Constructive Semantics for Description Logics

ASP Based Generation of Information Terms for Constructive $\mathcal{EL}$

Loris Bozzato

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

UniVR Logic Seminar

February 2, 2017 – Verona, Italy

Description logics

Description logics (DL)

A family of logic based Knowledge Representation formalisms

- Main features:
  - Expressive but decidable fragments of FOL
  - Formally defined semantics $\Rightarrow$ Reasoning
  - Efficient implementations for key problems
  - Relevant applications (Semantic Web)

Description logics

Description logics (DL)

A family of logic based Knowledge Representation formalisms

- Main features:
  - Expressive but decidable fragments of FOL
  - Formally defined semantics $\Rightarrow$ Reasoning
  - Efficient implementations for key problems
  - Relevant applications (Semantic Web)
- Elements:
  - Concepts: classes of objects **Human**
  - Roles: binary relations between objects **hasChild**
- Complex descriptions:
  - Concept constructors ($\sqcap$, $\sqcup$, $\neg$) $\texttt{Square} \sqcup \neg \texttt{Round}$
  - Role restrictions ($\exists$, $\forall$, $\geqslant$) $\geqslant 2\, \texttt{hasChild}$

Description logics

Description logics (DL)

A family of logic based Knowledge Representation formalisms

- Main features:
  - Expressive but decidable fragments of FOL
  - Formally defined semantics $\Rightarrow$ Reasoning
  - Efficient implementations for key problems
  - Relevant applications (Semantic Web)
- Elements:
  - Concepts: classes of objects **Human**
  - Roles: binary relations between objects **hasChild**
- Complex descriptions:
  - Concept constructors ($\sqcap$, $\sqcup$, $\neg$) $\mathtt{Square} \sqcup \neg \mathtt{Round}$
  - Role restrictions ($\exists$, $\forall$, $\geqslant$) $\geqslant 2\, \mathtt{hasChild}$

$$\forall \mathtt{hasBrother}.(\exists \mathtt{isChildOf}.\mathtt{Father})$$

Constructive logics

Constructive logics

Formalizations of ideas from constructivism

Constructive logics

Constructive logics

Formalizations of ideas from constructivism

Brouwer-Heyting-Kolmogorov (BHK) or proof interpretation

Semiformal presentation of a constructive semantics

E.g. propositional part:

- A proof of $A \wedge B$ is composed from proofs of A and B
- A proof of $A \vee B$ is composed of a proof for A or B
- A proof of $A \rightarrow B$ is construction transforming proofs of A in proofs for B
- $\bot$ is an unprovable formula (thus $A \rightarrow \bot = \neg A$)

- Possible formalizations:
  - Intuitionism
  - Recursive realizability
  - Information terms semantics

Constructive logics

- Characteristic properties:
  - Disjunction property (DP):
    "Whenever it proves a disjunction formula,
    it proves one of the disjoints"
  - Explicit definability property (ED):
    "Whenever it proves an existential formula,
    it presents a witness of the existence"
- Constructivism and Computer Science:
  - Formulas-as-types (Curry-Howard isomorphism)
  - Proofs-as-programs

Constructive description logics

Constructive description logics
Constructive interpretations of description logics

Motivations

- Computational interpretation of proofs and formulas
- Useful in domains with dynamic and incomplete knowledge

Proposals

- [de Paiva, 2005]: translations of DLs in constructive systems
- [Kaneiwa, 2005]: definitions for different constructive negations in DLs
- [Odintsov and Wansing, 2003]: inconsistency tolerant version of DLs
- [Mendler and Scheele, 2010]: Kripke semantics with "fallible" elements
- $\mathcal{BCDL}$ [Ferrari et al., 2010]: Information terms semantics + natural deduction
- $\mathcal{KALC}$ [Bozzato, 2011]: Kripke-style semantics + tableaux algorithm

Constructive DLs and applications

- Constructive DLs mostly studied from formal point of view...
- Limited proposals for application in KR and Semantic Web languages

Applications (examples)

- [Mendler and Scheele, 2009]: reasoning over incomplete data streams
- [Haeusler et al., 2011]: conflict management on legal ontologies
- [Hilia et al., 2012]: semantic services compositions (on $\mathcal{BCDL}$)

Idea: ...let's try to bridge the gap!

Proposal

- Theory: study relations between IT and ASP
- Practice: prototype over "off the shelf" tools (OWL API, dlv)

Contributions

- $\mathcal{EL}c$: IT semantics for description logic $\mathcal{EL}$
- ASP and IT semantics: formal relation and datalog rewriting
- Asp-it prototype: ASP based IT generator for OWL-EL ontologies

Overview

1. Constructive semantics for DLs
2. $\mathcal{EL}$c: constructive semantics for $\mathcal{EL}$
3. Answer sets and IT semantics
4. ASP based generation of IT
5. Asp-it prototype

Overview

1. Constructive semantics for DLs
2. $\mathcal{EL}c$: constructive semantics for $\mathcal{EL}$
3. Answer sets and IT semantics
4. ASP based generation of IT
5. Asp-it prototype

Proposals

Known proposals

- [de Paiva, 2005]: translations of DLs in constructive systems
- [Kaneiwa, 2005]: definitions for different constructive negations in DLs
- [Odintsov and Wansing, 2003]: inconsistency tolerant version of DLs
- [Mendler and Scheele, 2010]: Kripke semantics with “fallible” elements

- $\mathcal{BCDL}$ [Ferrari et al., 2010]: Information terms semantics + ND calculus
- $\mathcal{KALC}$ [Bozzato, 2011]: Kripke-style semantics + tableaux algorithm

Proposal [de Paiva, 2005]

Motivation

- Extend proof-theoretical results (Curry-Howard) on DLs
- Define a context-sensitive DL

Proposals

3 different interpretations of $\mathcal{ALC}$ in constructive systems:

- IALC: from $\mathcal{ALC}$ to IFOL (via $\mathcal{ALC} \rightarrow$ FOL translation)
- iALC: from $\mathcal{ALC}$ to IK (via $\mathcal{ALC} \rightarrow K_m$ translation)
- cALC: from $\mathcal{ALC}$ to CK (via $\mathcal{ALC} \rightarrow K_m$ translation)

Proposal [Kaneiwa, 2005]

Motivation

Representation of different notions of negative information in DLs
(contraries, contradictories and subcontraries)
E.g. difference between $Happy, Unhappy, \neg Happy, \neg Unhappy$

Proposals

- 2 different extensions to $\mathcal{ALC}$ semantics
  (different interactions between constructive and classical negation)
- tableaux algorithm for satisfiability

Similar works: [Kamide, 2010a, Kamide, 2010b]

Paraconsistent and temporal versions of $\mathcal{ALC}$, based on a similar semantics

Proposal [Odintsov and Wansing, 2003]

Ideas

- Paraconsistent versions of $\mathcal{ALC}$
- Constructive semantics to represent partial information

Proposal [Odintsov and Wansing, 2003]

- 3 constructive paraconsistent semantics for $\mathcal{ALC}$
  (Different translations to four valued logic $N4$)
- complete tableaux calculus for each logic

Further work [Odintsov and Wansing, 2008]

Reviews of calculi, tableaux procedure for one of the presented logics

Proposal [Mendler and Scheele, 2010]

Idea

- Representation of partial knowledge and consistency under abstraction
- Evolving OWA: stages of information with changing properties and abstract individuals

Proposal

$c\mathcal{ALC}$: Kripke semantics for $\mathcal{ALC}$ with fallible entities

- fallible entities $\perp^{\mathcal{I}}$: contradictory domain elements (maximal poset elements or undefined role fillers)
- complete and decidable Hilbert and tableaux calculi

