DEFAULTS IN DESCRIPTION LOGICS:
SO SIMPLE, SO DIFFICULT

Piero A. Bonatti

Università di Napoli Federico II

DL & NMR, Roma, June 9, 2012
Outline

- Specifications with exceptions are common in human life
  - laws, taxonomies in natural science
  - O.O. languages commonly support them because they are useful in practice
- It is surprising that major KR formalisms do not support them
  - Description Logics (DL), OWL and its profiles
- This talk:
  - provides evidence that complexity is probably the main obstacle
  - argues that unnecessary complexity is caused by “abuse” of nonmonotonic constructs
  - advocates a prototype-oriented perspective
The disappearance of defaults

Nonmonotonic DL

Complexity sources

A new perspective

The End

Some history

NONMONOTONIC REASONING
& DESCRIPTION LOGICS

~ 30 years of attraction
Frame systems (starting from mid ’70s)

- The ancestors of Description Logics
- Support default attribute values / role fillers
- Inheritance with overriding
- LOOM (mid 80s–mid 90s)
  - (default Quaker Pacifist)
  - (get-role-default-values ...)

- Some old applications of defaults
  - Schank’s Memory Organization Packets
  - NLP, Linguistics (e.g. default verb tense formation)
Phase 2: The logical cleanup (from the end of the 70s on)

- Description logics provide
  - Unambiguous semantics to frame-like languages
  - Soundness and completeness criteria for engine validation
  - Complexity analysis for engine evaluation
  - ...

- Defaults disappear
  - WHY?
Why did defaults disappear?

- Not because they are useless
- New applications:
  - Prototype-driven ontology authoring (biomedical ontologies)
  - Default policy encoding in Description Logics (DL) (semantic web policies)
Prototypes and defaults in biomedical ontologies (I)

- In biomedical domains most properties/rules have exceptions

**Situs Inversus [Rector 2004]**

Human hearts are usually located on the left-hand side of the body. In humans with *situs inversus* the heart is located on the right-hand side.

**Eukaryotic cells [Stevens et al. 2007]**

Eukaryotic cells are—by definition—those with a proper nucleus. However, some cells that lack a proper nucleus—e.g. mammalian red blood cells—are considered eukaryotic, too.
Defaults in semantic web policies (I)

- Encoding policies in Description Logics (KAoS, Rei, ...)
  - e.g. access control policies
  - based on semantic metadata
  - uniform language for knowledge and policies

- Major expressiveness issues [B., Datalog 2.0 Workshop 2010]
  - including lack of support to default policies
  - open/closed policies (grant/deny access by default)
  - combinations thereof (relativized to groups, roles, ...)

- Other uses of defaults in this context
  - Emergency handling (overriding standard rules)
  - Conflict resolution
  - Incremental refinements
Defaults in semantic web policies (II)

Example: Policy formulation by refinements

- Users should not access confidential files
- Staff members should, instead
- Blacklisted staff cannot access confidential files

Emergency handling examples

- Blacklisted users cannot use the system
- This investigator is allowed to read admin files
- Replacement can do whatever Doctor-on-duty could
Monotonic workarounds (I)

- Linguistic primitives would help in managing defaults/exceptions

- Advantages in terms of usability / maintainability:
  - Incremental ontology refinements
  - Biologists etc. spontaneously adopt prototypes
  - Simpler ontologies
  - Less opportunities for errors

- Currently no support for prototypes / defaults
  - Workarounds based on monotonic DL
  - E.g. ontology design patterns
Monotonic workaround for eukaryotic cells [Stevens et al.’07]
- Introduce 2 disjoint classes for regular / exceptional instances
- Make them a partition of EukaryoticCell with covering axioms
- Distribute conditions across the two subclasses as appropriate, using negation and restrictions

⇒ Even if the original KB belongs to a low-complexity fragment (like OWL2-EL or OWL2-QL), the modified KB is not
Monotonic workarounds (III)

- Beyond design patterns [Rector 2004]

- Eukaryotic cells are one of the Simple cases:
  - Small number of (dimensions of) exceptions

- Complex cases: unbounded / unpredictable exceptions
  - Hybrid approaches (defaults processed outside the classifier)
  - Frame systems as high-level specification languages
  - to be “compiled” into description logic
Philosophical objections to defaults (I)

The purist point of view

- Ontologies should be sound, universal knowledge about the world
- If something is not *necessarily* true, then it is no ontology business

- Intellectually respectable viewpoint
- but it clashes with real-life user requirements
### Philosophical objections to defaults (II)

**Brachman’s point of view**

- Defaults interfere with structural classification
- which is based on necessary characterizing properties
- overriding mechanisms clash with this idea
- if every property can be overridden, which characterizing properties can a classifier exploit?

