Building Medical Ontologies Using Description Logics:

What does it buy us?

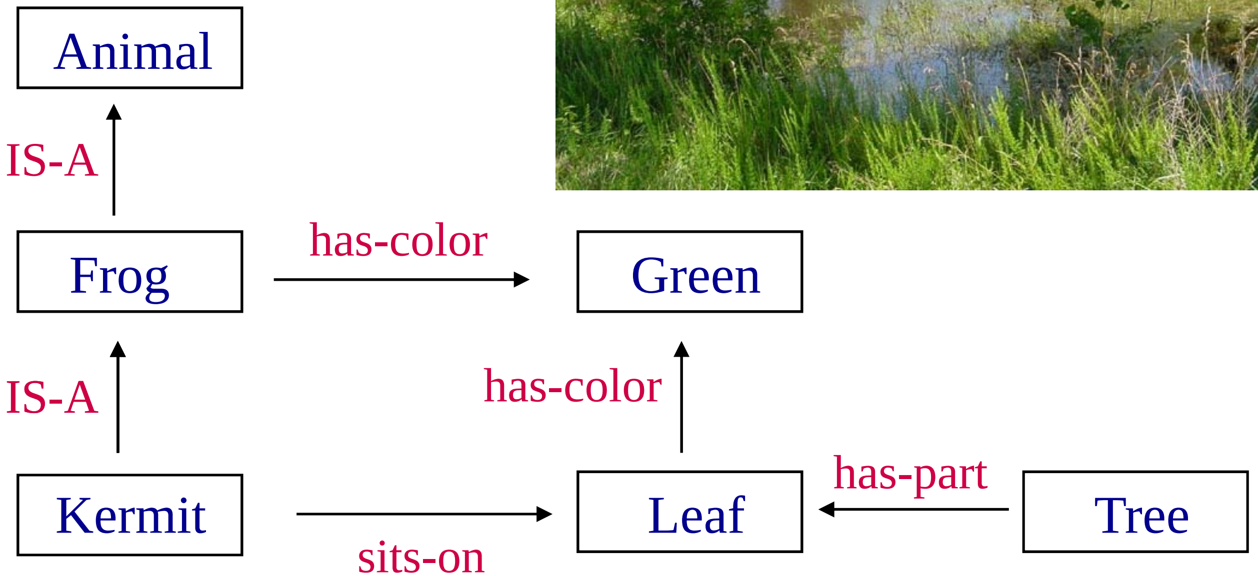
Building Medical Ontologies Using Description Logics:
What does it buy us?

Franz Baader
Theoretical Computer Science
TU Dresden
Germany

Semantic networks

[Quillian, 1967]

one of the origins of DLs



Dresden

Problems

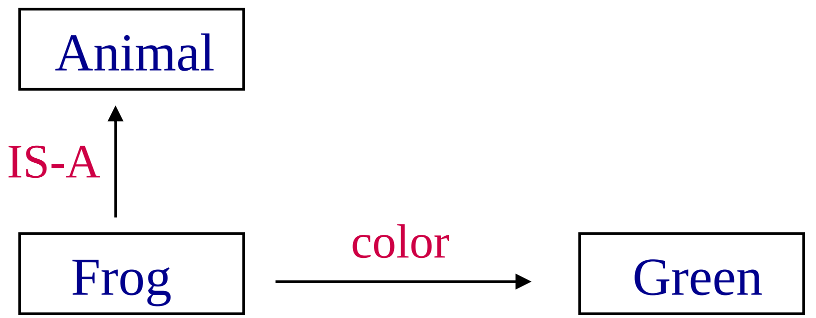
with Semantic Networks

- no formal semantics
- meaning is defined by the processes operating on the network
- identical networks may lead to different results, depending on which system is employed

- attempts to formalize the meaning of semantic networks use first-order predicate logic (e.g., [Schubert et al., 1979])
- development of DLs follows the same idea, but tries to find useful decidable fragments

