A new n-ary existential quantifier in Description Logics
or
How syntactic sugar can speed up reasoning*

Franz Baader
Theoretical Computer Science
TU Dresden
Germany

- A short introduction into Description Logics.
- Motivation for new constructor from chemical process engineering.
- The new constructor and how it can be expressed in DLs.
- Complexity of reasoning with the new constructor.

* Joint work with M. Theißen, RWTH Aachen, and C. Lutz, E. Karabaev, TU Dresden

Description Logics

class of knowledge representation formalisms

Descended from semantic networks and frames via the system KL-ONE [Brachman&Schmolze 85]. Emphasis on well-defined basic inference procedures: subsumption and instance problem.

Phase 1:

- implementation of incomplete systems (Back, Classic, Loom)
- based on structural subsumption algorithms

Phase 2:

- development of tableau-based algorithms and complexity results
- first implementation of tableau-based systems (Kris, Crack)
- first formal investigation of optimization methods

Phase 3:

- tableau-based algorithms for very expressive DLs
- highly optimized tableau-based systems (FaCT, Racer)
- relationship to modal logic and decidable fragments of FOL

Description logic system

structure

description language

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

knowledge base

TBox

defines the terminology of the application domain

ABox

states facts about a specific “world”

reasoning component

- derive implicitly respresented knowledge (e.g., subsumption)
- “practical” algorithms

Description language

Constructors of the DL $\mathcal{ALCQ}$:

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

A man	$Human \sqcap \neg Female \sqcap$
that has a rich or beautiful wife	$\exists married\_to.(Rich \sqcup Beautiful) \sqcap$
and at least 2 sons,	$(\geq 2\ child.\neg Female) \sqcap$
all of whom are happy	$\forall child.(Female \sqcup Happy)$

TBox

definition of concepts

$Happy\_man \equiv Human \sqcap \ldots$

more complex constraints

$\exists married\_to.Doctor \sqsubseteq Doctor$

ABox

properties of individuals

$Happy\_man(Franz)$

$married\_to(Franz, Inge)$

$child(Franz, Luisa)$

Dresden

Formal semantics

An interpretation $\mathcal{I}$ consist of a domain $\Delta^{\mathcal{I}}$ and it associates

- concepts C with sets $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$,
- roles r with binary relations $r^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$, and
- individuals a with elements $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.

The semantics of the constructors is defined through identities:

- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}, \ \ldots$
- $(\exists r.C)^{\mathcal{I}} = \{d \mid \exists e.(d,e) \in r^{\mathcal{I}} \land e \in C^{\mathcal{I}}\}, \ \ldots$
- $(\geq n\, r.C)^{\mathcal{I}} = \{d \mid \sharp\{e \mid (d,e) \in r^{\mathcal{I}} \land e \in C^{\mathcal{I}}\} \geq n\}, \ \ldots$

The interpretation $\mathcal{I}$ is a model of the concept definition/inclusion axiom/assertion

$$\begin{array}{rcl} A \equiv C & \text{iff} & A^{\mathcal{I}} = C^{\mathcal{I}}, \\ C \sqsubseteq D & \text{iff} & C^{\mathcal{I}} \subseteq D^{\mathcal{I}}, \\ C(a) & \text{iff} & a^{\mathcal{I}} \in C^{\mathcal{I}}, \\ r(a,b) & \text{iff} & (a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}. \end{array}$$

Reasoning

makes implicitly represented knowledge explicit, provided as service by the DL system, e.g.:

polynomial reductions

Subsumption: Is C a subconcept of D?

$C \sqsubseteq_{\mathcal{T}} D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of the TBox $\mathcal{T}$.

Satisfiability: Is the concept C non-contradictory?

C is satisfiable w.r.t. $\mathcal{T}$ iff $C^{\mathcal{I}} \neq \emptyset$ for some model $\mathcal{I}$ of $\mathcal{T}$.

Consistency: Is the ABox $\mathcal{A}$ non-contradictory?