Application [Mendler and Scheele, 2009]

Reasoning on data streams in auditing domain

Our proposals

$\mathcal{BCDL}$ [Bozzato et al., 2007, Bozzato et al., 2009b, Ferrari et al., 2010]

Information terms semantics + natural deduction calculus

➔ computational interpretation of proofs (*Proofs as programs*)

$\mathcal{KALC}^{\infty}$ [Bozzato et al., 2009a, Villa, 2010]

Kripke-style semantics + tableaux calculus

➔ possibly infinite models, efficient treatment of implications

$\mathcal{KALC}$ [Bozzato et al., 2010, Bozzato, 2011]

Kripke-style semantics + tableaux algorithm

➔ finite models, decidability from terminating tableau procedure

$\mathcal{BCDL}$: information terms semantics for $\mathcal{ALC}$

$\mathcal{BCDL}$: Basic Constructive Description Logic [Ferrari et al., 2010]

- Information terms semantics for $\mathcal{ALC}$
- Natural deduction calculus $\mathcal{ND}_c$

Information terms (IT) [Miglioli et al., 1989]

Syntactic objects justifying validity of formulas in classical models

- realization of BHK interpretation
- related to realizability interpretations

$\mathcal{BCDL}$: information terms semantics for $\mathcal{ALC}$

Information terms (IT) justification

E.g.: Truth of $\exists R.C(a)$ in $\mathcal{M}$ justified by IT (b, α) s.t.

- $\mathcal{M} \models R(a, b)$ and
- α justifies truth of $C(b)$

- Features:
  - Classical reading of DL formulas
  - Simple proof theoretical characterization by $\mathcal{ND}_c$
  - Computational interpretation of proofs
  - Natural notion of state

$\mathcal{ALCG}$ syntax and classical semantics

Syntax is the same as $\mathcal{ALC}$, adding a set of generators $\mathtt{NG}$

Concepts

$$C \ ::= \ A \mid G \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \exists R.C \mid \forall R.C$$

Formulas

$$K \ ::= \ \bot \mid R(t,s) \mid C(t) \mid \forall_G C$$

where $A \in \mathtt{NC}$, $R \in \mathtt{NR}$, $t,s \in \mathtt{NI}$ and $G \in \mathtt{NG}$

Generators $\mathtt{NG}$

Concepts over fixed set of individual names $dom(G) = \{c_1, \ldots, c_n\}$
➔ limited form of subsumption: $\forall_G C \ \equiv \ G \sqsubseteq C$

$\mathcal{ALCG}$ syntax and classical semantics

Validity of formulas: given a model $\mathcal{M} = (\Delta^{\mathcal{M}}, \cdot^{\mathcal{M}})$

$$\mathcal{M} \not\models \bot$$

$$\mathcal{M} \models R(t,s) \text{ iff } (t^{\mathcal{M}}, s^{\mathcal{M}}) \in R^{\mathcal{M}}$$

$$\mathcal{M} \models H(t) \text{ iff } t^{\mathcal{M}} \in H^{\mathcal{M}}$$

$$\mathcal{M} \models \forall_G H \text{ iff } G^{\mathcal{M}} = \{c_1^{\mathcal{M}}, \ldots, c_n^{\mathcal{M}}\} \subseteq H^{\mathcal{M}}$$

$\mathcal{BCDL}$ information terms semantics

Information terms $\mathsf{IT}_{\mathcal{N}}(K)$

Structured objects that constructively justify the truth of a formula K

$\mathsf{IT}_{\mathcal{N}}(K) = \{tt\}$, if K is atomic

$\mathsf{IT}_{\mathcal{N}}(C_1 \sqcup C_2(c)) = \{(k, \alpha) \mid k \in \{1, 2\}$ and $\alpha \in \mathsf{IT}(C_k(c))\}$

$\mathcal{BCDL}$ information terms semantics

Information terms $\mathsf{IT}_{\mathcal{N}}(K)$

Structured objects that constructively justify the truth of a formula K

$\mathsf{IT}_{\mathcal{N}}(K) = \{tt\}$, if K is atomic
$\mathsf{IT}_{\mathcal{N}}(C_1 \sqcup C_2(c)) = \{(k, \alpha) \mid k \in \{1,2\}$ and $\alpha \in \mathsf{IT}(C_k(c))\}$

Realizability $\mathcal{M} \rhd \langle \alpha \rangle\, K$

Truth of K in a model $\mathcal{M}$ justified w.r.t. α

$\mathcal{M} \rhd \langle tt \rangle\, K$ iff $\mathcal{M} \models K$
$\mathcal{M} \rhd \langle (k, \alpha) \rangle\, C_1 \sqcup C_2(c)$ iff $\mathcal{M} \rhd \langle \alpha \rangle\; C_k(c)$

$\mathcal{BCDL}$ information terms semantics

Information terms $\mathsf{IT}_{\mathcal{N}}(K)$

Structured objects that constructively justify the truth of a formula K

$\mathsf{IT}_{\mathcal{N}}(K) = \{tt\}$, if K is atomic
$\mathsf{IT}_{\mathcal{N}}(C_1 \sqcup C_2(c)) = \{(k, \alpha) \mid k \in \{1,2\}$ and $\alpha \in \mathsf{IT}(C_k(c))\}$

Realizability $\mathcal{M} \rhd \langle \alpha \rangle\, K$

Truth of K in a model $\mathcal{M}$ justified w.r.t. α

$\mathcal{M} \rhd \langle tt \rangle\, K$ iff $\mathcal{M} \models K$
$\mathcal{M} \rhd \langle (k, \alpha) \rangle\, C_1 \sqcup C_2(c)$ iff $\mathcal{M} \rhd \langle \alpha \rangle\; C_k(c)$

Theorem (classical and IT semantics)

$\mathcal{M} \models K$ *iff there exists* $\alpha \in \mathsf{IT}(K)$ *such that* $\mathcal{M} \rhd \langle \alpha \rangle\, K$

$\mathcal{BCDL}$ information terms semantics

$\mathsf{IT}_{\mathcal{N}}(K) = \{\mathtt{tt}\}$ for K atomic or negated	$\mathcal{M} \rhd \langle \mathtt{tt} \rangle\, K$ iff $\mathcal{M} \models K$
$\mathsf{IT}_{\mathcal{N}}(C_1 \sqcap C_2(c)) = \mathsf{IT}(C_1(c)) \times \mathsf{IT}(C_2(c))$	$\mathcal{M} \rhd \langle (\alpha, \beta) \rangle\, C_1 \sqcap C_2(c)$ iff $\mathcal{M} \rhd \langle \alpha \rangle\, C_1(c)$ and $\mathcal{M} \rhd \langle \beta \rangle\, C_2(c)$
$\mathsf{IT}_{\mathcal{N}}(C_1 \sqcup C_2(c)) = \mathsf{IT}(C_1(c)) \uplus \mathsf{IT}(C_2(c))$	$\mathcal{M} \rhd \langle (k, \alpha) \rangle\, C_1 \sqcup C_2(c)$ iff $\mathcal{M} \rhd \langle \alpha \rangle\, C_k(c)$
$\mathsf{IT}_{\mathcal{N}}(\exists R.C(c)) = \mathcal{N} \times \bigcup_{d \in \mathcal{N}} \mathsf{IT}(C(d))$	$\mathcal{M} \rhd \langle (d, \alpha) \rangle\, \exists R.C(c)$ iff $\mathcal{M} \models R(c,d)$ and $\mathcal{M} \rhd \langle \alpha \rangle\, C(d)$
$\mathsf{IT}_{\mathcal{N}}(\forall R.C(c)) = (\bigcup_{d \in \mathcal{N}} \mathsf{IT}(C(d)))^{\mathcal{N}}$	$\mathcal{M} \rhd \langle \phi \rangle\, \forall R.C(c)$ iff $\mathcal{M} \models \forall R.C(c)$ and, for every $d \in \mathcal{N}$, if $\mathcal{M} \models R(c,d)$ then $\mathcal{M} \rhd \langle \phi(d) \rangle\, C(d)$
$\mathsf{IT}_{\mathcal{N}}(\forall_G C) = (\bigcup_{d \in dom(G)} \mathsf{IT}(C(d)))^{dom(G)}$	$\mathcal{M} \rhd \langle \phi \rangle\, \forall_G C$ iff, for every $d \in dom(G)$, $\mathcal{M} \rhd \langle \phi(d) \rangle\, C(d)$