Ambiguities

in Semantic Networks

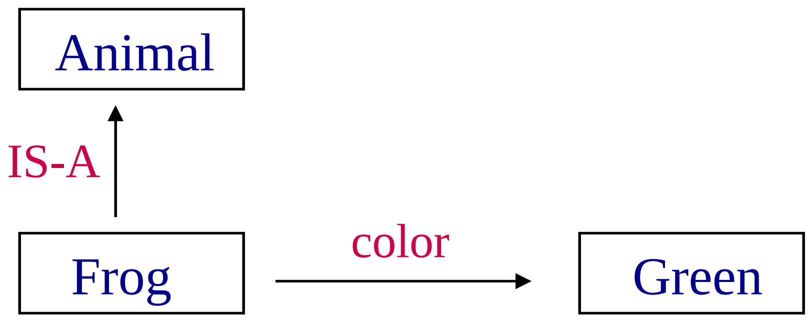


- value restriction: green is the only possible color;

Dresden

Ambiguities

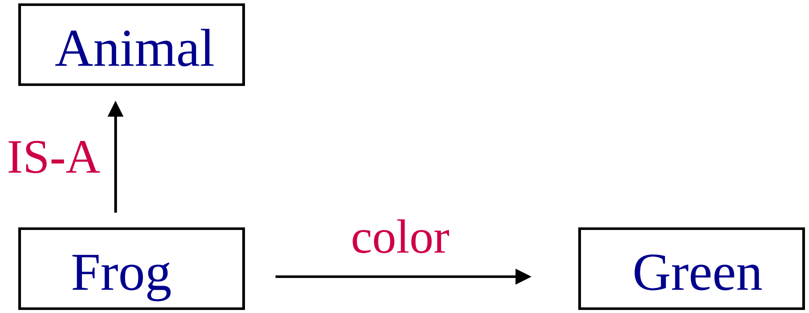
in Semantic Networks



- existential restriction: green is one of its colors;

Ambiguities

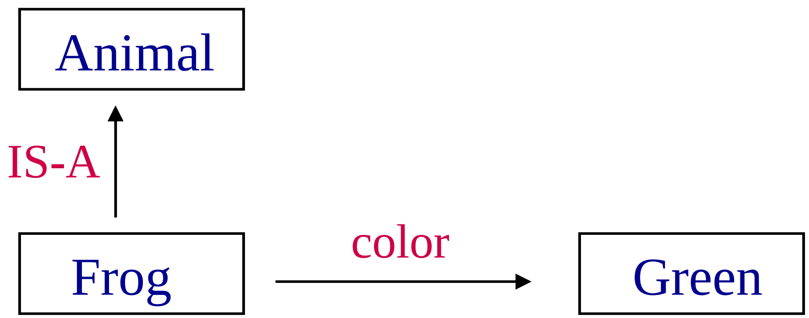
in Semantic Networks



- default reading: assume that green is its color, unless you know to the contrary

Ambiguities

resolved in DLs and SNOMED CT



- value restriction: green is the only possible color
- existential restriction: green is one of its colors;
- default reading: assume that green is its color, unless you know to the contrary