- Accordingly, KL-ONE’s terminologies do not support defaults
Philosophical objections to defaults (III)

- Eventually Brachman “surrenders” to user requirements
- In [Brachman, Schmolze 95] a 2-phase approach is suggested
  - Phase 1: Monotonic, structural classification
  - Phase 2: Add, propagate & override default attributes
  - Add default properties with Reiter’s default rules
- In the same year Baader & Hollunder investigate the idea
Undecidability issues

- Nonmonotonic Description Logics may be undecidable
  - Skolemized default rules [Baader, Hollunder]
  - Minimized/fixed roles under circumscription
  - Even if the underlying monotonic DL is very simple, e.g.
    - $\mathcal{ALC}$ with empty TBox [B., Lutz, Wolter 2009]
    - $\mathcal{EL}$ (F. Wolter)
Complexity issues

- Decidable Nonmonotonic Description Logics are typically very complex
  - $\mathcal{ALC} + \text{MKNF}$ (with restrictions): $3\text{-EXPTIME}$ algorithm
  - $\mathcal{ALC} + \text{Prioritized Circumscription}$: $\text{NExp}^{NP}$

- In some cases, same complexity as the monotonic version
  - $\mathcal{ALC} + \text{MKNF}$ (with restrictions) without quantifying in
  - e.g. $\exists R. K C$ is not allowed

- This is not expected to hold for tractable fragments
  - $\mathcal{EL}$, DL-lite, corresponding to OWL2-EL, OWL2-QL
  - Surely it does not hold under circumscription (more later)
Optimizations

- Asymptotic complexity is bad for classical DLs, too
  - (monotonic) $\mathcal{ALC}$ with general KBs is EXPTIME-complete

- Optimization and heuristics are necessary and available. Some of the most effective are:
  - Lazy unfolding + absorption
  - Tableaux caching
    - particularly effective for TBox classification
    - i.e. finding all mutual inclusions over a set of defined concepts
    - a concept may be considered many times
  - Dependency directed backtracking
Effectiveness of optimizations in classical DL

- Classification of a medium-size KB (4-5K concepts) as reported by the GALEN project:
  - mutual comparison over all pairs of concepts in the KB
  - without optimizations: weeks of CPU time
  - with optimizations: \(~60\) seconds
Lack of optimizations for nonmonotonic DL

- Classical optimizations rely heavily on monotonicity
- only lazy evaluation + absorption work out of the box
- caching and dependency directed backtracking don’t
Tableaux caching

Suppose open complete tableaux branches for $C$ and $D$ are cached.

A quick satisfiability test for $C \cap D$ can be obtained as follows.
Tableaux caching

Suppose open complete tableaux branches for $C$ and $D$ are cached.

A quick satisfiability test for $C \sqcap D$ can be obtained as follows.
Tableaux caching

Merge the two roots
Tableaux caching
Tableaux caching

If no rule is applicable in the new root then the new tableaux is complete as well $\Rightarrow C \sqcap D$ is satisfiable
Tableaux caching

If no rule is applicable in the new root then the new tableaux is complete as well \( \Rightarrow C \sqcap D \) is satisfiable

Note: only the root and its children need to be cached
Tableaux caching in nonmonotonic DL

Branch **closure depends on underivability** (new “clashes”)

- **Propositional circumscription** [Niemelä TABLEAUX’96]: a branch is closed if *ungrounded*
  - negative literals in the branch + KB $\not\models$ positive facts
  - adapted to DL in [Grimm, Hitzler 2009]

- **Default logics** [Amati et al JLC’96]: branch not *saturated* if some default is applicable ($\Delta^-, \alpha^- \not\models$) and ...

