Nonstandard Inferences in Description Logics

Franz Baader
Theoretical Computer Science
RWTH Aachen
Germany

- Short introduction to Description Logics
- Application in chemical process engineering
- Non-standard inferences least common subsumer, most specific concept, rewriting, and matching

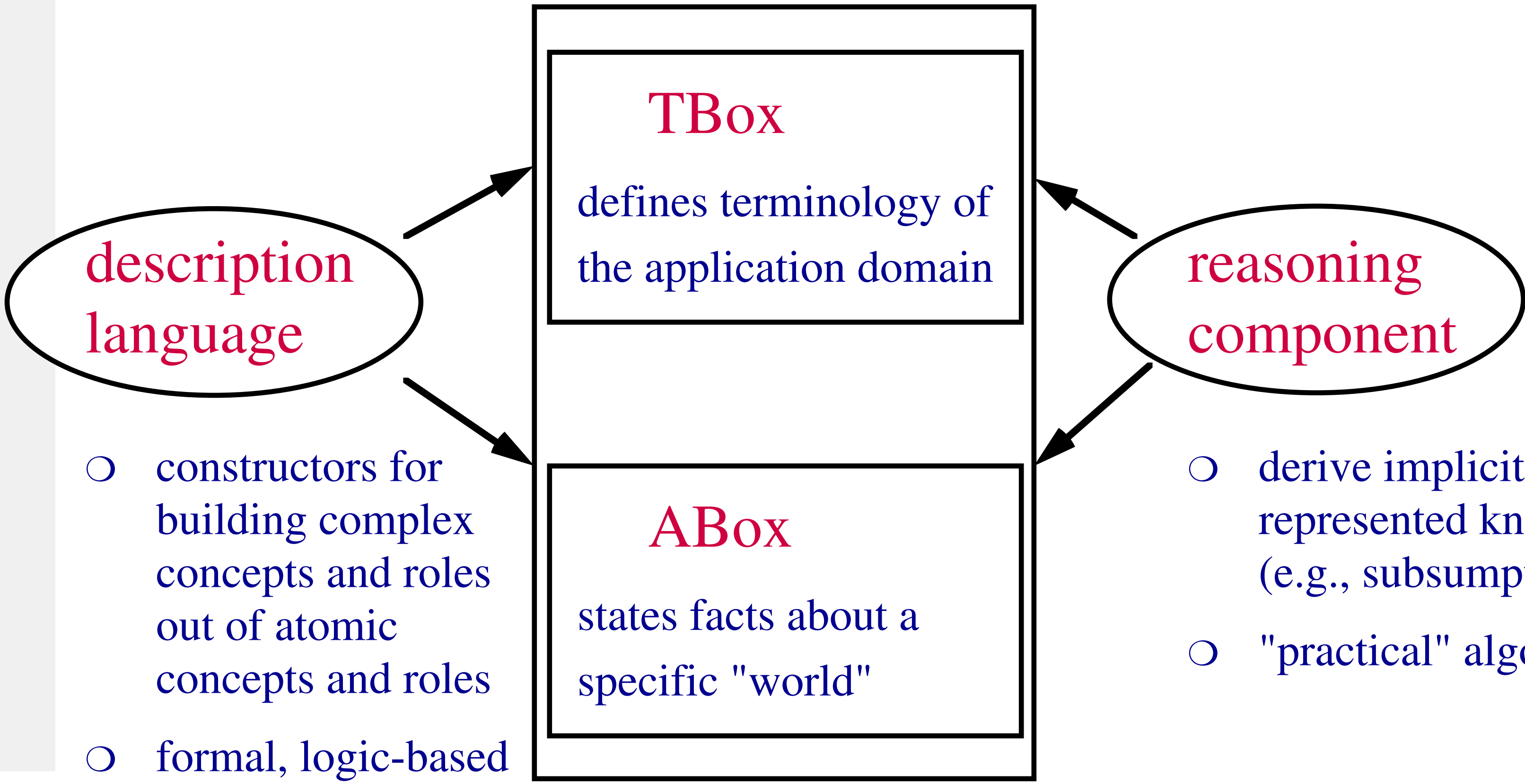
Description logics

class of knowledge representation formalisms

- Descended from structured inheritance networks [Brachman 78].
- Tried to overcome ambiguities in semantic networks and frames
  that were due to their lack of a formal semantics.
- Restriction to a small set of "epistemologically adequate" operators
  for defining concepts (classes).
- Importance of well-defined basic inference procedures:
  subsumption and instance problem.
- First realization: system KL-ONE [Brachman&Schmolze 85],
  many successor systems (Classic, Crack, FaCT, Flex, Kris, Loom, Race...).
- First application: natural language processing;
  now also other domains (configuration, medical terminology, databases,
  chemical process engineering, ontologies for the semantic web,...)

