Description Logic: A Formal Foundation for Languages and Tools

Ian Horrocks
<ian.horrocks@comlab.ox.ac.uk>
Information Systems Group
Oxford University Computing Laboratory

Contents

DL Basics

What Are Description Logics?

What Are Description Logics?

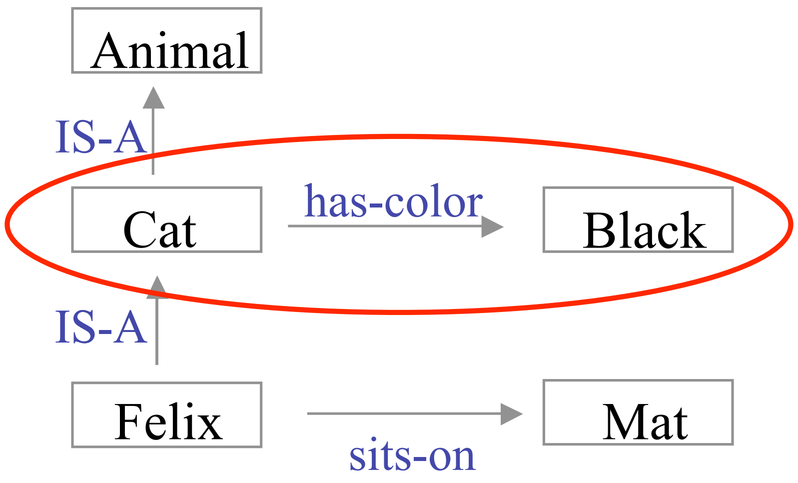
- Decidable fragments of First Order Logic

Thank you for listening

Any questions?

What Are Description Logics?

- A family of logic based Knowledge Representation formalisms
  - Originally descended from semantic networks and KL-ONE
  - Describe domain in terms of concepts (aka classes), roles (aka properties, relationships) and individuals



[Quillian, 1967]

What Are Description Logics?

- Modern DLs (after Baader et al) distinguished by:
  - Fully fledged logics with formal semantics
    - Decidable fragments of FOL (often contained in C_2)
    - Closely related to Propositional Modal & Dynamic Logics
    - Closely related to Guarded Fragment
  - Provision of inference services
    - Practical decision procedures (algorithms) for key problems (satisfiability, subsumption, etc)
    - Implemented systems (highly optimised)

and now:

A Word from our Sponsors

Crash Course in (simplified) FOL

- Syntax
  - Non-logical symbols (signature)
    - Constants: Felix, MyMat
    - Predicates(arity): Animal(1), Cat(1), has-color(2), sits-on(2)
  - Logical symbols:
    - Variables: x, y
    - Operators: ∧, ∨, →, ¬, …
    - Quantifiers: ∃, ∀
    - Equality: =
  - Formulas:
    - $\mathrm{Cat}(\mathrm{Felix}),\ \ \mathrm{Mat}(\mathrm{MyMat}),\ \ \text{sits-on}(\mathrm{Felix}, \mathrm{MyMat})$
    - $\mathrm{Cat}(x),\ \ \mathrm{Cat}(x) \vee \mathrm{Human}(x),\ \ \exists y.\mathrm{Mat}(y) \wedge \text{sits-on}(x, y)$
    - $\forall x.\mathrm{Cat}(x) \rightarrow \mathrm{Animal}(x),\ \ \forall x.\mathrm{Cat}(x) \rightarrow (\exists y.\mathrm{Mat}(y) \wedge \text{sits-on}(x, y))$