- **Autoepistemic DLs** [Donini et al TOCL’02]: branches are closed when $Ob_K(\mathcal{B}) \not\models Ob_A(\mathcal{B})$ or ...

- ...

P. A. Bonatti
Università di Napoli

Defaults in Description Logics – DL & NMR 2012
Impact of underivability tests on tableaux caching

Since branch closure depends on underivability

- The very idea of caching becomes inapplicable
- When two roots are merged, their formulae entail *more consequences*
  - some underivability test may be affected
  - the union of two open tableaux might not be open
- One should cache the *whole* tableaux and re-check openness
  - much more expensive, where is the advantage?
Dependency directed backtracking

- Label each concept with the nondeterministic choices that cause its introduction
- Labels identify which choices led to a clash
- During backtracking, skip the choice points not involved in the clash
A branch may be closed because some test $T \not\models \alpha$ succeeds.

How to track the dependency of $T \not\models \alpha$ from previous nondeterministic choices?

Underderivability depends on the absence of information:

- a nondeterministic choice is responsible for $T \not\models \alpha$ if some alternative choice helps in proving $\alpha$.

$\Rightarrow$ the proof that a nondeterministic choice is involved in a failure is not available when needed.

$\Rightarrow$ novel dependency analysis needed (if at all possible).
Diagnosis

- Defaults / NMR not (yet) supported in DL mainly because:
  - high computational complexity
  - not moderated by any effective heuristics
  - *complexity sources?*
NONMONOTONIC DESCRIPTION LOGICS

Some syntax and semantics
Description Logics in one slide (by translation in FOL)

- Some constructs require second order logic (but not today)

Translation of $\mathcal{ALC}$ concepts $C \leadsto C^i$ using variables $x_1, x_2, \ldots$

<table>
<thead>
<tr>
<th>$A^i$</th>
<th>$A(x_i)$</th>
<th>(A a concept name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\neg C)^i$</td>
<td>$\neg C^i$</td>
<td></td>
</tr>
<tr>
<td>$(C \sqcap D)^i$</td>
<td>$C^i \land D^i$</td>
<td></td>
</tr>
<tr>
<td>$(C \sqcup D)^i$</td>
<td>$C^i \lor D^i$</td>
<td></td>
</tr>
<tr>
<td>$(\exists R.C)^i$</td>
<td>$\exists x_{i+1}. R(x_i, x_{i+1}) \land C^{i+1}$</td>
<td>(R a role name)</td>
</tr>
<tr>
<td>$(\forall R.C)^i$</td>
<td>$\forall x_{i+1}. R(x_i, x_{i+1}) \rightarrow C^{i+1}$</td>
<td></td>
</tr>
</tbody>
</table>

- Fragment of FOL $\Rightarrow$ nonmonotonic extensions semantically straightforward

$C \sqsubseteq D \leadsto \forall x_0.C^0 \rightarrow D^0$ (inclusions)

$C(a) \leadsto C^0[a/x_0]$ (assertions)
## Main approaches until 2002

<table>
<thead>
<tr>
<th>Ref</th>
<th>DL</th>
<th>NM features</th>
<th>Complexity</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Brewka, 1987)</td>
<td>frame lang.</td>
<td>Circ</td>
<td>n.a.</td>
<td>Y</td>
</tr>
<tr>
<td>(Cadoli, Donini, &amp; Schaerf, 1990)</td>
<td>$&lt; \mathcal{ALE}$</td>
<td>Circ</td>
<td>in $\Sigma^p_2$</td>
<td>N</td>
</tr>
<tr>
<td>(Padgham &amp; Zhang, 1993)</td>
<td>$\mathcal{ALC}$ with concrete domains</td>
<td>inheritance networks</td>
<td>n.a.</td>
<td>Y</td>
</tr>
<tr>
<td>(Straccia, 1993)</td>
<td>$\mathcal{ALC}$</td>
<td>prioritized default logic</td>
<td>decidable</td>
<td>Y</td>
</tr>
<tr>
<td>(Baader &amp; Hollunder, 1995a)</td>
<td>$\mathcal{ALCF}$</td>
<td>default logic</td>
<td>decidable</td>
<td>N</td>
</tr>
<tr>
<td>(Baader &amp; Hollunder, 1995b)</td>
<td>$\mathcal{ALC}$</td>
<td>prioritized default logic</td>
<td>decidable</td>
<td>Y</td>
</tr>
<tr>
<td>(Lambrix, Shahmehri, &amp; Wahlloef, 1998)</td>
<td>$\mathcal{ALCQO}+$ feature agreement</td>
<td>prioritized default logic</td>
<td>n.a.</td>
<td>Y</td>
</tr>
<tr>
<td>(Donini et al., 1997)</td>
<td>any decidable DL</td>
<td>MKNF with restrictions</td>
<td>depends on DL</td>
<td>N</td>
</tr>
<tr>
<td>(Donini et al., 2002)</td>
<td>$\mathcal{ALC}$</td>
<td>MKNF with restrictions</td>
<td>in 3-ExpTime</td>
<td>N</td>
</tr>
</tbody>
</table>
Description logics + Default logic [Baader & Hollunder’95]