Value restrictions vs existential restrictions in DLs

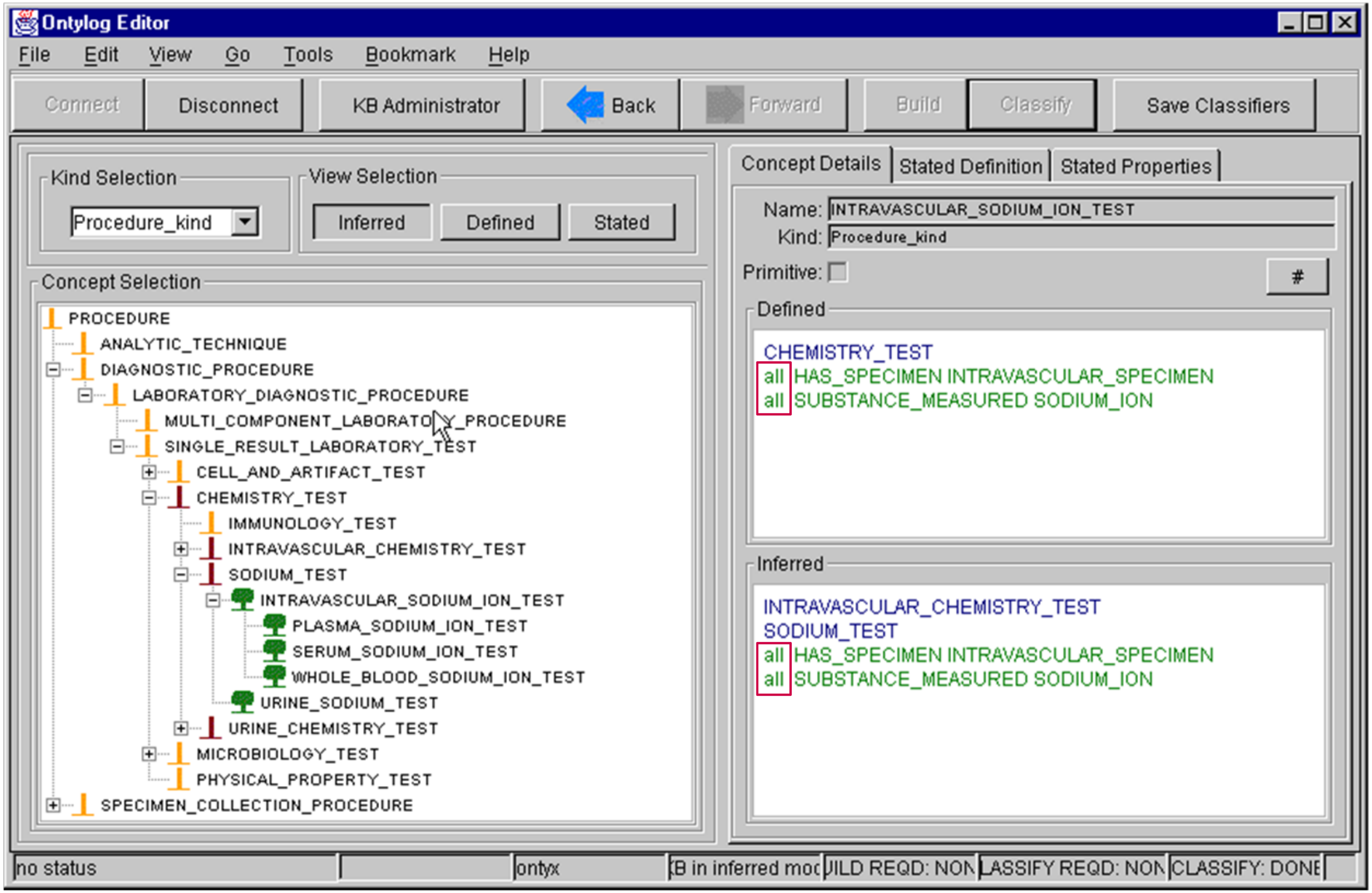
- the seminal system KL-ONE and other early DL systems used value restrictions as reading for property edges.
- Schmidt-Schauß and Smolka [1988] introduce negation and thus implicitly existential restrictions.
- Value-restrictions and conjunction until recently considered to be indispensable for DLs: $\mathcal{FL}_0$ minimal such DL.
- DLs with existential restrictions, but without value restrictions:
  - have been investigated in the DL community only since about 2000;
  - have better algorithmic properties than the corresponding languages with value restrictions;
  - are useful for representing bio-medical ontologies.

Value restrictions vs existential restrictions in SNOMED

- SNOMED RT and CT use existential restrictions as reading for property edges.