Description logic systems

structure



description language

- constructors for building complex concepts and roles out of atomic concepts and roles
- formal, logic-based semantics

knowledge base

reasoning component

- derive implicitly represented knowledge (e.g., subsumption)
- "practical" algorithms

Rheinisch-
Westfälische
Technische
Hochschule
Aachen
RWTH

Description language

examples of typical constructors:

$C \sqcap D$, $\neg C$, $\forall r . C$, $\exists r . C$, $(\geq n\ r)$

A man	Human $\sqcap$ $\neg$ Female $\sqcap$
that is married to a doctor, and	$\exists$ married-to . Doctor $\sqcap$
has at least 5 children,	($\geq$ 5 child) $\sqcap$
all of whom are professors.	$\forall$ child . Professor

TBox

definition of concepts

Happy-man = Human $\sqcap$ ...

ABox

properties of individuals

Happy-Man(Franz)

child(Franz,Luisa)

child(Franz,Julian)

Formal semantics

based on interpretations as in predicate logic

An interpretation I associates

- concepts C with sets C^I and
- roles r with binary relations r^I.

The semantics of the constructors is defined through identities:

- $(C \sqcap D)^I = C^I \cap D^I$
- $(\geq n\ r)^I = \{ d \mid \#\{e \mid (d,e) \in r^I\} \geq n \}$
- $(\forall r . C)^I = \{ d \mid \forall e: (d,e) \in r^I \Rightarrow e \in C^I \}$
- $(\exists r . C)^I = \{ d \mid \exists e: (d,e) \in r^I \wedge e \in C^I \}$

$I \models A = C$ iff $A^I = C^I$

$I \models C(a)$ iff $a^I \in C^I$

$I \models r(a,b)$ iff $(a^I, b^I) \in r^I$

model

Inferences

make implicit knowledge explicit;
are available as system services of DL systems

Subsumption

Is C a sub-concept of D?

$C \sqsubseteq D$ iff $C^I \subseteq D^I$ for all interpretations I.

Instance

Is e an instance of C w.r.t. the ABox $\mathcal{A}$?

$\mathcal{A} \models C(e)$ iff $e^I \in C^I$ for all models I of $\mathcal{A}$.

Focus of DL research

- decidability/complexity of reasoning
- requires restricted description language
- systems and complexity results available for various combinations of constructors

- application relevant concepts must be definable
- some application domains require very expressive DLs
- efficient algorithms in practice for very expressive DLs?

Reasoning feasible ⟷ versus ⟷ Expressivity sufficient

DL research

historical overview

Phase 1

mostly system development (KL-ONE, LOOM, ...)

early eighties

- expressive description languages, but no disjunction, negation, exist. quant.
- use of so-called structural subsumption algorithms (polynomial)
- no formal investigation of reasoning problems and properties of algorithms

Phase 2

first formal investigations

mid-eighties

- formal, logic-based semantics
- first undecidability and complexity results
- incompleteness of structural subsumption algorithms
  - incompleteness as feature (Loom, Back)
  - restrict expressive power (Classic)

Phase 3

tableau algorithms for DLs and thorough complexity analysis

end eighties to mid-nineties

- Schmidt-Schauß and Smolka describe the first complete (tableau-based) subsumption algorithm for a non-trivial DL;

  $\mathcal{ALC}$: propositionally closed (negation, disjunction, existential restrictions);

  complexity result: subsumption in $\mathcal{ALC}$ is PSPACE-complete.

- Exact worst-case complexity of satisfiability and subsumption for various DLs (DFKI, University of Rome I).

- Development of tableau-based algorithms for a great variety of DLs (DFKI, University of Rome I, RWTH Aachen, ...).

- First DL systems with tableau algorithms: Kris (DFKI), Crack (IRST Trento); first optimization techniques for DL systems with tableau algorithms.

- Schild notices a close connection between DLs and modal logics.