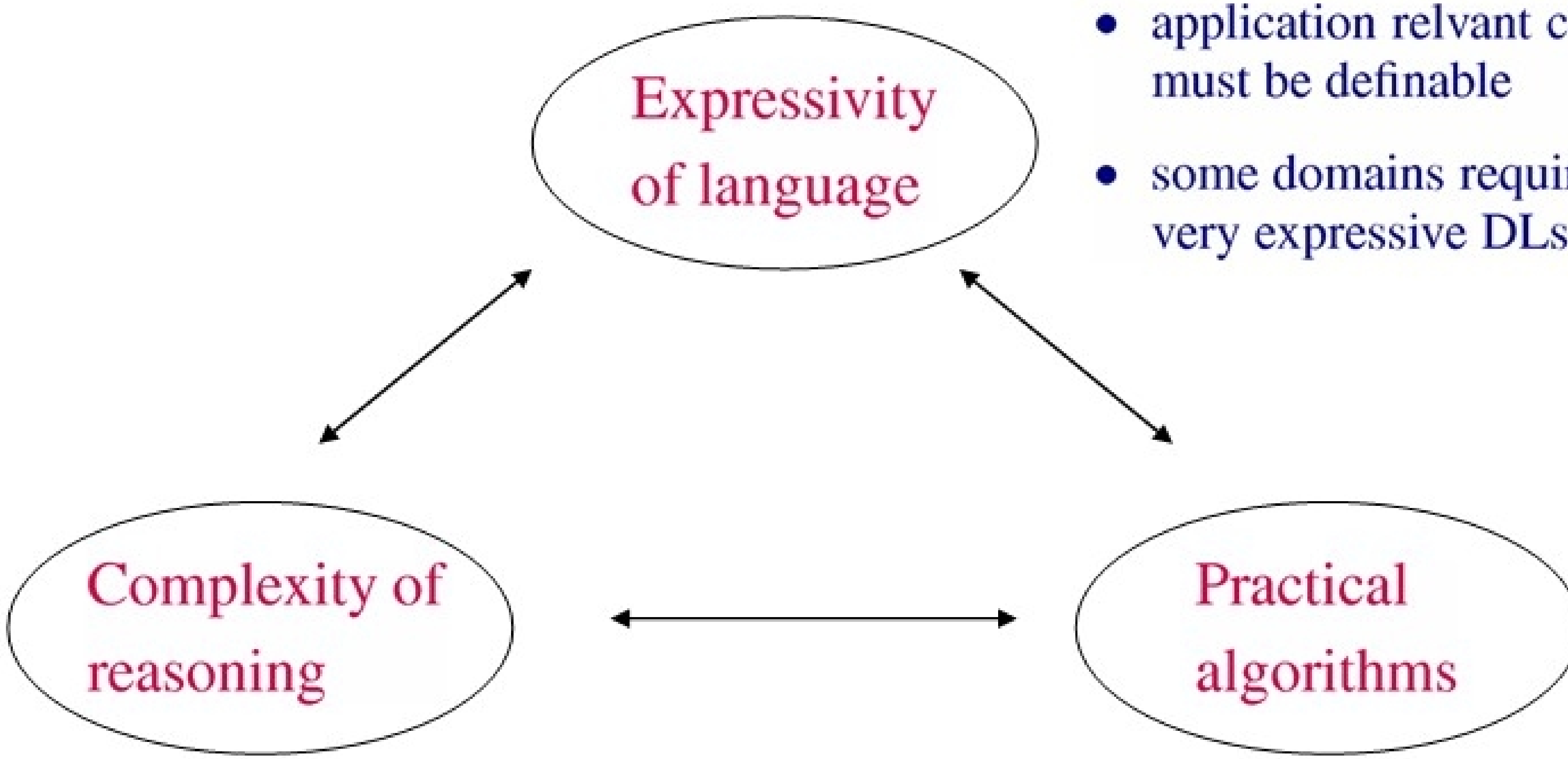
$\mathcal{A}$ is consistent w.r.t. $\mathcal{T}$ iff it has a model that is also a model of $\mathcal{T}$.

Instantiation: Is e an instance of C?

