Description Logics Primer

Luciano Serafini

FBK-IRST, Trento, Italy

July 17, 2012

Origins of Description Logics

Description Logics stem from early days knowledge representation formalisms (late '70s, early '80s):

- Semantic Networks: graph-based formalism, used to represent the meaning of sentences.
- Frame Systems: frames used to represent prototypical situations, antecedents of object-oriented formalisms.

Problems: **no clear semantics**, reasoning not well understood.
Description Logics (a.k.a. Concept Languages, Terminological Languages) developed starting in the mid '80s, with the aim of providing semantics and inference techniques to knowledge representation system

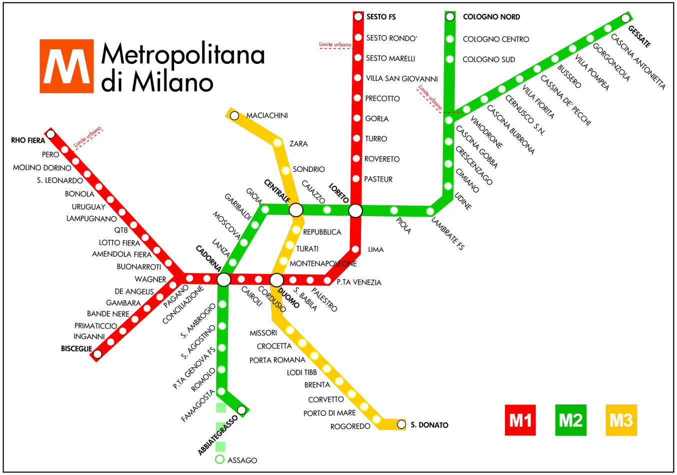
What are Description Logics today?

In the modern view, description logics are a family of logics that allow to speak about a domain composed of a set of generic (pointwise) objects, organized in classes, and related one another via various binary relations.
Abstractly, description logics allows to predicate about labeled directed graphs

- vertexes represents real world objects
- vertexes's labels represents qualities of objects
- edges represents relations between (pairs of) objects
- vertexes' labels represents the types of relations between objects.

Every piece of world that can be abstractly represented in terms of a labeled directed graph is a good candidate for being formalized by a DL.

What are Description Logics about?

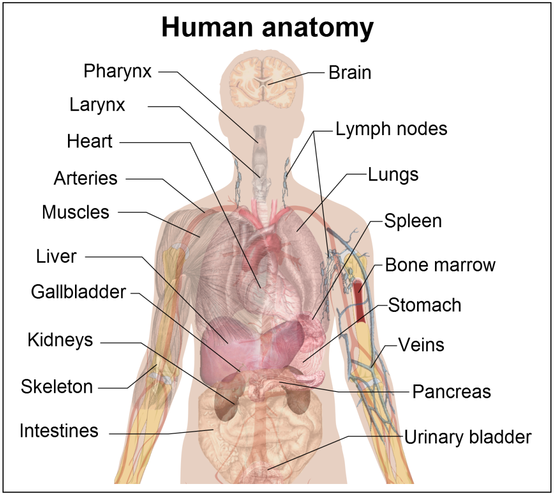


Exercise

What are Description Logics about?

Exercise

What are Description Logics about?



Exercise

What are Description Logics about?

Exercise

Represent some aspects of document classification as a labelled

Ingredients of a Description Logic

A DL is characterized by:

1. A description language: how to form concepts and roles

$$\text{Human} \sqcap \text{Male} \sqcap \exists \text{hasChild}.\top \sqcap \forall \text{hasChild}.(\text{Doctor} \sqcup \text{Lawyer})$$

2. A mechanism to specify knowledge about concepts and roles (i.e., a TBox)

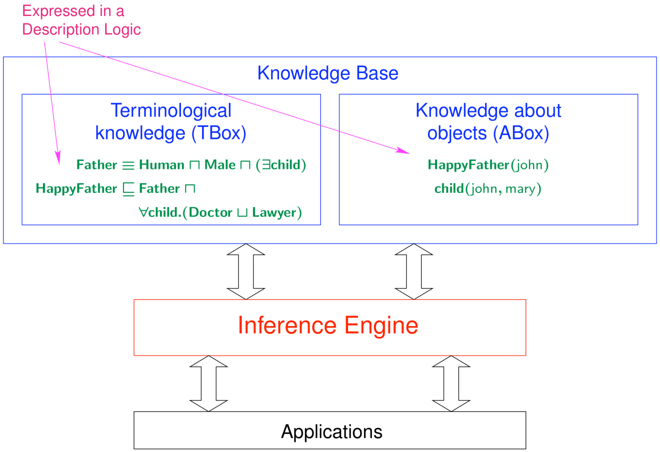
$$\mathcal{T} = \left\{ \begin{array}{l} \text{Father} \equiv \text{Human} \sqcap \text{Male} \sqcap \exists \text{hasChild}.\top \\ \text{HappyFather} \sqsubseteq \text{Father} \sqcap \forall \text{hasChild}.(\text{Doctor} \sqcup \text{Lawyer}) \\ \textit{hasFather} \sqsubseteq \textit{hasParent} \end{array} \right\}$$

3. A mechanism to specify properties of objects (i.e., an ABox)

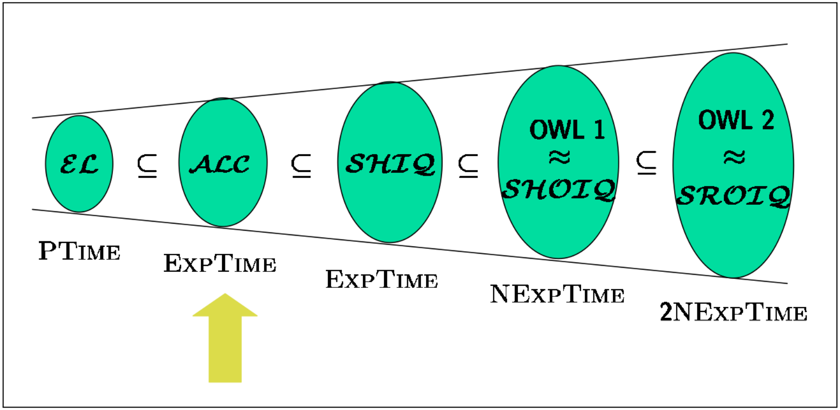
$$\mathcal{A} = \{\textit{HappyFather}(\textit{john}),\ \textit{hasChild}(\textit{john}, \textit{mary})\}$$

4. A set of inference services that allow to infer new properties on concepts, roles and objects, which are logical consequences of those explicitly asserted in the T-box and in the A-box

Architecture of a Description Logic system


Expressed in a Description Logic
Knowledge Base
Terminological knowledge (TBox)
Father ≡ Human ⊓ Male ⊓ (∃child)
HappyFather ⊑ Father ⊓ ∀child.(Doctor ⊔ Lawyer)
Knowledge about objects (ABox)
HappyFather(john)
child(john, mary)
Inference Engine
Applications

Many description logics


$\mathcal{EL}$ ⊆ $\mathcal{ALC}$ ⊆ $\mathcal{SHIQ}$ ⊆ OWL 1 ≈ $\mathcal{SHOIQ}$ ⊆ OWL 2 ≈ $\mathcal{SROIQ}$
PTime
ExpTime
ExpTime
NExpTime
2NExpTime

The description logics $\mathcal{ALC}$: Syntax

Alphabet

The alphabet Σ of $\mathcal{ALC}$ is composed of:

Σ_C:	Concept names	corresponding to node labels
Σ_R:	Role names	corresponding to arc labels
Σ_I:	Individual names	nodes identifiers

Grammar

Concept	$C := A \mid \neg C \mid C \sqcap C \mid \exists R.C$	$A \in \Sigma_C,\ R \in \Sigma_R$
Definition	$A \doteq C$	$A \in \Sigma_C$
Subsumption	$C \sqsubseteq C$	
Assertion	$C(a) \mid R(a,b)$	$a, b \in \Sigma_I,\ R \in \Sigma_R$

The description logics $\mathcal{ALC}$: Syntax

Abbreviations

$\top$	$A \sqcap \neg A$	for some $A \in \Sigma_C$
$\bot$	$\neg\top$	
$C \sqcup D$	$\neg(\neg C \sqcap \neg D)$	
$\forall R.C$	$\neg\exists R.(\neg C)$	
$C \equiv D$	$\{C \sqsubseteq D, D \sqsubseteq C\}$	

Exercise

Define Σ for speaking about the metro in Milan, and give examples of Concepts, Definitions, Subsumptions, and Assertions

The description logics $\mathcal{ALC}$: Syntax

Solution

	Concept Names (Σ_C):
Station	the set of metro stations
RedLineStation	the set of metro stations on the red line
ExchangeStation	the set of metro stations in which it is possible to exchange line

	Role Names (Σ_R):
Next	the relation between one station and its next stations

	Individual Names (Σ_I):
Centrale	the station called "Centrale" ...
Gioia	...
⋮	

The description logics $\mathcal{ALC}$: Syntax

Solution (Cont'd)

Concepts	
$RedLineStation \sqcap GreenLineStation$	*the set of stations which are on both red and green line*
$ExchangeStation \sqcap RedLineStation$	*the set of exchange stations of the red line*
$Station \sqcap \exists Next.RedLineStation$	*the set of stations which has a next station on the red line*
$Station \sqcap \forall Next.\bot$	*The set of End stations*

Definition

$$RGExchangeStation \doteq RedLineStation \sqcap GreenLineStation$$
$$RYExchangeStation \doteq RedLineStation \sqcap YellowLineStation$$
$$GYExchangeStation \doteq GreenLineStation \sqcap YellowLineStation$$
$$ExchangeStation \doteq RGExchangeStation \sqcup RYExchangeStation \sqcup GYExchangeStation$$

The description logics $\mathcal{ALC}$: Syntax

Solution (Cont'd)

	Subsumptions
$RedLineStation \sqsubseteq Station$	*A red line station is a station*
$\top \sqsubseteq \forall Next.Station$	*everything next to something is a station*
$\exists Next.\top \sqsubseteq Station$	*everything that has something next must be a station*

	Subsumptions
$GreenLineStation(Gioia)$	*"Gioia" is a station of the green line*
$RGExchangeStation(Loreto)$	*"Loreto" is an exchange station between the green and the red line*
$Next(Loreto,Lima)$	*"Lima" is a next stop of "Loreto"*
$\neg Next(Loreto, Duomo)$	*"Duomo" is not next to "Loreto"*

The description logics $\mathcal{ALC}$: Semantics

Definition

A DL interpretation $\mathcal{I}$ is pair $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ where:

- $\Delta^{\mathcal{I}}$ is a non empty set called interpretation domain
- $\cdot^{\mathcal{I}}$ is an interpretation function of the alphabet Σ such that
  - $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, every concept name is mapped into a subset of the interpretation domain
  - $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, every role name is mapped into a binary relation on the interpretation domain
  - $o^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ every individual is mapped into an element of the interpretation domain.