Natural deduction calculus $\mathcal{ND}$

Calculus $\mathcal{ND}$: natural deduction calculus for $\mathcal{ALCG}$

Rules

$$\frac{\begin{array}{c}\Gamma \\ \vdots\ \pi' \\ A_k(t)\end{array}}{A_1 \sqcup A_2(t)}\ \sqcup I_k \qquad \frac{\begin{array}{c}\Gamma_1 \\ \vdots\ \pi_1 \\ \exists R.A(t)\end{array} \qquad \begin{array}{c}\Gamma_2, [R(t,p), A(p)] \\ \vdots\ \pi_2 \\ K\end{array}}{K}\ \exists E \qquad \frac{\forall_G A \quad \begin{array}{c}\Gamma' \\ \vdots\ \pi' \\ G(t)\end{array}}{A(t)}\ \forall_G E \qquad \frac{\begin{array}{c}\Gamma, [\neg H(t)] \\ \vdots\ \pi' \\ \bot\end{array}}{H(t)}\ \neg E$$

Theorem

- $\mathcal{ND}$ *is sound and complete w.r.t.* $\mathcal{ALCG}$
- *leaving aside generators rules,* $\mathcal{ND}$ *is sound and complete w.r.t.* $\mathcal{ALC}$

Natural deduction calculus $\mathcal{ND}_c$

Calculus $\mathcal{ND}_c$: natural deduction calculus for $\mathcal{BCDL}$

Rules

Every rule from $\mathcal{ND}$ (minus $\neg E$)

$$\dfrac{\begin{matrix}\Gamma \\ \vdots\ \pi' \\ \neg\neg C(t)\end{matrix}}{C(t)}\ \text{At} \qquad \dfrac{\begin{matrix}\Gamma \\ \vdots\ \pi' \\ \forall R.\neg\neg H(t)\end{matrix}}{\neg\neg\forall R.H(t)}\ \text{KUR}$$

Note

KUR corresponds to the Kuroda axiom schema Kur

$$\text{Kur} \equiv \forall x.\neg\neg A(x) \rightarrow \neg\neg\forall x.A(x)$$

Soundness of $\mathcal{ND}_c$

Operator $\Phi^{\pi}_{\mathcal{N}}$: Given a proof $\pi : \Gamma \vdash K$ over $\mathcal{N}$:

$$\Phi^{\pi}_{\mathcal{N}} : \mathsf{IT}_{\mathcal{N}}(\Gamma) \to \mathsf{IT}_{\mathcal{N}}(K)$$

Note: computable function, inductively defined on depth of π

Example: $\sqcup I_k$

If last rule in π is $\sqcup I_k$ with $k \in \{1, 2\}$, then:

$$\Phi^{\pi}_{\mathcal{N}} : \mathsf{IT}_{\mathcal{N}}(\Gamma) \to \mathsf{IT}_{\mathcal{N}}(C_1 \sqcup C_2(t))$$

defined as:

$$\Phi^{\pi}_{\mathcal{N}}(\overline{\gamma}) \;=\; (\,k,\, \Phi^{\pi'}_{\mathcal{N}}(\overline{\gamma})\,)$$

$\sqcup I_k$ rule

$$\dfrac{\begin{array}{c}\Gamma \\ \vdots\ \pi' \\ A_k(t)\end{array}}{A_1 \sqcup A_2(t)}\ \sqcup I_k$$

Soundness of $\mathcal{ND}_c$: computational interpretation

Theorem (Soundness)

If $\pi : \Gamma \vdash K$*, then:*

- $\Gamma \models K$.
- *If* $\mathcal{M} \rhd \langle \overline{\gamma} \rangle \, \Gamma$ *then* $\mathcal{M} \rhd \langle \Phi^{\pi}_{\mathcal{N}}(\overline{\gamma}) \rangle \, K$*.* (constructive consequence)

Computational interpretation

- Given a proof π of a formula K...
- ...its $\Phi^{\pi}_{\mathcal{N}}$ provides a "program" to compute an IT for K

Properties of $\mathcal{ND}_c$

Theorem (Completeness)

$\Gamma \vdash_{\mathcal{BCDL}} K$ *iff* K *is a constructive consequence of* Γ.

Constructive properties

For Γ of Harrop formulas (no $\sqcup$ and $\exists$):

- Disjunction property (DP):
  If $\Gamma \vdash_{\mathcal{BCDL}} A \sqcup B(c)$, then $\Gamma \vdash_{\mathcal{BCDL}} A(c)$ or $\Gamma \vdash_{\mathcal{BCDL}} B(c)$.
- Explicit definability property (EDP):
  If $\Gamma \vdash_{\mathcal{BCDL}} \exists R.A(c)$, then there exists $d \in \mathtt{NI}$ such that $\Gamma \vdash_{\mathcal{BCDL}} R(c,d)$ and $\Gamma \vdash_{\mathcal{BCDL}} A(d)$.

Overview

1. Constructive semantics for DLs
2. $\mathcal{EL}$c: constructive semantics for $\mathcal{EL}$
3. Answer sets and IT semantics
4. ASP based generation of IT
5. Asp-it prototype

Overview

1. Constructive semantics for DLs
2. $\mathcal{EL}$c: constructive semantics for $\mathcal{EL}$
3. Answer sets and IT semantics
4. ASP based generation of IT
5. Asp-it prototype

$\mathcal{EL}$c

- $\mathcal{EL}$c: information terms semantics for $\mathcal{EL}$ [Baader, 2003]
- Simple restriction of $\mathcal{BCDL}$ to $\mathcal{EL}$
- Why $\mathcal{EL}$?
  - The most simple DL s.t. semantics with constructive properties (ED) can be defined
  - Well-known and used DL language, base of OWL EL profile

DL language $\mathcal{L}$

Syntax is the same as $\mathcal{EL}$, adding a set of generators $\texttt{NG}$

Concepts

$$C \quad ::= \quad A \mid G \mid C \sqcap C \mid \exists R.C$$

Formulas

$$K \quad ::= \quad R(s,t) \mid C(t) \mid \forall_G C$$

where $A \in \texttt{NC}$, $R \in \texttt{NR}$, $t, s \in \texttt{NI}$ and $G \in \texttt{NG}$

DL language $\mathcal{L}$

Syntax is the same as $\mathcal{EL}$, adding a set of generators $\texttt{NG}$

Concepts

$$C \;::=\; A \mid G \mid C \sqcap C \mid \exists R.C$$

Formulas

$$K \;::=\; R(s,t) \mid C(t) \mid \forall_G C$$

where $A \in \texttt{NC}$, $R \in \texttt{NR}$, $t, s \in \texttt{NI}$ and $G \in \texttt{NG}$

Generators $\texttt{NG}$

Concepts over fixed set of individual names $\textsc{dom}(G) = \{c_1, \ldots, c_n\}$