Crash Course in (simplified) FOL

- Semantics

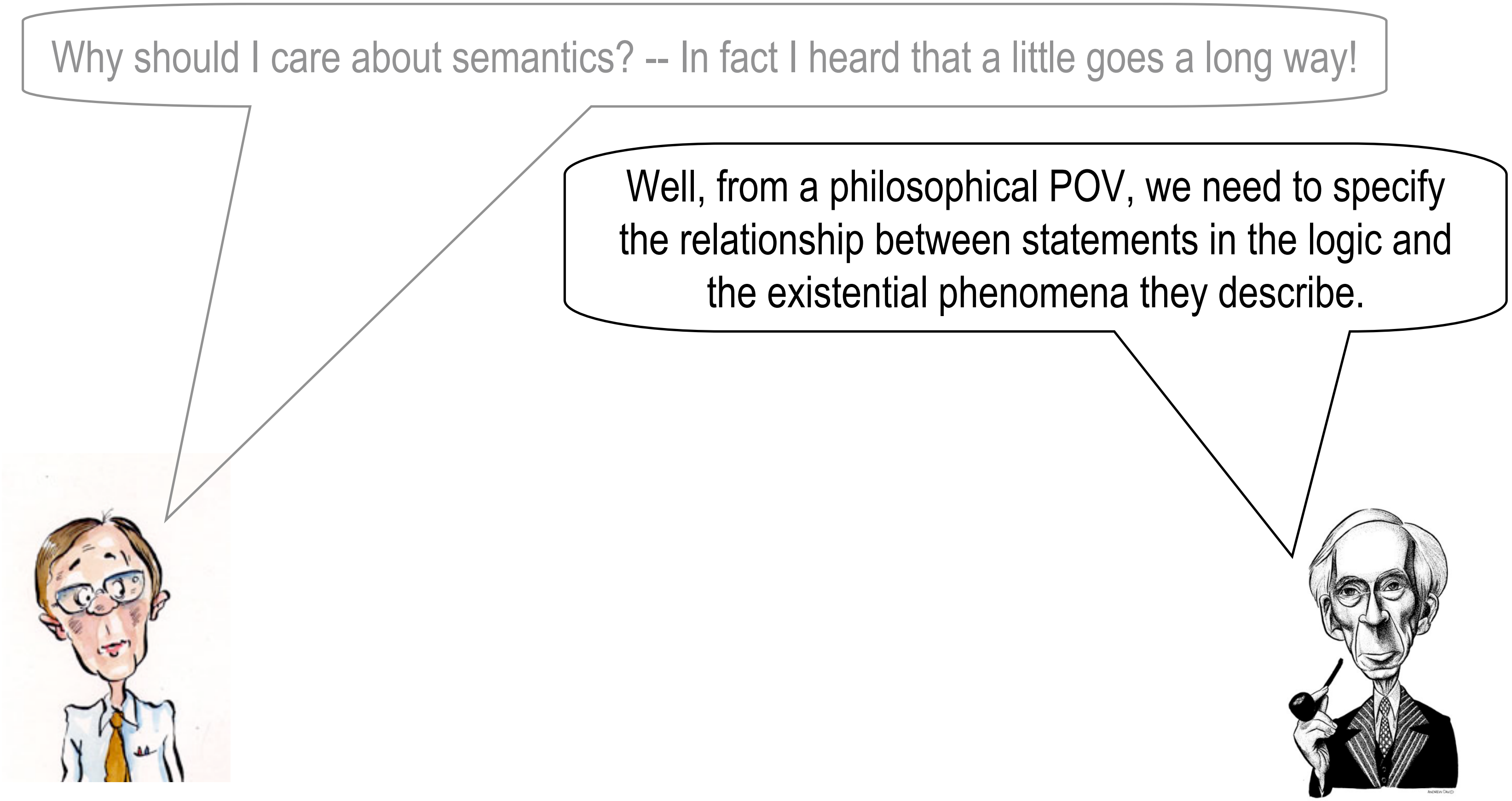
Crash Course in (simplified) FOL

- Semantics

Why should I care about semantics? -- In fact I heard that a little goes a long way!

Crash Course in (simplified) FOL

- Semantics



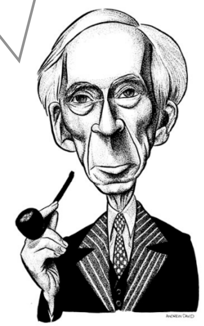
Crash Course in (simplified) FOL

- Semantics

Why should I care about semantics? -- In fact I heard that a little goes a long way!

Well, from a philosophical POV, we need to specify the relationship between statements in the logic and the existential phenomena they describe.

That's OK, but I don't get paid for philosophy.



Crash Course in (simplified) FOL

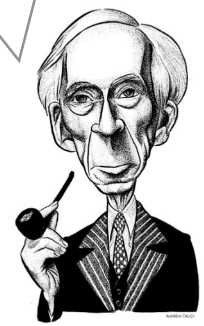
- Semantics

Why should I care about semantics? -- In fact I heard that a little goes a long way!

Well, from a philosophical POV, we need to specify the relationship between statements in the logic and the existential phenomena they describe.

That's OK, but I don't get paid for philosophy.

From a practical POV, we need to define relationships (like entailment) between logical statements -- without such a definition we can't spec software such as a reasoner.