The description logics $\mathcal{ALC}$: Semantics

Interpretation of Complex concepts

$$\begin{aligned}
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(\exists R.C)^{\mathcal{I}} &= \{d \in \Delta^{\mathcal{I}} \mid \text{exists } d', \langle d, d' \rangle \in R^{\mathcal{I}} \text{ and } d' \in C^{\mathcal{I}}\}
\end{aligned}$$

Exercise

Provide the definition of the interpretations of the abbreviations:

$$\begin{aligned}
(\top)^{\mathcal{I}} &= \ldots \\
(\bot)^{\mathcal{I}} &= \ldots \\
(C \sqcup D)^{\mathcal{I}} &= \ldots \\
(\forall R.C)^{\mathcal{I}} &= \ldots
\end{aligned}$$

The description logics $\mathcal{ALC}$: Semantics

Satisfaction relation $\models$

$$\begin{aligned}
\mathcal{I} \models A \doteq C & \quad \textit{iff} \quad A^{\mathcal{I}} = C^{\mathcal{I}} \\
\mathcal{I} \models C \sqsubseteq D & \quad \textit{iff} \quad C^{\mathcal{I}} \subseteq D^{\mathcal{I}} \\
\mathcal{I} \models C(a) & \quad \textit{iff} \quad a^{\mathcal{I}} \in C^{\mathcal{I}} \\
\mathcal{I} \models R(a,b) & \quad \textit{iff} \quad \langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}}
\end{aligned}$$

Satisfiability of a concept

A concept C is satisfiable if there is an interpretation $\mathcal{I}$, such that

$$C^{\mathcal{I}} \neq \emptyset$$

$\mathcal{ALC}$ knowledge base

Definition (Knowledge Base)

A knowledge base $\mathcal{K}$ is a pair $(\mathcal{T}, \mathcal{A})$, wehre

- $\mathcal{T}$, called the Terminological box (T-box), is a set of concept definition and subsumptions
- $\mathcal{A}$, called the Assertional box (A-box), is a set of assertions

Logical Consequence $\models$

A subsumption/assertion ϕ is a logical consequence of $\mathcal{T}$, $\mathcal{T} \models \phi$, if ϕ is satisfied by all interpretations that satisfies $\mathcal{T}$,

Satisfiability of a concept w.r.t, $\mathcal{T}$

A concept C is satisfiable w.r.t., $\mathcal{T}$ if there is an interpretation that satisfies $\mathcal{T}$ and such that

$$C^{\mathcal{I}} \neq \emptyset$$

$\mathcal{ALC}$ and Modal Logics

Remark

There is a strict relation between $\mathcal{ALC}$ and multi modal logics

$\mathcal{ALC}$	$\longleftrightarrow$	Multi Modal Logics
$\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$	$\longleftrightarrow$	$\mathcal{M} = \langle W, R_1, \dots, R_n, \nu \rangle$
object o	$\longleftrightarrow$	world w
domain $\Delta^{\mathcal{I}}$	$\longleftrightarrow$	set of possible worlds W
concept name A	$\longleftrightarrow$	propositional variable A
concept interpretation $A^{\mathcal{I}}$	$\longleftrightarrow$	evaluation $\nu(A)$
role name R	$\longleftrightarrow$	modality $\Box_i$
role interpretation $R^{\mathcal{I}}$	$\longleftrightarrow$	accessibility relation R_i
$\exists R \dots$	$\longleftrightarrow$	$\Diamond_i \dots$
$\neg C$	$\longleftrightarrow$	$\neg C$
$C \sqcap D$	$\longleftrightarrow$	$C \wedge D$
$\mathcal{I} \models C(a)$	$\longleftrightarrow$	$\mathcal{M}, w_a \models C$

$\mathcal{ALC}$ and Modal Logics

$\mathcal{ALC}$ and Multi Modal Logics are equivalent

The logic $\mathcal{ALC}$ in the language $\Sigma = \Sigma_C \cup \Sigma_R$ (i.e., with no individuals), is equivalent to the multi-modal logic K defined on the set of propositions Σ_C and the set of modalities $\Diamond_R$ with $R \in \Sigma_R$.

Theorem (From $\mathcal{ALC}$ to multi modal K)

Let $\cdot^*$ be a transformation that replace $\sqcap$ with $\wedge$, and $\exists R$ with $\Diamond_R$,

$$\models_{\mathcal{ALC}} C \sqsubseteq D \quad \Rightarrow \quad \models_K C^* \rightarrow D^*$$

Theorem (From multi modal K to $\mathcal{ALC}$)

Let $\cdot^+$ be a transformation that replace $\wedge$ with $\sqcap$, and $\Diamond_R$ with $\exists R$,

$$\models_K C \quad \Rightarrow \quad \models_{\mathcal{ALC}} \top \sqsubseteq C^+$$

Axiomatization of $\mathcal{ALC}$ (via Modal Logic)

Axioms for $\mathcal{ALC}$

- $\top \sqsubseteq \phi[p_1, \dots, p_n / C_1, \dots, C_n]$
  where ϕ is a propositional **valid formula** on the propositional variables $p_1, \dots, p_n$, $C_1, \dots, C_n$ are $\mathcal{ALC}$ concept expressions for, and $\phi[p_1, \dots, p_n / C_1, \dots, C_n]$, denotes the simultaneous substitution of $p_1, \dots, p_n$ with $C_1, \dots, C_n$, and of $\wedge$ with $\sqcap$.
- $\top \sqsubseteq \neg\forall R.(\neg C \sqcup B) \sqcup \neg\forall R.C \sqcup \forall R.D$
  (Translation of $\Box_R(C \to D) \to (\Box_R C \to \Box_R D)$ K axiom)
- $\dfrac{\top \sqsubseteq C \qquad C \sqsubseteq D}{\top \sqsubseteq D} MP$ (translation of $\dfrac{C \qquad C \to D}{D} MP$)
- $\dfrac{\top \sqsubseteq C}{\top \sqsubseteq \forall R.C} Nec$ (translation of $\dfrac{C}{\Box_R C} Nec$)

$\mathcal{ALC}$ and First Order Logic

Remark

There is also a strong relation between $\mathcal{ALC}$ and function free first order logics with unary and binary predicates

$\mathcal{ALC}$	$\longleftrightarrow$	First order logic
	$\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$	
concept name A	$\longleftrightarrow$	unary predicate $A(x)$
role name R	$\longleftrightarrow$	binary predicate $R(x, y)$
$\exists R.C$	$\longleftrightarrow$	$\exists y(R(x, y) \wedge C(y))$
$\neg C$	$\longleftrightarrow$	$\neg C(x)$
$C \sqcap D$	$\longleftrightarrow$	$C(x) \wedge D(x)$
	$\mathcal{I} \models C(a)$	
$\mathcal{I} \models C \sqsubseteq D$	$\longleftrightarrow$	$\mathcal{I} \models \forall x(C(x) \rightarrow D(x))$

$\mathcal{ALC}$ and First Order Logics

Exercise

Define a transformation $\cdot^*$ from $\mathcal{ALC}$ concepts to first order formulas such that the following proposition is true

$$\models_{\mathcal{ALC}} \top \sqsubseteq C \quad \Rightarrow \quad \models_{FOL} C^*$$

Solution

$$\begin{aligned} ST^x(A) &= A(x) \\ ST^x(A \sqcap B) &= ST^x(A) \wedge ST^x(B) \\ ST^x(\neg A) &= \neg ST^x(A) \\ ST^x(\exists R.A) &= \exists y(R(x,y) \wedge ST^y(A)) \end{aligned}$$

$\mathcal{ALC}$ and First Order Logics

Exercise

Define a transformation $\cdot^*$ from $\mathcal{ALC}$ concepts to first order formulas such that the following proposition is true

$$\models_{\mathcal{ALC}} \top \sqsubseteq C \quad \Rightarrow \quad \models_{FOL} C^*$$

Solution

$$\begin{array}{rcl} ST^x(A) & = & A(x) \\ ST^x(A \sqcap B) & = & ST^x(A) \wedge ST^x(B) \\ ST^x(\neg A) & = & \neg ST^x(A) \\ ST^x(\exists R.A) & = & \exists y(R(x,y) \wedge ST^y(A)) \end{array}$$

$\mathcal{ALC}$ and First Order Logics

Exercise

Define a transformation $\cdot^*$ from $\mathcal{ALC}$ concepts to first order formulas such that the following proposition is true

$$\models_{\mathcal{ALC}} \top \sqsubseteq C \quad \Rightarrow \quad \models_{FOL} C^*$$

Solution

$$\begin{array}{rcl}
ST^x(A) & = & A(x) \\
ST^x(A \sqcap B) & = & ST^x(A) \wedge ST^x(B) \\
ST^x(\neg A) & = & \neg ST^x(A) \\
ST^x(\exists R.A) & = & \exists y(R(x,y) \wedge ST^y(A))
\end{array}$$

From First Order Logic to $\mathcal{ALC}$

Exercise

Is it possible to define a transformation $\cdot^{+}$ from function free first order formulas on unary and binary predicates such that the following is true?

$$\models_{FOL} \phi \quad \Rightarrow \quad \models_{\mathcal{ALC}} \top \sqsubseteq \phi^{+}$$

- if yes specify the transformation
- if not provide a formal proof

$\mathcal{ALC}$ Basic Inference Problems

Concept subsumption

$\models C \sqsubseteq D$ $C^\mathcal{I} \subseteq D^\mathcal{I}$ in all interpretations $\mathcal{I}$?

Concept Subsumption w.r.t. T-Box $\mathcal{T}$

$\models C \sqsubseteq_\mathcal{T} D$ $C^\mathcal{I} \subseteq D^\mathcal{I}$ in all interpretations $\mathcal{I}$ that satisfies $\mathcal{T}$?

Concept consistency

Is C consistent? there exists an interpretation $\mathcal{I}$ such that $C^\mathcal{I} \neq \emptyset$?

Consistency w.r.t a Tbox $\mathcal{T}$

Is C consistent w.r.t. $\mathcal{T}$? Is there a model $\mathcal{I}$ of $\mathcal{T}$ with $C^\mathcal{I} \neq \emptyset$?

$\mathcal{ALC}$ Basic Inference Problems

Concept subsumption

$\models C \sqsubseteq D$ $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$?

Concept Subsumption w.r.t. T-Box $\mathcal{T}$

$\models C \sqsubseteq_{\mathcal{T}} D$ $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$ that satisfies $\mathcal{T}$?

Concept consistency

Is C consistent? there exists an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$?

Consistency w.r.t a Tbox $\mathcal{T}$

Is C consistent w.r.t. $\mathcal{T}$? Is there a model $\mathcal{I}$ of $\mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$?

$\mathcal{ALC}$ Basic Inference Problems

Concept subsumption

$\models C \sqsubseteq D$ $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$?

Concept Subsumption w.r.t. T-Box $\mathcal{T}$

$\models C \sqsubseteq_{\mathcal{T}} D$ $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$ that satisfies $\mathcal{T}$?

Concept consistency

Is C consistent? there exists an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$?

Consistency w.r.t a Tbox $\mathcal{T}$

Is C consistent w.r.t. $\mathcal{T}$? Is there a model $\mathcal{I}$ of $\mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$?

$\mathcal{ALC}$ Basic Inference Problems

Concept subsumption

$\models C \sqsubseteq D$ $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$?

Concept Subsumption w.r.t. T-Box $\mathcal{T}$

$\models C \sqsubseteq_{\mathcal{T}} D$ $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all interpretations $\mathcal{I}$ that satisfies $\mathcal{T}$?

Concept consistency

Is C consistent? there exists an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}} \neq \emptyset$?

Consistency w.r.t a Tbox $\mathcal{T}$

Is C consistent w.r.t. $\mathcal{T}$? Is there a model $\mathcal{I}$ of $\mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$?