➔ limited form of subsumption: $\forall_G C \;\equiv\; G \sqsubseteq C$

$\top$ operator: special generator $\top_{\mathcal{N}}$ s.t. $\textsc{dom}(\top_{\mathcal{N}}) = \mathcal{N}$

$\mathcal{EL}$: classical semantics

- Classical model: $\mathcal{M} = (\Delta^{\mathcal{M}}, \cdot^{\mathcal{M}})$

Individuals:	$c^{\mathcal{M}} \in \Delta^{\mathcal{M}}$
Atomic concepts:	$A^{\mathcal{M}} \subseteq \Delta^{\mathcal{M}}$
Roles:	$R^{\mathcal{M}} \subseteq \Delta^{\mathcal{M}} \times \Delta^{\mathcal{M}}$
Generators:	$G^{\mathcal{M}} = \{c_1^{\mathcal{M}}, \ldots, c_n^{\mathcal{M}}\}$

- Non-atomic concepts:

$$(C \sqcap D)^{\mathcal{M}} = C^{\mathcal{M}} \cap D^{\mathcal{M}}$$

$$(\exists R.C)^{\mathcal{M}} = \{ c \in \Delta^{\mathcal{M}} \mid \text{there is } d \text{ s.t. } (c,d) \in R^{\mathcal{M}} \text{ and } d \in C^{\mathcal{M}} \}$$

- Validity of formulas:

$$\mathcal{M} \models R(s,t) \text{ iff } (s^{\mathcal{M}}, t^{\mathcal{M}}) \in R^{\mathcal{M}}$$

$$\mathcal{M} \models C(t) \text{ iff } t^{\mathcal{M}} \in C^{\mathcal{M}}$$

$$\mathcal{M} \models \forall_G C \text{ iff } G^{\mathcal{M}} \subseteq C^{\mathcal{M}}$$

Example: Food and Wines

Scenario: food and wines knowledge base [Brachman et al., 1991]

Task: pair each food with the correct wine

Conditions:

- Pairings:
  - Meat with red wines
  - Fish with white wines
- For every food, a correct wine color
- For every wine color, at least one wine

Example: Food and Wines KB

Knowledge base $\mathcal{K}_W$:

TBox $\mathcal{T}$:

$$(Ax_1) : \forall_{\mathtt{Food}} \exists \mathtt{goesWith.Color} \equiv \mathtt{Food} \sqsubseteq \exists \mathtt{goesWith.Color}$$

$$(Ax_2) : \forall_{\mathtt{Color}} \exists \mathtt{isColorOf.Wine} \equiv \mathtt{Color} \sqsubseteq \exists \mathtt{isColorOf.Wine}$$

$$\mathsf{DOM}(\mathtt{Food}) = \{\mathtt{fish}, \mathtt{meat}\} \qquad \mathsf{DOM}(\mathtt{Color}) = \{\mathtt{red}, \mathtt{white}\}$$

Example: Food and Wines KB

Knowledge base $\mathcal{K}_W$:

TBox $\mathcal{T}$:

$$(Ax_1): \forall_{\texttt{Food}} \exists \texttt{goesWith.Color} \equiv \texttt{Food} \sqsubseteq \exists \texttt{goesWith.Color}$$

$$(Ax_2): \forall_{\texttt{Color}} \exists \texttt{isColorOf.Wine} \equiv \texttt{Color} \sqsubseteq \exists \texttt{isColorOf.Wine}$$

$$\mathsf{DOM}(\texttt{Food}) = \{\texttt{fish}, \texttt{meat}\} \qquad \mathsf{DOM}(\texttt{Color}) = \{\texttt{red}, \texttt{white}\}$$

ABox $\mathcal{A}$:

Wine(barolo)
Wine(chardonnay)

isColorOf(red,barolo)
isColorOf(white,chardonnay)
goesWith(fish,white)
goesWith(meat,red)

$\mathcal{EL}$c: information terms semantics

Given $\mathcal{N} \subseteq \mathtt{NI}$ finite and a closed formula $K \in \mathcal{L}_{\mathcal{N}}$

Information terms $\mathsf{IT}_{\mathcal{N}}(K)$

Structured objects that constructively justify the truth of a formula K

$\mathcal{EL}$c: information terms semantics

Given $\mathcal{N} \subseteq \mathtt{NI}$ finite and a closed formula $K \in \mathcal{L}_{\mathcal{N}}$

Information terms $\mathrm{IT}_{\mathcal{N}}(K)$

Structured objects that constructively justify the truth of a formula K

Realizability $\mathcal{M} \rhd \langle \alpha \rangle\, K$

Truth of K in a model $\mathcal{M}$ justified w.r.t. α

$\mathcal{EL}$c: information terms semantics

Given $\mathcal{N} \subseteq \mathtt{NI}$ finite and a closed formula $K \in \mathcal{L}_{\mathcal{N}}$

Information terms $\mathsf{IT}_{\mathcal{N}}(K)$

$\mathsf{IT}_{\mathcal{N}}(K) = \{\mathtt{tt}\}$, if K is atomic

Realizability $\mathcal{M} \rhd \langle \alpha \rangle\, K$

$\mathcal{M} \rhd \langle \mathtt{tt} \rangle\, K$ iff $\mathcal{M} \models K$

$\mathcal{EL}$c: information terms semantics

Given $\mathcal{N} \subseteq \mathtt{NI}$ finite and a closed formula $K \in \mathcal{L}_{\mathcal{N}}$

Information terms $\mathsf{IT}_{\mathcal{N}}(K)$

$\mathsf{IT}_{\mathcal{N}}(K) = \{\mathtt{tt}\}$, if K is atomic

$\mathsf{IT}_{\mathcal{N}}(C_1 \sqcap C_2(c)) = \{(\alpha, \beta) \mid \alpha \in \mathsf{IT}(C_1(c))$ and $\beta \in \mathsf{IT}(C_2(c))\}$

Realizability $\mathcal{M} \rhd \langle\alpha\rangle\, K$

$\mathcal{M} \rhd \langle\mathtt{tt}\rangle\, K$ iff $\mathcal{M} \models K$

$\mathcal{M} \rhd \langle(\alpha, \beta)\rangle\, C_1 \sqcap C_2(c)$ iff $\mathcal{M} \rhd \langle\alpha\rangle\, C_1(c)$ and $\mathcal{M} \rhd \langle\beta\rangle\, C_2(c)$

$\mathcal{EL}$c: information terms semantics

Given $\mathcal{N} \subseteq \mathtt{NI}$ finite and a closed formula $K \in \mathcal{L}_{\mathcal{N}}$

Information terms $\mathsf{IT}_{\mathcal{N}}(K)$

$\mathsf{IT}_{\mathcal{N}}(K) = \{\mathtt{tt}\}$, if K is atomic

$\mathsf{IT}_{\mathcal{N}}(C_1 \sqcap C_2(c)) = \{(\alpha, \beta) \mid \alpha \in \mathsf{IT}(C_1(c))$ and $\beta \in \mathsf{IT}(C_2(c))\}$

$\mathsf{IT}_{\mathcal{N}}(\exists R.C(c)) = \{(d, \alpha) \mid d \in \mathcal{N}$ and $\alpha \in \mathsf{IT}(C(d))\}$

Realizability $\mathcal{M} \rhd \langle \alpha \rangle\, K$

$\mathcal{M} \rhd \langle \mathtt{tt} \rangle\, K$ iff $\mathcal{M} \models K$

$\mathcal{M} \rhd \langle (\alpha, \beta) \rangle\, C_1 \sqcap C_2(c)$ iff $\mathcal{M} \rhd \langle \alpha \rangle\, C_1(c)$ and $\mathcal{M} \rhd \langle \beta \rangle\, C_2(c)$