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

in presence of negation

Focus of DL research



Expressivity of language

- application relvant concepts must be definable
- some domains require very expressive DLs

Complexity of reasoning

- reasoning must be decidable/of low complexity
- requires restricted description languages

Practical algorithms

- easy to implement and optimize
- behave well in practice

Complexity of reasoning in $\mathcal{ALCQ}$ [Tobies, 2001]

- All four inference problems have the same worst-case complexity in $\mathcal{ALCQ}$.
- This complexity depends on the presence of complex constraints in the TBox:
  - PSPACE-complete without TBox and also w.r.t. acyclic TBoxes
  - EXPTIME-complete w.r.t. complex constraints and already w.r.t. cyclic TBoxes
- Optimized implementations of tableau-based algorithms for extensions of $\mathcal{ALCQ}$ in the systems FaCT and Racer behave quite well in applications.

Application

in chemical process systems engineering

- mathematical models important for simulating and optimizing chemical process systems
- industrial use of detailed models limited due to high development costs
- reuse of existing models is a promising approach
- which depends on good tools for storing and retrieving building blocks for models:
  - represent models (building blocks) as classes
  - that are automatically inserted in a class hierarchy
  - retrieval by browsing the hierarchy or by formulating query classes

Class descriptions

[Theißen&von Wedel, 2004]

use a simple frame-based formalism

$$\begin{pmatrix} \text{Metaclass} & & \\ \text{slot}_1 : & \text{Class}_{1,1}, \ldots, & \text{Class}_{1,k_1} \\ \vdots & \vdots & \vdots \\ \text{slot}_m : & \text{Class}_{m,1}, \ldots, & \text{Class}_{m,k_m} \end{pmatrix}$$

Example:

a plant that has a reactor with main reaction
and, in addition, a reactor with main and side reaction

$$\begin{pmatrix} \text{Plant} & \\ \text{has-apparatus :} & \text{Reactor-with-Main-Reaction,} \\ & \text{Reactor-with-Main-and-Side-Reaction} \end{pmatrix}$$

Class descriptions

intended semantics

$$\left(\begin{array}{lll} \text{Metaclass} & & \\ \text{slot}_1: & \text{Class}_{1,1}, \ldots, & \text{Class}_{1,k_1} \\ \vdots & \vdots & \vdots \\ \text{slot}_m: & \text{Class}_{m,1}, \ldots, & \text{Class}_{m,k_m} \end{array}\right)$$

Metaclasses:

are equipped with a predefined class hierarchy

Slots and their fillers:

Slot_i has k_i distinct fillers belonging to the respective classes $\text{Class}_{i,1}, \ldots, \text{Class}_{i,k_i}$

Dresden

Class descriptions

translation into DLs

Metaclasses:

and the metaclass hierarchy can be expressed using conjunctions of concept names

Slots and their fillers:

require an n-ary variant of the usual existential restrictions

$$\exists r.(C_1, \ldots, C_k)$$

with the semantics

$$\begin{aligned}\exists r.(C_1, \ldots, C_k)^{\mathcal{I}} \; = \; \{d \mid \exists e_1, \ldots, e_k.\, & (d, e_1) \in r^{\mathcal{I}} \wedge \ldots \wedge (d, e_k) \in r^{\mathcal{I}} \wedge \\ & e_1 \in C_1^{\mathcal{I}} \wedge \ldots \wedge e_k \in C_k^{\mathcal{I}} \wedge \\ & \textstyle\bigwedge_{i \neq j} e_i \neq e_j\}\end{aligned}$$

Can this n-ary existential restriction be expressed within $\mathcal{ALCQ}$?

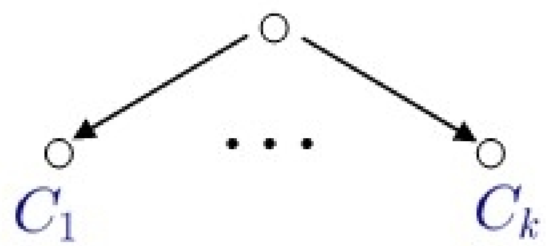
Dresden

First attempt

using unary existential restrictions

$\exists r.(C_1, \ldots, C_k)$ $\exists r.C_1 \sqcap \ldots \sqcap \exists r.C_k$

Only works if the concepts $C_1, \ldots, C_n$ are pairwise disjoint.