$\mathcal{ALC}$ Basic Inference Problems

Concept subsumption

$\models C \sqsubseteq D \qquad \Longrightarrow \qquad C \sqcap \neg D$ is not Consistent

Concept Subsumption w.r.t. T-Box $\mathcal{T}$

$\models C \sqsubseteq_{\mathcal{T}} D \qquad \Longrightarrow \qquad C \sqcap \neg D$ is not Consistent w.r.t. $\mathcal{T}$

$\mathcal{ALC}$ Basic Inference Problems

Concept subsumption problem can be reduced into Concept consistency as follows:

Consistency of an A-Box $\mathcal{A}$

Is $\mathcal{A}$ consistent i.e., is there a model $\mathcal{I}$ of $\mathcal{A}$?

Consistency of a T-Box $\mathcal{T}$

Is $\mathcal{T}$ consistent i.e., is there a model $\mathcal{I}$ of $\mathcal{T}$?

Consistency of a knowledge base $\mathcal{K}$

Is $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consistent i.e., is there a model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$?

$\mathcal{ALC}$ Basic Inference Problems

Concept subsumption problem can be reduced into Concept consistency as follows:

Consistency of an A-Box $\mathcal{A}$

Is $\mathcal{A}$ consistent i.e., is there a model $\mathcal{I}$ of $\mathcal{A}$?

Consistency of a T-Box $\mathcal{T}$

Is $\mathcal{T}$ consistent i.e., is there a model $\mathcal{I}$ of $\mathcal{T}$?

Consistency of a knowledge base $\mathcal{K}$

Is $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consistent i.e., is there a model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$?

$\mathcal{ALC}$ Basic Inference Problems

Concept subsumption problem can be reduced into Concept consistency as follows:

Consistency of an A-Box $\mathcal{A}$

Is $\mathcal{A}$ consistent i.e., is there a model $\mathcal{I}$ of $\mathcal{A}$?

Consistency of a T-Box $\mathcal{T}$

Is $\mathcal{T}$ consistent i.e., is there a model $\mathcal{I}$ of $\mathcal{T}$?

Consistency of a knowledge base $\mathcal{K}$

Is $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consistent i.e., is there a model $\mathcal{I}$ of $\mathcal{T}$ and $\mathcal{A}$?

$\mathcal{ALC}$ Complex Inference Tasks

Concept hierarchy

The subsumption hierarchy of $\mathcal{T}$, is a partial order on the set of primitive concepts defined as follows:

$$\{A \prec B | A, B \in \Sigma_C \text{ and } A \sqsubseteq_{\mathcal{T}} B\}$$

Individual classification

For all individual $o \in \Sigma_I$ determine all the primitive concepts $A \in \Sigma_C$, such that $\mathcal{T} \models A(o)$.

Negation Normal Form

Definition

A concept C is in negation normal form (NNF) if the $\neg$ operator is applied only to atomic concepts

Lemma

Every concept C can be reduced in an equivalent concept in NNF.

proof

A concept C can be reduced in NNF by the following rewriting rules that push inside the $\neg$ operator:

$$\begin{array}{rcl}
\neg(C \sqcap D) & \equiv & \neg C \sqcup \neg D \\
\neg(C \sqcup D) & \equiv & \neg C \sqcap \neg D \\
\neg(\neg C) & \equiv & C \\
\neg \forall R.C & \equiv & \exists R.\neg C \\
\neg \exists R.C & \equiv & \forall R.\neg C
\end{array}$$

Checking satisfiability of a concept in $\mathcal{ALC}$

Tableaux

Let C_0 be an $\mathcal{ALC}$-concept in NNF. In order to test satisfiability of C_0, the algorithm starts with $\mathcal{A}_0 := \{C_0(x_0)\}$, and applies the following rules:

Rule	Condition	$\longrightarrow$	Effect
$\rightarrow_{\sqcap}$	$C_1 \sqcap C_2(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C_1(x), C_2(x)\}$
$\rightarrow_{\sqcup}$	$C_1 \sqcup C_2(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C_1(x)\}$ or $\mathcal{A} \cup \{C_2(x)\}$
$\rightarrow_{\exists}$	$\exists R.C(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{R(x,y), C(y)\}$
$\rightarrow_{\forall}$	$\forall R.C(x), R(x,y) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C(y)\}$

Every rule is applicable only if it has an effect on $\mathcal{A}$, i.e., if it adds some new assertion; otherwise it's not applicable.

Checking satisfiability of a concept in $\mathcal{ALC}$

Definition

An ABox $\mathcal{A}$

- is complete iff none of the transformation rules applies to it.
- has a clash iff $\{C(x), \neg C(x)\} \subseteq \mathcal{A}$
- is closed if it contains a clash
- is open if it is not closed

Checking satisfiability of a concept in $\mathcal{ALC}$

Lemma

- *There cannot be an infinite sequence of rule applications*

$$\{C_0(x_0)\} \rightarrow \mathcal{A}_1 \rightarrow \mathcal{A}_2 \rightarrow \dots$$

- *If $\mathcal{A}'$ is obtained by applying a deterministic rule to $\mathcal{A}$, then $\mathcal{A}$ is consistent iff $\mathcal{A}'$ is consistent*
- *If $\mathcal{A}'$ and $\mathcal{A}''$ can be obtained by applying a non-deterministic rule to $\mathcal{A}$, then $\mathcal{A}$ is consistent iff either $\mathcal{A}'$ or $\mathcal{A}''$ are consistent*
- *Any closed ABox $\mathcal{A}$ is inconsistent.*
- *Any complete and open ABox $\mathcal{A}$ is consistent.*

Canonical model

Satisfiability of complete and open A-box

To show item 5 of previous lemma, we describe a method for generating an interpretation $\mathcal{I}_\mathcal{A}$ starting from a complete and closed A-box $\mathcal{A}$. This model is called Canonical interpretation

Canonical interpretation $\mathcal{I}_\mathcal{A}$

1. $\Delta^{\mathcal{I}_\mathcal{A}} = \{x | \text{either } C(x),\ r(x,y), \text{ or } r(y,x) \in \mathcal{A}\}$
2. $A^{\mathcal{I}_\mathcal{A}} = \{x | A(x) \in \mathcal{A}\}$
3. $R^{\mathcal{I}_\mathcal{A}} = \{(x,y) | R(x,y) \in \mathcal{A}\}$.

Theorem

It is decidable whether or not an $\mathcal{ALC}$-concept is satisfiable

Complexity of reasoning in $\mathcal{ALC}$

Exercise

Consider the concept C_n inductively defined as follows;

$$\begin{aligned} C_1 &= \exists R.A \sqcup \exists R.\neg A \\ C_{n+1} &= \exists R.A \sqcup \exists R.\neg A \sqcap \forall R.C_n \end{aligned}$$

Check the form of the canonical interpretation of the A-box generated starting form $\{C_n(x_0)\}$.

Solution

Given the input description C_n the satisfiability algorithm generates a complete and open ABox whose canonical interpretation is a binary tree of depth n, and thus consists of $2^{n+1} - 1$ individuals.

So in principle the complexity of checking sat in $\mathcal{ALC}$ is exponential in space

Complexity of reasoning in $\mathcal{ALC}$

Tree model property implies that each single branch of the A-box can be elaborated separately $\rightarrow$ NPSPACE

The fact that NPSPACE = PSPACE allows us to show that

Theorem

Satisfiability of $\mathcal{ALC}$-concept descriptions is PSPACE-COMPLETE

Exercises: Satisfiability in $\mathcal{ALC}$

Exercise

Check the satisfiability of the following concepts:

1. $\neg(\forall R.A \sqcup \exists R.(\neg A \sqcap \neg B))$
2. $\exists R.(\forall S.C) \sqcap \forall R.(\exists S.\neg C)$
3. $(\exists S.C \sqcap \exists S.D) \sqcap \forall S.(\neg C \sqcup \neg D)$
4. $\exists S.(C \sqcap D) \sqcap (\forall S.\neg C \sqcup \exists S.\neg D)$
5. $C \sqcap \exists R.A \sqcap \exists R.B \sqcap \neg\exists R.(A \sqcap B)$

Check if the following subsumption is valid

$$\neg\forall R.A \sqcap \forall R((\forall R.B) \sqcup A) \sqsubseteq \forall R.\neg(\exists R.A) \sqcap \exists R.(\exists R.B)$$

Consistency of $\mathcal{ALC}$ A-boxes

Consistency of $\mathcal{ALC}$-ABoxe

Let $\mathcal{A}_0$ be an $\mathcal{ALC}$-ABox in NNF. To test $\mathcal{A}_0$ for consistency, we simply apply the rules given above to $\mathcal{A}_0$.

Theorem

Consistency of ALC ABoxes is PSPACE-COMPLETE.

Exercise

Which of the following statements are true? Explain your answer.

1. $\forall R.(A \sqcap B) \sqsubseteq \forall R.A \sqcap \forall R.B$
2. $\forall R.A \sqcap \forall R.B \sqsubseteq \forall R.(A \sqcap B)$
3. $\forall R.A \sqcup \forall R.B \sqsubseteq \forall R.(A \sqcup B)$
4. $\forall R.(A \sqcup B) \sqsubseteq \forall R.A \sqcup \forall R.B$
   $R^{\mathcal{I}} = \{(x,y),(x,z)\},\ A^{I} = \{y\},\ B^{\mathcal{I}} = \{z\}$
5. $\exists R.(A \sqcap B) \sqsubseteq \exists R.A \sqcap \exists R.B$
6. $\exists R.(A \sqcup B) \sqsubseteq \exists R.A \sqcup \exists R.B$
7. $\exists R.A \sqcup \exists R.B \sqsubseteq \exists R.(A \sqcup B)$
8. $\exists R.A \sqcap \exists R.B \sqsubseteq \exists R.(A \sqcap B)$
   $R^{\mathcal{I}} = \{(x,y),(x,z)\},\ A^{I} = \{y\},\ B^{\mathcal{I}} = \{z\}$

Exercise

Which of the following statements are true? Explain your answer.

1. $\forall R.(A \sqcap B) \sqsubseteq \forall R.A \sqcap \forall R.B$
2. $\forall R.A \sqcap \forall R.B \sqsubseteq \forall R.(A \sqcap B)$
3. $\forall R.A \sqcup \forall R.B \sqsubseteq \forall R.(A \sqcup B)$
4. $\forall R.(A \sqcup B) \sqsubseteq \forall R.A \sqcup \forall R.B$
   $R^{\mathcal{I}} = \{(x,y),(x,z)\},\ A^{I} = \{y\},\ B^{\mathcal{I}} = \{z\}$
5. $\exists R.(A \sqcap B) \sqsubseteq \exists R.A \sqcap \exists R.B$
6. $\exists R.(A \sqcup B) \sqsubseteq \exists R.A \sqcup \exists R.B$
7. $\exists R.A \sqcup \exists R.B \sqsubseteq \exists R.(A \sqcup B)$
8. $\exists R.A \sqcap \exists R.B \sqsubseteq \exists R.(A \sqcap B)$
   $R^{\mathcal{I}} = \{(x,y),(x,z)\},\ A^{I} = \{y\},\ B^{\mathcal{I}} = \{z\}$

Reasoning in $\mathcal{ALC}$ with T-box

Subsumption w.r.t. TBoxes A subsumption $C \sqsubseteq D$ follows from a TBox $\mathcal{T}$, in symbols $\mathcal{T} \models C \sqsubseteq D$, if for every interpretation $\mathcal{I}$, if $\mathcal{I} \models \mathcal{T}$ then $\mathcal{I} \models C \sqsubseteq D$

Concept satisfiability w.r.t. TBoxes A concept C is satisfiable w.r.t. a TBox $\mathcal{T}$ if there exists an interpretation $\mathcal{I} \models \mathcal{T}$ and such that $C^{\mathcal{I}} \neq \emptyset$.