$\mathcal{M} \rhd \langle (d, \alpha) \rangle\, \exists R.C(c)$ iff $\mathcal{M} \models R(c, d)$ and $\mathcal{M} \rhd \langle \alpha \rangle\, C(d)$

$\mathcal{E}\mathcal{L}$c: information terms semantics

Given $\mathcal{N} \subseteq \mathtt{NI}$ finite and a closed formula $K \in \mathcal{L}_{\mathcal{N}}$

Information terms $\mathsf{IT}_{\mathcal{N}}(K)$

$\mathsf{IT}_{\mathcal{N}}(K) = \{\mathtt{tt}\}$, if K is atomic

$\mathsf{IT}_{\mathcal{N}}(C_1 \sqcap C_2(c)) = \{(\alpha, \beta) \mid \alpha \in \mathsf{IT}(C_1(c))$ and $\beta \in \mathsf{IT}(C_2(c))\}$

$\mathsf{IT}_{\mathcal{N}}(\exists R.C(c)) = \{(d, \alpha) \mid d \in \mathcal{N}$ and $\alpha \in \mathsf{IT}(C(d))\}$

$\mathsf{IT}_{\mathcal{N}}(\forall_G C) = \{\, \phi : \mathsf{DOM}(G) \to \bigcup_{d \in dom(G)} \mathsf{IT}_{\mathcal{N}}(C(d)) \mid \phi(d) \in \mathsf{IT}_{\mathcal{N}}(C(d)) \,\}$

Realizability $\mathcal{M} \rhd \langle \alpha \rangle\, K$

$\mathcal{M} \rhd \langle \mathtt{tt} \rangle\, K$ iff $\mathcal{M} \models K$

$\mathcal{M} \rhd \langle (\alpha, \beta) \rangle\, C_1 \sqcap C_2(c)$ iff $\mathcal{M} \rhd \langle \alpha \rangle\, C_1(c)$ and $\mathcal{M} \rhd \langle \beta \rangle\, C_2(c)$

$\mathcal{M} \rhd \langle (d, \alpha) \rangle\, \exists R.C(c)$ iff $\mathcal{M} \models R(c, d)$ and $\mathcal{M} \rhd \langle \alpha \rangle\, C(d)$

$\mathcal{M} \rhd \langle \phi \rangle\, \forall_G C$ iff, for every $d \in \mathsf{DOM}(G)$, $\mathcal{M} \rhd \langle \phi(d) \rangle\, C(d)$

Example: IT for Food and Wines

(Ax_1) :

$$\forall_{\texttt{Food}}\exists\texttt{goesWith.Color} \equiv \texttt{Food} \sqsubseteq \exists\texttt{goesWith.Color}$$

$\phi_1 \in \mathsf{IT}_{\mathcal{N}}(Ax_1)$:

$$[\ \texttt{fish} \mapsto (\texttt{white}, \texttt{tt}),\ \texttt{meat} \mapsto (\texttt{red}, \texttt{tt})\ ]$$

Example: IT for Food and Wines

(Ax_2) :

$$\forall_{\texttt{Color}} \exists \texttt{isColorOf.Wine} \equiv \texttt{Color} \sqsubseteq \exists \texttt{isColorOf.Wine}$$

$\phi_2 \in \mathsf{IT}_{\mathcal{N}}(Ax_2)$:

$$[\ \texttt{red} \mapsto (\texttt{barolo}, \texttt{tt}),\ \texttt{white} \mapsto (\texttt{chardonnay}, \texttt{tt})\ ]$$

Let $\mathcal{M}$ be a model of $\mathcal{K}_W$: then, $\mathcal{M} \rhd \langle(\phi_1, \phi_2)\rangle\, \mathcal{T}$

Constructive consequence

Theorem (classical and IT semantics)

If $\alpha \in \mathsf{IT}(K)$, $\mathcal{M} \rhd \langle \alpha \rangle\, K$ *implies* $\mathcal{M} \models K$

Constructive consequence $\Gamma \overset{c}{\models} K$:
$\Gamma \overset{c}{\models} K$ iff $\mathcal{M} \rhd \langle \gamma \rangle\, \Gamma$ implies $\mathcal{M} \rhd \langle \eta \rangle\, K$

Constructive consequence

Theorem (classical and IT semantics)

If $\alpha \in \mathsf{IT}(K)$, $\mathcal{M} \rhd \langle \alpha \rangle\, K$ *implies* $\mathcal{M} \models K$

Constructive consequence $\Gamma \overset{c}{\models} K$:
$\Gamma \overset{c}{\models} K$ iff $\mathcal{M} \rhd \langle \gamma \rangle\, \Gamma$ implies $\mathcal{M} \rhd \langle \eta \rangle\, K$

Computational interpretation

- $\Gamma \overset{c}{\models} K$ implicitly defines a semantic map $\Phi_{\mathcal{N}}$ such that:

$$\text{if } \mathcal{M} \rhd \langle \gamma \rangle\, \Gamma \text{ then } \mathcal{M} \rhd \langle \Phi_{\mathcal{N}}(\gamma) \rangle\, K$$

- In $\mathcal{BCDL}$ we can extract $\Phi_{\mathcal{N}}$ from natural deduction proofs [Bozzato et al., 2007, Ferrari et al., 2010]

Overview

1. Constructive semantics for DLs
2. $\mathcal{EL}$c: constructive semantics for $\mathcal{EL}$
3. Answer sets and IT semantics
4. ASP based generation of IT
5. Asp-it prototype

Overview

1. Constructive semantics for DLs
2. $\mathcal{EL}c$: constructive semantics for $\mathcal{EL}$
3. Answer sets and IT semantics
4. ASP based generation of IT
5. Asp-it prototype

Answer sets for formulas and information terms

Task

Compute information terms of input KB Γ in $\mathcal{EL}$

Idea

Use relations across IT and Answer Sets semantics

[Fiorentini and Ornaghi, 2007] on propositional nested expressions

➔ We extend these results to $\mathcal{EL}$c formulas

Answer set

lp-interpretation

Set of closed atomic formulas in $\mathcal{L}_{\mathcal{N}}$

Given a closed $K \in \mathcal{L}_{\mathcal{N}}$:

$$
\begin{array}{rl}
I \models K, & \text{iff } K \in I \text{ and } K \text{ is atomic} \\
I \models C \sqcap D(c) & \text{iff } I \models C(c) \text{ and } I \models D(c) \\
I \models \exists R.C(c) & \text{iff } R(c,d) \in I \text{ for } d \in \mathcal{N} \text{ and } I \models C(d) \\
I \models \forall_G C & \text{iff for every } e \in \text{DOM}(G), I \models C(e) \\
I \models \Gamma & \text{iff } I \models K \text{ for } K \in \Gamma
\end{array}
$$

Answer set

lp-interpretation

Set of closed atomic formulas in $\mathcal{L}_{\mathcal{N}}$

Given a closed $K \in \mathcal{L}_{\mathcal{N}}$:

$$
\begin{array}{rl}
I \models K, & \text{iff } K \in I \text{ and } K \text{ is atomic} \\
I \models C \sqcap D(c) & \text{iff } I \models C(c) \text{ and } I \models D(c) \\
I \models \exists R.C(c) & \text{iff } R(c,d) \in I \text{ for } d \in \mathcal{N} \text{ and } I \models C(d) \\
I \models \forall_G C & \text{iff for every } e \in \text{DOM}(G), I \models C(e) \\
I \models \Gamma & \text{iff } I \models K \text{ for } K \in \Gamma
\end{array}
$$