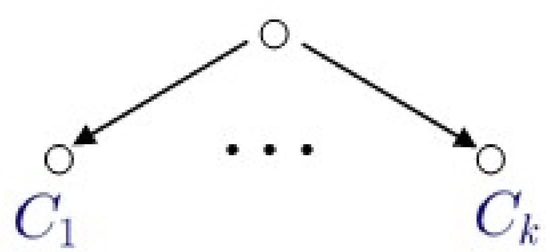


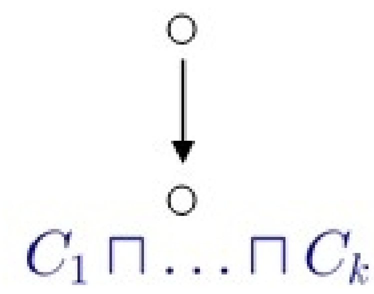
First attempt

using unary existential restrictions

$\exists r.(C_1, \ldots, C_k)$ $\exists r.C_1 \sqcap \ldots \sqcap \exists r.C_k$

Only works if the concepts $C_1, \ldots, C_n$ are pairwise disjoint.





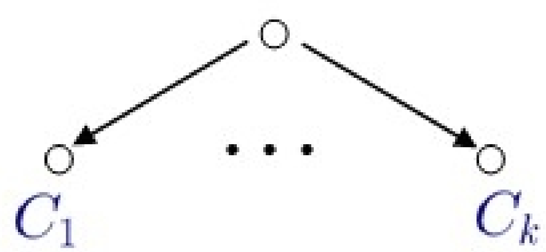
Dresden

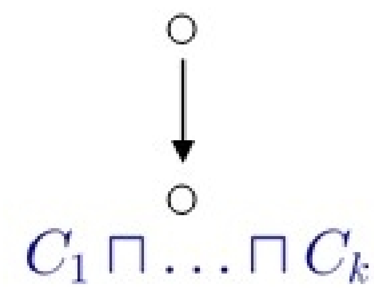
First attempt

using unary existential restrictions

$\exists r.(C_1, \ldots, C_k)$ $\exists r.C_1 \sqcap \ldots \sqcap \exists r.C_k$

Only works if the concepts $C_1, \ldots, C_n$ are pairwise disjoint.





Disjointness cannot be assumed in the process engineering application:

Plant ⊓ ∃has-apparatus.(Reactor-with-Main-Reaction,
Reactor-with-Main-and-Side-Reaction)

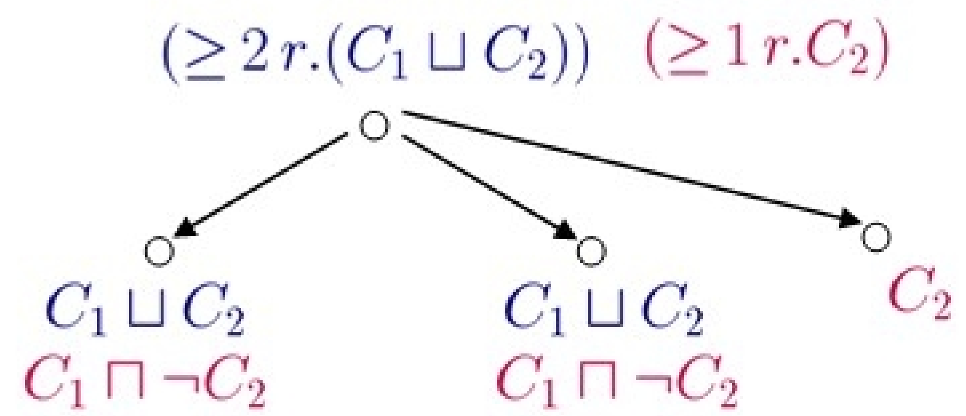
Second attempt

using number restrictions

$$\exists r.(C_1, C_2) \quad \stackrel{?}{\equiv} \quad (\geq 1\, r.C_1) \sqcap (\geq 1\, r.C_2) \sqcap (\geq 2\, r.(C_1 \sqcup C_2))$$

$\sqsubseteq$: obvious

$\sqsupseteq$:



Does this work in general, i.e., also for $n > 2$?