TBox satisfiability A TBox $\mathcal{T}$ is satisfiable if, there is a model of $\mathcal{T}$.

We have the following reductions to concept satisfiability w.r.t. T-Boxes:

- $\mathcal{T} \models C \sqsubseteq D$ if and only if $C \sqcap \neg D$ is not consistent w.r.t. $\mathcal{T}$.
- $\mathcal{T}$ is satisfiable if $\top$ is consistent w.r.t. $\mathcal{T}$.

$\mathcal{ALC}$ concept satisfiability w.r.t. Acyclic T-box

Definition (Acyclic T-box)

A TBox is acyclic if it is a set of concept definitions that neither contains multiple definitions nor cyclic definitions.

Multiple definitions are of the form $A \doteq C$ and $A \doteq D$ for distinct concept descriptions C and D

cyclic definitions are of the form

$$A_1 \doteq C_1[A_2],\ A_2 \doteq C_2[A_3],\ \dots,\ A_n \doteq C_n[A_1]$$

where $C[A]$ means that the atomic concept A occurs in the complex concept description C.

$\mathcal{ALC}$ concept satisfiability w.r.t. Acyclic T-box

Naive reduction to $\mathcal{ALC}$ satisfiability

Satisfiability w.r.t. acyclic T-box can be reduced to $\mathcal{ALC}$ satisfiability without T-Boxes by unfolding the definitions

Unfolding: repeatedly replacing defined names by their defining concepts until no more defined names occur.

Exponential blow up

Unfolding may lead to an exponential blow-up,

$$\begin{aligned} A_0 &\doteq \forall R.A_1 \sqcap \forall S.A_1 \\ A_1 &\doteq \forall R.A_2 \sqcap \forall S.A_2 \\ &\vdots \\ A_{n-1} &\doteq \forall R.A_n \sqcap \forall S.A_n \end{aligned}$$

$\mathcal{ALC}$ concept satisfiability w.r.t. Acyclic T-box

Smarter strategy - Unfolding on demand

Rule	Condition	$\longrightarrow$	Effect
$\rightarrow_\sqcap$	$C_1 \sqcap C_2(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C_1(x), C_2(x)\}$
$\rightarrow_\sqcup$	$C_1 \sqcup C_2(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C_1(x)\}$ or $\mathcal{A} \cup \{C_2(x)\}$
$\rightarrow_\exists$	$\exists R.C(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{R(x,y), C(y)\}$
$\rightarrow_\forall$	$\forall R.C(x), R(x,y) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C(y)\}$
$\rightarrow_\mathcal{T}$	$A(x) \in \mathcal{A}$ and $A \doteq C \in \mathcal{T}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup NNF(C)(x)$

Theorem

Satisfiability w.r.t. acyclic terminologies is PSPACE-COMPLETE *in* $\mathcal{ALC}$.

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Intuition

1. $C \sqsubseteq D$ is equivalent to $\top \sqsubseteq \neg C \sqcup D$
2. $\top \sqsubseteq \neg C \sqcup D$ is an axiom of $\mathcal{T}$ then for every A-box generated by the tableaux and for ever occurrence of x in $\mathcal{A}$, we have to add also the fact $\neg C \sqcup D(x)$
3. so it's just a matter of extending the set of rules as follows:

Rule	Condition	$\longrightarrow$	Effect
$\rightarrow_\sqcap$	$C_1 \sqcap C_2(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C_1(x), C_2(x)\}$
$\rightarrow_\sqcup$	$C_1 \sqcup C_2(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C_1(x)\}$ or $\mathcal{A} \cup \{C_2(x)\}$
$\rightarrow_\exists$	$\exists R.C(x) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{R(x,y), C(y)\}$
$\rightarrow_\forall$	$\forall R.C(x), R(x,y) \in \mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup \{C(y)\}$
$\rightarrow_\mathcal{T}$	x occurs in $\mathcal{A}$	$\longrightarrow$	$\mathcal{A} := \mathcal{A} \cup NNF(\neg C \sqcup D)(x)$

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$
\begin{aligned}
\{C(x_0)\} \quad & \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\
& \rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\} \\
& \rightarrow_{\mathcal{T}} \{C(x_0), R(x_0, x_1), C(x_1), C \sqcup \exists R.C(x_1)\} \\
& \rightarrow_{\sqcup} \{C(x_0), R(x_0, x_1), C(x_1) \exists R.C(x_1)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1), R(x_1, x_2), C(x_2)\} \\
& \rightarrow_{\mathcal{T}} \ldots
\end{aligned}
$$

Termination

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$\begin{aligned}
\{C(x_0)\} \quad &\rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\
&\rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\
&\rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\} \\
&\rightarrow_{\mathcal{T}} \{C(x_0), R(x_0, x_1), C(x_1), C \sqcup \exists R.C(x_1)\} \\
&\rightarrow_{\sqcup} \{C(x_0), R(x_0, x_1), C(x_1) \exists R.C(x_1)\} \\
&\rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1), R(x_1, x_2), C(x_2)\} \\
&\rightarrow_{\mathcal{T}} \dots
\end{aligned}$$

Termination

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$
\begin{aligned}
\{C(x_0)\} \quad & \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\
& \rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\} \\
& \rightarrow_{\mathcal{T}} \{C(x_0), R(x_0, x_1), C(x_1), C \sqcup \exists R.C(x_1)\} \\
& \rightarrow_{\sqcup} \{C(x_0), R(x_0, x_1), C(x_1)\exists R.C(x_1)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1), R(x_1, x_2), C(x_2)\} \\
& \rightarrow_{\mathcal{T}} \ldots
\end{aligned}
$$

Termination

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$
\begin{aligned}
\{C(x_0)\} \quad & \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\
& \rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\} \\
& \rightarrow_{\mathcal{T}} \{C(x_0), R(x_0, x_1), C(x_1), C \sqcup \exists R.C(x_1)\} \\
& \rightarrow_{\sqcup} \{C(x_0), R(x_0, x_1), C(x_1) \exists R.C(x_1)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1), R(x_1, x_2), C(x_2)\} \\
& \rightarrow_{\mathcal{T}} \dots
\end{aligned}
$$

Termination

[illegible]

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$
\begin{aligned}
\{C(x_0)\} \quad & \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\
& \rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\} \\
& \rightarrow_{\mathcal{T}} \{C(x_0), R(x_0, x_1), C(x_1), C \sqcup \exists R.C(x_1)\} \\
& \rightarrow_{\sqcup} \{C(x_0), R(x_0, x_1), C(x_1)\exists R.C(x_1)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1), R(x_1, x_2), C(x_2)\} \\
& \rightarrow_{\mathcal{T}} \dots
\end{aligned}
$$

Termination

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$\begin{aligned}
\{C(x_0)\} \quad & \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\
& \rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\} \\
& \rightarrow_{\mathcal{T}} \{C(x_0), R(x_0, x_1), C(x_1), C \sqcup \exists R.C(x_1)\} \\
& \rightarrow_{\sqcup} \{C(x_0), R(x_0, x_1), C(x_1)\exists R.C(x_1)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1), R(x_1, x_2), C(x_2)\} \\
& \rightarrow_{\mathcal{T}} \dots
\end{aligned}$$

Termination

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$
\begin{aligned}
\{C(x_0)\} \quad & \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\
& \rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\} \\
& \rightarrow_{\mathcal{T}} \{C(x_0), R(x_0, x_1), C(x_1), C \sqcup \exists R.C(x_1)\} \\
& \rightarrow_{\sqcup} \{C(x_0), R(x_0, x_1), C(x_1) \exists R.C(x_1)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1), R(x_1, x_2), C(x_2)\} \\
& \rightarrow_{\mathcal{T}} \dots
\end{aligned}
$$

Termination

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$
\begin{aligned}
\{C(x_0)\} \quad & \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\
& \rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\} \\
& \rightarrow_{\mathcal{T}} \{C(x_0), R(x_0, x_1), C(x_1), C \sqcup \exists R.C(x_1)\} \\
& \rightarrow_{\sqcup} \{C(x_0), R(x_0, x_1), C(x_1)\exists R.C(x_1)\} \\
& \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1), R(x_1, x_2), C(x_2)\} \\
& \rightarrow_{\mathcal{T}} \ldots
\end{aligned}
$$

Termination

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Blocking

- y is an ancestor of y in an A-box $\mathcal{A}$, if $\mathcal{A}$ contains

$$R_0(y, x_1), R_1(x_1, x_2), \dots, R_n(x_n, x)$$

- $\mathcal{L}(x) = \{C | C(x) \in \mathcal{A}\}$
- x is directly blocked in $\mathcal{A}$ if it has an ancestor y with $\mathcal{L}(x) \subseteq \mathcal{L}(y)$
- if y is the closest such node to x, we say that x is blocked by y
- A node is blocked if it is directly blocked or one of its ancestors is blocked

Restriction

Restrict the application of all rules to nodes which are not blocked

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$\begin{array}{ll} \{C(x_0)\} & \rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\ & \rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\ & \rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\} \end{array}$$

x_1 is blocked by x_0 since

$$\mathcal{L}(x_1) = \{C\} \subseteq \mathcal{L}(x_0) = \{C, \exists R.C\}$$

Termination

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Exercise

Check if C is satisfiable w.r.t. the T-box $\{C \sqsubseteq \exists R.C\}$

Solution

$$\begin{aligned}\{C(x_0)\} \quad &\rightarrow_{\mathcal{T}} \{C(x_0), \neg C \sqcup \exists R.C(x_0)\} \\ &\rightarrow_{\sqcup} \{C(x_0), \exists R.C(x_0)\} \\ &\rightarrow_{\exists} \{C(x_0), R(x_0, x_1), C(x_1)\}\end{aligned}$$

x_1 *is blocked by* x_0 *since*

$$\mathcal{L}(x_1) = \{C\} \subseteq \mathcal{L}(x_0) = \{C, \exists R.C\}$$

Termination

$\mathcal{ALC}$ concept satisfiability w.r.t. T-box

Cyclic interpretations

The interpretation $\mathcal{I}_\mathcal{A}$ generated from an A-box $\mathcal{A}$ obtained by the tableaux algorithm with blocking strategy is defined as follows:

- $\Delta^{\mathcal{I}_\mathcal{A}} = \{x \mid C(x) \in \mathcal{A}$ and x is not blocked$\}$
- $A^{\mathcal{I}_\mathcal{A}} = \{x \in \Delta^{\mathcal{I}_\mathcal{A}} \mid A(x) \in \mathcal{A}\}$
- $R^{\mathcal{I}_\mathcal{A}} = \{(x,y) \in \Delta^{\mathcal{I}_\mathcal{A}} \times \Delta^{\mathcal{I}_\mathcal{A}} \mid R(x,y) \in \mathcal{A}\} \cup$
  $\{(x',x) \mid x' \in \Delta^{\mathcal{I}_\mathcal{A}},\ R(x',x) \in \mathcal{A}$, and x is blocked by $y\}$

Complexity

The algorithm is no longer in PSPACE since it may generate role paths of exponential length before blocking occurs.