Crash Course in (simplified) FOL

- Semantics

In FOL we define the semantics in terms of models (a model theory). A model is supposed to be an analogue of (part of) the world being modeled. FOL uses a very simple kind of model, in which “objects” in the world (not necessarily physical objects) are modeled as elements of a set, and relationships between objects are modeled as sets of tuples.

Crash Course in (simplified) FOL

- Semantics

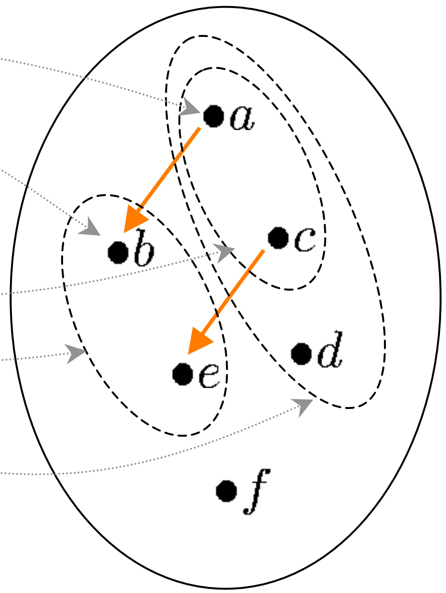
In FOL we define the semantics in terms of models (a model theory). A model is supposed to be an analogue of (part of) the world being modeled. FOL uses a very simple kind of model, in which “objects” in the world (not necessarily physical objects) are modeled as elements of a set, and relationships between objects are modeled as sets of tuples.

Note that this is exactly the same kind of model as used in a database: objects in the world are modeled as values (elements) and relationships as tables (sets of tuples).

Crash Course in (simplified) FOL

- Semantics
  - Model: a pair $\langle D, \cdot^I \rangle$ with D a non-empty set and $\cdot^I$ an interpretation
    - C^I is an element of D for C a constant
    - v^I is an element of D for v a variable
    - P^I is a subset of D^n for P a predicate of arity n
  - E.g., $D = \{a, b, c, d, e, f\}$, and

$$\begin{aligned}
\mathrm{Felix}^I &= a \\
\mathrm{MyMat}^I &= b \\
\mathrm{Cat}^I &= \{a, c\} \\
\mathrm{Mat}^I &= \{b, e\} \\
\mathrm{Animal}^I &= \{a, c, d\} \\
\text{sits-on}^I &= \{\langle a, b\rangle, \langle c, e\rangle\}
\end{aligned}$$



Crash Course in (simplified) FOL

- Semantics
  - Evaluation: truth value in a given model M = $\langle D, \cdot^I \rangle$
    - $P(t_1, \ldots, t_n)$ is *true* iff $\langle t_1^I, \ldots, t_n^I \rangle \in P^I$
    - $A \wedge B$ is *true* iff A is *true* and B is *true*

      $\neg A$ is *true* iff A is not *true*
  - E.g.,

$\mathrm{Cat(Felix)}$	true
$\mathrm{Cat(MyMat)}$	false
$\neg\mathrm{Mat(Felix)}$	true
$\mathrm{sits\text{-}on(Felix, MyMat)}$	true
$\mathrm{Mat(Felix)} \vee \mathrm{Cat(Felix)}$	true

$$
\begin{aligned}
&D = \{a, b, c, d, e, f\}\\
&\mathrm{Felix}^I = a\\
&\mathrm{MyMat}^I = b\\
&\mathrm{Cat}^I = \{a, c\}\\
&\mathrm{Mat}^I = \{b, e\}\\
&\mathrm{Animal}^I = \{a, c, d\}\\
&\mathrm{sits\text{-}on}^I = \{\langle a, b\rangle, \langle c, e\rangle\}
\end{aligned}
$$

Crash Course in (simplified) FOL

- Semantics
  - Evaluation: truth value in a given model M = $\langle D, \cdot^I \rangle$
    - $\exists x.A$ is *true* iff exists $\cdot^{I'}$ s.t. $\cdot^I$ and $\cdot^{I'}$ differ only w.r.t. x, and A is *true* w.r.t. $\langle D, \cdot^{I'} \rangle$
    - $\forall x.A$ is *true* iff for all $\cdot^{I'}$ s.t. $\cdot^I$ and $\cdot^{I'}$ differ only w.r.t. x, A is *true* w.r.t. $\langle D, \cdot^{I'} \rangle$