Dresden

Theorem

The new operator can be expressed in $\mathcal{ALCQ}$.

$$\exists r.(C_1,\ldots,C_k) \;\equiv\; \bigsqcap_{\{i_1,\ldots,i_\ell\}\subseteq\{1,\ldots,k\}} (\geq \ell\, r.(C_{i_1}\sqcup\ldots\sqcup C_{i_\ell}))$$

$\sqsubseteq$: obvious

$\sqsupseteq$: is an easy consequence of Hall's Theorem
on the existence of systems of distinct representatives

Hall's theorem

[Hall, 1935]

Let $F = (S_1, \ldots, S_k)$ be a finite family of sets.

Definition

This family has a system of distinct representatives (SDR)

iff

there are k distinct elements $s_1, \ldots, s_k$ such that $s_i \in S_i$ for $i = 1, \ldots, k$.

Theorem

The family $F = (S_1, \ldots, S_k)$ has an SDR iff

$$|S_{i_1} \cup \ldots \cup S_{i_\ell}| \geq \ell$$

for all $\{i_1, \ldots, i_\ell\} \subseteq \{1, \ldots, k\}$.

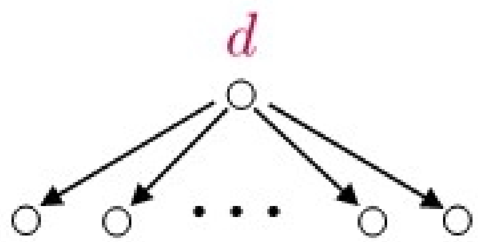
Dresden

Theorem

The new operator can be expressed in $\mathcal{ALCQ}$

$$\exists r.(C_1,\ldots,C_k) \;\equiv \bigsqcap_{\{i_1,\ldots,i_\ell\}\subseteq\{1,\ldots,n\}} (\geq \ell\, r.(C_{i_1} \sqcup \ldots \sqcup C_{i_\ell}))$$

$\sqsupseteq$: is an easy consequence of Hall's Theorem



- Let S_i be the set of r-successors of d belonging to C_i.
- If d belongs to the rhs, then the precondition of Hall's Theorem is statisfied.
- The existence of an SDR implies that d belongs to the lhs.

Dresden

Consequences of this theorem

Computing the hierarchy of class descriptions can be reduced to subsumption in $\mathcal{ALCQ}$, however

- The reduction is exponential.
- Together with PSPACE-completeness of subsumption in $\mathcal{ALCQ}$, this yields an EXPSPACE-upper bound.
- The reduction introduces many disjunctions and number restrictions, which are hard to handle for tableau-based subsumption algorithms.

In practice, this leads to an unacceptable run-time behaviour:

- For some inputs of size about 10, Racer runs for 30 minutes on the translation.

Can we do better?

The DL $\mathcal{EL}_n$

is sufficient to express class descriptions

Concept descriptions of $\mathcal{EL}_n$ are built using

- concept names,
- conjunction $\sqcap$,
- n-ary existential restrictions $\exists r.(C_1, \ldots, C_n)$

with the additional restriction that a conjunction does not contain different restrictions on the same role.