$\mathcal{ALC}$ is a syntactic variant of multi-modal K

[Schild 91]

concept name A		propositional variable A
role name r		modal parameter r
$C \sqcap D$	translation	$t(C) \wedge t(D)$
$C \sqcup D$	$\longrightarrow$	$t(C) \vee t(D)$
$\neg C$	t	$\neg\, t(C)$
$\exists r \,.\, C$		$\langle r \rangle t(C)$
$\forall r \,.\, C$		$[r]t(C)$

interpretation I	Kripke structure $\mathcal{K} = (\mathcal{W}, \mathcal{R})$
set of individuals dom(I)	set of worlds $\mathcal{W}$
interpretation of role names r^I	accessibility relation R_r
interpretation of concept names A^I	worlds in which A is true

Rheinisch-
Westfälische
Technische
Hochschule
Aachen
RWTH

Phase 4

algorithms and systems for very expressive DLs
(e.g., without finite model property)

late nineties

- Decidability results for very expressive DLs by translation into PDL (propositional dynamic logic) (Uni Roma I), strong complexity results; motivated by database applications.
- Intensive optimization of tableau algorithms (Uni Manchester, IRST Trento, Bell Labs, Uni Hamburg): very efficient systems for expressive DLs.
- Design of practical tableau algorithms for very expressive DLs (Uni Manchester, RWTH Aachen); application to ontology reasoning for the semantic web.

Old and new inference problems

- Standard inference problems (like subsumption, instance) in Description Logics are well-investigated:
  - Complexity results for a great variety of DLs.
  - Optimized implementations for expressive DLs.

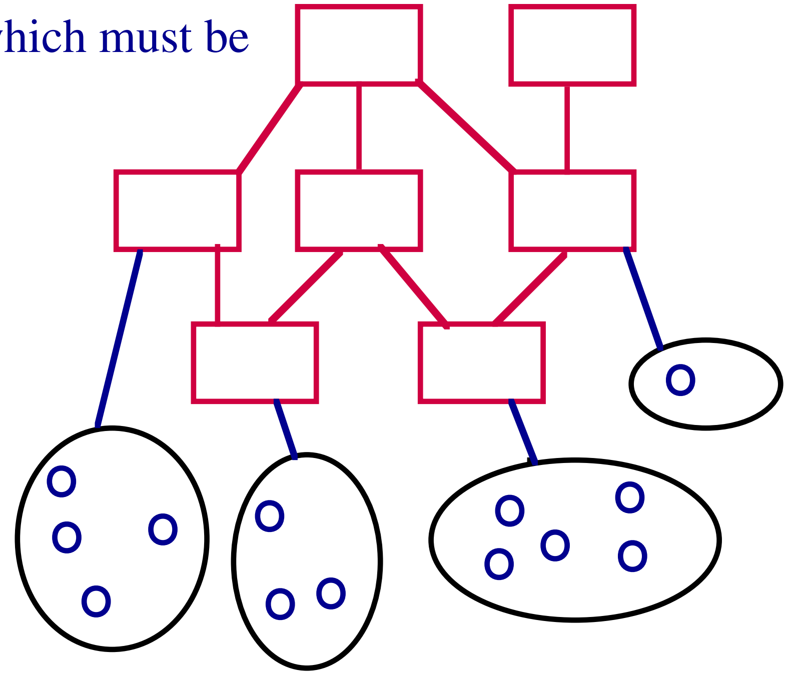
- Building and maintaining large knowledge bases requires support by additional nonstandard inference methods; e.g.:
  - Bottom-up construction of knowledge bases requires least common subsumer, most specific concept, and rewriting.
  - Search for partially specified concepts requires matching and unification.

Rheinisch-
Westfälische
Technische
Hochschule
Aachen
RWTH

Modelling chemical processes and plants

Process Systems Engineering
RWTH Aachen (Prof. Marquardt)

- Computer-aided modelling of chemical processes to analyze, simulate, and optimize the processes.
- Modelling tool ModKit that allows to build process models from standard building blocks.
- Library of standard building blocks, which must be extended continuously.
- Description of building blocks in a frame-based formalism (VeDa).
- Structured representation of the building blocks in a class hierarchy supports searching and browsing.



Applying standard inferences

for the automated structuring of the library.

