description logic $\mathcal{DL}^N$: [Bonatti et al., 2015]
  - extends a generic base DL $\mathcal{DL}$ with an operator NC for *normality concepts*
  - *defeasible inclusions (DIs)* $C \sqsubseteq_n D$ between concepts,
  - a polynomial rewriting procedure to base $\mathcal{DL}$
  - can not handle reasoning by cases (Nixon Diamond)

Other approaches

- Normality in DLs: cf. [Britz and Varzinczak, 2016]
  no complex contextual structure with contextual reasoning inside modules
- Typicality in DL: $\mathcal{ALC} + \mathbf{T}_{min}$ [Giordano et al., 2013]
  - defeasible membership similar to typical instances of C ($\mathbf{T}C$)
  - model-based, our approach is syntax-sensitive
- Normality via Circumscription: [Bonatti et al., 2006]
  similar notion of abnormality, model based minimization
- Nonmonotonic description logic $\mathcal{DL}^N$: [Bonatti et al., 2015]
  - extends a generic base DL $\mathcal{DL}$ with an operator NC for *normality concepts*
  - *defeasible inclusions (DIs)* $C \sqsubseteq_n D$ between concepts,
  - a polynomial rewriting procedure to base $\mathcal{DL}$
  - can not handle reasoning by cases (Nixon Diamond)
- Non-monotonic multi-context systems (MCS):
  [Brewka and Eiter, 2007, Bikakis and Antoniou, 2010]
  - translate CKR to MCS with bridge rules

Justifiable exceptions

Summary:

- Extension of CKR semantics to represent clashing assumptions and justifications
- Extension of CKR datalog translation with defeasible propagation
- CKR*ew* datalog rewriter implementation

Reasoning in $\mathcal{EL}_\bot$ and *DL-Lite*$_\mathcal{R}$

Introduce problem of reasoning with existential axioms and exceptions

- CKR in $\mathcal{EL}_\bot$ [Bozzato et al., 2019c]
- Justifiable exceptions in *DL-Lite*$_\mathcal{R}$ KB [Bozzato et al., 2019b]

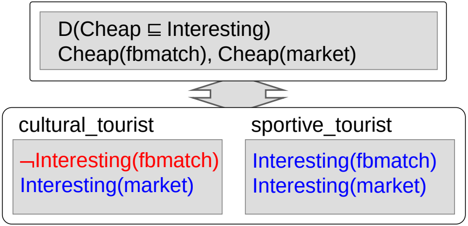
Overview

Overview

Limits of the model

CKR with Justifiable Exceptions

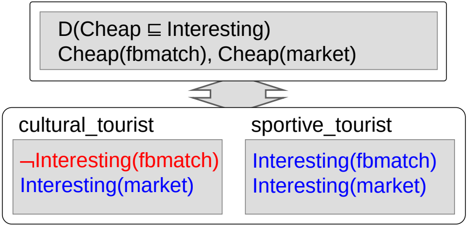
- Global context:
  Structure of contexts and object knowledge shared by all contexts
  Defeasible axioms: allow exceptional instances in local contexts
- (Local) contexts: Local object knowledge (with references)



Limits of the model

CKR with Justifiable Exceptions

- Global context:
  Structure of contexts and object knowledge shared by all contexts
  Defeasible axioms: allow exceptional instances in local contexts
- (Local) contexts: Local object knowledge (with references)



- Limited to 2 level hierarchy
- No further refinements allowed (e.g. sportive_cultural_tourist)

Proposal: contextual hierarchies

Idea

- Allow local defeasible axioms
- Contexts organized in a coverage hierarchy
- Axiom preference defined by context position:
  "more specific axioms are stronger"

Proposal: contextual hierarchies

Idea

- Allow local defeasible axioms
- Contexts organized in a coverage hierarchy
- Axiom preference defined by context position: "more specific axioms are stronger"

➔ sCKR with ranked contextual hierarchies [Bozzato et al., 2018b]

- Syntax and semantics for simple CKRs with ranked contextual hierarchies
- Study of reasoning problems and their complexity
- Extended datalog translation for OWL-RL based sCKR with rules for model preference (weak constraints)

Simple CKR: idea

sCKR: idea

- Global context: poset representing context hierarchy
- Local contexts: local context KBs with defeasible axioms