Vintage example

\[
Penguin \sqsubseteq Bird
\]
\[
Penguin \sqsubseteq \forall \text{behavior}. \neg \text{Fly}
\]

\[
\text{Bird}(x) : (\exists \text{behavior}. \text{Fly})(x)
\]
\[
(\exists \text{behavior}. \text{Fly})(x)
\]
Description Logics + MKNF [Donini et al.’97+2002]

- The logic of Minimal Knowledge and Negation as Failure generalizes Autoepistemic Logic
- it has 2 modal operators:
  - **K**: (∼standard) knowledge
  - **A**: nonmonotonic assumptions
- negation as failure encoded as: \( \neg A \)

**Vintage example in MKNF**

The default rule can be equivalently encoded as:

\[
\text{KB}ird \sqcap \neg A \neg (\exists \text{behavior}.Fly) \sqsubseteq \exists \text{behavior}.Fly
\]

Note that variables are implicit, as in DL
Description Logics + Rules [Motik & ...]

- **Rules**: Extensions of Datalog with
  - nonmonotonic negation (stable models, Answer Set Programs)
  - disjunctions
  - special operators for querying / updating terminologies
  - ...
- Implemented (e.g. with the DLV ASP engine)
- It would deserve a talk in itself
Known issues of Default Logic, MKNF, DL+Rules

- For decidability defaults/rules must be restricted to
  - *named individuals* only (the constants occurring in the KB)
  - no defaults apply to the implicit individuals generated by $\exists$

- MKNF and DL+Rules do not support priorities / specificity
  - difficult to model overriding
  - the embeddings in ASP are complicated
  - sometimes unsatisfactory / restricted

- MKNF must support quantification through modal operators
  - as in $\exists R.KC$
  - one fixed domain for all interpretations:
  - a denumerable set of *standard names*
Description Logics + Circumscription

- One of the earliest approaches (also in DL)
  - [Brewka’87, Frame language]
  - Cadoli et al. 1990
  - ⟨gap⟩
  - Resumed in 2006 by Lutz, Wolter, myself

- A preferential semantics
  - A partial preference ordering $<$ over interpretations
  - Preferential entailment:

$$\text{Is } \varphi \text{ true in all } <\text{-minimal models of KB?}$$
Description Logics + Circumscription

- Less problematic than other semantics
- No restrictions on interpretation domains
- No restrictions to named individuals
- Priorities supported since the earliest versions [McCarthy’80]
- Some restrictions still needed for decidability
  - but not so much on axioms
  - rather on the parameters that control circumscription...
Description Logics + Circumscription [B,Lutz,Wolter’06]

- Circumscription pattern CP
  (i.e. circumscription’s parameters)
  - Fixed predicates $F$: retain their classical semantics
  - Minimized predicates $M$: represent “abnormal” situations
  - Variable predicates $V$: may be affected by $M$’s minimization
  - Priority $\prec$ over $M$ (useful for specificity/overriding)

- CP determines the preference ordering $<$ over interpretations.
  If $\mathcal{I} < \mathcal{J}$ then:
  - fixed predicates have the same extension in $\mathcal{I}$ and $\mathcal{J}$
  - the extension of minimized predicates is smaller in $\mathcal{I}$ than in $\mathcal{J}$
  - exception: if this is not true then some higher-priority predicate has a smaller extension in $\mathcal{I}$
    (“smaller” means “subset”)
Description Logics + Circumscription [B,Lutz,Wolter’06]

- If no roles are minimized or fixed then:
  - Finite model property
    - of course, the underlying DL must enjoy it, too...
    - we proved it up to $ALCQO$ and $ALCIO$
    - (monotonic $ALCQIO$ does not have the f.m.p.)