VeDa / ModKit

Class descriptions

Building block descriptions

trans- late →

DL $\mathcal{ALE}$ / FaCT

Concept definitions (TBox)

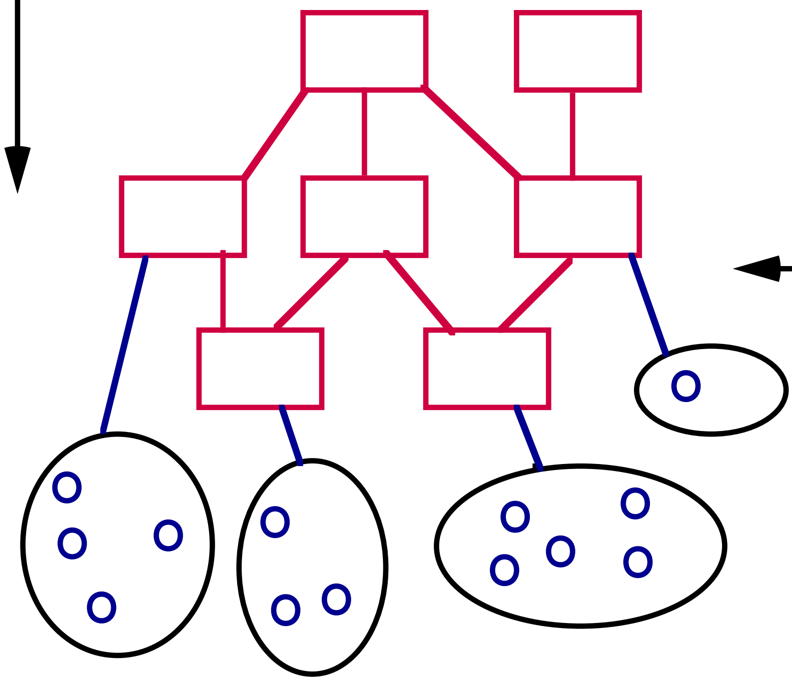
Individuals (ABox)