Dresden

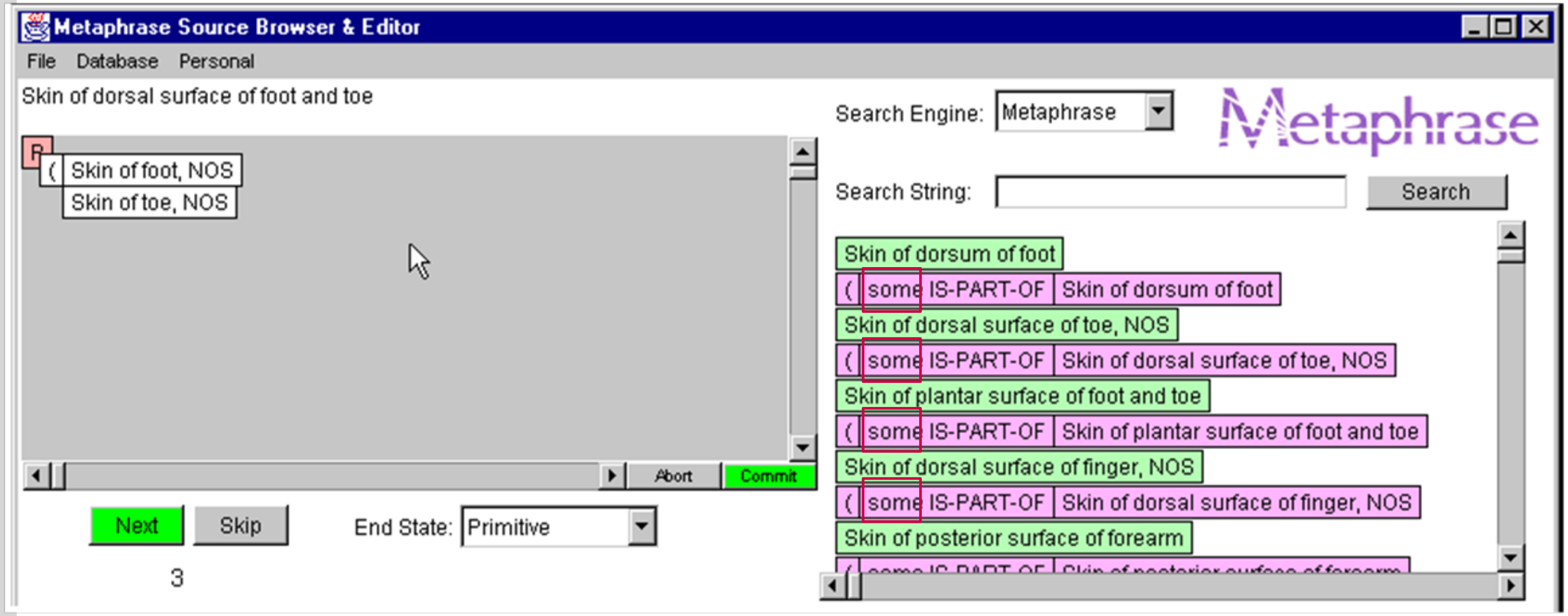
- SNOMED RT and CT use existential restrictions as reading for property edges, though this decision was not that clear in the beginning ...



Screenshot of Ontyx editor taken from DL2008 invited talk of Kent Spackman

Dresden

- SNOMED RT and CT use existential restrictions as reading for property edges, though this decision was not that clear in the beginning ...



Screenshot of Metaphrase editor taken from DL2008 invited talk of Kent Spackman

Value restrictions vs existential restrictions in SNOMED

- SNOMED RT and CT use existential restrictions as reading for property edges.

- $\mathcal{EL}$: DL that has only existential restrictions, conjunction, and the top-concept as concept constructors.

