

An Introduction to Description Logics

0. Some RDFS Limitations

G. Falquet

Limitations

- global domain/range
- no number restrictions
- no union
- no existential assertion/inference
- no negation

global domain/range

- The parents of a Human are Humans
 - parent domain Human
 - parent range Human
- The parents of a Human are Humans and the parents of a Cat are Cats
 - impossible on this vocabulary
 - must add properties
 - humanParent subproperty of parent ; domain Human ; range Human
 - catParent subproperty of parent ; domain Cat ; range Cat

no number restrictions

Impossible to express

- an individual has at most one address
- a house has at least one owner
- an individual has exactly two biological parents

no "exact" union

- Students are either bachelor or master or PhD students, and nothing else
- All we can do in RDFS is
 - BacStd `rdfs:subClassOf Student`
 - MasterStd `rdfs:subClassOf Student`
 - PhDStd `rdfs:subClassOf Student`
- Impossible to express "and nothing else"
- Impossible to express the disjointness of these classes (if it's the case)

no existential statement and inference

knowing that

- A car necessarily has an owner (not expressible)
- c is a car (:c a :Car)

the query

```
select ?x  
where {?x :hasOwner ?y}
```

should answer {c}

even if no triple (:c :hasOwner :o) is present in the graph

no negation

- Impossible to state that something is false
- In knowledge bases we often consider that what is not expressed is not known, it can be true or false

```
:ChemicalProduct a rdfs:Class .  
:ToxicProduct rdfs:subClassOf ChemicalProduct .  
:p1 a :ToxicProduct .  
:p2 a :ChemicalProduct
```

does not mean that :p2 is not toxic

An Introduction to Description Logics

1. Syntax and Semantics

G. Falquet

Contents

- Individuals, classes and properties
- Class constructors
- Ontologies
- Reasoning

The world according to DL

- There are **individuals**
- Individuals may be interconnected through **roles**
- Individuals belong to **concepts**

property
clauses

Similar to RDF/S (resources, properties, classes)

But different

- individuals, classes, and roles are disjoint
 - a class may not be an instance of another class
 - an individual may not have instances

DL languages

a family of logical languages with different

- class constructors
- role constructors

aims

- formalize the knowledge representation languages of the 80-90's (CLASSIC, KL-ONE, semantic networks, ...)
- decidable
 - there is an algorithm to check the consistency of any set of formulas
 - ⇒ not as expressive as First order logic

\mathcal{ALC} – concept constructors

close

proper

Start with a vocabulary of **concept names** (N_C), **role names** (N_R) and **individual names** (N_O)

Concept expressions may be

- the top concept (everything): T
- the bottom (impossible) concept: \perp
- a concept name: A
- a conjunction of concepts: C \sqcap D
- a disjunction: C \sqcup D
- a complement: \neg C
- an existential restriction: \exists R.C
- a universal restriction: \forall R.C

~~Set (VR)~~

Manchester notation

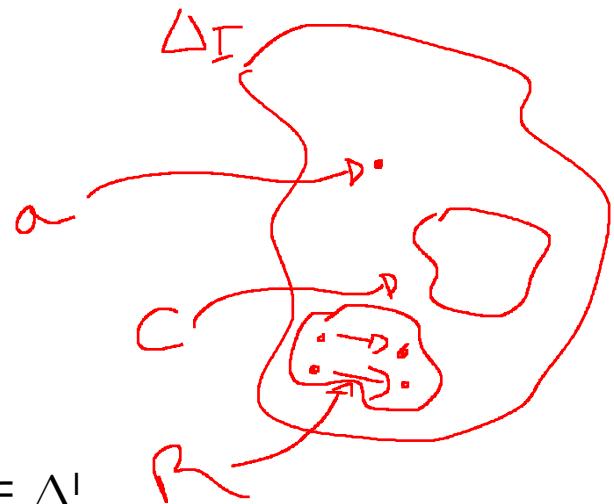
class name	Building
conjunction	Student and Employee
disjunction	Man or Woman
complement	not Student
existential restriction	child some Student
universal restriction	child only Student

Semantics of ALC

An interpretation I consists of

- a domain Δ^I
- an interpretation function I such that maps:

- every individual name a to an element $I(a) \in \Delta^I$
- every concept C to a subset of Δ^I
- every role R to a binary relation on Δ^I (a subset of $\Delta^I \times \Delta^I$)



The semantics of non atomic concepts and roles is (recursively) defined in terms of atomic concept and role interpretations.