⇒ No inconsistency from nonmonotonicity

- i.e. if KB classically consistent, then so is its circumscription
- Default Logic and MKNF do not enjoy this property
- no infinite $<$-chain below a finite model

⇒ Decidability

- reasoning is very complex, though: $NExp^{NP}$
Description logics + Typicality [Giordano et al.’08–]

- Preferential semantics
- Typicality relation over *individuals*
- Intriguing mix of similarities & differences with Circumscription
  - yet to be investigated
- Unfortunately, no time for describing it here
Some further interesting open questions

- Completing the complexity analysis of nonmonotonic DL
  - including circumscribed DL without the finite model property

- Priorities in DL + Default Logic / MKNF
  - impact on complexity / decidability
  - assessment of different semantics
    (Baader & Hollunder’s, Brewka’s, lexicographic, ...)

- Optimization techniques for all nonmonotonic DL
TOWARDS LOW-COMPLEXITY NONMONOTONIC DL

Complexity sources
Attacking complexity issues

- Three main possible lines of attack:
  1. Less complex semantics
  2. Optimizations & heuristics
  3. Syntax restrictions

- Here we focus on 3
  - little to say about 2
  - semantics should be motivated by modelling needs / expressiveness

- within DL + Circumscription
  - circumscription has nice properties
  - comparatively little to say about the other semantics
Starting point: tractable monotonic fragments

**DL-lite** — underlying the OWL2-QL profile

restricts concept inclusions to expressions $C_L \sqsubseteq C_R$, where

$$
C_L ::= A \mid \exists R. \top \\
C_R ::= C_L \mid \neg C_L
$$

**EL** — foundation of the OWL2-EL profile

$$
C ::= A \mid \top \mid C_1 \sqcap C_2 \mid \exists P.C
$$

$\mathcal{EL}^\perp$ is the extension of $\mathcal{EL}$ with the empty concept $\bot$
Additional restrictions [B,Faella,Sauro’09–]

- Restricted & implicit minimized predicates
- Replaced by defeasible inclusions
  \[ C \sqsubseteq_n D \] (“a C is normally a D”)
- Preferred models maximize the sets of individuals satisfying
  \[ C \sqsubseteq D \]
  \[ \text{sat}_I(C \sqsubseteq_n D) = \{ x \in \Delta^I \mid x \notin C^I \text{ or } x \in D^I \} \]
- Conflicts resolved using priority \( \prec \) over defeasible inclusions
Defeasible inclusions, $\prec$, and fixed predicates determine preferences over interpretations

**Definition** ($D =$ defeasible inclusion set; $F =$ fixed predicates)

For all interpretations $\mathcal{I}$ and $\mathcal{J}$, let $\mathcal{I} < \mathcal{J}$ iff:

1. $\Delta^\mathcal{I} = \Delta^\mathcal{J}$;
2. $a^\mathcal{I} = a^\mathcal{J}$, for all constants $a$;
3. $A^\mathcal{I} = A^\mathcal{J}$, for all $A \in F$;
4. for all $\delta \in D$, if sat$_\mathcal{I}(\delta) \nsubseteq$ sat$_\mathcal{J}(\delta)$ then there exists $\delta' \in D$ such that $\delta' \prec \delta$ and sat$_\mathcal{I}(\delta') \supset$ sat$_\mathcal{J}(\delta')$;
5. there exists a $\delta \in D$ such that sat$_\mathcal{I}(\delta) \supset$ sat$_\mathcal{J}(\delta)$.
Example: Eukaryotic cells

Knowledge base

EukaryoticCell ⊑ n ∃has nucleus
MammalianRedBloodCell ⊑ EukaryoticCell
MammalianRedBloodCell ⊑ ¬∃has nucleus