subsumption / instance algorithm ↓

Computed subsumption hierarchy

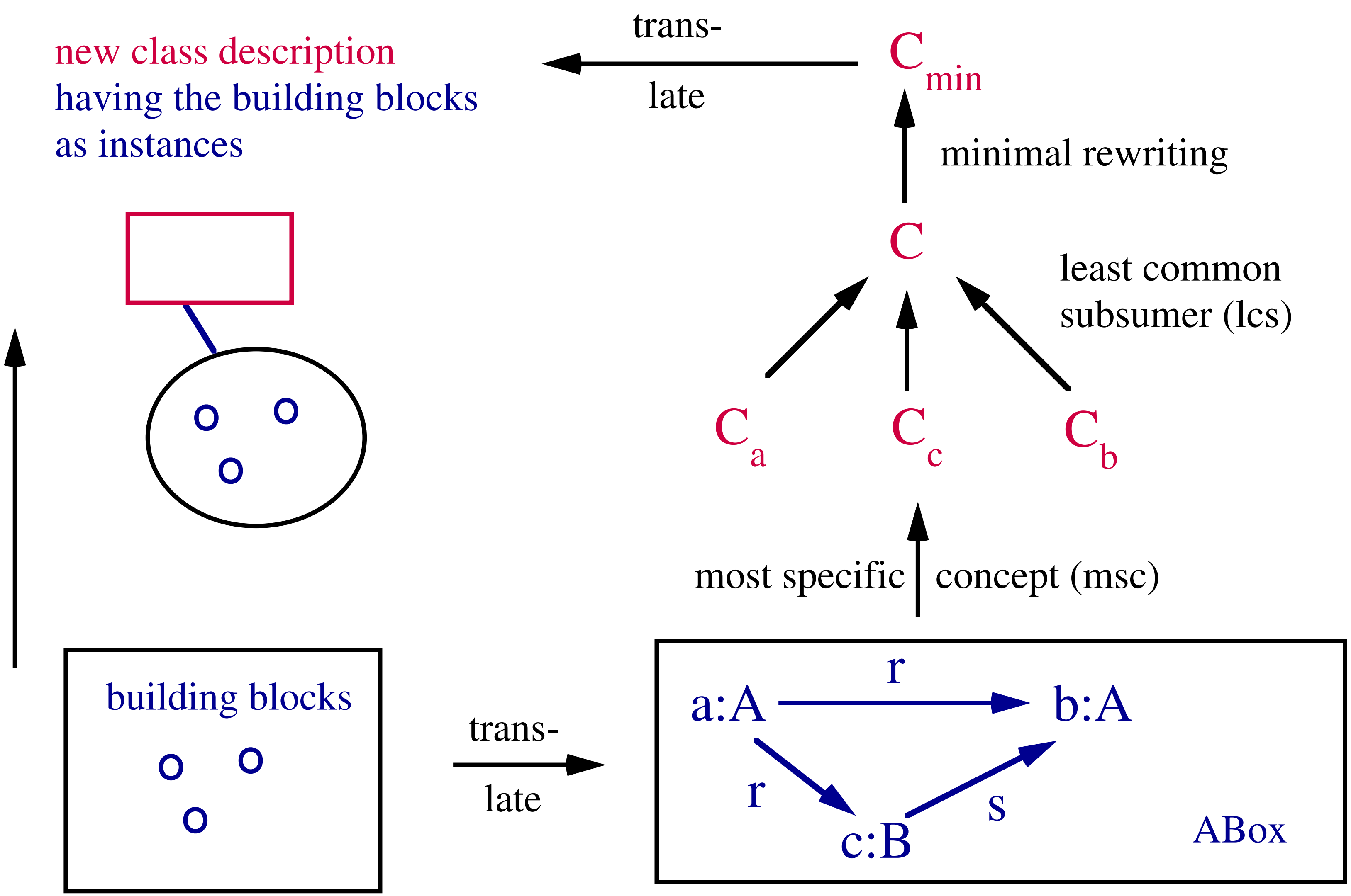
Computed instance relationships

← trans- fer



Applying nonstandard inferences

to support the bottom-up construction of knowledge bases



Research work

necessary to realize this approach

- Least common subsumer: existence and computation
  - First results for Classic (AT&T) [Cohen&Hirsh94, Frazier&Pitt96]: incorrect treatment of inconsistency and attributes.
  - First sound and complete treatment for $\mathcal{ALN}$, $\mathcal{ALE}$, and $\mathcal{ALEN}$ [B.&Küsters98, B.,Küsters&Molitor99, Küsters&Molitor01].
- Most specific concept: existence, computation or approximation
  - Existence and computation for $\mathcal{ALN}$ with cyclic definitions [B.&Küsters98].
  - Non-existence and approximation in $\mathcal{ALE}$ [Küsters&MolitorKI01].
- Rewriting: general framework, complexity and computation
  - Results for $\mathcal{ALE}$, $\mathcal{ALN}$, $\mathcal{ALC}$ and sub-languages [B.,Küsters&Molitor00].
  - Related results in database research: rewriting queries using views.

Least common subsumer

of concept descriptions

- The concept description E is the least common subsumer (lcs) of the concept descriptions $C_1, \ldots, C_n$ iff
  - $C_1 \sqsubseteq E, \ldots, C_n \sqsubseteq E$, and — *subsumes all C_i*
  - $C_1 \sqsubseteq F, \ldots, C_n \sqsubseteq F$ implies $E \sqsubseteq F$ — *is the least such description*

- $\exists$child.Male $\sqcap$ $\exists$child.Doctor is in $\mathcal{ALE}$ the lcs of

$\exists$child.Top $\sqcap$
$\forall$child.(Male $\sqcap$ Doctor)

and

$\exists$child.(Male $\sqcap$ Student) $\sqcap$
$\exists$child.(Doctor $\sqcap$ Female)

- In DLs with disjunction, lcs = disjunction and thus not interesting.

- Questions to be answered:
  Existence of the lcs, its size (binary, n-ary), how to compute it.

Results

for the lcs in sublanguages of $\mathcal{ALEN}$

	$\mathcal{EL}$ C ⊓ D, ∃r.C, Top	$\mathcal{ALE}$ $\mathcal{EL}$ + ∀r.C + atomic negation	$\mathcal{ALEN}$ $\mathcal{ALE}$ + number restrictions
Existence	yes	yes	yes
Size (2)	polynomial	exponential	doubly exponential
Size (n)	exponential	exponential	doubly exponential
Computation	Ptime/Exptime	Exptime	2-Exptime

Approach

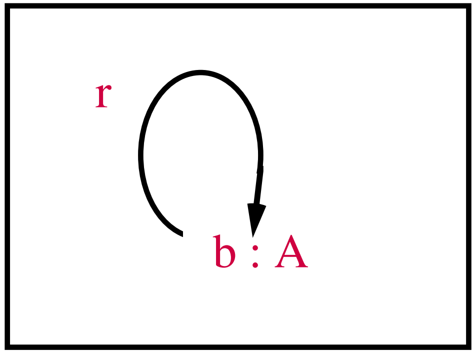
- Translate concept descriptions into description trees.
- Characterize subsumption via homomorphisms.
- Lcs as product of the description trees.