E.g.,

$\exists x.\mathrm{Cat}(x)$	true
$\forall x.\mathrm{Cat}(x)$	false
$\exists x.\mathrm{Cat}(x) \wedge \mathrm{Mat}(x)$	false
$\forall x.\mathrm{Cat}(x) \rightarrow \mathrm{Animal}(x)$	true
$\forall x.\mathrm{Cat}(x) \rightarrow (\exists y.\mathrm{Mat}(y) \wedge \text{sits-on}(x, y))$	true

$$\boxed{\begin{array}{l} D = \{a, b, c, d, e, f\} \\ \mathrm{Felix}^I = a \\ \mathrm{MyMat}^I = b \\ \mathrm{Cat}^I = \{a, c\} \\ \mathrm{Mat}^I = \{b, e\} \\ \mathrm{Animal}^I = \{a, c, d\} \\ \text{sits-on}^I = \{\langle a, b\rangle, \langle c, e\rangle\} \end{array}}$$

Crash Course in (simplified) FOL

- Semantics
  - Given a model M and a formula F, M is a model of F (written $\mathrm{M} \models \mathrm{F}$) iff F evaluates to true in M
  - A formula F is **satisfiable** iff there exists a model M s.t. $\mathrm{M} \models \mathrm{F}$
  - A formula F **entails** another formula G (written $\mathrm{F} \models \mathrm{G}$) iff every model of F is also a model of G (i.e., $\mathrm{M} \models \mathrm{F}$ implies $\mathrm{M} \models \mathrm{G}$)

E.g.,

$$M \models \exists x.\mathrm{Cat}(x)$$
$$M \not\models \forall x.\mathrm{Cat}(x)$$
$$M \not\models \exists x.\mathrm{Cat}(x) \wedge \mathrm{Mat}(x)$$
$$M \models \forall x.\mathrm{Cat}(x) \rightarrow \mathrm{Animal}(x)$$
$$M \models \forall x.\mathrm{Cat}(x) \rightarrow (\exists y.\mathrm{Mat}(y) \wedge \text{sits-on}(x, y))$$

$$\boxed{\begin{aligned}
&D = \{a, b, c, d, e, f\} \\
&\mathrm{Felix}^I = a \\
&\mathrm{MyMat}^I = b \\
&\mathrm{Cat}^I = \{a, c\} \\
&\mathrm{Mat}^I = \{b, e\} \\
&\mathrm{Animal}^I = \{a, c, d\} \\
&\text{sits-on}^I = \{\langle a, b\rangle, \langle c, e\rangle\}
\end{aligned}}$$

Crash Course in (simplified) FOL

- Semantics
  - Given a model M and a formula F, M is a model of F (written $\mathrm{M} \models \mathrm{F}$) iff F evaluates to true in M
  - A formula F is **satisfiable** iff there exists a model M s.t. $\mathrm{M} \models \mathrm{F}$
  - A formula F **entails** another formula G (written $\mathrm{F} \models \mathrm{G}$) iff every model of F is also a model of G (i.e., $\mathrm{M} \models \mathrm{F}$ implies $\mathrm{M} \models \mathrm{G}$)

E.g.,

✓ $\mathrm{Cat}(\mathrm{Felix}) \models \exists x.\mathrm{Cat}(x) \quad (\mathrm{Cat}(\mathrm{Felix}) \wedge \neg\exists x.\mathrm{Cat}(x) \text{ is not satisfiable})$

✓ $(\forall x.\mathrm{Cat}(x) \rightarrow \mathrm{Animal}(x)) \wedge \mathrm{Cat}(\mathrm{Felix}) \models \mathrm{Animal}(\mathrm{Felix})$

✓ $(\forall x.\mathrm{Cat}(x) \rightarrow \mathrm{Animal}(x)) \wedge \neg\mathrm{Animal}(\mathrm{Felix}) \models \neg\mathrm{Cat}(\mathrm{Felix})$

✘ $\mathrm{Cat}(\mathrm{Felix}) \models \forall x.\mathrm{Cat}(x)$

✘ $\text{sits-on}(\mathrm{Felix}, \mathrm{Mat1}) \wedge \text{sits-on}(\mathrm{Tiddles}, \mathrm{Mat2}) \models \neg\text{sits-on}(\mathrm{Felix}, \mathrm{Mat2})$