➔ Simplifies presentation of coverage, representable in "regular" CKR

Coverage and language

- Context names: $\mathbf{N} \subseteq \mathrm{NI}$
- Coverage: strict partial order $\prec \subseteq \mathbf{N} \times \mathbf{N}$
  if $c_1 \prec c_2$, c_2 covers c_1 (i.e. c_2 is more general than c_1)

Contextual language $\mathcal{L}^{\mathrm{D}}_{\mathbf{N}}$

DL language $\mathcal{L}$ extended with:

- eval expressions: $\mathit{eval}(X, \mathsf{c})$ ("the interpretation of X in context c")
- defeasible axioms: $\mathrm{D}(\alpha)$ for $\alpha \in \mathcal{L}$

Simple CKR

Simple Contextualized Knowledge Repository (sCKR):

$$\mathfrak{K} = \langle \mathfrak{C}, \mathrm{K}_{\mathbf{N}} \rangle$$

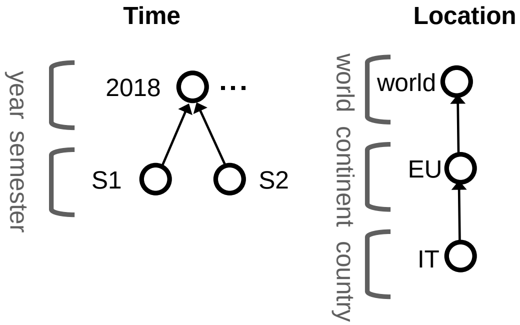
- $\mathfrak{C}$ is a poset $(\mathbf{N}, \prec)$
- $\mathrm{K}_{\mathbf{N}} = \{\mathrm{K}_{\mathsf{c}}\}_{\mathsf{c} \in \mathbf{N}}$ for every context name $\mathsf{c} \in \mathbf{N}$,
  K_{c} is a local DL knowledge base over $\mathcal{L}^{\mathrm{D}}_{\mathbf{N}}$

Example: introduction

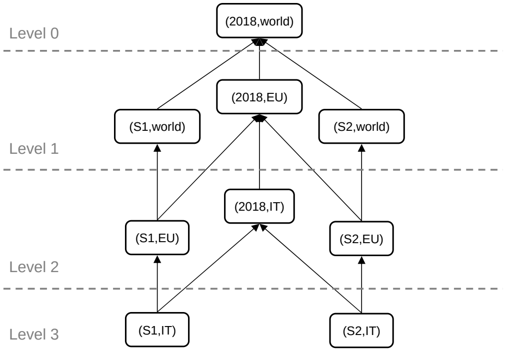
➔ Example of coverage structure defined by contextual dimensions [Serafini and Homola, 2012]

- A large organization has different policies with respect to
  - local branches (location dimension)
  - time period (time dimension)
- Active in different fields:
  Electronics (E), Robotics (R), Musical instruments (M)
- A local Supervisor (S) can manage only one of the fields

Example: dimensions



Example: hierarchy and local contexts



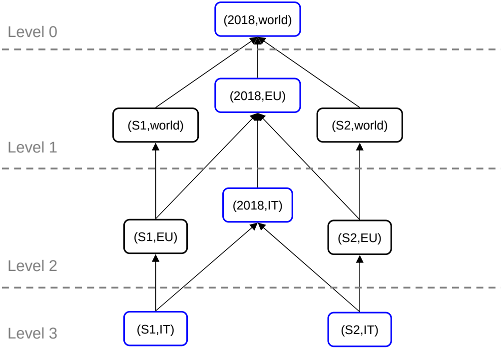
$\mathsf{C}_{(2018,world)}$: $\{M \sqcap E \sqsubseteq \bot, M \sqcap R \sqsubseteq \bot, E \sqcap R \sqsubseteq \bot\}$

$\mathsf{C}_{(2018,EU)}$: $\{\mathrm{D}(S \sqsubseteq E)\}$

$\mathsf{C}_{(2018,IT)}$: $\{\mathrm{D}(S \sqsubseteq M)\}$

$\mathsf{C}_{(S1,IT)}$: $\{S(i), R(i)\}$ $\mathsf{C}_{(S2,IT)}$: $\{S(i)\}$

Example: hierarchy and local contexts



$\mathsf{C}_{(2018,world)} : \{M \sqcap E \sqsubseteq \bot, M \sqcap R \sqsubseteq \bot, E \sqcap R \sqsubseteq \bot\}$