Semantics of concept expressions

$$I(\perp)$$

$$= \emptyset \quad \checkmark$$

$$I(\top)$$

$$= \Delta^I \quad \checkmark$$

$$I(C \sqcap D)$$

$$= I(C) \cap I(D)$$



$$I(C \sqcup D)$$

$$= I(C) \cup I(D)$$



$$I(\neg C)$$

$$= \Delta^I - I(C)$$

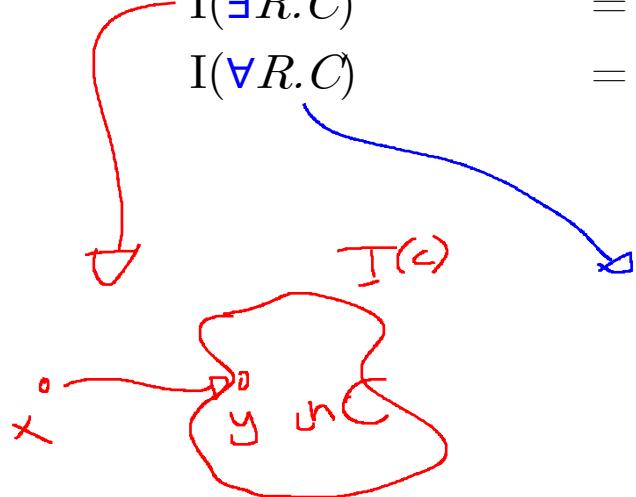


$$I(\exists R.C)$$

$$= \{x \in \Delta^I : \exists y: (x, y) \in I(R) \text{ and } y \in I(C)\}$$

$$I(\forall R.C)$$

$$= \{x \in \Delta^I : \forall y: (x, y) \in I(R) \Rightarrow y \in I(C)\}$$



X

Exercise

Consider the vocabulary

concepts: Human, Cat ; roles: hasParent
and the interpretation

- $\Delta^I = \{a, b, c, d, e, f, k, l, m\}$
- $I(\text{Human}) = \{a, b, c, d, e\}$
- $I(\text{Cat}) = \{k, l, m\}$
- $I(\text{hasParent}) = \{(a, b), (a, c), (c, d), (b, e), (b, f), (k, m)\}$

$x \xrightarrow{\text{hasParent}} y \in I(T)$

What are the formal interpretations (and informal meaning) of

- $\text{Human} \sqcap \text{Cat} \rightarrow \emptyset$
- $\neg \text{Human} \rightarrow \{f, k, m, l\}$
- $\exists \text{ hasParent}. T$ $\{a, c, b, l, m\}$
- $\exists \text{ hasParent}. \text{Cat} \rightarrow \{k\}$

ici y is in cat

Exercise (cont.)

If

- $\Delta^I = \{a, b, c, d, e, f\}$
- $I(\text{Human}) = \{a, b, c, d, e\}$
- $I(\text{Cat}) = \{k, l, m\}$
- $I(\text{hasParent}) = \{(a, b), (a, c), (c, d), (b, e), (b, f), (k, m)\}$

what are the interpretations of

- $\forall . \text{hasParent} . \text{Cat}$ on veut les individus ou tous leurs parents sont des chats vu que (d) n'a pas de parent, il est dans la classe
- $\exists \text{ hasParent} . \top \sqcap \forall . \text{hasParent} . \text{Cat}$ $\rightarrow \{k\}$
- $\exists \text{ hasParent} . (\exists \text{ hasParent} . \text{Human})$
- $\exists \text{ hasParent} . (\forall \text{ hasParent} . \perp)$

$\{k, l, m\}$

on veut les individus ou tous leurs parents sont des chats
vu que (d) n'a pas de parent, il est dans la classe

$\{k\}$

$\{a, b, c, d, e\}$

$\{b, c, k\}$

$\{a, c, b\}$

More class constructors (\mathcal{ALCOQ})

Number restrictions on properties

$\geq n$ R.C

$$\{x \in \Delta^I : \#\{y | (x, y) \in I(R) \text{ and } y \in I(C)\} \geq n\}$$