✘ $\text{sits-on}(\mathrm{Felix}, \mathrm{Mat1}) \wedge \text{sits-on}(\mathrm{Tiddles}, \mathrm{Mat1}) \models \exists^{\geq 2}x.\text{sits-on}(x, \mathrm{Mat1})$

Decidable Fragments

- FOL (satisfiability) well known to be undecidable
  - A sound, complete and terminating algorithm is impossible
- Interesting decidable fragments include, e.g.,
  - C2: FOL with 2 variables and Counting quantifiers $(\exists^{\geq n}, \exists^{\leq n})$
    - Counting quantifiers abbreviate pairwise (in-) equalities, e.g.:

      $\exists^{\geq 3} x.\mathrm{Cat}(x)$ equivalent to

      $\exists x, y, z.\mathrm{Cat}(x) \wedge \mathrm{Cat}(y) \wedge \mathrm{Cat}(z) \wedge x \neq y \wedge x \neq z \wedge y \neq z$

      $\exists^{\leq 2} x.\mathrm{Cat}(x)$ equivalent to

      $\forall x, y, z.\mathrm{Cat}(x) \wedge \mathrm{Cat}(y) \wedge \mathrm{Cat}(z) \rightarrow x = y \vee x = z \vee y = z$
  - Propositional modal and description logics
  - Guarded fragment

Back to our Scheduled Program

DL Syntax

- Signature
  - **Concept** (aka class) names, e.g., Cat, Animal, Doctor
    - Equivalent to FOL unary predicates
  - **Role** (aka property) names, e.g., sits-on, hasParent, loves
    - Equivalent to FOL binary predicates
  - **Individual** names, e.g., Felix, John, Mary, Boston, Italy
    - Equivalent to FOL constants

DL Syntax

- Operators
  - Many kinds available, e.g.,
    - Standard FOL Boolean operators ($\sqcap$, $\sqcup$, $\neg$)
    - Restricted form of quantifiers ($\exists$, $\forall$)
    - Counting ($\geq$, $\leq$, $=$)
    - …

DL Syntax

- Concept expressions, e.g.,
  - $\text{Doctor} \sqcup \text{Lawyer}$
  - $\text{Rich} \sqcap \text{Happy}$
  - $\text{Cat} \sqcap \exists \text{sits-on}.\text{Mat}$
- Equivalent to FOL formulae with one free variable
  - $\text{Doctor}(x) \vee \text{Lawyer}(x)$
  - $\text{Rich}(x) \wedge \text{Happy}(x)$
  - $\exists y.(\text{Cat}(x) \wedge \text{sits-on}(x, y))$

DL Syntax

- Special concepts
  - $\top$ (aka top, Thing, most general concept)
  - $\bot$ (aka bottom, Nothing, inconsistent concept)

  used as abbreviations for
  - $(A \sqcup \neg A)$ for any concept A
  - $(A \sqcap \neg A)$ for any concept A

DL Syntax

- Role expressions, e.g.,
  - loves^{-}
  - $\text{hasParent} \circ \text{hasBrother}$
- Equivalent to FOL formulae with two free variables
  - $\text{loves}(y, x)$
  - $\exists z.(\text{hasParent}(x, z) \wedge \text{hasBrother}(z, y))$

DL Syntax

- "Schema" Axioms, e.g.,
  - $\text{Rich} \sqsubseteq \neg\text{Poor}$ (concept inclusion)
  - $\text{Cat} \sqcap \exists\text{sits-on.Mat} \sqsubseteq \text{Happy}$ (concept inclusion)
  - $\text{BlackCat} \equiv \text{Cat} \sqcap \exists\text{hasColour.Black}$ (concept equivalence)
  - $\text{sits-on} \sqsubseteq \text{touches}$ (role inclusion)
  - Trans(part-of) (transitivity)
- Equivalent to (particular form of) FOL sentence, e.g.,
  - $\forall x.(\text{Rich}(x) \rightarrow \neg\text{Poor}(x))$
  - $\forall x.(\text{Cat}(x) \wedge \exists y.(\text{sits-on}(x,y) \wedge \text{Mat}(y)) \rightarrow \text{Happy}(x))$
  - $\forall x.(\text{BlackCat}(x) \leftrightarrow (\text{Cat}(x) \wedge \exists y.(\text{hasColour}(x,y) \wedge \text{Black}(y)))$
  - $\forall x,y.(\text{sits-on}(x,y) \rightarrow \text{touches}(x,y))$
  - $\forall x,y,z.((\text{sits-on}(x,y) \wedge \text{sits-on}(y,z)) \rightarrow \text{sits-on}(x,z))$