Most specific concept

of an ABox individual

The concept description E is the most specific concept (msc) of the individual b in the ABox $\mathcal{A}$ iff

- $\mathcal{A} \models E(b)$, and — *b is an instance of E*
- $\mathcal{A} \models F(b)$ implies $E \sqsubseteq F$ — *E is the least such description*



b is an instance of

$A, \ \exists r.A, \ \exists r.\exists r.A, \ \exists r.\exists r.\exists r.A, \ldots$

There is no most specific $\mathcal{ALE}$-concept description C such that b is an instance of C.

A similar example shows that the msc need not exist in $\mathcal{ALN}$.

Ways out

two approaches to deal with the non-existence of the msc

- Allow for cyclic TBoxes with an appropriate fixpoint semantics:

  $C = A \sqcap \exists r.C$ with gfp-semantics yields msc in the example.

  For $\mathcal{ALN}$ with cyclic definitions and gfp-semantics, the msc always exists and can effectively be computed [B.&Küsters98].

- Use k-approximation of the msc: most specific concept of role depth at most k having b as an instance:

  $A \sqcap \exists r.(A \sqcap \exists r.A)$ is the 2-approximation of the msc in the example.

  For $\mathcal{ALE}$, the k-approximation always exists and can effectively be computed. More efficient approaches for sub-languages of $\mathcal{ALE}$ [Küsters&MolitorKI01].

Rewriting in DL

our motivation

Non-standard inference procedures for DLs, like

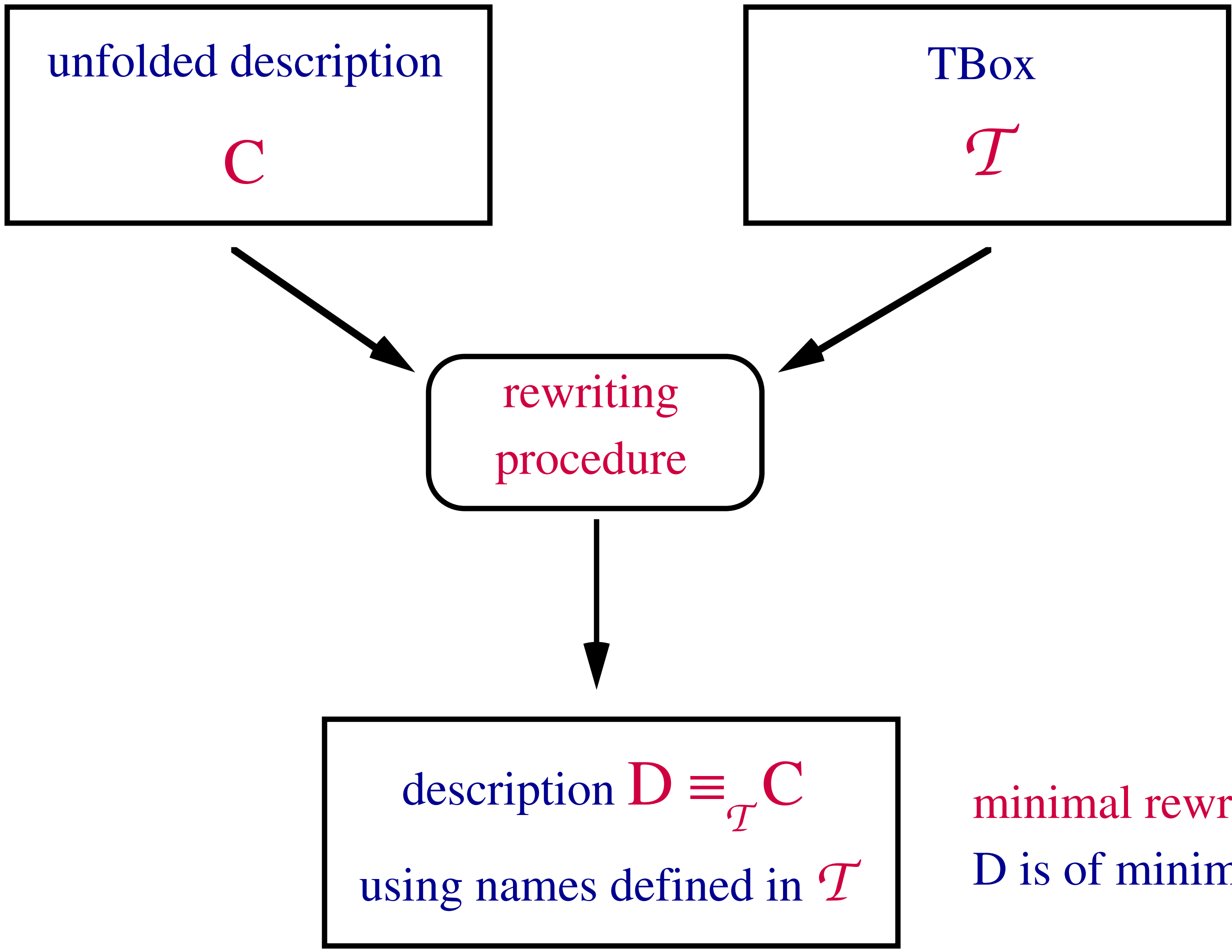
- computing the least common subsumer (lcs),
- matching and unification of concept descriptions,