$\leq n$ R.C

$$\{x \in \Delta^I : \#\{y | (x, y) \in I(R) \text{ and } y \in I(C)\} \leq n\}$$

$= n$ R.C \leftrightarrow $\geq n$ R.C and $\leq n$ R.C

Enumeration of individuals

$$\{i_1, i_2, \dots, i_n\}$$

$$\{x \in \Delta : x = I(i_1) \text{ or } x = I(i_2) \text{ or } \dots, \text{ or } x = I(i_n)\}$$

DL Knowledge Base \sim RDFS

Vocabulary:

class names, property names, individual names

Terminological axioms (TBox):

provide class definitions and relationships between classes

Role axioms (RBox):

about roles

Assertional axioms (ABox):

about individuals

Terminological Axioms (TBox)

Axioms of the form

$$C \sqsubseteq D$$

$$C \equiv D$$

$$C \text{ disjoint } D$$

Axiom satisfaction by an interpretation I (notation: $I \models \text{Axiom}$)

$I \models C \sqsubseteq D$ if and only if $I(C) \subseteq I(D)$

$I \models C \equiv D$ if and only if $I(C) = I(D)$

$I \models C \text{ disjoint } D$ if and only if $I(C) \cap I(D) \neq \emptyset$

Exercises

Cat $\xrightarrow{I} \{f\}$
Mammal $\xrightarrow{I} \{a, b, c, d, e, f\}$
Human $\xrightarrow{I} \{a, b, c\}$
hasParent $\xrightarrow{I} \{f, a, b\}$

Find an interpretation I of the vocabulary

- Cat, Mammal, Human (concept names),
- hasParent (role names),
- felix, bob, alice (individual names)
$$\begin{array}{ccc} f & \downarrow & a \\ & & \end{array}$$

that has $\Delta^I = \{a, b, c, d, e, f\}$ and satisfies the axioms

- Cat \sqsubseteq Mammal
- Human \sqsubseteq Mammal
- Human $\sqsubseteq \forall \text{hasParent}.\text{Human}$
- {bob, alice} \sqsubseteq Human
- {felix} \sqsubseteq Cat
- Cat $\sqsubseteq \exists \text{hasParent}.T$
- T \sqsubseteq Mammal

Axioms on roles (RBox) (*ALCHOQ*)

$$P \sqsubseteq R$$

if **a** is linked to **b** through P

then **a** is linked to **b** through R

$$\text{I} \models P \sqsubseteq R \text{ if and only if } \text{I}(P) \subseteq \text{I}(R)$$

Examples

- mother \sqsubseteq parent
- primaryFunction \sqsubseteq function

Axioms on roles: property chains

$$R_1 \circ R_2 \circ \dots \circ R_n \sqsubseteq P$$

if **a** is linked to **b** through a property chain $R_1 \circ R_2 \circ \dots \circ R_n$
then **a** is linked to **b** through **P**

$$\text{I} \vDash R_1 \circ R_2 \circ \dots \circ R_n \sqsubseteq P$$

if and only if

$$\forall y_0, y_1, \dots, y_n : (y_0, y_1) \in \text{I}(R_1), \dots, (y_{n-1}, y_n) \in \text{I}(R_n) \rightarrow (y_0, y_n) \in \text{I}(P)$$

Examples

- parent \circ parent \sqsubseteq grandParent
- parent \circ parent \circ child \circ child \sqsubseteq cousinOrSiblingOrSelf

More axioms on roles

functional(R)

$$(x, y) \in I(R) \text{ and } (x, z) \in I(R) \rightarrow y = z$$

inverse(R, S)

$$(x, y) \in I(R) \rightarrow (y, x) \in I(S)$$

symmetric(R)

$$(x, y) \in I(R) \rightarrow (y, x) \in I(R)$$

transitive(R)

$$(x, y) \in I(R) \text{ and } (y, z) \in I(R) \rightarrow (x, z) \in I(R)$$

reflexive(R)

$$\forall x \in \Delta^I : (x, x) \in I(R)$$

Examples

functional(biologicalMother)

inverse(mother , child)

symmetric(friend)

transitive(before)

reflexive(knows)

Assertions on individuals (ABox)