DL Syntax

- "Data" Axioms (aka Assertions or Facts), e.g.,
  - BlackCat(Felix) (concept assertion)
  - Mat(Mat1) (concept assertion)
  - Sits-on(Felix,Mat1) (role assertion)
- Directly equivalent to FOL "ground facts"
  - Formulae with no variables

DL Syntax

- A set of axioms is called a TBox, e.g.:

$\{$Doctor $\sqsubseteq$ Person,
Parent $\equiv$ Person $\sqcap$ $\exists$hasChild.Pers
HappyParent $\equiv$ Parent $\sqcap$ $\forall$hasChil

- A set of facts is called an AB

$\{$HappyParent(John),
hasChild(John,Mary)$\}$

Note

Facts sometimes written
John:HappyParent,
John hasChild Mary,
$\langle$John,Mary$\rangle$:hasChild

- A Knowledge Base (KB) is just a TBox plus an Abox
  - Often written $\mathcal{K} = \langle\mathcal{T}, \mathcal{A}\rangle$

The DL Family

- Many different DLs, often with “strange” names
  - E.g., $\mathcal{EL}$, $\mathcal{ALC}$, $\mathcal{SHIQ}$
- Particular DL defined by:
  - Concept operators ($\sqcap$, $\sqcup$, $\neg$, $\exists$, $\forall$, etc.)
  - Role operators ($^-$, $\circ$, etc.)
  - Concept axioms ($\sqsubseteq$, $\equiv$, etc.)
  - Role axioms ($\sqsubseteq$, Trans, etc.)

The DL Family

- E.g., $\mathcal{EL}$ is a well known "sub-Boolean" DL
  - Concept operators: $\sqcap$, $\neg$, $\exists$
  - No role operators (only atomic roles)
  - Concept axioms: $\sqsubseteq$, $\equiv$
  - No role axioms
- E.g.:

$$\text{Parent} \equiv \text{Person} \sqcap \exists \text{hasChild}.\text{Person}$$

The DL Family

- $\mathcal{ALC}$ is the smallest propositionally closed DL
  - Concept operators: $\sqcap$, $\sqcup$, $\neg$, $\exists$, $\forall$
  - No role operators (only atomic roles)
  - Concept axioms: $\sqsubseteq$, $\equiv$
  - No role axioms
- E.g.:

$$\text{ProudParent} \equiv \text{Person} \sqcap \forall \text{hasChild}.(\text{Doctor} \sqcup \exists \text{hasChild}.\text{Doctor})$$

The DL Family

- $\mathcal{S}$ used for $\mathcal{ALC}$ extended with (role) transitivity axioms
- Additional letters indicate various extensions, e.g.:
  - $\mathcal{H}$ for role hierarchy (e.g., hasDaughter $\sqsubseteq$ hasChild)
  - $\mathcal{R}$ for role box (e.g., $\mathrm{hasParent} \circ \mathrm{hasBrother} \sqsubseteq$ hasUncle)
  - $\mathcal{O}$ for nominals/singleton classes (e.g., {Italy})
  - $\mathcal{I}$ for inverse roles (e.g., isChildOf $\equiv$ hasChild^{-})
  - $\mathcal{N}$ for number restrictions (e.g., $\geqslant 2$hasChild, $\leqslant 3$hasChild)
  - $\mathcal{Q}$ for qualified number restrictions (e.g., $\geqslant 2$hasChild.Doctor)
  - $\mathcal{F}$ for functional number restrictions (e.g., $\leqslant 1$hasMother)
- E.g., $\mathcal{SHIQ}$ = $\mathcal{S}$ + role hierarchy + inverse roles + QNRs

The DL Family

- Numerous other extensions have been investigated
  - Concrete domains (numbers, strings, etc)
  - DL-safe rules (Datalog-like rules)
  - Fixpoints
  - Role value maps
  - Additional role constructors ($\cap$, $\cup$, $\neg$, $\circ$, id, …)
  - Nary (i.e., predicates with arity >2)
  - Temporal
  - Fuzzy
  - Probabilistic
  - Non-monotonic
  - Higher-order
  - …

DL Semantics

Via translaton to FOL, or directly using FO model theory:

Interpretation function $\mathcal{I}$ Interpretation domain $\Delta^{\mathcal{I}}$