$\mathsf{C}_{(2018,EU)} : \{\mathrm{D}(S \sqsubseteq E)\}$

$\mathsf{C}_{(2018,IT)} : \{\mathrm{D}(S \sqsubseteq M)\}$

$\mathsf{C}_{(S1,IT)} : \{S(i), R(i)\}$ $\mathsf{C}_{(S2,IT)} : \{S(i)\}$

Ranked hierarchies

Idea

Hierarchies with a notion of level

Ranked hierarchy

A contextual hierarchy $\mathfrak{C} = (\mathbf{N}, \prec)$ is ranked iff,
for every root context r $\in \mathfrak{C}$ and every context c with c $\prec$ r,
all paths from c to r have the same length

Ranked hierarchies

Idea

Hierarchies with a notion of level

Ranked hierarchy

A contextual hierarchy $\mathfrak{C} = (\mathbf{N}, \prec)$ is ranked iff,
for every root context $\mathsf{r} \in \mathfrak{C}$ and every context c with $\mathsf{c} \prec \mathsf{r}$,
all paths from c to r have the same length

Level function: $l : \mathbf{N} \to \mathbb{N}$

$$l(\mathsf{c}) = \begin{cases} 0, & \text{if c is root} \\ 1 + \max(\{l(\mathsf{c}') \mid \mathsf{c} \prec \mathsf{c}'\}), & \text{otherwise} \end{cases}$$

Ranked hierarchies

Idea

Hierarchies with a notion of level

Ranked hierarchy

A contextual hierarchy $\mathfrak{C} = (\mathbf{N}, \prec)$ is ranked iff,
for every root context r $\in \mathfrak{C}$ and every context c with c $\prec$ r,
all paths from c to r have the same length

Level function: $l : \mathbf{N} \rightarrow \mathbb{N}$

$$l(\mathrm{c}) = \begin{cases} 0, & \text{if c is root} \\ 1 + \max(\{l(\mathrm{c}') \mid \mathrm{c} \prec \mathrm{c}'\}), & \text{otherwise} \end{cases}$$

Example: products of ranked dimension hierarchies
(like our example hierarchy in previous slide...)

sCKR interpretation

Idea

Set of interpretations for each local context

sCKR interpretation $\mathfrak{I}$

- $\mathfrak{I} = \{\mathcal{I}(\mathsf{c})\}_{\mathsf{c} \in \mathbf{N}}$
- For $\mathsf{c}, \mathsf{c}' \in \mathbf{N}$, $\mathcal{I}(\mathsf{c})$ is a DL interpretation:
  - $\Delta^{\mathcal{I}(\mathsf{c})} = \Delta^{\mathcal{I}(\mathsf{c}')}$
  - for $a \in \mathrm{NI}$, $a^{\mathcal{I}(\mathsf{c})} = a^{\mathcal{I}(\mathsf{c}')}$

Clashing assumptions

- CAS-interpretation $\mathfrak{I}_{CAS} = \langle \mathfrak{I}, \chi \rangle$:
  $\chi(c)$: set of clashing assumptions of context c

CAS-model $\mathfrak{I}_{CAS} \models \mathfrak{K}$

$\mathfrak{I}_{CAS}$ is a CAS-model for $\mathfrak{K}$ if:

- $\mathcal{I}(\mathsf{c}') \models \mathrm{K}_{\mathsf{c}}$, if $\mathsf{c}' \preceq \mathsf{c}$
- for every $\mathrm{D}(\alpha) \in \mathrm{K}_{\mathsf{c}}$, $\mathcal{I}(\mathsf{c}) \models \alpha$
- for every $\mathrm{D}(\alpha) \in \mathrm{K}_{\mathsf{c}}$ and $\mathsf{c}' \prec \mathsf{c}$, if $\langle \alpha, \mathbf{e} \rangle \notin \chi(\mathsf{c}')$, then $\mathcal{I}(\mathsf{c}') \models \alpha(\mathbf{e})$