Axioms asserting that

- a named individual belongs to a class

$$C(i) \rightsquigarrow i \text{ rdf:type } C$$

$I \models C(i)$ if and only if $I(i) \in I(C)$

- a named individual is linked to another one through a role

$$R(i,j)$$

$I \models R(i,j)$ if and only if $(I(i), I(j)) \in I(R)$

Assertions on individuals (ABox)

Axioms asserting that

- two named individuals are different

$i \text{ differentFrom } j$

$I \models i \text{ differentFrom } j$ if and only if $I(i) \neq I(j)$

- two named individuals are equal

$i \text{ sameAs } j$

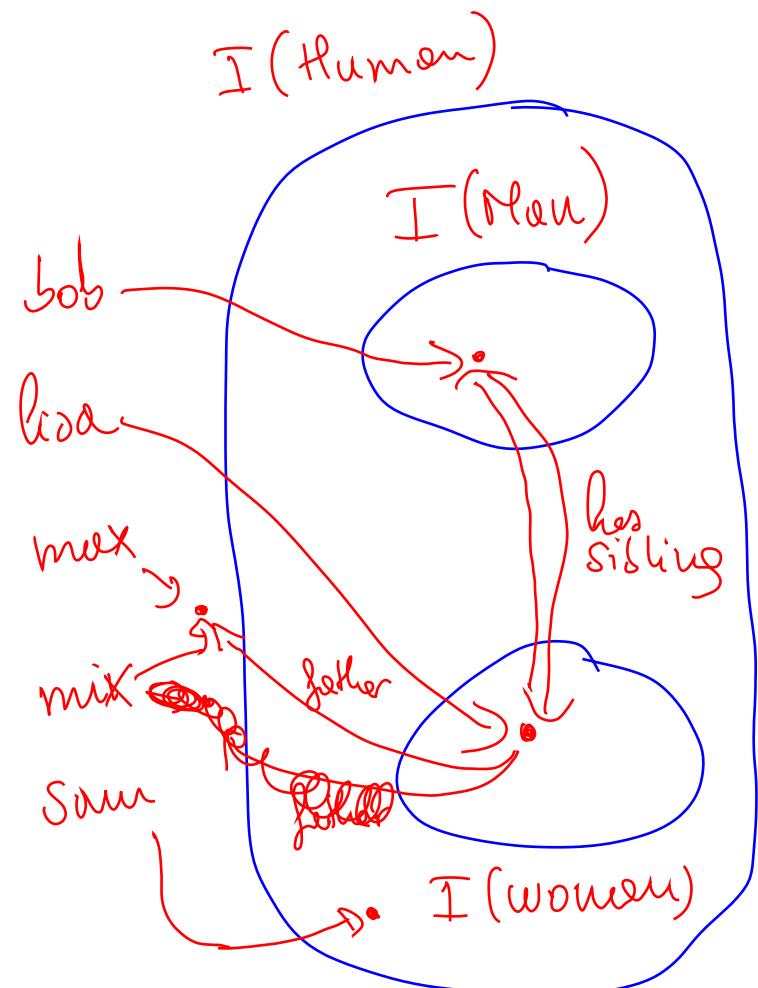
$I \models i \text{ sameAs } j$ if and only if $I(i) = I(j)$

Exercise

Find a minimal interpretation that satisfies

- Man(bob)
- Woman(lisa)
- Human(sam)
- $\text{Man} \sqcup \text{Woman} \sqsubseteq \text{Human}$
- Man disjointFrom Woman
- hasSibling(bob, lisa)
- symmetric(hasSibling)
- father(lisa, max)
- father(lisa, mix)
- mix differentFrom sam
- functional(father)