Individuals $\mathrm{i}^{\mathcal{I}} \in \Delta^{\mathcal{I}}$

John

Mary

Concepts $\mathrm{C}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$

Lawyer

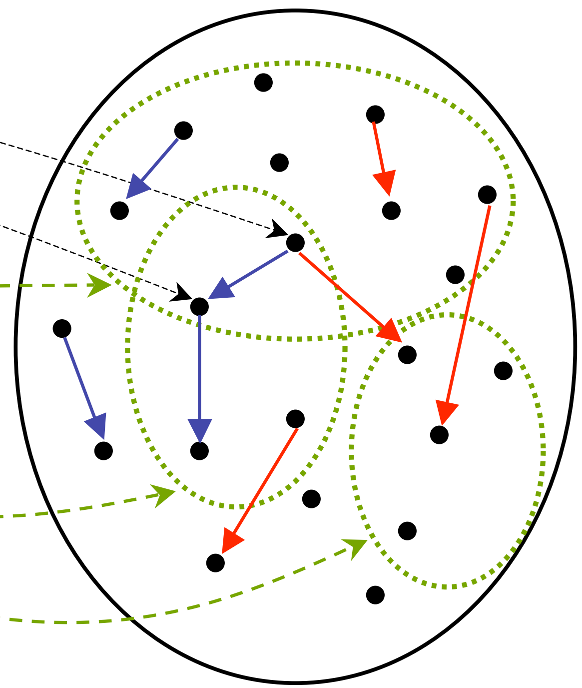
Doctor

Vehicle

Roles $\mathrm{r}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

hasChild

owns



DL Semantics

- Interpretation function extends to **concept expressions** in the obvious(ish) way, e.g.:

$$
\begin{aligned}
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
\{x\}^{\mathcal{I}} &= \{x^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} &= \{x \mid \exists y. \langle x, y \rangle \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} &= \{x \mid \forall y. (x, y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\} \\
(\leqslant n R)^{\mathcal{I}} &= \{x \mid \#\{y \mid \langle x, y \rangle \in R^{\mathcal{I}}\} \leqslant n\} \\
(\geqslant n R)^{\mathcal{I}} &= \{x \mid \#\{y \mid \langle x, y \rangle \in R^{\mathcal{I}}\} \geqslant n\}
\end{aligned}
$$

DL Semantics

- Given a model M = $\langle D, \cdot^{I} \rangle$
  - $M \models C \sqsubseteq D$ iff $C^{I} \subseteq D^{I}$
  - $M \models C \equiv D$ iff $C^{I} = D^{I}$
  - $M \models C(a)$ iff $a^{I} \in C^{I}$
  - $M \models R(a, b)$ iff $\langle a^{I}, b^{I} \rangle \in R^{I}$
  - $M \models \langle \mathcal{T}, \mathcal{A} \rangle$ iff for every axiom $ax \in \mathcal{T} \cup \mathcal{A}$, $M \models ax$

DL Semantics

- Satisfiability and entailment
  - A KB $\mathcal{K}$ is satisfiable iff there exists a model M s.t. $\mathrm{M} \models \mathcal{K}$
  - A concept C is satisfiable w.r.t. a KB $\mathcal{K}$ iff there exists a model $\mathrm{M} = \langle \mathrm{D}, \cdot^{\mathrm{I}} \rangle$ s.t. $\mathrm{M} \models \mathcal{K}$ and $\mathrm{C}^{\mathrm{I}} \neq \emptyset$
  - A KB $\mathcal{K}$ entails an axiom ax (written $\mathcal{K} \models \mathrm{ax}$) iff for every model M of $\mathcal{K}$, $\mathrm{M} \models \mathrm{ax}$ (i.e., $\mathrm{M} \models \mathcal{K}$ implies $\mathrm{M} \models \mathrm{ax}$)

DL Semantics

E.g.,

$\mathcal{T}$ = {Doctor $\sqsubseteq$ Person, Parent $\equiv$ Person $\sqcap$ $\exists$hasChild.Person,
HappyParent $\equiv$ Parent $\sqcap$ $\forall$hasChild.(Doctor $\sqcup$ $\exists$hasChild.Doctor)}

$\mathcal{A}$ = {John:HappyParent, John hasChild Mary, John hasChild Sally,
Mary:$\neg$Doctor, Mary hasChild Peter, Mary:($\leq$ 1 hasChild)}

✓ – $\mathcal{K} \models$ John:Person ?