produce concept descriptions that

- are shown to the user for inspection,
- may be quite large,
- are unfolded, i.e., do not use concept names defined in the TBox.

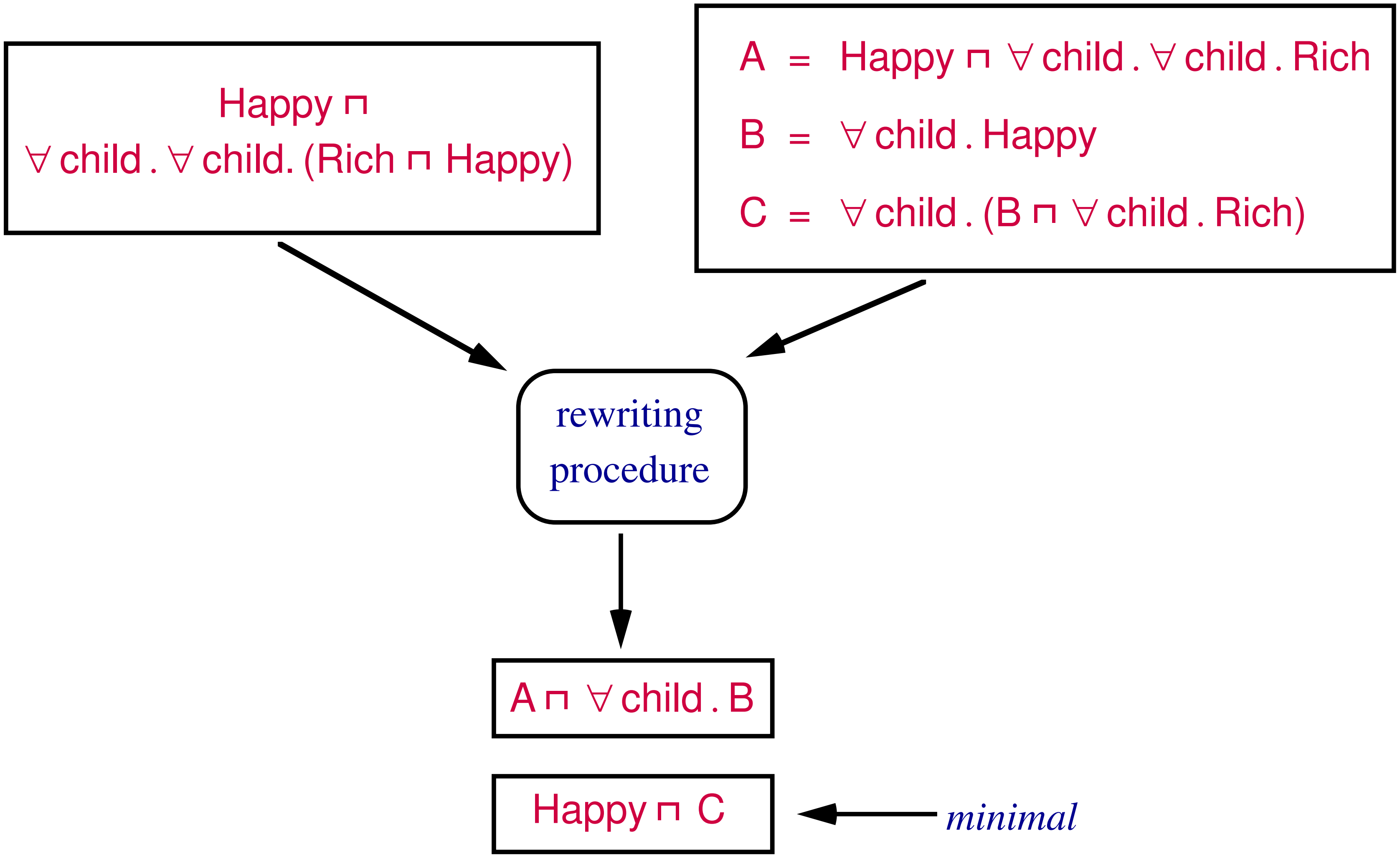
Inference service that automatically increases readability of concept descriptions?