Theorem

Satisfiability of an $\mathcal{ALC}$ concept w.r.t. general T-box is EXPTIME-COMPLETE

Finite model property

Theorem

A consistent T-box in $\mathcal{ALC}$ has a finite model

proof

The model constructed via tableaux is finite. Completeness of the tableaux procedure implies that if a T-box is consistent, then the algorithm will find a model, which is indeed finite

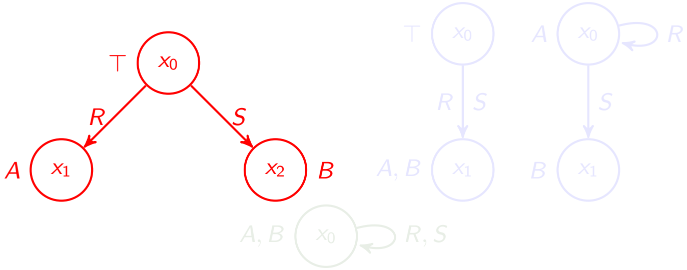
Finite model property

Remark

The tableaux method reported here does not generate neither the smallest, nor the more intuitive model. For instance, to check the consistency of the following T-box

$$\{\top \sqsubseteq \exists R.A \sqcap \exists S.B\}$$

the algorithm returns the following model, But there are also the following, maybe more appropriate, and smaller models



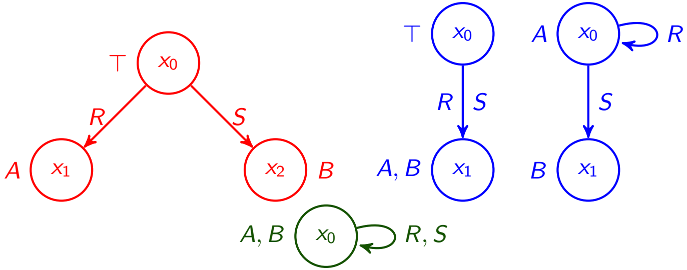
Finite model property

Remark

The tableaux method reported here does not generate neither the smallest, nor the more intuitive model. For instance, to check the consistency of the following T-box

$$\{\top \sqsubseteq \exists R.A \sqcap \exists S.B\}$$

the algorithm returns the following model, But there are also the following, maybe more appropriate, and smaller models



Extensions of $\mathcal{ALC}$

Number restrictions $\mathcal{ALCN}$ $(\leq n)R$ $[(\geq n)R]$

$$\mathit{Persons} \sqsubseteq (\leq 1)\text{is\_merried\_with}$$

Number restriction allows to impose that a relation is a function

Qualified Number restrictions $\mathcal{ALCQ}$ $(\leq n)R.C$ $[(\geq n)R.C]$

$$\begin{aligned}\text{football\_team} \quad \sqsubseteq \quad & (\geq 1)\text{has\_player.Golly} \sqcap \\ & (\leq 2)\text{has\_player.Golly} \sqcap \\ & (\geq 2)\text{has\_player.Defensor} \sqcap \\ & (\geq 4)\text{has\_player.Defensor} \sqcap \\ & \ldots\end{aligned}$$

Extensions of $\mathcal{ALC}$

Inverse roles $\mathcal{ALCI}$ R^-. make it possible to use the inverse of a role. For example, we can specify has_Parent as the inverse of has_Child,

$$\text{has\_Parent} \equiv \text{has\_Child}^-$$

meaning that
$\text{hasParent}^\mathcal{I} = \{(y, x) \mid (x, y) \in \mathit{has\_ChildI}^\mathcal{I}\}$

Transitive roles tr(R) used to state that a given relation is transitive

$$Tr(hasAncestor)$$

meaning that $(x, y), (y, z) \in \text{hasAncestor}^\mathcal{I} \rightarrow (x, z) \in \text{hasAncestor}^\mathcal{I}$

Subsumptions between roles $R \sqsubseteq S$ used to state that a relation is contained in another relation.

$$hasMother \sqsubseteq hasParent$$

Modeling with Inverse role

Exercise

Try to model the following facts in $\mathcal{ALCI}$. (notice that not all the statements are modellable in $\mathcal{ALCI}$)

1. Lonely people do not have friends and are not friends of anybody
2. An intermediate stop is a stop which has a predecessor stop and a successor stop
3. one of the next stop of the next stop is the current stop

Modeling with Inverse role

Solution

1. *Lonely people do not have friends and are not friends of anybody*

   $$lonely\_person \equiv person \sqcap \neg\exists has\_friend^{-}.\top \sqcap \neg\exists has\_friend.\top$$

2. *An intermediate stop is a stop which has a predecessor stop and a successor stop*

   $$Intermediate\_stop \equiv Stop \sqcap \exists next.Stop \sqcap \exists next^{-}.Stop$$

3. *one of the next stop of the next stop is the current stop*

   Not modellable

Modeling with Inverse role

Solution

1. *Lonely people do not have friends and are not friends of anybody*

   $$lonely\_person \equiv person \sqcap \neg\exists has\_friend^{-}.\top \sqcap \neg\exists has\_friend.\top$$

2. *An intermediate stop is a stop which has a predecessor stop and a successor stop*

   $$Intermediate\_stop \equiv Stop \sqcap \exists next.Stop \sqcap \exists next^{-}.Stop$$

3. *one of the next stop of the next stop is the current stop*

   Not modellable

Modeling with Inverse role

Solution

1. *Lonely people do not have friends and are not friends of anybody*

$$lonely\_person \equiv person \sqcap \neg\exists has\_friend^{-}.\top \sqcap \neg\exists has\_friend.\top$$

2. *An intermediate stop is a stop which has a predecessor stop and a successor stop*

$$Intermediate\_stop \equiv Stop \sqcap \exists next.Stop \sqcap \exists next^{-}.Stop$$

3. *one of the next stop of the next stop is the current stop*

Not modellable

Encoding number restriction with inverse and functional roles

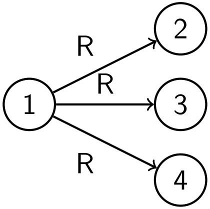
Exercise

Suppose that the concept C and T-box $\mathcal{T}$ contains number restrictions only on a single role R. Define set of axioms $\mathcal{T}_R$ such and a transformation τ from concepts of $\mathcal{ALCN}$ and $\mathcal{ALCIF}$ such that the following fact holds: C is satisfiable w.r.t. $\mathcal{T}$ in $\mathcal{ALCN}$ iff $\tau(C)$ is satisfiable w.r.t. $\tau(\mathcal{T}) \cup \mathcal{T}_R$ in $\mathcal{ALCIF}$

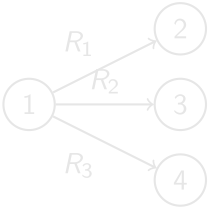
Encoding number restriction with inverse and functional roles

Intuitive solution

Replace the role R with $R_1, \ldots, R_n$ used for counting the number of R's successors.



$1 \models (\leq 3)R$
$1 \models \neg(\geq 4)R$

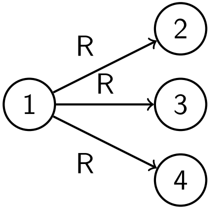


$1 \models \exists R_1.\top$
$1 \models \exists R_2.\top$
$1 \models \exists R_3.\top$
$1 \models \neg\exists R_4.\top$

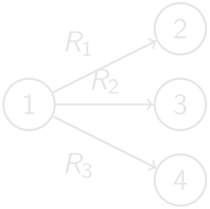
Encoding number restriction with inverse and functional roles

Intuitive solution

Replace the role R with $R_1, \ldots, R_n$ used for counting the number of R's successors.



$1 \models (\leq 3)R$
$1 \models \neg(\geq 4)R$

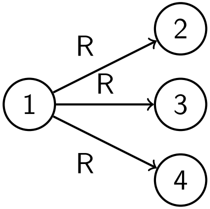


$1 \models \exists R_1.\top$
$1 \models \exists R_2.\top$
$1 \models \exists R_3.\top$
$1 \models \neg\exists R_4.\top$

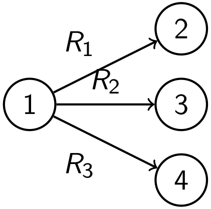
Encoding number restriction with inverse and functional roles

Intuitive solution

Replace the role R with $R_1, \ldots, R_n$ used for counting the number of R's successors.



$1 \models (\leq 3)R$
$1 \models \neg(\geq 4)R$



$1 \models \exists R_1.\top$
$1 \models \exists R_2.\top$
$1 \models \exists R_3.\top$
$1 \models \neg\exists R_4.\top$

Encoding number restriction with inverse and functional roles

Solution (Formal)

1. *n is the maximum number occurring in a number restriction of C*
2. *for every role R introduce $R_1, \dots, R_{n+1}$*
3. *for every role R_i, $\mathcal{T}_R$ contains the axioms:*
   - **3.1** $\exists R_{i+1}.\top \sqsubseteq \exists R_i.\top$ *for* $1 \leq i \leq n$
   - **3.2** $\top \sqsubseteq (\leq 1)R_i$ *for* $1 \leq i \leq n$ *(NB: R_{n+1} is not functional)*
   - **3.3** $\top \sqsubseteq \forall R_i.(\forall R_j^-.\bot)$ *for* $1 \leq i \neq j \leq n$
4. $\tau((\geq m)R) = \exists R_m.\tau(A)$
5. $\tau((\leq m)R) = \forall R_{m+1}.\neg\tau(A)$
6. $\tau(\exists R.A) = \exists R_1.\tau(A) \sqcup \cdots \sqcup \exists R_{n+1}.\tau(A)$
7. $\tau(\forall R.A) = \forall R_1.\tau(A) \sqcap \cdots \sqcap \forall R_{n+1}.\tau(A)$

Encoding number restriction with inverse and functional roles

Solution (Formal (cont'd))

We have to prove that if C is satisfiable, then $\tau(C)$ is satisfiable in $\mathcal{T}_R$.

1. *If C is satisfiable in $\mathcal{ALCN}$, then it has a tree-shaped model $\mathcal{I}$*
2. *Extend $\mathcal{I}$ into $\mathcal{J}$ with the interpretation of $R_1, \dots, R_{n+1}$ as follows. For all $d \in \Delta^{\mathcal{I}}$, let $R^{\mathcal{I}}(d) = \{d_1, \dots, d_m, \dots\}$ is the set of R-successors of d in $\mathcal{I}$, then*
   - *if $|D| < n$, then add (d, d_i) to $R_i^{\mathcal{J}}$ for $1 \leq i \leq |D|$*
   - *if $|D| \geq n$, then add (d, d_i) to $R_i^{\mathcal{J}}$ for $1 \leq i < n$ and also add (d, d_j) to $R_{n+1}^{\mathcal{J}}$ for $j \geq n+1$*
3. *Prove that $\mathcal{J}$ is a model of $\mathcal{T}_R$*
4. *Prove that $\mathcal{J}$ is a model of $\tau(C)$*

Encoding number restriction with inverse and functional roles

Solution (Formal (cont'd))

We have to prove that if C is satisfiable, then $\tau(C)$ is satisfiable in $\mathcal{T}_R$.

1. *If C is satisfiable in $\mathcal{ALCN}$, then it has a tree-shaped model $\mathcal{I}$*
2. *Extend $\mathcal{I}$ into $\mathcal{J}$ with the interpretation of $R_1, \dots, R_{n+1}$ as follows. For all $d \in \Delta^{\mathcal{I}}$, let $R^{\mathcal{I}}(d) = \{d_1, \dots, d_m, \dots\}$ is the set of R-successors of d in $\mathcal{I}$, then*
   - *if $|D| < n$, then add (d, d_i) to $R_i^{\mathcal{J}}$ for $1 \leq i \leq |D|$*
   - *if $|D| \geq n$, then add (d, d_i) to $R_i^{\mathcal{I}}$ for $1 \leq i \leq n$ and also add (d, d_j) to $R_{n+1}^{\mathcal{I}}$ for $j \geq n+1$*
3. *Prove that $\mathcal{J}$ is a model of $\mathcal{T}_R$*
4. *Prove that $\mathcal{J}$ is a model of $\tau(C)$*

Encoding number restriction with inverse and functional roles

Solution (Formal (cont'd))

We have to prove that if C is satisfiable, then $\tau(C)$ is satisfiable in $\mathcal{T}_R$.