↳ 1 seule



An Introduction to Description Logics

2. Reasoning Tasks

G. Falquet

Reasoning Tasks

- Consistency
- Subsumption
- Open world
- Unique name
- Instance checking

Consider the axioms

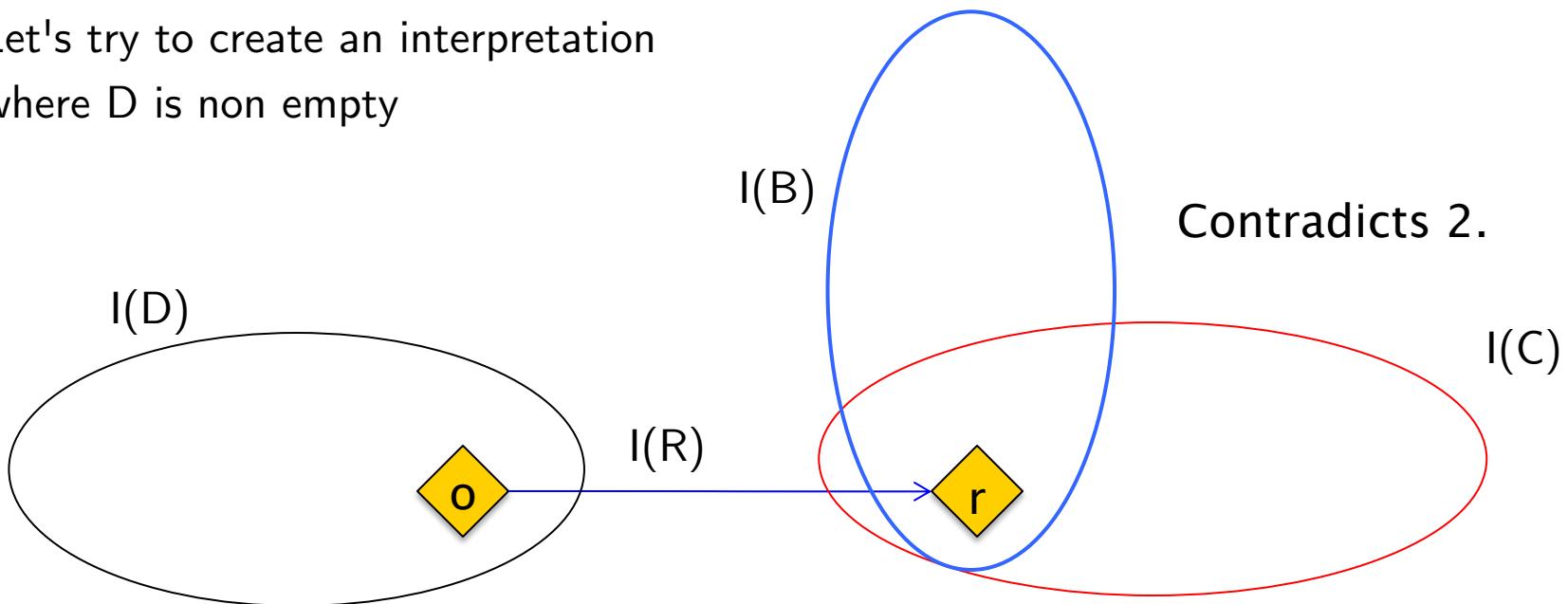
1. $A \sqsubseteq (\forall R . B)$
2. C disjoint B
3. $D \sqsubseteq ((\exists R . C) \sqcap A)$

Let's try to create an interpretation
where D is non empty

Consider the axioms

1. $A \sqsubseteq (\forall R . B)$
2. C disjoint B
3. $D \sqsubseteq ((\exists R . C) \sqcap A)$

Let's try to create an interpretation
where D is non empty



Consistency \rightsquigarrow cohérence

- a knowledge base is consistent if there is an interpretation such that all the axioms are satisfied
- a concept C is consistent if we can populate the ontology so as to
 - satisfy all the axioms
 - have at least one object in TC)