- Until recently, the DL community was largely oblivious of the fact that SNOMED uses $\mathcal{EL}$:
  - no publication about algorithm used for classification
  - no publication of experimental results
  - no access to the system used for classifying SNOMED

Complexity

of reasoning

A commonly held belief in the 1980ies:

> reasoning in KR systems should be tractable,
> i.e., of polynomial time complexity

- KL-ONE and its early successor systems (BACK, MESON, K-Rep, ...) employed polynomial-time algorithms
- reasoning in KL-ONE is undecidable [Schmidt-Schauß, 1989]
- even in very inexpressive DLs, reasoning may be intractable [Brachman&Levesque, 1987]
- reasoning w.r.t. a TBox is intractable even in the minimal DL $\mathcal{FL}_0$ (value-restriction, conjunction) [Nebel, 1990]
- the early DL systems employed sound, but incomplete algorithms

Ways out

of this dilemma

expressive DL
sound, but incomplete
tractable algorithms

expressive DL
sound and complete
intractable algorithms

approach followed by main-stream DL research in the last 15 years

inexpressive DL
sound and complete
tractable algorithms

SNOMED®

recent research on leight-weight DLs:
$\mathcal{EL}$, DL-Lite, Horn-$\mathcal{SHIQ}$

Ways out

of this dilemma

expressive DL
sound, but incomplete
tractable algorithms

expressive DL
sound and complete
intractable algorithms

inexpressive DL
sound and complete
tractable algorithms

The complexity monster

Description Logics

research of the last 20 years

Phase 1:

- implementation of systems (Back, K-Rep, Loom, Meson, . . .)
- based on incomplete 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:

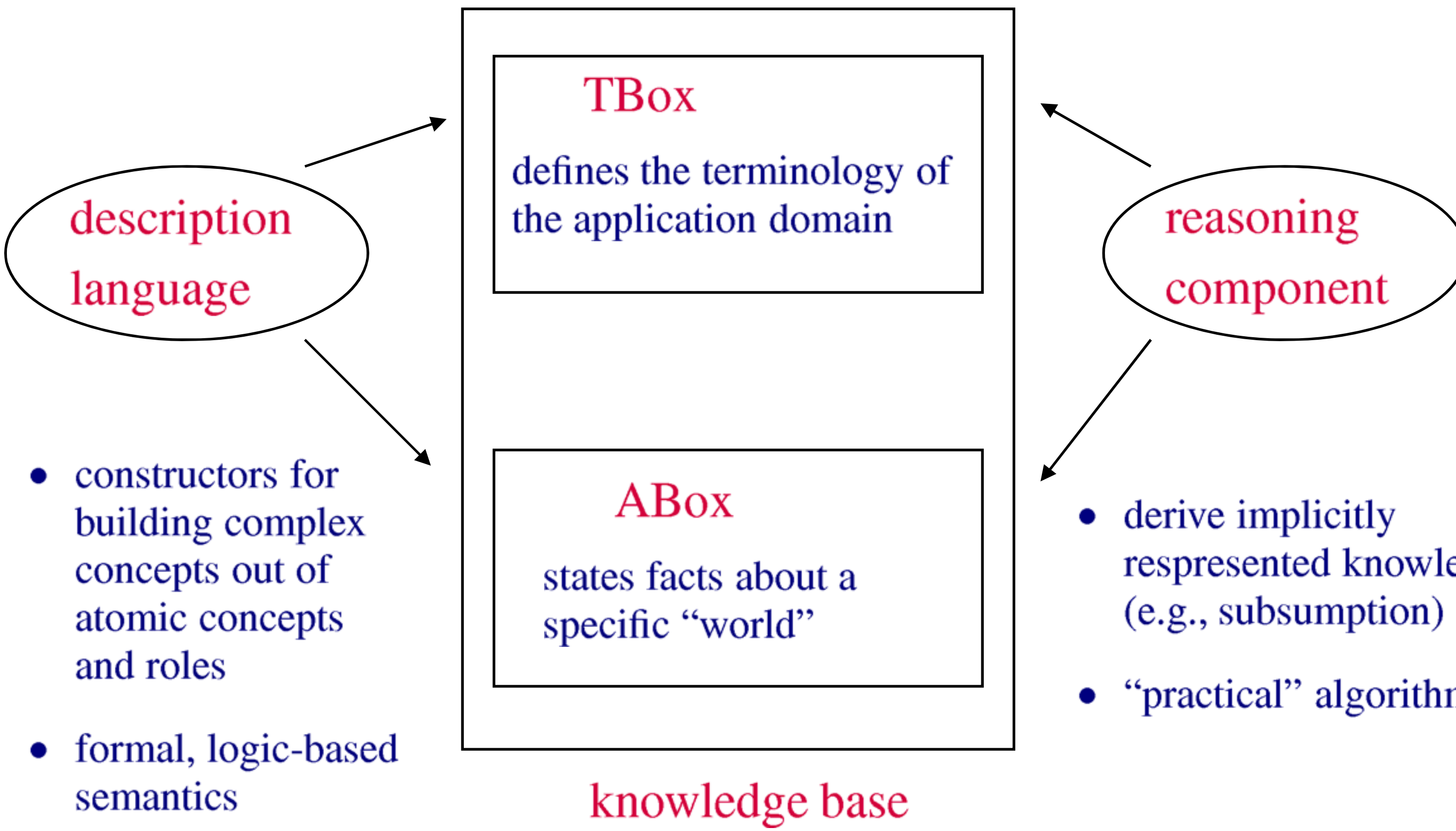
- tableau-based algorithms for very expressive DLs
- highly optimized tableau-based systems (FaCT, Racer)
- relationship to modal logic and decidable fragments of FOL