Clashing assumptions

- CAS-interpretation $\mathfrak{I}_{CAS} = \langle \mathfrak{I}, \chi \rangle$:
  $\chi(c)$: set of clashing assumptions of context c

CAS-model $\mathfrak{I}_{CAS} \models \mathfrak{K}$

$\mathfrak{I}_{CAS}$ is a CAS-model for $\mathfrak{K}$ if:

- $\mathcal{I}(\mathsf{c}') \models \mathrm{K}_{\mathsf{c}}$, if $\mathsf{c}' \preceq \mathsf{c}$
- for every $\mathrm{D}(\alpha) \in \mathrm{K}_{\mathsf{c}}$, $\mathcal{I}(\mathsf{c}) \models \alpha$
- for every $\mathrm{D}(\alpha) \in \mathrm{K}_{\mathsf{c}}$ and $\mathsf{c}' \prec \mathsf{c}$, if $\langle \alpha, \mathbf{e} \rangle \notin \chi(\mathsf{c}')$, then $\mathcal{I}(\mathsf{c}') \models \alpha(\mathbf{e})$

Justification

$\mathfrak{I}_{\chi} = \langle \mathfrak{I}, \chi \rangle$ model of $\mathfrak{K}$ is justified, if for every context $\mathsf{c} \in \mathbf{N}$ and clashing assumption $\langle \alpha, \mathbf{e} \rangle \in \chi(\mathsf{c})$,

1. some clashing set $S = S_{\langle \alpha, \mathbf{e} \rangle, \mathsf{c}}$ exists such that $\mathcal{I}(\mathsf{c}) \models S_{\langle \alpha, \mathbf{e} \rangle, \mathsf{c}}$, and
2. for every model $\mathfrak{I}'_{\chi}$ of $\mathfrak{K}$ that is NI-congruent with $\mathfrak{I}_{\chi}$ $\mathcal{I}'(\mathsf{c}) \models S_{\langle \alpha, \mathbf{e} \rangle, \mathsf{c}}$

Assumption profile and ordering

Idea

- We want to give priority to more specific axioms
- ➔ Maximize the level of overridden axioms
- ➔ Order models using level of clashing assumptions

- Global profile $p(\chi)$: vector $(l_n, \ldots, l_0)$,
  each l_i is the number of clashing assumptions for axioms at level i
- Ordering $p(\chi) < p(\chi')$: lexicographical ordering
  e.g. (0,1,0,1) < (0,1,5,0)

sCKR model

Idea

sCKR models are justified and "maximize the rank" of overridings

Model preference:

$\mathfrak{I}_\chi = \langle \mathfrak{I}, \chi \rangle$ is preferred to $\mathfrak{I}'_\chi = \langle \mathfrak{I}, \chi' \rangle$ iff $p(\chi) < p(\chi')$

sCKR model

Idea

sCKR models are justified and "maximize the rank" of overridings

Model preference:

$\mathfrak{I}_\chi = \langle \mathfrak{I}, \chi \rangle$ is preferred to $\mathfrak{I}'_\chi = \langle \mathfrak{I}, \chi' \rangle$ iff $p(\chi) < p(\chi')$

sCKR model $\mathfrak{I} \models \mathfrak{K}$

$\mathfrak{I}$ is a sCKR model of $\mathfrak{K}$ if

- some $\mathfrak{I}_{CAS}$ is a justified CAS-model of $\mathfrak{K}$
- there exists no $\mathfrak{I}'_{CAS}$ that is preferred to $\mathfrak{I}_{CAS}$

Example: preferred models

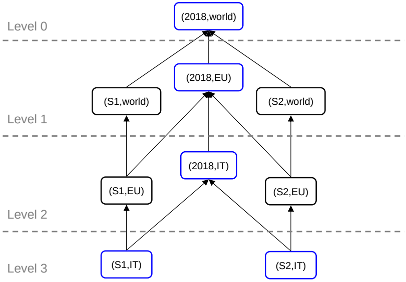
$\mathsf{c}_{(2018,world)}$: $\{M \sqcap E \sqsubseteq \bot,$
$M \sqcap R \sqsubseteq \bot, E \sqcap R \sqsubseteq \bot\}$