Rewriting concepts using a TBox



minimal rewriting:
D is of minimal size

Example


Happy ⊓
∀ child . ∀ child. (Rich ⊓ Happy)
A = Happy ⊓ ∀ child . ∀ child . Rich
B = ∀ child . Happy
C = ∀ child . (B ⊓ ∀ child . Rich)
rewriting
procedure
A ⊓ ∀ child . B
Happy ⊓ C
minimal

Results

for the minimal rewriting problem in DLs

[B.,Küsters&Molitor00]

- Complexity of the corresponding optimization problem:
  - $\mathcal{ALN}$: NP-hard, in Σ_2^p
  - $\mathcal{ALE}$: NP-hard, in PSPACE
  - $\mathcal{ALC}$: PSPACE-complete
- Nondeterministic algorithms computing all minimal rewritings in $\mathcal{ALE}$ and $\mathcal{ALN}$.
- Heuristic rewriting algorithm for $\mathcal{ALE}$ behaves quite well in practice:

  descriptions of size 800 obtained as lcs in the process engineering application are rewritten into descriptions of size 10.

Matching in DL

our motivation

- Design by modification:
  before defining a new building block from scratch, the process engineers try to locate a structurally similar one in the knowledge base, and then modify the existing block.

- Matching allows to look for concepts having a certain (partially specified) structure:

  Assume we look for concepts concerned with individuals having a son and a daughter sharing some characteristic:

  $$\exists \text{child}.(\text{Male} \sqcap X) \sqcap \exists \text{child}.(\text{Female} \sqcap X)$$ *variable*

  is a concept pattern expressing this.

  The substitution {X —> Tall, Y —> Tall} shows that this pattern matches the description

  `
  $$\exists \text{child}.(\text{Male} \sqcap \text{Tall}) \sqcap \exists \text{child}.(\text{Female} \sqcap \text{Tall})$$

Matching in DL

definition and results

Let C be a concept and D be a pattern.

The substitution σ is a matcher of $C \equiv^? D$ iff $C \equiv \sigma(D)$.

- Matching in $\mathcal{ALN}$ [B., Küsters, Borgida, McGuinness 99]:
  - Existence of matchers can be decided in polynomial time.
  - Solvable problems have a unique least matcher, which can be computed in polynomial time.
  - Matching problems are translated into formal language equations.
- Matching in $\mathcal{ALE}$ [B.&Küsters00]:
  - Existence of matchers is an NP-complete problem.
  - Solvable problems have finitely many minimal matchers, which can be computed in exponential time. Both the number and the size of matchers may be exponential.
  - Approach depends on partial homomorphisms and lcs.

Conclusion

- Compared to the body of results for standard inferences, the research on nonstandard inferences is just at the beginning.
- For lcs, msc, and matching, we have a relatively clear picture for (sub-languages of) $\mathcal{ALE}$ and $\mathcal{ALN}$.
- Other interesting nonstandard inferences:
  - Unification, i.e., patterns on both sides of the equation: considerably harder than matching; Exptime-complete for $\mathcal{FL}_0$ ($C \sqcap D$, $\forall r.C$).
  - Approximation of concepts in one DL by concepts in another DL: we just started to work on this.