$$\exists r.(A, \exists r.(B, C)) \;\sqcap\; \exists s.(A, A)$$

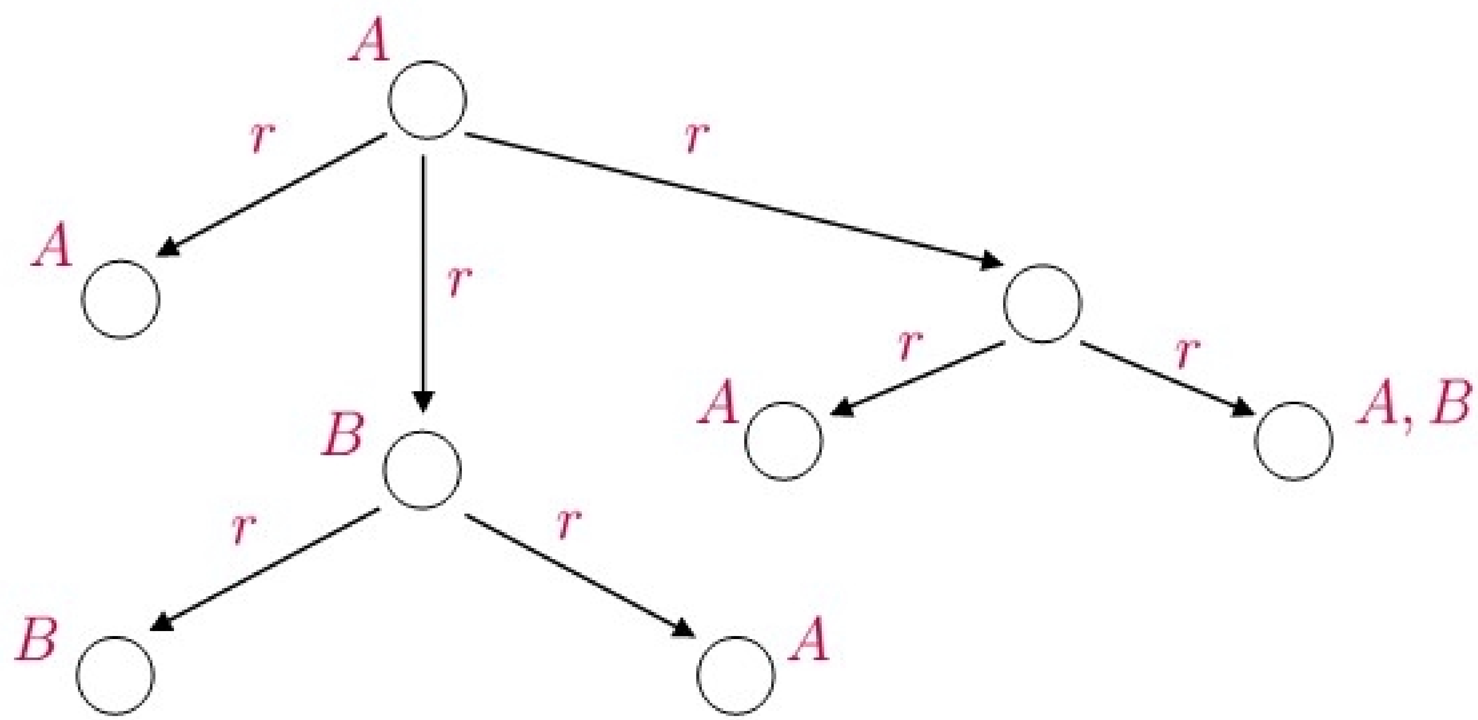
$$\exists r.(A, \exists r.(B, C)) \;\sqcap\; \exists r.(A, A)$$

Dresden

$\mathcal{EL}_n$-description trees

Every $\mathcal{EL}_n$-concept description C can be translated into an $\mathcal{EL}_n$-description tree $\mathcal{T}_C$.

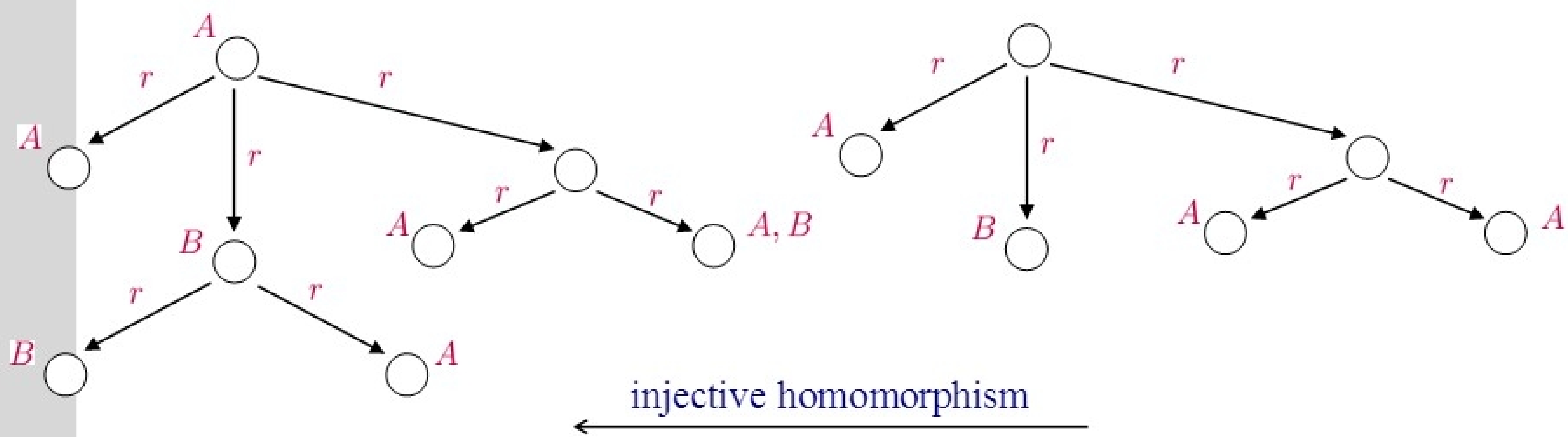
$$A \sqcap \exists r.(A, \quad B \sqcap \exists r.(B, A), \quad \exists r.(A, A \sqcap B))$$