$\mathsf{c}_{(2018,EU)}$: $\{\mathrm{D}(S \sqsubseteq E)\}$

$\mathsf{c}_{(2018,IT)}$: $\{\mathrm{D}(S \sqsubseteq M)\}$

$\mathsf{c}_{(S1,IT)}$: $\{S(i), R(i)\}$

$\mathsf{c}_{(S2,IT)}$: $\{S(i)\}$



- 2 justified models:

$$\chi_1(\mathsf{c}_{(S1,IT)}) = \{\langle S \sqsubseteq E, i\rangle, \langle S \sqsubseteq M, i\rangle\} \quad \chi_1(\mathsf{c}_{(S2,IT)}) = \{\langle S \sqsubseteq E, i\rangle\}$$

$$\chi_2(\mathsf{c}_{(S1,IT)}) = \{\langle S \sqsubseteq E, i\rangle, \langle S \sqsubseteq M, i\rangle\} \quad \chi_2(\mathsf{c}_{(S2,IT)}) = \{\langle S \sqsubseteq M, i\rangle\}$$

Example: preferred models

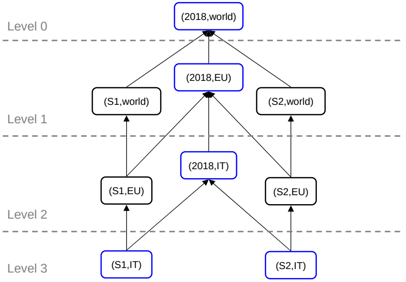
$\mathsf{c}_{(2018,world)}: \{M \sqcap E \sqsubseteq \bot,\ M \sqcap R \sqsubseteq \bot, E \sqcap R \sqsubseteq \bot\}$

$\mathsf{c}_{(2018,EU)}: \{\mathrm{D}(S \sqsubseteq E)\}$

$\mathsf{c}_{(2018,IT)}: \{\mathrm{D}(S \sqsubseteq M)\}$

$\mathsf{c}_{(S1,IT)}: \{S(i), R(i)\}$

$\mathsf{c}_{(S2,IT)}: \{S(i)\}$



- 2 justified models:

$$\chi_1(\mathsf{c}_{(S1,IT)}) = \{\langle S \sqsubseteq E, i\rangle, \langle S \sqsubseteq M, i\rangle\} \quad \chi_1(\mathsf{c}_{(S2,IT)}) = \{\langle S \sqsubseteq E, i\rangle\}$$

$$\chi_2(\mathsf{c}_{(S1,IT)}) = \{\langle S \sqsubseteq E, i\rangle, \langle S \sqsubseteq M, i\rangle\} \quad \chi_2(\mathsf{c}_{(S2,IT)}) = \{\langle S \sqsubseteq M, i\rangle\}$$

- Profile ordering: $p(\chi_1) = (0,1,2,0) < p(\chi_2) = (0,2,1,0)$

→ Model based on χ_1 is the preferred model

Reasoning and complexity

1. Satisfiability (does $\mathfrak{K}$ have a CKR model)
2. Model checking (is $\mathfrak{I}_{CAS}$ a model for $\mathfrak{K}$)
3. Axiom entailment $\mathfrak{K} \models \mathrm{c} : \alpha$
4. Conjunctive query answering $\mathfrak{K} \models \exists \mathbf{y} \gamma(\mathbf{y})$

Complexity results

- Satisfiability is NP-complete (was NP-complete)
- Model checking is coNP-complete (was polynomial)
- Axiom entailment is Δ_2^p-complete (was coNP-complete)
- (Boolean) CQ answering is Π_2^p-complete (was Π_2^p-complete)

Datalog translation with preferences

Main idea:

- Materialization calculus for instance checking and CQ answering in sCKR based on $\mathcal{SROIQ}$-RL (OWL-RL)
- Extends the datalog translation for CKR with justifiable exceptions in [Bozzato et al., 2018a]
- Interpreted under Answer Set semantics

➔ Rules for model preference: weak constraints [Leone et al., 2002]

Overriding level rules

Level preference rules: attach level info to overridings

$\texttt{ovrlevel\_subClass}(x, A, B, c, n) \leftarrow \texttt{ovr}(\texttt{subClass}, x, A, B, c_1, c), \texttt{level}(c_1, n).$