Some consequences (preferential entailment)

EukaryoticCell ⊓ ¬MammalianRedBloodCell ⊑ ∃has nucleus
EukaryoticCell ⊓ ∃has nucleus ⊑ ¬MammalianRedBloodCell
Example: Access control policy

Knowledge base

\[
\begin{align*}
\text{Staff } &\sqsubseteq \text{Users} \quad \text{Blacklisted } \sqsubseteq \text{Staff} \\
\text{UserRequest} &\equiv \exists \text{subject}.\text{Users} \sqcap \exists \text{target}.\text{Proj} \sqcap \exists \text{op}.\text{Read} \\
\text{StaffRequest} &\equiv \exists \text{subject}.\text{Staff} \sqcap \exists \text{target}.\text{Proj} \sqcap \exists \text{op}.\text{Read} \\
\delta_1 : \text{UserRequest} &\sqsubseteq_n \exists \text{decision}.\text{Deny} \\
\delta_2 : \text{StaffRequest} &\sqsubseteq_n \exists \text{decision}.\text{Permit} \\
\exists \text{subject}.\text{Blacklisted} &\sqsubseteq \exists \text{decision}.\text{Deny} \\
\exists \text{decision}.\text{Permit} \sqcap \exists \text{decision}.\text{Deny} &\sqsubseteq \bot
\end{align*}
\]

Some consequences

\[
\text{StaffRequest} \sqcap \neg \exists \text{subject}.\text{Blacklisted} \sqsubseteq \exists \text{decision}.\text{Permit}
\]
Simulating nominals in $\mathcal{EL}^\bot$

\[ \top \sqsubseteq \exists aux. N \quad \text{make } N \text{ nonempty} \]
\[ N \sqsubseteq_n \bot \quad \text{minimize } N's \text{ extension} \]

If $aux, N \notin F$ (not fixed) then $\forall <$-minimal models $I$ of the KB,

\[ |N^I| = 1 \]

**Fragile:** if you add $N(a)$ and $N(b)$ with $a \neq b$ then $|N^I| = 2$
Simulating exclusive disjunctions in $\mathcal{EL}^\perp$

\[
\top \sqsubseteq_n C \\
\top \sqsubseteq_n D \\
C \sqcap D \sqsubseteq \bot
\]

If $C, D \not\in F$ (not fixed) then all $\prec$-minimal models $\mathcal{I}$ of the KB satisfy

\[
\top \sqsubseteq C \sqcup D
\]

Fragile: e.g. $C \sqsubseteq \bot$ and $D \sqsubseteq \bot$ destroy this

Similarly with $C \sqsubseteq_n \bot$ and $D \sqsubseteq_n \bot$ if $\prec$ based on specificity
Searching for individuals with property $C$

$\top \sqsubseteq_n \exists aux. C$

$\top \sqsubseteq \exists aux. C$ preferentially entailed iff for all $\prec$-minimal models $\mathcal{I}$ of the KB

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

It performs a (non-guarded) quantification over the whole domain!
Simulating $\mathcal{EL} +$ atomic negation (ExpTime-hardness)

1. Replace literals $\neg A$ with fresh atoms $\bar{A}$
2. Entail disjoint union “$\forall x, A(x)$ is true, false, or undefined” using defeasible inclusions

$$T \subseteq A \cup \bar{A} \cup U_A$$

3. Search for undefined values

$$T \subseteq_n \exists aux. U_A$$

4. Create a single “sensor” $U$ for undefined values

$$U_A \subseteq U$$

The given KB is classically inconsistent iff its translation entails

$$T \subseteq \exists aux. U$$
Complexity sources (supporting the above constructions)

- Fixed predicates
- Qualified existentials on the l.h.s. of inclusions
- Nested quantifiers
- Priorities, including specificity vs. general priorities
- Subconcept defaults
  - $A \sqsubseteq C$ is valid depending on
  - the default properties of all the subclasses $B \sqsubseteq A$
  - more about this later
# Complexity analysis – current state