Phase 4:

- Web Ontology Language (OWL-DL) based on very expressive DL
- industrial-strength reasoners and ontology editors for OWL-DL
- investigation of leight-weight DLs with tractable reasoning problems

Description logic system

structure



Dresden

Description logic system

structure

description language

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

TBox

defines the terminology of the application domain

only TBox

knowledge base

reasoning component

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

Dresden

Description language

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

$\top, C \sqcap D, \exists r.C$

A man	$Human \sqcap Male \sqcap$
that has a rich and beautiful wife,	$\exists married\_to.(Rich \sqcap Beautiful) \sqcap$
a son and a daughter,	$\exists has\_child.Male \sqcap \exists has\_child.Female \sqcap$
and a job	$\exists has\_job.\top$

TBox

full definitions

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

primitive definitions

$Happy\_man \sqsubseteq Human \sqcap \ldots$

GCIs

general concept inclusions

additional constraints

$\exists has\_child.Human \sqsubseteq Human$

SNOMED currently no GCIs in DL version

Formal semantics

An interpretation $\mathcal{I}$ has a domain $\Delta^{\mathcal{I}}$ and associates

- concepts C with sets $C^{\mathcal{I}}$, and
- roles r with binary relations $r^{\mathcal{I}}$.

The semantics of the constructors is defined through identities:

- $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$,
- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$,
- $(\exists r.C)^{\mathcal{I}} = \{d \mid \exists e.(d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$.

The interpretation $\mathcal{I}$ is a model of

- the full definition $A \equiv C$ iff $A^{\mathcal{I}} = C^{\mathcal{I}}$,
- the primitive definition $A \sqsubseteq C$ iff $A^{\mathcal{I}} \subseteq C^{\mathcal{I}}$,
- the general concept inclusion (GCI) $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

Subsumption

is concept C a subconcept of concept D?

$$\boxed{C \sqsubseteq D \quad \text{iff} \quad C^{\mathcal{I}} \subseteq D^{\mathcal{I}}}$$

for all interpretations $\mathcal{I}$

$$\boxed{C \sqsubseteq_{\mathcal{T}} D \quad \text{iff} \quad C^{\mathcal{I}} \subseteq D^{\mathcal{I}}}$$

for all models $\mathcal{I}$ of $\mathcal{T}$

$\mathcal{T}$ can be

- an acyclic TBox: finite set of unambiguous and acyclic concept definitions
- a cyclic TBox: finite set of unambiguous concept definitions
- a general TBox: finite set of GCIs