✓ – $\mathcal{K} \models$ Peter:Doctor ?

✓ – $\mathcal{K} \models$ Mary:HappyParent ?

– What if we add "Mary hasChild Jane" ?

$\mathcal{K} \models$ Peter = Jane

– What if we add "HappyPerson $\equiv$ Person $\sqcap$ $\exists$hasChild.Doctor" ?

$\mathcal{K} \models$ HappyPerson $\sqsubseteq$ Parent

DL and FOL

- Most DLs are subsets of C2
  - But reduction to C2 may be (highly) non-trivial
    - Trans(R) naively reduces to $\forall x, y, z. R(x, y) \land R(y, z) \rightarrow R(x, z)$
- Why use DL instead of C2?
  - Syntax is succinct and convenient for KR applications
  - Syntactic conformance guarantees being inside C2
    - Even if reduction to C2 is non-obvious
  - Different combinations of constructors can be selected
    - To guarantee decidability
    - To reduce complexity
  - DL research has mapped out the decidability/complexity landscape in great detail
    - See Evgeny Zolin's DL Complexity Analyzer 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: *A*ttributive *L*anguage with *C*omplements

$\mathcal{ALC}$ ::= ⊥ | A | ¬C | $C \wedge D$ | $C \vee D$ | ∃$R.C$ | ∀$R.C$

Concept constructors:

- [ ] $\mathcal{F}$ – functionality[2]: (≤1 R)
- [x] $\mathcal{N}$ – (unqualified) number restrictions: (≥n R), (≤n R)
- [ ] $\mathcal{Q}$ – qualified number restrictions: (≥n $R.C$), (≤n $R.C$)
- [x] $\mathcal{O}$ – nominals: {a} or {a_1,...,a_n} ("one-of" constructor)
- [ ] μ – least fixpoint operator: $\mu X.C$
- [ ] $R \subseteq S$ – role-value-maps
- [ ] $f = g$ – agreement of functional role chains ("same-as")

Role constructors: trans reg

- [x] $\mathcal{I}$ – role inverses: R^{-}
- [ ] ∩ – role intersection[3]: $R \cap S$
- [ ] ∪ – role union: $R \cup S$
- [ ] ¬ – role complement: full
- [ ] o – role chain (composition): $R o S$
- [ ] * – reflexive-transitive closure[4]: R^*
- [ ] *id* – concept identity: $id(C)$

Forbid complex roles[5] in number restrictions[6]

TBox is *internalized* in extensions of $\mathcal{ALCIO}$, see [76, Lemma 4.12], [54, 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)

Role axioms (RBox): OWL-Lite OWL-DL OWL 1.1

- [x] $\mathcal{S}$ – Role transitivity: Trans(R)
- [x] $\mathcal{H}$ – Role hierarchy: $R \subseteq S$
- [ ] $\mathcal{R}$ – Complex role inclusions: $R o S \subseteq R$, $R o S \subseteq S$
- [ ] s – some additional features

Reset You have selected the Description Logic: $\mathcal{SHOIN}$

Complexity of reasoning problems[7]

Reasoning problem	Complexity[8]	Comments and references
Concept satisfiability	**NExpTime-complete**	• Hardness of even $\mathcal{ALCFIO}$ is proved in [76, Corollary 4.13]. In that paper, the result is formulated for $\mathcal{ALCQIO}$, but only number restrictions of the form (≤1R) are used in the proof. • A different proof of the NExpTime-hardness for $\mathcal{ALCFIO}$ is given in [54] (even with 1 nominal, and role inverses not used in number restrictions). • Upper bound for $\mathcal{SHOIQ}$ is proved in [77, Corollary 6.31] with numbers coded in unary (for binary coding, the upper bound remains an open problem for all logics in between $\mathcal{ALCNIO}$ and $\mathcal{SHOIQ}$). • **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 [46]. • **Remark:** recently [47] it was observed that, in many cases, one can use transitive roles in number restrictions – and still have a decidable logic! So the above notion of a *simple* role could be substantially extended.
ABox consistency	**NExpTime-complete**	By reduction to concept satisfiability problem in presence of nominals shown in [69, Theorem 3.7].

Complexity Measures

- Taxonomic complexity

  Measured w.r.t. total size of “schema” axioms

- Data complexity

  Measured w.r.t. total size of “data” facts

- Query complexity

  Measured w.r.t. size of query

- Combined complexity

  Measured w.r.t. total size of KB (plus query if appropriate)