i.e. there is an interpretation I such that

$$1. I \models \text{TBox}$$

$$2. I \not\models C \sqsubseteq \perp$$

Example : TBox vs. Concept Consistency

TBox $T =$

$$W \sqsubseteq \{w\}$$

$$W \sqsubseteq \exists r. \top$$

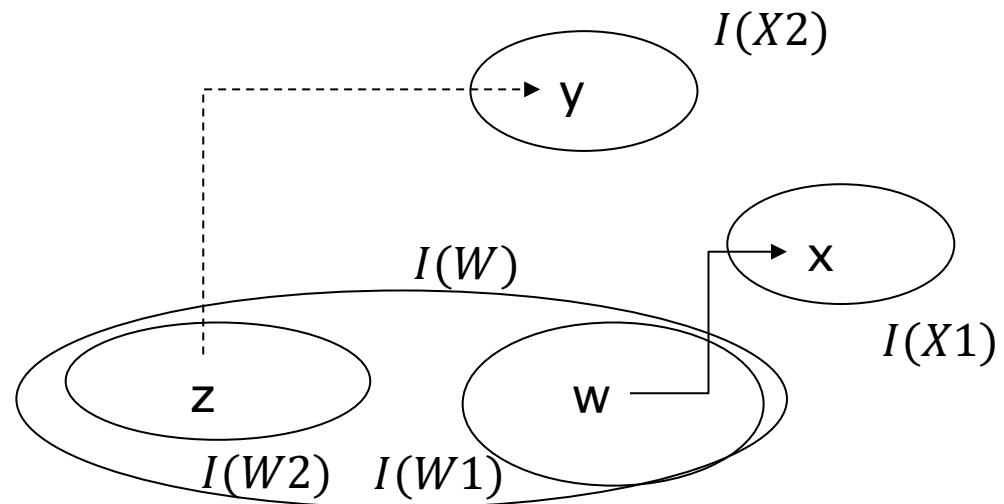
$$W \sqsubseteq (\forall r. X_1) \sqcup (\forall r. X_2)$$

$$W_1 \sqsubseteq W \sqcap (\forall r. X_1)$$

$$W_2 \sqsubseteq W \sqcap (\forall r. X_2)$$

W_1 disjoint W_2

X_1 disjoint X_2



T is consistent but in every model I of T ,
if $I(W_1)$ is non-empty then $I(W_2)$ is empty, and vice versa.

Reasoning tasks: subsumption

Given a TBox T , C subsumes D if
context

for every model I of T , $I(D) \subseteq I(C)$

or equivalently

$T \cup \{D \sqcap \neg C\}$ is inconsistent

Reasoning task:

input: a Tbox T , two classes C, D

output: true iff C subsumes D for T

Reasoning tasks: Instance checking

1. check if $C(o)$ is a consequence of the axioms and asserted facts

amounts to check if C subsumes the concept $\{o\}$

2. find all the individuals that belong to C

similar to query answering in (deductive) databases

Example

Find facts about individuals belonging to classes.

1. Parent $\equiv \exists$ hasChild . Person
2. hasChild(Bob, Alice)
3. Woman(Alice)
4. Woman \sqsubseteq Person

consequence

Parent(Bob)

Open World Semantics

What is not explicitly asserted is unknown (maybe true maybe false). Leads to counter intuitive results:

1. GoParent $\equiv \forall$ hasChild . Girl
2. hasChild(Bob, Alice)
3. Girl(Alice)

can we infer GoParent(Bob) ?

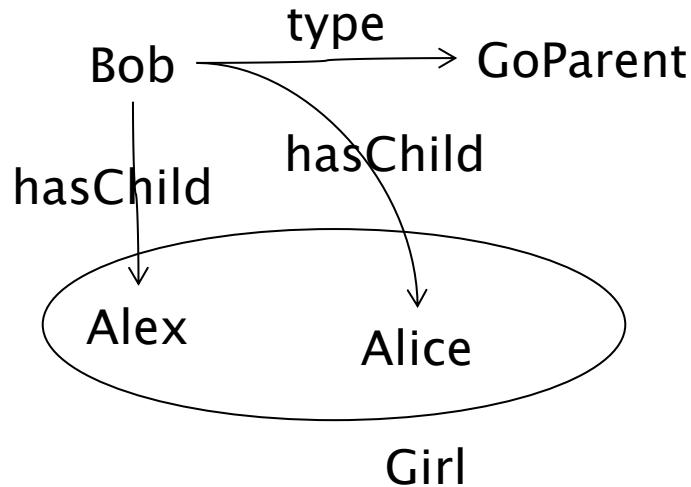
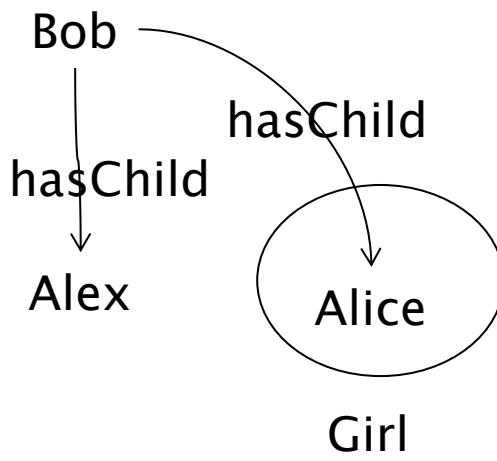
No, (Bob may have other children who are not girls)