Weak constraints: prefer models with ovr. at higher level

$:\sim \texttt{ovrlevel\_subClass}(x, y, z, c, n).\ [1 : n]$

Overriding level rules

Level preference rules: attach level info to overridings

$\texttt{ovrlevel\_subClass}(x, A, B, c, n) \leftarrow \texttt{ovr}(\texttt{subClass}, x, A, B, c_1, c), \texttt{level}(c_1, n).$

Weak constraints: prefer models with ovr. at higher level

$:\sim \texttt{ovrlevel\_subClass}(x, y, z, c, n).\ [1 : n]$

Weak constraints

- $[1 : n]$: weight 1, priority level n
- wc intepretation: “minimize weight of violations at higher levels”

➔ prefer models with less overridings and at the higher levels

Translation process

1. Global program $PG(\mathfrak{C})$: translation for global context $\mathfrak{C}$
2. Local programs $PC(\mathsf{c}, \mathfrak{K})$: translation for local contexts K_{c}
3. CKR program $PK(\mathfrak{K})$: union of global and local programs

Translation process

1. Global program $PG(\mathfrak{C})$: translation for global context $\mathfrak{C}$
2. Local programs $PC(\mathsf{c}, \mathfrak{K})$: translation for local contexts K_{c}
3. CKR program $PK(\mathfrak{K})$: union of global and local programs

Translation Correctness

1. $\mathfrak{K} \models \mathsf{c} : \alpha$ iff $PK(\mathfrak{K}) \models O(\alpha, \mathsf{c})$ (axiom α in context c)
2. $\mathfrak{K} \models \exists \mathbf{y} \gamma(\mathbf{y})$ iff $PK(\mathfrak{K}) \models O(\exists \mathbf{y} \gamma(\mathbf{y}), \mathsf{c})$ (Boolean CQ in context c)

Contextual hierarchies

Summary:

- CKR extension with local defeasible axioms and knowledge propagation across coverage structure
- For ranked hierarchies: global model preference relation
- Datalog translation extending [Bozzato et al., 2018a] for instance checking based on weak constraints

sCKR with general hierarchies [Bozzato et al., 2019a]

- Semantics: local ordering on models
- Reasoning: selection procedure for preferred answer sets

Conclusion

Summary:

- Contextual model formalized in DL and AI theory of context
- Reasoning formalized as datalog materialization calculus
- Different (RDF based) implementations
- Extension with defeasible global axioms and justifiable exceptions
- Extension with defeasible local axioms in contextual hierarchies

Current and future directions:

- Application to OLAP operations on RDF cubes [Schuetz et al., 2020]
- Extension to different DL languages (see $\mathcal{EL}_\bot$ [Bozzato et al., 2019c])
- Study of alternative translations and implementation (CKRew)
- Different preference relations (e.g. for representation, efficiency)
- Interaction of different contextual relations (e.g. temporal, revision...)

Thank you for listening

Reasoning with Exceptions in Contextualized Knowledge Repositories

Loris Bozzato

bozzato[at]fbk.eu

https://dkm.fbk.eu/

Data and Knowledge Management Research Unit,

Fondazione Bruno Kessler - Trento, Italy

References I

Bao, J., Tao, J., and McGuinness, D. (2010).
Context representation for the semantic web.
In *Procs. of WebSci10*.

Benerecetti, M., Bouquet, P., and Ghidini, C. (2000).
Contextual reasoning distilled.
J. Exp. Theor. Artif. Intell., 12(3):279–305.

Bikakis, A. and Antoniou, G. (2010).
Defeasible contextual reasoning with arguments in ambient intelligence.
IEEE Trans. Knowl. Data Eng., 22(11):1492–1506.

Bonatti, P. A., Faella, M., Petrova, I., and Sauro, L. (2015).
A new semantics for overriding in description logics.
Artificial Intelligence, 222:1–48.

Bonatti, P. A., Lutz, C., and Wolter, F. (2006).
Description logics with circumscription.
In *KR2006*, pages 400–410. AAAI Press.

Bozzato, L., Eiter, T., and Serafini, L. (2018a).
Enhancing context knowledge repositories with justifiable exceptions.
Artif. Intell., 257:72–126.