Answer set

I answer set for set of closed formulas $\Gamma \subseteq \mathcal{L}_{\mathcal{N}}$ iff:

- $I \models \Gamma$
- for every $I' \subseteq I$, $I' \models \Gamma$ implies $I' = I$ (minimality)

Answers of pieces of information

Piece of information (POI)

$\langle \eta \rangle K$ with closed $K \in \mathcal{L}_{\mathcal{N}}$ and $\eta \in \mathsf{IT}_{\mathcal{N}}(K)$
(Idea: "state" of K defined by η)

Answers of pieces of information

Piece of information (POI)

$\langle \eta \rangle K$ with closed $K \in \mathcal{L}_{\mathcal{N}}$ and $\eta \in \mathsf{IT}_{\mathcal{N}}(K)$

(Idea: "state" of K defined by η)

Answers $\mathtt{ans}(\langle \eta \rangle K)$

Set of closed atomic formulas: (Idea: "information content" of $\langle \eta \rangle K$)

$$\begin{aligned}
\mathtt{ans}(\langle \mathtt{tt} \rangle K) &= \{K\}, \text{ with } K \text{ atomic} \\
\mathtt{ans}(\langle (\alpha, \beta) \rangle A_1 \sqcap A_2(c)) &= \mathtt{ans}(\langle \alpha \rangle A_1(c)) \cup \mathtt{ans}(\langle \beta \rangle A_2(c)) \\
\mathtt{ans}(\langle (d, \alpha) \rangle \exists R.A(c)) &= \{R(c,d)\} \cup \mathtt{ans}(\langle \alpha \rangle A(d)) \\
\mathtt{ans}(\langle \phi \rangle \forall_G A) &= \bigcup_{d \in \mathsf{DOM}(G)} \mathtt{ans}(\langle \phi(d) \rangle A(d))
\end{aligned}$$

Answers of pieces of information

Piece of information (POI)

$\langle \eta \rangle K$ with closed $K \in \mathcal{L}_{\mathcal{N}}$ and $\eta \in \mathsf{IT}_{\mathcal{N}}(K)$

(Idea: "state" of K defined by η)

Answers $\mathtt{ans}(\langle \eta \rangle K)$

Set of closed atomic formulas: (Idea: "information content" of $\langle \eta \rangle K$)

$$\begin{aligned}
\mathtt{ans}(\langle \mathtt{tt} \rangle K) &= \{K\}, \text{ with } K \text{ atomic} \\
\mathtt{ans}(\langle (\alpha, \beta) \rangle A_1 \sqcap A_2(c)) &= \mathtt{ans}(\langle \alpha \rangle A_1(c)) \cup \mathtt{ans}(\langle \beta \rangle A_2(c)) \\
\mathtt{ans}(\langle (d, \alpha) \rangle \exists R.A(c)) &= \{R(c,d)\} \cup \mathtt{ans}(\langle \alpha \rangle A(d)) \\
\mathtt{ans}(\langle \phi \rangle \forall_G A) &= \bigcup_{d \in \mathsf{DOM}(G)} \mathtt{ans}(\langle \phi(d) \rangle A(d))
\end{aligned}$$

Minimal POI

$\langle \eta \rangle K$ is minimal iff $\nexists \mu$ with $\mathtt{ans}(\langle \mu \rangle K) \subset \mathtt{ans}(\langle \eta \rangle K)$

Example: answers of POIs

$(Ax_1) : \forall_{\texttt{Food}}\exists\texttt{goesWith}.\texttt{Color} \equiv \texttt{Food} \sqsubseteq \exists\texttt{goesWith}.\texttt{Color}$

$\phi_1 \in \mathsf{IT}_{\mathcal{N}}(Ax_1) : [\ \texttt{fish} \mapsto (\texttt{white}, \texttt{tt}), \texttt{meat} \mapsto (\texttt{red}, \texttt{tt})\ ]$

$$\texttt{ans}(\langle\phi_1\rangle Ax_1) = \{\texttt{Color(white)}, \texttt{goesWith(fish,white)}, \texttt{Color(red)}, \texttt{goesWith(meat,red)}\}$$

Example: answers of POIs

$(Ax_1) : \forall_{\mathtt{Food}} \exists \mathtt{goesWith.Color} \equiv \mathtt{Food} \sqsubseteq \exists \mathtt{goesWith.Color}$

$\phi_1 \in \mathsf{IT}_{\mathcal{N}}(Ax_1) : [\ \mathtt{fish} \mapsto (\mathtt{white}, \mathtt{tt}),\ \mathtt{meat} \mapsto (\mathtt{red}, \mathtt{tt})\ ]$

$$\mathtt{ans}(\langle \phi_1 \rangle Ax_1) = \{\mathtt{Color(white)}, \mathtt{goesWith(fish, white)}, \mathtt{Color(red)}, \mathtt{goesWith(meat, red)}\}$$

$(Ax_2) : \forall_{\mathtt{Color}} \exists \mathtt{isColorOf.Wine} \equiv \mathtt{Color} \sqsubseteq \exists \mathtt{isColorOf.Wine}$

$\phi_2 \in \mathsf{IT}_{\mathcal{N}}(Ax_2) : [\ \mathtt{red} \mapsto (\mathtt{barolo}, \mathtt{tt}),\ \mathtt{white} \mapsto (\mathtt{chardonnay}, \mathtt{tt})\ ]$

$$\mathtt{ans}(\langle \phi_2 \rangle Ax_2) = \{\mathtt{Wine(barolo)}, \mathtt{isColorOf(red, barolo)}, \mathtt{Wine(chardonnay)}, \mathtt{isColorOf(white, chardonnay)}\}$$

Result: answer sets and minimal POI

Theorem

For every model $\mathcal{M}$, $\mathcal{M} \rhd \langle\eta\rangle\, K$ iff $\mathcal{M} \models \mathtt{ans}(\langle\eta\rangle K)$ □

Theorem

I answer set for K iff
$\exists$ a minimal $\langle\eta\rangle K$ such that $I = \mathtt{ans}(\langle\eta\rangle K)$ □

Overview

1. Constructive semantics for DLs
2. $\mathcal{EL}$c: constructive semantics for $\mathcal{EL}$
3. Answer sets and IT semantics
4. ASP based generation of IT
5. Asp-it prototype

Overview

1. Constructive semantics for DLs
2. $\mathcal{EL}$c: constructive semantics for $\mathcal{EL}$
3. Answer sets and IT semantics
4. ASP based generation of IT
5. Asp-it prototype

ASP based generation of information terms

Solution to generate (minimal) IT:

1. Compute answer sets of input KB Γ
2. For each formula $K \in \Gamma$, use recursive definition of $\mathtt{ans}(\langle \eta \rangle K)$ to reconstruct IT η

Idea: computation of answer sets

Translate input KB to datalog program

➔ Translation includes rules recursively constructing ITs (generate-and-test approach)

Model generating rewriting

Model generating rewriting (P_1)

Generates interpretations for input $\mathcal{EL}$ formulas:

$$
\begin{aligned}
A(b) &\mapsto \{\, is(b,A) \leftarrow is(b,l_A).\,\} \\
R(a,b) &\mapsto \{\, rel(a,R,b).\,\} \\
C \sqcap D(a) &\mapsto \{\, is(a,l_C) \leftarrow is(a,l_{C\sqcap D}). \\
&\qquad is(a,l_D) \leftarrow is(a,l_{C\sqcap D}).\,\} \cup P_1(C(a)) \cup P_1(D(a)) \\
\exists R.C(a) &\mapsto \{\, is(x,l_C) \leftarrow rel(a,R,x), is(a,l_{\exists R.C}).\,\} \cup P_1(C(x)) \\
\forall_G C &\mapsto \{\, is(x,l_C) \leftarrow is(x,G).\,\} \cup P_1(C(x))
\end{aligned}
$$