Open World Semantics

Some models of

1. GoParent $\equiv \forall$ hasChild . Girl
2. hasChild(Bob, Alice)
3. Girl(Alice)



closing the world

1. GoParent $\equiv \forall$ hasChild . Girl
2. hasChild(Bob, Alice)
3. Girl(Alice)
4. ParentOf1 \sqsubseteq hasChild $=_1$ Thing
5. ParentOf1(Bob)

now we can infer Bob **a** GoParent

No Unique Name Assumption (UNA)

1. BusyParent \equiv hasChild \geq_2 Person
2. hasChild (Cindy, Bob)
3. hasChild (Cindy, John)

consequence: BusyParent (Cindy) ?

no, because *Bob* and *John* may be the same person

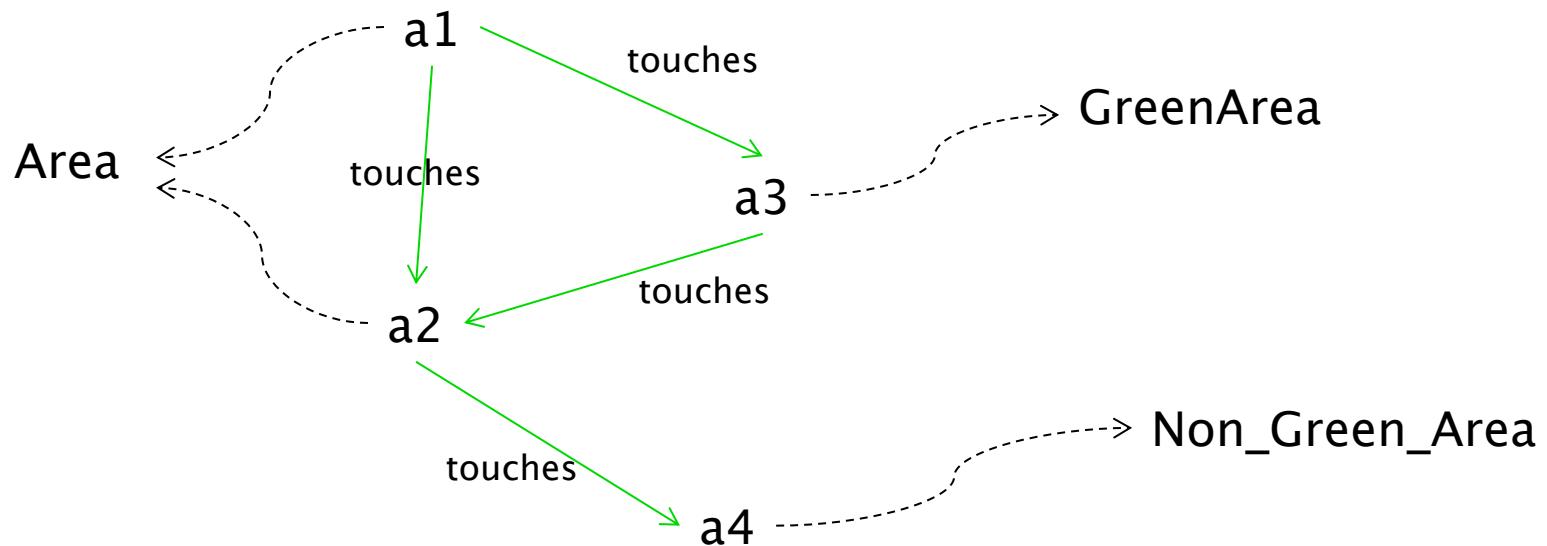
yes if we add the axiom
Bob \neq John

Sophisticated “open world” reasoning

Terminological Axioms (TBox)

1. $\text{Green_Area} \sqsubseteq \text{Area}$
2. $\text{Non_Green_Area} \equiv \text{Area} \sqcap (\neg \text{Green_Area})$

ABox



Q: Does **a1** touch some **Green Area** that touches some non **Green Area**?