<table>
<thead>
<tr>
<th>DL-lite$_R$</th>
<th>$\mathcal{EL}$</th>
<th>$\mathcal{EL}^\perp$</th>
</tr>
</thead>
<tbody>
<tr>
<td>var</td>
<td>fix$^{(\dagger)}$</td>
<td>var</td>
</tr>
<tr>
<td>concept satisfiability</td>
<td>$\Sigma^p_2$</td>
<td>$\leq$</td>
</tr>
<tr>
<td>subsumption &amp; instance checking</td>
<td>$\Pi^p_2$</td>
<td>$\leq$</td>
</tr>
</tbody>
</table>

$(\dagger)$ with specificity-based priorities; general priorities make var at least as complex as fix
$(\dagger)$ if quantifier nesting is bounded in the r.h.s. of subsumptions and in instance checking problems
$(\dagger)$ if DIIs are left-fixed or the priority relation is empty
$(\#)$ membership holds if the priority relation is empty and condition $(\dagger)$ holds

- $LL_f$ (*full left local*): no qualified existentials ($\exists R.C$) in the l.h.s. of inclusions, only $\exists R.\top$
- $aLL$: add acyclic definitions $A \equiv C$ whose unfolding is $LL_f$
- $LL_2$: $LL_f$ without quantifier nesting
- $LL$: $LL_2$ without default conflicts
HOW CAN SOMETHING AS SIMPLE AS PROTOTYPES BE SO COMPLEX?

On the wrong use of formalism
Question 1: Is this a prototype?

Example 1 (simulating nominals)

\[ N \sqsubseteq_n \bot \]

Example 2 (simulating disjunction)

\[ \top \sqsubseteq_n C \quad \top \sqsubseteq_n D \quad C \cap D \sqsubseteq \bot \]

- Prototypes are expected to admit exceptions
- Not to be be *incoherent*
- Why computing the invariants across all optimal repairs?
Q2: Is preferential entailment what you really want?

Knowledge base

- EukaryoticCell ⊑∃ has_nucleus
- MammalianRedBloodCell ⊑ EukaryoticCell ⊓¬∃ has_nucleus
- EukaryoticCell(c)

Not a consequence

∃ has_nucleus(c)
because c might be a MammalianRedBloodCell!

- You want the prototypical properties of A
- *not* the common properties of all prototypical B s.t. B ⊑ A...
Q3: How much circumscription do you really want?

Knowledge base

\[
\text{EukaryoticCell} \sqsubseteq_n \exists \text{has\_nucleus} \\
\text{Mammalian\_Red\_Blood\_Cell} \sqsubseteq \text{EukaryoticCell} \sqcap \neg \exists \text{has\_nucleus}
\]

An undesirable consequence

\[
\text{Mammalian\_Red\_Blood\_Cell} \sqsubseteq \bot
\]

because \text{Mammalian\_Red\_Blood\_Cells} never satisfy the defeasible inclusion!

- In general, a CWA effect on all exceptional concepts
- Extending legacy ontologies is very difficult
A modest proposal

- Reject inconsistent prototypes
- Reason about prototype properties, not preferential entailment
- Fix all concept names
- Circumscribed $\mathcal{EL}^{++}$ becomes almost tractable
Pushing the $\mathcal{EL}$ envelope [Baader et al.'05]

$\mathcal{EL}^{++} = \mathcal{EL}^\bot$ plus:

- nominals $\{a\}$
- role inclusions $R_1 \circ \cdots \circ R_n \sqsubseteq R$
- concrete domains (numbers, strings, ...)
  - concrete features $f, f_1, \ldots$ (functional roles ranging over c.d.)
  - suitable predicates $p(f_1, \ldots, f_n)$
  - some restrictions (convexity) for tractability
Further restrictions for circumscription tractability

- Under Circumcription, for tractability we have to exclude
  - role composition
  - variable formulae in $\exists$ (it affects $\exists$-nesting)

Operational definition (Standard KB)

Any set of $\mathcal{EL}^{++}$ inclusions that satisfy the following conditions:

- no quantifier nesting, e.g. $\exists R. \exists S. \top$
  - but you can use $\exists R. A$ and $A \equiv \exists S. \top$

- no role composition

- if $\exists R. A$ occurs in $\sqsubseteq_n$, then $A$ is nonempty
  - e.g. by $\top \sqsubseteq \exists aux. A$