Bozzato, L., Eiter, T., and Serafini, L. (2019a).
Justifiable exceptions in general contextual hierarchies.
In *CONTEXT 2019*, volume 11939 of *LNCS*, pages 26–39. Springer.

References II

Bozzato, L., Eiter, T., and Serafini, L. (2019b).
Reasoning on *DL-Lite*$_\mathcal{R}$ with defeasibility in ASP.
In *RuleML+RR 2019*, volume 11784 of *LNCS*, pages 19–35. Springer.

Bozzato, L., Eiter, T., and Serafini, L. (2019c).
Reasoning with justifiable exceptions in $\mathcal{EL}_\bot$ contextualized knowledge repositories.
In *Description Logic, Theory Combination, and All That*, volume 11560 of *LNCS*, pages 110–134. Springer.

Bozzato, L., Ghidini, C., and Serafini, L. (2013).
Comparing contextual and flat representations of knowledge: a concrete case about football data.
In *K-CAP 2013*, pages 9–16. ACM.

Bozzato, L. and Serafini, L. (2013).
Materialization Calculus for Contexts in the Semantic Web.
In *DL2013*, volume 1014 of *CEUR-WP*. CEUR-WS.org.

Bozzato, L. and Serafini, L. (2014).
Knowledge propagation in contextualized knowledge repositories: An experimental evaluation - (extended paper).
In *EKAW 2014 (Satellite Events)*, volume 8982 of *LNCS*, pages 35–51.

Bozzato, L., Serafini, L., and Eiter, T. (2018b).
Reasoning with justifiable exceptions in contextual hierarchies.
In *KR2018*, pages 329–338. AAAI Press.

Brewka, G. and Eiter, T. (2007).
Equilibria in heterogeneous nonmonotonic multi-context systems.
In *AAAI-07*, pages 385–390, Vancouver, Canada. AAAI Press.

References III

Britz, K. and Varzinczak, I. J. (2016).
Introducing role defeasibility in description logics.
In *JELIA 2016*, pages 174–189.

Ghidini, C. and Giunchiglia, F. (2001).
Local models semantics, or contextual reasoning = locality + compatibility.
Artif. Intell., 127:221–259.

Giordano, L., Gliozzi, V., Olivetti, N., and Pozzato, G. L. (2013).
A non-monotonic description logic for reasoning about typicality.
Artif. Intell., 195:165–202.

Klarman, S. and Gutiérrez-Basulto, V. (2011).
Two-dimensional description logics for context-based semantic interoperability.
In *AAAI-11*. AAAI Press.

Krötzsch, M. (2010).
Efficient inferencing for OWL EL.
In *JELIA 2010*, volume 6341 of *LNCS*, pages 234–246. Springer.

Lenat, D. (1998).
The Dimensions of Context Space.
Technical report, CYCorp.
Published online `http://www.cyc.com/doc/context-space.pdf`.

Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., and Scarcello, F. (2002).
The DLV system for knowledge representation and reasoning.
CoRR, cs.AI/0211004.

References IV

McCarthy, J. (1993).
Notes on formalizing context.
In *IJCAI-93*, pages 555–560. Morgan Kaufmann.

Schuetz, C. G., Bozzato, L., Neumayr, B., Schrefl, M., and Serafini, L. (2020).
Knowledge Graph OLAP: A Multidimensional Model and Query Operations for Contextualized Knowledge Graphs.
Semantic Web Journal.
In press. http://www.semantic-web-journal.net/content/
knowledge-graph-olap-multidimensional-model-and-query-operations-contextualized-knowledge-0.

Serafini, L. and Homola, M. (2012).
Contextualized knowledge repositories for the semantic web.
J. of Web Semantics, 12:64–87.

Straccia, U., Lopes, N., Lukacsy, G., and Polleres, A. (2010).
A general framework for representing and reasoning with annotated Semantic Web data.
In *AAAI-10*. AAAI Press.

Tamilin, A., Magnini, B., Serafini, L., Girardi, C., Joseph, M., and Zanoli, R. (2010).
Context-driven semantic enrichment of italian news archive.
In *ESWC 2010*, volume 6088 of *LNCS*, pages 364–378. Springer.