1. *If C is satisfiable in $\mathcal{ALCN}$, then it has a tree-shaped model $\mathcal{I}$*
2. *Extend $\mathcal{I}$ into $\mathcal{J}$ with the interpretation of $R_1, \ldots, R_{n+1}$ as follows. For all $d \in \Delta^{\mathcal{I}}$, let $R^{\mathcal{I}}(d) = \{d_1, \ldots, d_m, \ldots\}$ is the set of R-successors of d in $\mathcal{I}$, then*
   - *if $|D| < n$, then add (d, d_i) to $R_i^{\mathcal{J}}$ for $1 \leq i \leq |D|$*
   - *if $|D| \geq n$, then add (d, d_i) to $R_i^{\mathcal{I}}$ for $1 \leq i \leq n$ and also add (d, d_j) to $R_{n+1}^{\mathcal{I}}$ for $j \geq n+1$*
3. *Prove that $\mathcal{J}$ is a model of $\mathcal{T}_R$*
4. *Prove that $\mathcal{J}$ is a model of $\tau(C)$*

Encoding number restriction with inverse and functional roles

Solution (Formal (cont'd))

We have to prove that if C is satisfiable, then $\tau(C)$ is satisfiable in $\mathcal{T}_R$.

1. *If C is satisfiable in $\mathcal{ALCN}$, then it has a tree-shaped model $\mathcal{I}$*
2. *Extend $\mathcal{I}$ into $\mathcal{J}$ with the interpretation of $R_1, \dots, R_{n+1}$ as follows. For all $d \in \Delta^{\mathcal{I}}$, let $R^{\mathcal{I}}(d) = \{d_1, \dots, d_m, \dots\}$ is the set of R-successors of d in $\mathcal{I}$, then*
   - *if $|D| < n$, then add (d, d_i) to $R_i^{\mathcal{J}}$ for $1 \leq i \leq |D|$*
   - *if $|D| \geq n$, then add (d, d_i) to $R_i^{\mathcal{I}}$ for $1 \leq i \leq n$ and also add (d, d_j) to $R_{n+1}^{\mathcal{I}}$ for $j \geq n+1$*
3. *Prove that $\mathcal{J}$ is a model of $\mathcal{T}_R$*
4. *Prove that $\mathcal{J}$ is a model of $\tau(C)$*

Encoding number restriction with inverse and functional roles

Solution (Formal (cont'd))

We have to prove that if C is satisfiable, then $\tau(C)$ is satisfiable in $\mathcal{T}_R$.

1. *If C is satisfiable in $\mathcal{ALCN}$, then it has a tree-shaped model $\mathcal{I}$*
2. *Extend $\mathcal{I}$ into $\mathcal{J}$ with the interpretation of $R_1, \ldots, R_{n+1}$ as follows. For all $d \in \Delta^{\mathcal{I}}$, let $R^{\mathcal{I}}(d) = \{d_1, \ldots, d_m, \ldots\}$ is the set of R-successors of d in $\mathcal{I}$, then*
   - *if $|D| < n$, then add (d, d_i) to $R_i^{\mathcal{J}}$ for $1 \leq i \leq |D|$*
   - *if $|D| \geq n$, then add (d, d_i) to $R_i^{\mathcal{I}}$ for $1 \leq i \leq n$ and also add (d, d_j) to $R_{n+1}^{\mathcal{I}}$ for $j \geq n+1$*
3. *Prove that $\mathcal{J}$ is a model of $\mathcal{T}_R$*
4. *Prove that $\mathcal{J}$ is a model of $\tau(C)$*

Encoding number restriction with inverse and functional roles

Solution (Formal (cont'd))

We have to prove that if C is satisfiable, then $\tau(C)$ is satisfiable in $\mathcal{T}_R$.

1. *If C is satisfiable in $\mathcal{ALCN}$, then it has a tree-shaped model $\mathcal{I}$*
2. *Extend $\mathcal{I}$ into $\mathcal{J}$ with the interpretation of $R_1, \dots, R_{n+1}$ as follows. For all $d \in \Delta^{\mathcal{I}}$, let $R^{\mathcal{I}}(d) = \{d_1, \dots, d_m, \dots\}$ is the set of R-successors of d in $\mathcal{I}$, then*
   - *if $|D| < n$, then add (d, d_i) to $R_i^{\mathcal{J}}$ for $1 \leq i \leq |D|$*
   - *if $|D| \geq n$, then add (d, d_i) to $R_i^{\mathcal{I}}$ for $1 \leq i \leq n$ and also add (d, d_j) to $R_{n+1}^{\mathcal{I}}$ for $j \geq n+1$*
3. *Prove that $\mathcal{J}$ is a model of $\mathcal{T}_R$*
4. *Prove that $\mathcal{J}$ is a model of $\tau(C)$*

Encoding number restriction with inverse and functional roles

Solution (Formal (cont'd))

Finally we have to prove that if $\tau(C)$ is satisfiable in $\mathcal{T}_R$, then C is satisfiable.

1. *Let $\mathcal{J}$ be a tree-shaped model of $\mathcal{T}_R$ that satisfies C.*
2. *Let $\mathcal{I}$ be obtained by extending $\mathcal{J}$ with the interpretation of R as follows $R^{\mathcal{I}} = R_1^{\mathcal{I}} \cup \cdots \cup R_{n+1}^{\mathcal{I}}$*
3. *prove by induction on C, that $\mathcal{I}$ is a model of C.*

Role hierarchy $\mathcal{H}$

Definition

Role Hierarchy A role hierarchy $\mathcal{H}$ is a finite set of role subsumptions, i.e., expressions of the form

$$R \sqsubseteq S$$

for role symbols R and S We say that R is a subrole of S

Definition

$\mathcal{I} \models R \sqsubseteq S$ if and only if $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$.

Exercise

Explain why the construct $R \sqsubseteq S$ cannot be axiomatized by the subsumptions

$$\exists R.\top \sqsubseteq \exists S.\top$$
$$\forall S.\top \sqsubseteq \forall R.\top$$

Role hierarchy $\mathcal{H}$

Definition

Role Hierarchy A role hierarchy $\mathcal{H}$ is a finite set of role subsumptions, i.e., expressions of the form

$$R \sqsubseteq S$$

for role symbols R and S We say that R is a subrole of S

Definition

$\mathcal{I} \models R \sqsubseteq S$ if and only if $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$.

Exercise

Explain why the construct $R \sqsubseteq S$ cannot be axiomatized by the subsumptions

$$\exists R.\top \sqsubseteq \exists S.\top$$
$$\forall S.\top \sqsubseteq \forall R.\top$$

Transitive roles $\mathcal{S}$

Semantic condition

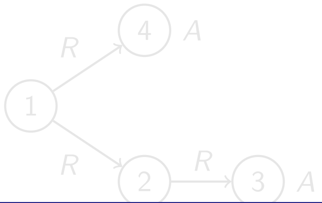
$\mathcal{I} \models tr(R)$ if $R^{\mathcal{I}}$ is a transitive relation.

Exercise

Explain why transitive roles cannot be axiomatized by the axiom

$$\exists R.(\exists R.A) \sqsubseteq \exists R.A$$

Solution



this model satisfies the axiom $\exists R.(\exists R.A) \sqsubseteq \exists R.A$ but R is not transitive

Transitive roles $\mathcal{S}$

Semantic condition

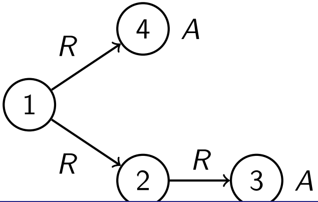
$\mathcal{I} \models tr(R)$ if $R^{\mathcal{I}}$ is a transitive relation.

Exercise

Explain why transitive roles cannot be axiomatized by the axiom

$$\exists R.(\exists R.A) \sqsubseteq \exists R.A$$

Solution



this model satisfies the axiom $\exists R.(\exists R.A) \sqsubseteq \exists R.A$ but R is not transitive

T-box internalization

Satisfiability w.r.t. T-box vs. concept satisfiability

Until now we have distinguished between the following two problems:

- Satisfiability of a concept C and
- Satisfiability of a concept C w.r.t. a T-box $\mathcal{T}$.

Clearly the first problem is a special case of the second, but with expressive languages that support role hierarchy and transitive role satisfiability w.r.t., T-box can be reduced to satisfiability.

This is like in propositional or first order logic where the problem of checking $\Gamma \models \phi$ (validity under a finite set of axioms Γ) reduces to the problem of checking the validity of a single formula. I.e., $\bigwedge \Gamma \rightarrow \phi$.

T-box internalization for logics stronger than $\mathcal{SH}$

Lemma

Representing the whole t-box in a single concept Let C a concept and $\mathcal{T} = \{A_1 \sqsubseteq B_1, \ldots, A_n \sqsubseteq B_n\}$ be a finite set of GCI.

$$C_{\mathcal{T}} = \prod_{i=1}^{n} \neg A_i \sqcup B_i$$

Let U be a new transitive role, and let

$$\mathcal{R}_U = \{R \sqsubseteq U | \textit{for all role } R \textit{ appearing in } C \textit{ and } \mathcal{T}\}$$

C is satisfiable w.r.t., $\mathcal{T}$ iff $C \sqcap C_{\mathcal{T}} \sqcap \forall U.C_{\mathcal{T}}$ is satisfiable w.r.t. $\mathcal{R}_U$

Nominals (a.k.a. Objects) $\mathcal{O}$

Explicit definition of concepts

In many cases it is convenient to define a set (concept) by explicitly enumerating its members. E.g.,

$$WeekDay \equiv \left\{ \begin{array}{c} Monday, Friday, Saturday, Sunday, \\ Thursday, Tuesday, Wednesday \end{array} \right\}$$

Nominals

A nominal is a concept with cardinality equal to 1, it represents a singleton set. If o is an individual then, the expression:

$$\{o\}$$

is a concept, called Nominal. The expression $\{o_1, \dots, o_n\}$ for $n \geq 0$ denotes $\bot$ if $n = 0$, and $\{o_1\} \sqcup \dots \sqcup \{o_n\}$ if $n > 0$.

Nominals (a.k.a. Objects) $\mathcal{O}$

Semantics of Nominals

The interpretation of a nominal, i.e., $(\{o\})^{\mathcal{I}}$ is the singleton set $\{o^{\mathcal{I}}\}$. As a consequence:

$$\{o_1, \dots, o_n\}^{\mathcal{I}} = \{o_1^{\mathcal{I}}, \dots, o_n^{\mathcal{I}}\}$$

Nominals (a.k.a. Objects) $\mathcal{O}$

Exercise

Modeling with Nominals: Express, in term of subsumptions between concepts with, the following statements, using Nominals, and all the DL constructs you studied so far:

- There are exactly 195 Countries;
- either John or Mary is a spy but not both;
- Alice loves either Bob or Calvin;
- Everything is created by God;
- Everybody agree in driving on the left or on the right;
- $\exists x A(x) \rightarrow \forall x B(x)$

Nominals (a.k.a. Objects) $\mathcal{O}$

Solution

- *There are exactly 195 Countries;*

$$Country \equiv \{Afghanistan, Albania, \dots, Zimbabwe\}$$
$$\{Afghanistan\} \sqsubseteq \neg\{Albania\}, \dots, \{Afghanistan\} \sqsubseteq \neg\{Zimbabwe\}$$
$$\{Albania\} \sqsubseteq \neg\{Algeria\}, \dots, \{Albania\} \sqsubseteq \neg\{Zimbabwe\}$$
$$\dots$$

- *either John or Mary is a spy, but not both;*

$$\{John\} \sqsubseteq \neg\{Mary\}$$
$$\{JohnOrMary\} \sqsubseteq \{John, Mary\}$$
$$\{JohnOrMary\} \sqsubseteq Spy$$

- *Alice loves either Bob or Calvin;*

Nominals (a.k.a. Objects) $\mathcal{O}$

Solution (Cont'ed)

- *Everything is created by God*

$$\top \sqsubseteq \exists creates^{-}.\{god\}$$

In this case god is called spy point, as every object of the domain can be observed (and predicated) by "god" through the relation "creates". Spy point allows for full universal/existential quantification.