A: Yes

- a2 is either green or non green (axioms 1 and 2)
- if it is green a1 satisfies the condition (using a3, a2)
- if it is non green a1 satisfies the condition (using a2, a4)

Berlin001.owl (http://cui.unige.ch/isi/ontologies/Berlin001.owl) – [/Users/falquet/sci/Ontologies/Berlin001.owl]

File Edit Ontologies Reasoner Tools Refactor Tabs View Window Help

Berlin001.owl

Active Ontology Entities Classes Object Properties Data Properties Individuals OWLViz DL Query

Asserted Class Hierarchy: A

Thing Area GreenArea IndustrialArea

Query:

Query (class expression)

touches some (GreenArea and (touches some (not GreenArea)))

Execute

Query results

Instances

a_1

Super classes
 Ancestor classes
 Equivalent classes
 Subclasses
 Descendant classes
 Individuals

Reasoning Services for DL Ontologies

- In most description logics consistency and subsumption can be computed (with sophisticated tableau algorithms), with different time and space complexities
- Consequences
 - the consistency of an ontology can be checked
 - it is possible to compute the class subsumption hierarchy
 - it is possible to find the closest concept corresponding to a query
- There are description logics for which consistency and subsumption can be computed in polynomical time or better
 - OWL-RL, OWL-QL

Everything about DL

- at <http://dl.kr.org/>
- and <http://www.cs.man.ac.uk/~ezolin/dl/>



Complexity of reasoning in Description Logics

Note: the information here is (always) incomplete and updated often

Base description logic: Attributive Language with Complements

$\mathcal{ALC} ::= \perp \mid T \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C$

trans reg

Concept constructors:

- \mathcal{F} - functionality²: $(\leq 1 R)$
- \mathcal{N} - (unqualified) number restrictions: $(\geq n R), (\leq n R)$
- \mathcal{Q} - qualified number restrictions: $(\geq n R.C), (\leq n R.C)$
- \mathcal{O} - nominals: $\{a\}$ or $\{a_1, \dots, a_n\}$ ("one-of")

- μ - least fixpoint operator: $\mu X.C$

Forbid complex roles⁵ in number restrictions⁶

Role constructors:

- \mathcal{I} - role inverse: R^-
- \cap - role intersection³: $R \cap S$
- \cup - role union: $R \cup S$
- \neg - role complement: $\neg R$ full
- \circ - role chain (composition): $R \circ S$
- $*$ - reflexive-transitive closure⁴: R^*
- id - concept identity: $id(C)$

TBox (concept axioms) is internalizable in extensions of \mathcal{ALCIO} , see

[82, Lemma 4.12], [61, p.3]

- empty TBox
- acyclic TBox ($A \equiv C$, A is a concept name; no cycles)
- general TBox ($C \sqsubseteq D$, for arbitrary concepts C and D)

OWL-Lite

OWL-DL

OWL 1.1

RBox (role axioms):

- \mathcal{S} - role transitivity: $Tr(R)$
- \mathcal{H} - role hierarchy: $R \sqsubseteq S$
- \mathcal{R} - complex role inclusions: $R \circ S \sqsubseteq R, R \circ S \sqsubseteq S$
- s - some additional features (click to see them)

Reset

You have selected a Description Logic: \mathcal{SHOIQ}

Complexity⁷ of reasoning problems⁸

Concept satisfiability	NExpTime-complete	<ul style="list-style-type: none"> • Hardness of even \mathcal{ALCFO} is proved in [82, Corollary 4.13]. • A different proof of the NExpTime-hardness for \mathcal{ALCFO} is given in [61] (even with 1 nominal, and inverse roles not used in number restrictions). • Upper bound for \mathcal{SHOIQ} is proved in [12, Corollary 6.31] with numbers coded in unary (for binary coding, the upper bound remains an open problem for all logics in between \mathcal{ALCNO} and \mathcal{SHOIQ}). • A tableaux algorithm for \mathcal{SHOIQ} is presented in [51]. • Important: in number restrictions, only simple roles (i.e. which are neither transitive nor have a transitive subroles) are allowed; otherwise we gain undecidability even in \mathcal{SHN}; see [54]. • Remark: recently [55] it was observed that, in many cases, one can use transitive roles in number restrictions –