- all concept names are fixed
Excluding inconsistent prototypes

All conflicts that cannot be resolved by specificity are forbidden:

**Definition (Strongly conflict safe KB)**

For $A$ and all $B_1 \sqsubseteq_n C_1, \ldots, B_n \sqsubseteq_n C_n$ applicable to $A$,

$$A \sqcap C_1 \sqcap \ldots \sqcap C_n$$

is consistent

Where $B_i \sqsubseteq_n C_i$ is applicable to $A$ if:

- $A \sqsubseteq B_i$
- $B_i \sqsubseteq_n C_i$ is not overridden
  - by any $B \sqsubseteq_n C$ applicable to $A$
  - with higher priority ($B \sqsubseteq_{KB} B_i$ and $B_i \not\sqsubseteq_{KB} B$)

Can be checked in $\text{PTIME}$
Reasoning about prototypes (not preferential entailment)

A mix of Closed World Assumption and preferential entailment:

**Definition (New reasoning task)**

$$\text{Circ}(KB) \models_{cw} A \sqsubseteq C \text{ iff for all } <\text{-minimal models } I \text{ of } KB$$

$$I \models CWA_{KB}(A) \sqsubseteq C$$

where $$CWA_{KB}(A) = A \cap \{\neg B \mid A \not\sqsubseteq_{KB} B\}$$

$$CWA_{KB}(A)$$ always consistent if $$KB$$ in $$\mathcal{EL}^{++}$$ or DL-lite
**Example**

**Knowledge base**

| EukaryoticCell $\subseteq_n \exists \text{has nucleus} | MammalianRedBloodCell \subseteq \text{EukaryoticCell} \cap \neg \exists \text{has nucleus} |
|---------------------------------------------------------|
| EukaryoticCell(c) | |

**New consequences with** $\models_{cw}$

| EukaryoticCell $\subseteq \exists \text{has nucleus} | \{c\} \subseteq \exists \text{has nucleus} |
|---------------------------------------------------------|

If all you know about $c$ is that it is a eukaryotic cell, then you can conclude that it has a nucleus.
Tractability

Assume that $KB$ is standard and strong conflict safe:

**Theorem**

*If $C$ does not contain nested quantifiers, $\text{Circ}(KB) \models_{cw} A \sqsubseteq C$ is in $\text{PTIME}$*

However...

**Theorem**

$\text{Circ}(KB) \models_{cw} A \sqsubseteq C$ is $\text{coNP}$-hard if some of the following conditions holds:

- $C$ contains nested quantifiers
- *role composition occurs in $KB$*
- *concept names may vary*
What if prototypes are rejected

New (tractable) debugging services

1. List all concepts $A$ with inconsistent prototypes
   - show the conjuncts of the prototype
   - can be fixed with specific axioms $A \sqsubseteq_n ...$

2. Automatically remove inconsistent prototypes
   - block all defeasible inclusions applicable to $A$
   - by adding specific axioms $A \sqsubseteq_n ...$ and constraints
   - alternatively: minimal repairs (higher cost)
Summarizing

- Nonmonotonic reasoning can simulate many constructs
  - disjunctions, negation, nominals, ...
- It makes little sense to remove them from the monotonic fragment and reintroduce them with NMR
- Formalism abuses/wrong uses (wrt. prototype-based design)
  - inconsistent prototypes, wrong reasoning task and circumscription parameters
- Removing them brings us much closer to our goal
  - practical inheritance with overriding, default role fillers
To-do list

- Investigating the new perspective
  - other language abuses
  - other nonmonotonic logics
  - including ASP and Datalog
- DL-lite / conjunctive queries (OWL2-QL)
- Relaxing constraints (esp. on roles & quantification)
- Extend to more expressive languages
  - $\models_{cw}$ in $\mathcal{ALC}$
  - more general circumscription patterns
  - optimizations needed
- Experimental evaluation of usability & scalability
  - where are the benchmarks?...
Time for questions & feedback

Acknowledgements

- Carsten Lutz, Frank Wolter
- Marco Faella, Luigi Sauro
- LoDeN
  http://loden.fisica.unina.it/

Remembering Marco Cadoli and all of his great work