Subsumption in $\mathcal{EL}_n$

corresponds to existence of injective homomorphisms

$$A \sqcap \exists r.(A,\; B \sqcap \exists r.(B, A),\; \exists r.(A, A \sqcap B)) \;\sqsubseteq\; \exists r.(A,\; B,\; \exists r.(A, A))$$



Subsumption in $\mathcal{EL}_n$

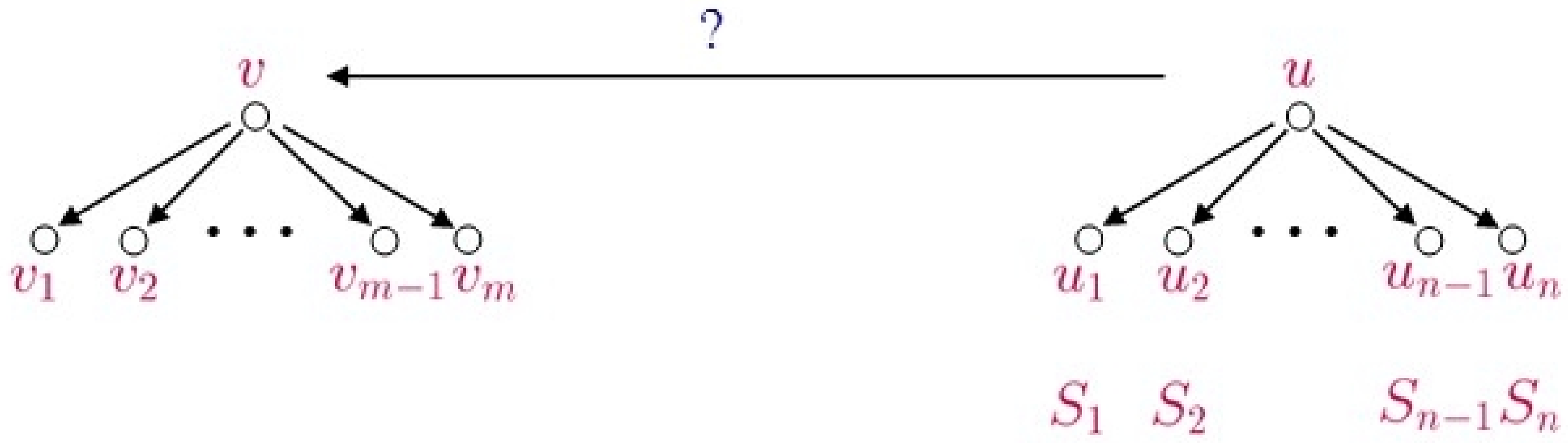
can be decided in polynomial time

Existence of injective homomorphisms between trees can be decided in polynomial time by modifying the well-known bottom-up algorithm that decides the existence of homomorphisms:

$$\mathcal{T}_D \longrightarrow \mathcal{T}_C$$

- For each node u in $\mathcal{T}_D$, compute the set S_u of nodes to which u can be mapped by an (injective) homomorphism, starting with the leafs.