- *Everybody agree in driving on the left or on the right;*

$$\begin{aligned}\{god\} \quad \sqsubseteq \quad & \forall creates(\neg Person \sqcup LeftDriver) \sqcup \\ & \forall creates(\neg Person \sqcup RightDriver)\end{aligned}$$

- $\exists x A(x) \rightarrow \forall x B(x)$

Boolean T-boxes

Definition (Boolean T-box)

A Boolean TBox is a propositional formula whose atomic components are concept subsumptions. More formally:

- $A \sqsubseteq B$ is a boolean T-box, for every concepts A and B;
- if α and β are boolean T-boxes then $\neg\alpha$, $\alpha \wedge \beta$, $\alpha \vee \beta$ and $\alpha \rightarrow \beta$ are boolean T-boxes

Example

The Boolean T-box:

$$\neg(\mathit{Driver} \sqsubseteq \mathit{Pilot}) \wedge ((\mathit{Driver} \sqsubseteq \mathit{LeftDriver}) \vee (\mathit{Driver} \sqsubseteq \mathit{RightDriver}))$$

states that not all the drivers are pilots and that either all drivers drive on the left or all drivers drive on the right side of the road.

Exercise

Show that boolean t-boxes cannot be represented in $\mathcal{SHIQ}$. (Suggestion: use the fact that $\mathcal{SHIQ}$ is invariant under disjoint union of models)

Boolean t-box, nominals and inverse

Theorem

In $\mathcal{ALCOI}$ a boolean T-box can be transformed in an equivalent standard T-box

Proof.

Let ϕ a boolean T. w.l.o.g we can assume that ϕ is CNF (w.r.t, the boolean operators) i.e., is a set of clauses C where each $c \in C$ is of the form:

$$c = \bigvee_{i=1}^{n} (A_i \sqsubseteq B_i) \vee \bigvee_{j=1}^{m} \neg(C_j \sqsubseteq D_j)$$

Let r and o be a new role and a new object respectively, not appearing in ϕ, $\mathcal{T}_\phi$ as the T-Box that contains the subsumption $\top \sqsubseteq \exists r^- \{o\}$ (i.e. o is a spy point) and the following subsumption for every clause c as above

$$\{o\} \sqsubseteq \bigsqcup_{i=1}^{n} (\forall r.(\neg A_i \sqcup B_i)) \sqcup \bigsqcup_{j=1}^{m} (\exists r.(C_j \sqcap \neg D_j))$$

Reasoning with Nominals

Tree-model property

The tree model property has been advantageous for DL tableau algorithms by allowing them to search for tree-like models

Example

The concept $\{a\} \sqcap \exists r.\{a\}$ is satisfiable only by a model that contains a cycle on a.

remark

The difficulty in extending the $\mathcal{SHIQ}$ algorithms to $\mathcal{SHOIQ}$ is due to the interaction between nominals, number restrictions, and inverse roles, which leads to the almost complete loss of the tree model property, and causes the complexity of the ontology consistency problem to jump from ExpTime *to* NExpTime

[Horrocks and Sattler]

Reasoning with Nominals

Example

Consider the T-box $\mathcal{T}$ that contains:

$$\top \sqsubseteq \exists r^{-}\{o\} \quad \{o\} \sqsubseteq (\leq 20)R.A$$

Completion Graph

Definition

Let $\mathcal{R}$ be a role hierarchy and D a $\mathcal{SHOIQ}$-concept in NNF. A completion graph for D with respect to $\mathcal{R}$ is a directed graph

$$\mathbf{G} = \langle V, E, \mathcal{L}, \neq \rangle$$

where:

$$\begin{array}{rcl} \mathcal{L}(v) & \subseteq & cl(D) \cup N_I \ \cup \\ & & \{(\leq m)R.C) \mid (\leq n)R.C \in cl(D) \text{ and } m < n\} \\ E(v,w) & \subseteq & \{R \mid R \text{ is a role of } D\} \\ \neq & \subseteq & V \times V \end{array}$$

Some terminology

Nominal Node

A nominal node (N-node) is a node x, such that $\mathcal{L}(x)$ contains a nominal o

Blockable Node

A Blockable node is any node which is not a nominal node

Clas

A completion graph G contain a clash if

1. $\{A, \neq A\} \subset \mathcal{L}(x)$ for some A and x; $(\mathcal{ALC})$
2. $(\leq n)S.C \in \mathcal{L}(x)$ and there are $n+1$ S-neighbours $y_0, \dots, y_n$ of x with $C \in \mathcal{L}(y_i)$, and $y_i \neq y_j$ for $0 \leq i < j \leq n$ $(\mathcal{ALCQ})$
3. $o \in \mathcal{L}(x) \cap \mathcal{L}(y)$, and $x \neq y$ for some x, y and o nominal. $(\mathcal{SHIQ})$

Tableau rules for $\mathcal{SHOIQ}$

$\rightarrow_{\sqcap}$:	if 1.	$C_1 \sqcap C_2 \in \mathcal{L}(x)$, x is not indirectly blocked, and
	2.	$\{C_1, C_2\} \not\subseteq \mathcal{L}(x)$
	then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{C1, C2\}$
$\rightarrow_{\sqcup}$:	if 1.	$C_1 \sqcup C_2 \in \mathcal{L}(x)$, x is not indirectly blocked, and
	2.	$\{C_1, C_2\} \cap \mathcal{L}(x) = \emptyset$
	then	$\mathcal{L}(x) := \mathcal{L}(x) \cup \{C\}$ for some $C \in \{C_1; C_2\}$
$\rightarrow_{\exists}$:	if 1.	$\exists S.C \in \mathcal{L}(x)$, x is not blocked, and
	2.	x has no safe S-neighbour y with $C \in \mathcal{L}(y)$,
	then	create a new node y with $L(x, y) = \{S\}$ and $\mathcal{L}(y) = \{C\}$
$\rightarrow_{\forall}$:	if 1.	$\forall S.C \in \mathcal{L}(x)$, x is not indirectly blocked, and
	2.	there is an S-neighbour y of x with $C \notin \mathcal{L}(y)$
	then	$\mathcal{L}(y) := \mathcal{L}(y) \cup \{C\}$
$\rightarrow_{\forall_+}$:	if 1.	$\forall S.C \in \mathcal{L}(x)$, x is not indirectly blocked, and
	2.	there is some R with $\textsc{Trans}(R)$ and $R \sqsubseteq^* S$, and
	3.	there is an R-neighbour y of x with $\forall R.C \notin \mathcal{L}(y)$
	then	$\mathcal{L}(y) := \mathcal{L}(y) \cup \{\forall R.C\}$

Tableau rules for $\mathcal{SHOIQ}$ (cont'd)

$\rightarrow_?$:	if 1.	$(\leq n)S.C) \in \mathcal{L}(x)$, x is not indirectly blocked, and
	2.	there is an S-neighbour y of x with $\{C, \neg C\} \cap L(y) = \emptyset$
	then	$\mathcal{L}(y) := \mathcal{L}(y) \cup \{E\}$ for some $E \in \{C, \neg C\}$
$\rightarrow_{\geq}$:	if 1.	$(\geq n.S.C) \in \mathcal{L}(x)$, x is not blocked, and
	2.	there are not n safe S-neighbors $y_1, \dots, y_n$ of x with $C \in \mathcal{L}(y_i)$ and $y_i \neq y_j$ for $1 \leq i < j \leq n$
	then	create n new nodes $y_1, \dots y_n$ with $\mathcal{L}(x, y_i) = \{S\}$, $\mathcal{L}(y_i) = \{C\}$, and $y_i \neq y_j$ for $1 \leq i < j \leq n$
$\rightarrow_{\leq}$:	if 1.	$(\leq n)S.C) \in \mathcal{L}(z)$, z is not indirectly blocked, and
	2.	$\#S^G(z, C) > n$ and there are two S-neighbours x, y of z with $C \in \mathcal{L}(x) \cap \mathcal{L}(y)$, and not $x \neq y$
	then 1.	if x is a nominal node, then $Merge(y, x)$
	2.	else if y is a nominal node or an ancestor of x, then $Merge(x, y)$
	3.	else $Merge(y; x)$

Blocking strategy in $\mathcal{SHOIQ}$

The blocking strategy is the same as in $\mathcal{SHIQ}$, restricted to the non-nominal nodes (i.e., blockable nodes).

Blocking in $\mathcal{SHOIQ}$

A node x is directly blocked if it has ancestors x', y and y' such that

1. x is a successor of x' and y is a successor of y',
2. y, x and all nodes on the path from y to x are blockable,
3. $\mathcal{L}(x) = \mathcal{L}(y)$ and $\mathcal{L}(x') = \mathcal{L}(y')$, and
4. $\mathcal{L}(x', x) = \mathcal{L}(y', y)$,

A node is blocked if either it is directly blocked or it is blockable and its predecessor is directly blocked, in this case we say that x is indirectly blocked.