Notes

- Fixed set $\mathcal{R}$ of roles assertions from input KB
- Labelling for complex concepts l_C
- Atomic assertions and generator domains added as facts

IT generating rewriting

IT generating rewriting (P_2)

Retrieves IT as complex terms, using definition of $\texttt{ans}(\langle \eta \rangle K)$:

$$A(b) \mapsto \{\, is\_it(\texttt{tt}, b, l_A) \leftarrow is(b, A).\, \}$$

$$R(a,b) \mapsto \{\, rel\_it(\texttt{tt}, a, R, b) \leftarrow rel(a, R, b).\, \}$$

$$C \sqcap D(a) \mapsto \{\, is\_it([x,y], a, l_{C\sqcap D}) \leftarrow is\_it(x, a, l_C), is\_it(y, a, l_D).\, \} \cup P_2(C(a)) \cup P_2(D(a))$$

$$\exists R.C(a) \mapsto \{\, is\_it([x,y], a, l_{\exists R.C}) \leftarrow rel\_it(\texttt{tt}, a, R, x), is\_it(y, x, l_C).\, \} \cup P_2(C(x))$$

$$\forall_G C \mapsto \{\, isa\_it([x,y], G, l_C) \leftarrow is(x, G), is\_it(y, x, l_C).\, \} \cup P_2(C(x))$$

Complete rewriting (P)

$$P(\Gamma) = P_1(\Gamma) \cup P_2(\Gamma)$$

Let $IT(K, I)$ be the set of IT "returned" by P_2 for formula K and interpretation I for $P(\Gamma)$

Theorem

Let I be the (unique) answer set for $P(\Gamma)$.
If $\eta \in IT(K, I)$, then $\exists$ lp-interpretation I' for Γ s.t. $\mathtt{ans}(\langle\eta\rangle K) \subseteq I'$ $\square$

Example

Suppose we add:

Wine(teroldego) **isColorOf(red,teroldego)**

Applying $P_2(Ax_2)$ to the model computed by $P_1(\mathcal{K}_W)$:

$[\mathtt{red}, [\mathtt{barolo}, \mathtt{tt}]]$, $[\mathtt{red}, [\mathtt{teroldego}, \mathtt{tt}]]$, $[\mathtt{white}, [\mathtt{chardonnay}, \mathtt{tt}]]$

Example

Suppose we add:

Wine(teroldego)   **isColorOf(red,teroldego)**

Applying $P_2(Ax_2)$ to the model computed by $P_1(\mathcal{K}_W)$:

[**red**, [**barolo**, **tt**]], [**red**, [**teroldego**, **tt**]], [**white**, [**chardonnay**, **tt**]]

$(Ax_3): \forall_{\texttt{Food}} \exists \texttt{goesWith}.(\texttt{Color} \sqcap \exists \texttt{isColorOf}.\texttt{Wine})$

Applying $P_2(Ax_3)$:

[**fish**, [**white**, [**tt**, [**chardonnay**, **tt**]]]],
[**meat**, [**red**, [**tt**, [**barolo**, **tt**]]]],
[**meat**, [**red**, [**tt**, [**teroldego**, **tt**]]]]

Overview

1. Constructive semantics for DLs
2. $\mathcal{EL}$c: constructive semantics for $\mathcal{EL}$
3. Answer sets and IT semantics
4. ASP based generation of IT
5. Asp-it prototype

Overview

1. Constructive semantics for DLs
2. $\mathcal{ELc}$: constructive semantics for $\mathcal{EL}$
3. Answer sets and IT semantics
4. ASP based generation of IT
5. Asp-it prototype

Asp-it prototype

Asp-it

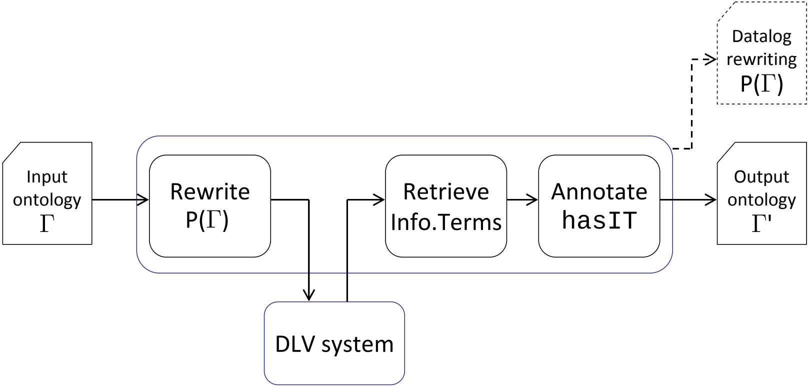
Java-based command line application:

- Input: OWL-EL ontology Γ
- Output: ontology Γ' annotated with IT (`elc:hasIT`)

Tools

- OWL API: ontology I/O, axioms annotation
- DLV: models computation (via DLVWrapper)

IT generation process



Prototype and examples available at:
https://github.com/dkmfbk/asp-it

Conclusions

Summary:

- $\mathcal{EL}$c: IT semantics for description logic $\mathcal{EL}$
- ASP and IT semantics: formal relation and datalog rewriting
- Asp-it prototype: ASP based IT generator for OWL-EL ontologies

Future works:

- Integrate procedures for transformation of IT
  (Calculus and proofs-as-programs)
- Applications: synthesis of Semantic Services [Bozzato and Ferrari, 2010a]
- Extend to larger DLs: $\mathcal{SROEL}$ (full OWL EL), $\mathcal{ALC}$ ($\mathcal{BCDL}$)

App. directions: IT and states

IT and states

- Information terms encode a natural notion of state
- Used in [Ferrari et al., 2008] to represent system snapshots

→ Action formalism for $\mathcal{ALC}$ [Bozzato et al., 2009b]

An action formalism based on IT semantics of $\mathcal{BCDL}$

App. directions: IT and states

IT and states

- Information terms encode a natural notion of state
- Used in [Ferrari et al., 2008] to represent system snapshots

➔ Action formalism for $\mathcal{ALC}$ [Bozzato et al., 2009b]

An action formalism based on IT semantics of $\mathcal{BCDL}$

System description and states

- Theory $\mathbf{T}$: description of a system
  - TBox: system constraints (general properties)
  - ABox: current state of the system
- State: $\alpha \in \mathrm{IT}(\mathbf{T})$
- State consistency: if there is a model $\mathcal{M}$ s.t. $\mathcal{M} \rhd \langle\alpha\rangle\, \mathbf{T}$

App. directions: action language

- Action: $\mathcal{P} \Rightarrow \mathcal{Q}$

Informal reading

- If the preconditions $\mathcal{P}$ hold in a state α, the action can be applied
- In the resulting state the postconditions $\mathcal{Q}$ must hold

Information content $\text{IC}(\langle\alpha\rangle\mathbf{T})$:
minimal set of atomic formulas encoding info. from $\langle\alpha\rangle\mathbf{T}$

- Applicability: an action is *active* if $\mathcal{P} \subseteq \text{IC}(\langle\alpha\rangle\mathbf{T})$
- Action output $\text{Out}(\alpha)$: update $\text{IC}(\langle\alpha\rangle\mathbf{T})$ with $\mathcal{Q}$

GenIt

- Algorithm to build up a state (IT) for a system, given an action output
- It can be used to trace reasons for inconsistency