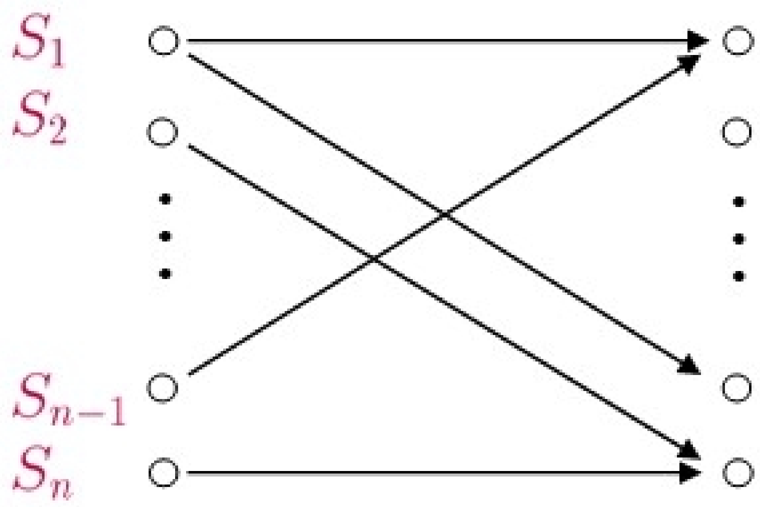
- Injectivity requires us to check the existence of a SDR.



Existence of an SDR

can be decided in polynomial time

Existence of an SDR is the same as the well-known bipartite matching problem:



The bipartite matching problem can be solved in polynomial time by reducing it to a network flow problem.

The DL $\mathcal{EL}_n\mathcal{C}$

is obtained by adding negation

$\mathcal{EL}_n\mathcal{C}$ is as expressive as $\mathcal{ALCQ}$ since it can express

- existential restrictions:

$$\exists r.C \equiv \exists r.(C)$$

- at-least number restrictions:

$$(\geq n\, r.C) \equiv \exists r.(C, \ldots, C)$$

- and thus also their duals $\forall r.C$ and $(\leq n\, r.C)$.

It can express n-ary existential restrictions $\exists r.(C_1, \ldots, C_n)$ in an exponentially more succinct way than $\mathcal{ALCQ}$.

?

Dresden

The DL $\mathcal{EL}_n\mathcal{C}$

why bother?

- Meta-classes are possibly described in an expressive DL such as $\mathcal{ALCQ}$.
- Until now, we have abstracted from their definition by looking only at the induced class hierarchy.
- We may lose some consequences that come from the interaction of the definitions of classes and meta-classes.
- If the meta-class definitions can be expressed in $\mathcal{ALCQ}$ (and thus in $\mathcal{EL}_n\mathcal{C}$), then reasoning in $\mathcal{EL}_n\mathcal{C}$ won't lose any consequences.

The DL $\mathcal{EL}_n\mathcal{C}$

complexity of the subsumption problem

The translation into $\mathcal{ALCQ}$ based on Hall's Theorem yields

- EXPSPACE for subsumption of concept descriptions.
- 2EXPTIME for subsumption w.r.t. general constraints.

By treating the new constructor directly one gets the same complexity as for $\mathcal{ALCQ}$:

- PSPACE for subsumption of concept descriptions (e.g., by an adaptation of the "Witness Algorithm" for $\mathcal{ALC}$).
- EXPTIME for subsumption w.r.t. general constraints (e.g., by an adaptation of the "Elimination of Hintikka Sets" algorithm for PDL).

Conclusion

- The new n-ary existential restriction operator is needed to represent (building blocks of) process models as classes.
- Adding it to the DL $\mathcal{ALCQ}$ does not increase the expressive power.
- Nevertheless, adding it explicitly decreases the complexity of reasoning.

Further work:

- Implementation of polynomial-time algorithm for $\mathcal{EL}_n$ is under way.
- Show that there is no polynomial translation of the new operator into $\mathcal{ALCQ}$.
- Develop and implement a "practical" tableau-based algorithm for $\mathcal{EL}_n\mathcal{C}$.