Merging Nodes

Intuition

Merge(y, x) is obtained by

- adding $\mathcal{L}(y)$ to $\mathcal{L}(x)$;
- redirecting all the edges leading to y to x
- redirecting all the edges leading from y to nominal nodes so that they lead from x to the same nominal nodes;
- removing y (and blockable sub-trees below y)

Tableau rules for $\mathcal{SHOIQ}$ (cont'd)

$\rightarrow_o$:	if	for some nominal o there are 2 nodes x, y with $o \in \mathcal{L}(x) \cap \mathcal{L}(y)$ and not $x \neq y$
	then	$Merge(x, y)$
$\rightarrow_{o?}$:	if 1.	$(\leq n)S.C \in \mathcal{L}(x)$, x is a nominal node, and there is a blockable S-neighbour y of x such that $\{C\} \in \mathcal{L}(y)$ and x is a successor of y and
	2.	there is no m with $1 \leq m \leq n$, $(\leq m)S.C \in \mathcal{L}(x)$ and there are m nominal S-neighbours $z_1, \ldots z_m$ of x with $C \in \mathcal{L}(z_i)$ and $z_i \neq z_j$ for all $1 \leq i < j \leq m$
	then 1.	guess $m \leq n$ and set $\mathcal{L}(x) := \mathcal{L}(x) \cup \{(\leq m)S.C\}$
	2.	create m new nodes $y_1, \ldots, y_m$ with $\mathcal{L}(x, y_i) := \{S\}$, $\mathcal{L}(y_i) = \{C, o_i\}$ for $o_i \in N_I$ new in G, and $y_i \neq y_j$ for all $1 \leq i < j \leq m$

More espressivity for roles

Boolean combination of roles $\mathcal{O}$ Role expressions can be boolean combination of roles $\neg R$, $R \sqcup S$, $R \sqcap S$. For instance:

$$HasParent \equiv HasMother \sqcup HasFather,$$
$$\neg Likes, \qquad hasColleque \sqcap hasFriend$$

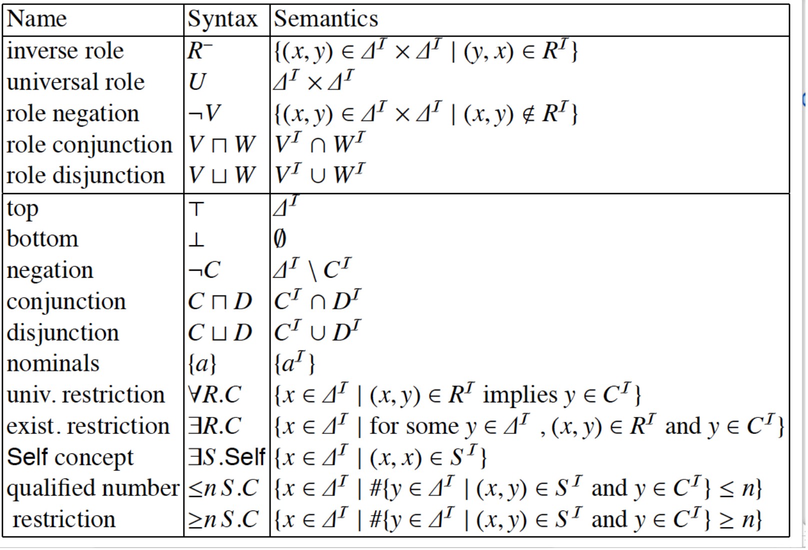
Role composition $\mathcal{R}$ Role expressions contains compositions of roles $R \circ S$. For instance

$$hasParent \circ hasBrother \sqsubseteq hasUncle$$

Role properties Direct statements about roles as $Transitive(R)$, $Symmetric(R)$, $Asymmetrichas(R)$, $Reflexive(R)$, $Irreflexive(R)$, $Functiona(R)$ and $InvFunctional(R)$

$$Transitive(hasAncestor) \qquad Symmetrichas(Spouse)$$

The description logics $\mathcal{SROIQB}$



Name	Syntax	Semantics
inverse role	R^-	$\{(x,y) \in \Delta^\mathcal{I} \times \Delta^\mathcal{I} \mid (y,x) \in R^\mathcal{I}\}$
universal role	U	$\Delta^\mathcal{I} \times \Delta^\mathcal{I}$
role negation	$\neg V$	$\{(x,y) \in \Delta^\mathcal{I} \times \Delta^\mathcal{I} \mid (x,y) \notin R^\mathcal{I}\}$
role conjunction	$V \sqcap W$	$V^\mathcal{I} \cap W^\mathcal{I}$
role disjunction	$V \sqcup W$	$V^\mathcal{I} \cup W^\mathcal{I}$
top	$\top$	$\Delta^\mathcal{I}$
bottom	$\bot$	$\emptyset$
negation	$\neg C$	$\Delta^\mathcal{I} \setminus C^\mathcal{I}$
conjunction	$C \sqcap D$	$C^\mathcal{I} \cap D^\mathcal{I}$
disjunction	$C \sqcup D$	$C^\mathcal{I} \cup D^\mathcal{I}$
nominals	$\{a\}$	$\{a^\mathcal{I}\}$
univ. restriction	$\forall R.C$	$\{x \in \Delta^\mathcal{I} \mid (x,y) \in R^\mathcal{I}$ implies $y \in C^\mathcal{I}\}$
exist. restriction	$\exists R.C$	$\{x \in \Delta^\mathcal{I} \mid$ for some $y \in \Delta^\mathcal{I}$, $(x,y) \in R^\mathcal{I}$ and $y \in C^\mathcal{I}\}$
Self concept	$\exists S.\mathsf{Self}$	$\{x \in \Delta^\mathcal{I} \mid (x,x) \in S^\mathcal{I}\}$
qualified number	$\leq n\, S.C$	$\{x \in \Delta^\mathcal{I} \mid \#\{y \in \Delta^\mathcal{I} \mid (x,y) \in S^\mathcal{I}$ and $y \in C^\mathcal{I}\} \leq n\}$
restriction	$\geq n\, S.C$	$\{x \in \Delta^\mathcal{I} \mid \#\{y \in \Delta^\mathcal{I} \mid (x,y) \in S^\mathcal{I}$ and $y \in C^\mathcal{I}\} \geq n\}$

Regular Role Inclusion Axioms (RIAs)

Undecidability

Role primitives cause undecidability when used without restrictions. Regularity restrictions ensure decidability

RIA as Grammar

An R-box R that contains a role R can be seen as a grammar

$$R_1 \circ \cdots \circ R_n \sqsubseteq R \quad \Longrightarrow \quad R \longrightarrow R_1 \dots R_n$$

$$L_{\mathcal{R}}(R) = \{R_1 \dots R_n \mid R \xrightarrow{*} R_1, \dots, R_n\}$$

Recovering from undecidability

Tableaux algorithm for $\mathcal{SROIQ}$ is based on the corresponding automata for $L_R(R)$. Decidability can be obtained by restricting to Regular context free grammars.

Regular R-box

Example (Regular language)

$$R \circ S \sqsubseteq R$$
$$S \circ R \sqsubseteq R$$

generates the language S^*RS^* which is regular. The sufficient condition is satisfied by the order $S \prec R$.

Example (Non regular language)

$$S \circ R \circ S \sqsubseteq R$$

is not regular as it generates the language S^nRS^n which is not a regular context free language.

Decidability in $\mathcal{SROIQ}$

How to ensure that the set of RIAs is regular?
Checking if a CFG is regular is undecidable
We can define a set of sufficient condition for regularity

Regular RIA

$$
\begin{array}{rcl}
R \circ R & \sqsubseteq & R \\
R^{-} & \sqsubseteq & R \\
S_1 \circ \cdots \circ S_n & \sqsubseteq & R \\
R \circ S_1 \circ \cdots \circ S_n & \sqsubseteq & R \\
S_1 \circ \cdots \circ S_n \circ R & \sqsubseteq & R
\end{array}
$$

if there is a partial order $\prec$ on roles such that

$S_i \prec R$ for $1 \leq i \leq n$, and
$R \prec S$ iff $R \prec S^{-}$

Regular R-box

Exercise

Check if the following Ria is Regular:

$$\begin{array}{rcl}
isProperPartOf & \sqsubseteq & isPartOf \\
isPartOf \circ isPartOf & \sqsubseteq & isPartOf \\
isPartOf \circ isProperPartOf & \sqsubseteq & isPartOf \\
isProperPartOf \circ isPartOf & \sqsubseteq & isPartOf
\end{array}$$

Then define $L_{\mathcal{R}}(isPartOf)$.

Exercise

Check if the following Ria is Regular:

$$\begin{array}{rcl}
R \circ R & \sqsubseteq & R \\
S & \sqsubseteq & R \\
R \circ S & \sqsubseteq & S
\end{array}$$

Reasoning in $\mathcal{SROIQ}$

- Satisfiability and subsumption of $\mathcal{SROIQ}$-concepts w.r.t. Tboxes, Aboxes, and Rboxes, are polynomially reducible to (un)satisfiability of $\mathcal{SROIQ}$-concepts w.r.t. Rboxes.
- W.l.o.g., we can assume that Rboxes do not contain role assertions of the form $\mathit{Irr}(R)$, $\mathit{Tra}(R)$, or $\mathit{Sym}(R)$, and that the universal role is not used.

Exercise

Show how $\mathit{Irr}(R)$, $\mathit{Tra}(R)$, and $\mathit{Sym}(R)$, can be expressed in RIA

Reasoning in $\mathcal{SROIQ}$

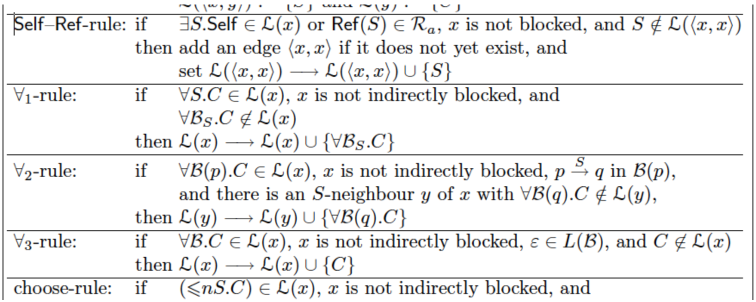
Universal role elimination

Consider U as any other role, (no special interpretation) and define following concept:

$$C_{\mathcal{T}} \equiv \forall U. \left(\bigsqcap_{A \sqsubseteq B \in T} \neg A \sqcup B \right) \sqcap \bigsqcap_{o \in N} \exists U.\{o\}$$

and extend the R-box with the following assertions: $R \sqsubseteq U$, $Tra(U)$, $Sym(U)$, and $Ref(U)$.

Tableaux for $\mathcal{SROIQ}$



Self–Ref-rule:	if	$\exists S.\mathsf{Self} \in \mathfrak{L}(x)$ or $\mathsf{Ref}(S) \in \mathcal{R}_a$, x is not blocked, and $S \notin \mathfrak{L}(\langle x, x\rangle)$
	then	add an edge $\langle x, x\rangle$ if it does not yet exist, and set $\mathfrak{L}(\langle x, x\rangle) \longrightarrow \mathfrak{L}(\langle x, x\rangle) \cup \{S\}$
$\forall_1$-rule:	if	$\forall S.C \in \mathfrak{L}(x)$, x is not indirectly blocked, and $\forall \mathcal{B}_S.C \notin \mathfrak{L}(x)$
	then	$\mathfrak{L}(x) \longrightarrow \mathfrak{L}(x) \cup \{\forall \mathcal{B}_S.C\}$
$\forall_2$-rule:	if	$\forall \mathcal{B}(p).C \in \mathfrak{L}(x)$, x is not indirectly blocked, $p \xrightarrow{S} q$ in $\mathcal{B}(p)$, and there is an S-neighbour y of x with $\forall \mathcal{B}(q).C \notin \mathfrak{L}(y)$,
	then	$\mathfrak{L}(y) \longrightarrow \mathfrak{L}(y) \cup \{\forall \mathcal{B}(q).C\}$
$\forall_3$-rule:	if	$\forall \mathcal{B}.C \in \mathfrak{L}(x)$, x is not indirectly blocked, $\varepsilon \in L(\mathcal{B})$, and $C \notin \mathfrak{L}(x)$
	then	$\mathfrak{L}(x) \longrightarrow \mathfrak{L}(x) \cup \{C\}$
choose-rule:	if	$(\leqslant n S.C) \in \mathfrak{L}(x)$, x is not indirectly blocked, and

- we have seen a set of DLs with increasing expressivity and reasoning complexity. There are many more, which can be obtained by different combinations of the presented primitives. For an overview of all the properties of the various DLs you can access to the interactive web site DL complexity navigator

  `http://www.cs.man.ac.uk/~ezolin/dl/`

  where you can define you preferred DL and check if somebody have studied it (if you are lucky you can find a paper on your preferred DL)
- We haven't dedicated much time to reasoning. For your reference, most (but not all) of the reasoning tools are based on tableaux, (Pellet, Fact++, Racer, Hermit,...). Some of them like MSPASS. All the DL reasoners are listed at the page

  `http://www.cs.man.ac.uk/~sattler/reasoners.html`