App. directions: web services composition

Service composition in $\mathcal{BCDL}$ [Bozzato and Ferrari, 2010b]

- Calculus for definition of Semantic Web Services compositions
- Related to program synthesis in constr. logics [Miglioli et al., 1986]
- Services as combined functions "computing" information terms

App. directions: web services composition

Service composition in $\mathcal{BCDL}$ [Bozzato and Ferrari, 2010b]

- Calculus for definition of Semantic Web Services compositions
- Related to program synthesis in constr. logics [Miglioli et al., 1986]
- Services as combined functions "computing" information terms

Composition calculus $\mathcal{SC}$

$$\frac{\mathsf{s}(x) :: P \Rightarrow Q}{\begin{array}{c}\Pi_1 : \mathsf{s}_1(x) :: P_1 \Rightarrow Q_1 \\ \ldots \\ \Pi_n : \mathsf{s}_n(x) :: P_n \Rightarrow Q_n\end{array}}\, r$$

- Applicability conditions (AC): constraints for correctness of rule application
- Computational interpretation (CI): computational reading of logical rule

App. directions: web services composition

Service composition in $\mathcal{BCDL}$ [Bozzato and Ferrari, 2010b]

- Calculus for definition of Semantic Web Services compositions
- Related to program synthesis in constr. logics [Miglioli et al., 1986]
- Services as combined functions "computing" information terms

Composition calculus $\mathcal{SC}$

$$\frac{\mathsf{s}(x) :: P \Rightarrow Q}{\begin{array}{c}\Pi_1 : \mathsf{s}_1(x) :: P_1 \Rightarrow Q_1 \\ \ldots \\ \Pi_n : \mathsf{s}_n(x) :: P_n \Rightarrow Q_n\end{array}} r$$

- Applicability conditions (AC): constraints for correctness of rule application
- Computational interpretation (CI): computational reading of logical rule

Result

If a composition meets the ACs of its rules, then its computational interpretation is sound

Thank you for listening

Constructive Semantics for Description Logics

ASP Based Generation of Information Terms for Constructive $\mathcal{EL}$

Loris Bozzato

bozzato@fbk.eu

https://dkm.fbk.eu/

DKM - Data and Knowledge Management Research Unit,

FBK - Fondazione Bruno Kessler, Trento, Italy

References I

Baader, F. (2003).
Terminological cycles in a description logic with existential restrictions.
In *IJCAI-03*, pages 325–330. Morgan Kaufmann.

Bozzato, L. (2011).
Kripke semantics and tableau procedures for constructive description logics.
PhD thesis, DICOM – Università degli Studi dell'Insubria.

Bozzato, L. and Ferrari, M. (2010a).
Composition of semantic web services in a constructive description logic.
In *RR2010*, volume 6333 of *Lecture Notes in Computer Science*, pages 223–226. Springer.

Bozzato, L. and Ferrari, M. (2010b).
Composition of Semantic Web Services in a Constructive Description Logic.
In *Proceedings of the 4th International Conference on Web Reasoning and Rule Systems (RR2010)*, volume 6333 of *Lecture Notes in Computer Science*, pages 223–226. Springer.

Bozzato, L., Ferrari, M., Fiorentini, C., and Fiorino, G. (2007).
A constructive semantics for $\mathcal{ALC}$.
In *DL2007*, volume 250 of *CEUR-WP*, pages 219–226. CEUR-WS.org.

Bozzato, L., Ferrari, M., Fiorentini, C., and Fiorino, G. (2010).
A decidable constructive description logic.
In *Proceedings of the 12th European Conference on Logics in Artificial Intelligence (JELIA 2010)*, Lecture Notes in Computer Science. Springer.

Bozzato, L., Ferrari, M., and Villa, P. (2009a).
A note on constructive semantics for description logics.
In *CILC09 - 24-esimo Convegno Italiano di Logica Computazionale*.

References II

Bozzato, L., Ferrari, M., and Villa, P. (2009b).
Actions Over a Constructive Semantics for Description Logics.
Fundamenta Informaticae, 96(3):253–269.

Brachman, R., McGuinness, D., Patel-Schneider, P., Resnick, L., and Borgida, A. (1991).
Living with CLASSIC: When and how to use a KL-ONE-like language.
In *Principles of Semantic Networks*, pages 401–456. Morgan Kauffman.

de Paiva, V. (2005).
Constructive description logics: what, why and how.
Technical report, Xerox Parc.

Ferrari, M., Fiorentini, C., and Fiorino, G. (2010).
$\mathcal{BCDL}$: basic constructive description logic.
J. of Automated Reasoning, 44(4):371–399.

Ferrari, M., Fiorentini, C., Momigliano, A., and Ornaghi, M. (2008).
Snapshot generation in a constructive object-oriented modeling language.
In *LOPSTR 2007, Selected Papers*, volume 4915 of *Lecture Notes in Computer Science*, pages 169–184. Springer.

Fiorentini, C. and Ornaghi, M. (2007).
Answer set semantics vs. information term semantics.
In *ASP2007: Answer Set Programming, Advances in Theory and Implementation.*

Haeusler, E. H., de Paiva, V., and Rademaker, A. (2011).
Intuitionistic description logic and legal reasoning.
In *DEXA 2011 Workshops*, pages 345–349. IEEE Computer Society.

References III

Hilia, M., Chibani, A., Djouani, K., and Amirat, Y. (2012).
Semantic service composition framework for multidomain ubiquitous computing applications.
In *ICSOC 2012*, volume 7636 of *Lecture Notes in Computer Science*, pages 450–467. Springer.

Kamide, N. (2010a).
A compatible approach to temporal description logics.
In *DL2010 - International Workshop on Description Logics*.

Kamide, N. (2010b).
Paraconsistent description logics revisited.
In *DL2010 - International Workshop on Description Logics*, pages 197–208.

Kaneiwa, K. (2005).
Negations in description logic – contraries, contradictories, and subcontraries.
In Dau, F., Mugnier, M.-L., and Stumme, G., editors, *Common Semantics for Sharing Knowledge: Contributions to the 13th International Conference on Conceptual Structures (ICCS '05)*, pages 66–79. Kassel University Press.

Mendler, M. and Scheele, S. (2009).
Towards a type system for semantic streams.
In *SR2009 - Stream Reasoning Workshop (ESWC 2009)*, volume 466 of *CEUR-WP*. CEUR-WS.org.

Mendler, M. and Scheele, S. (2010).
Towards Constructive DL for Abstraction and Refinement.
J. Autom. Reasoning, 44(3):207–243.

Miglioli, P., Moscato, U., and Ornaghi, M. (1986).
PAP: A Logic Programming System Based on a Constructive Logic.
In *Foundations of Logic and Functional Programming*, volume 306 of *Lecture Notes in Computer Science*, pages 143–156. Springer.

References IV

Miglioli, P., Moscato, U., Ornaghi, M., and Usberti, G. (1989).
A constructivism based on classical truth.
Notre Dame Journal of Formal Logic, 30(1):67–90.

Odintsov, S. and Wansing, H. (2003).
Inconsistency-tolerant description logic. Motivation and basic systems.
In Hendricks, V. and Malinowski, J., editors, *Trends in Logic. 50 Years of Studia Logica*, pages 301–335. Kluwer Academic Publishers, Dordrecht.

Odintsov, S. and Wansing, H. (2008).
Inconsistency-tolerant description logic. Part II: A tableau algorithm for $CALC^C$.
J. of Applied Logic, 6(3):343–360.

Villa, P. (2010).
Semantics foundations for constructive description logics.
PhD thesis, DICOM – Università degli Studi dell'Insubria.