What does it buy us?

formal logic-based semantics

The meaning of concepts is unambiguously determined by the semantics of the constructors:

- $(\exists r.C)^{\mathcal{I}} = \{d \mid \exists e.(d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$
- $(\forall r.C)^{\mathcal{I}} = \{d \mid \forall e.(d,e) \in r^{\mathcal{I}} \rightarrow e \in C^{\mathcal{I}}\}$

$\exists has\_child.Male \sqcap \exists has\_child.Female$

are usually interpreted by different sets of individuals

$\forall has\_child.Male \sqcap \forall has\_child.Female$

and behave differently w.r.t. subsumption:

$$\exists has\_child.Male \sqcap \exists has\_child.Female \not\sqsubseteq \exists has\_child.(Male \sqcap Female)$$

$$\forall has\_child.Male \sqcap \forall has\_child.Female \sqsubseteq \forall has\_child.(Male \sqcap Female)$$

What does it buy us?

subsumption reasoning

- A new concept can be introduced by defining the necessary conditions (primitive definition) or the necessary and sufficient conditions (full definition) for an individual to belong to this concept.

- Its place in the hierarchy of existing concepts is found automatically by the subsumption reasoner.

- Subsumption reasoning can also be used to test whether the definition of a new concept captures the underlying intuitition:
  - unintuitive subsumption relationships indicate that there is a modeling error

What does it buy us?

other reasoning

Understanding the reasons for unintuitive or unintended consequences can be difficult:

- W.r.t. the DL version of SNOMED, the concept *Amputation-of-finger* is classified as a subconcept of *Amputation-of-hand*.
- Finding the definitions that are responsible for this among the ∼350 000 definitions in SNOMED is not easy.

Pinpointing

my talk tomorrow

- identifies the source of a consequence by computing a minimal subset of the TBox from which this consequence already follows
- in the amputation example, this set consists of 6 definitions

What does it buy us?

other reasoning

- Assume that a user is interested in using only a subset of the concepts and roles from SNOMED to define new (post-coordinated) concepts.
- What part of the DL version of SNOMED does this user need to obtain the same subsumption consequences as with all of SNOMED?

Modularization

in Description Logics

- formal definition of module based on the notion of conservative extensions from logic
- extraction of minimal modules in polynomial time

⇒ *recent work by*
Cuenca Grau, Lutz, Sattler, Suntisrivaraporn, Wolter
and others

CEL

classifier for $\mathcal{EL}$

Experimental system developed at TU Dresden, which supports

- classification, i.e., computation of the subsumption hierarchy [Baader, Lutz, Suntisrivaraporn; 2005, 2006];
- incremental classification, i.e., recomputation of the subsumption hierarchy after the TBox has been extended [Suntisrivaraporn; 2008];
- pinpointing [Baader, Suntisrivaraporn; 2008] and modularization [Suntisrivaraporn; 2008].

http://lat.inf.tu-dresden.de/systems/cel/

What does it buy us?

formal investigation of
algorithmic properties

- Complexity of a problem: how hard is it in principle to solve the reasoning problems (like subsumption) in a given Description Logic?
- Complexity of an algorithm: is the employed algorithm optimal w.r.t. the complexity of the problem?
- Complexity versus expressivity: which concept constructors are "expensive" in the sense that adding/using them increases the complexity?

DL community has obtained such results for a great variety of Description Logics of different expressive power

Complexity of subsumption

$\mathcal{FL}_0$ versus $\mathcal{EL}$

	$\mathcal{FL}_0$	$\mathcal{EL}$
no TBox	polynomial [Brachman, Levesque, 84]	polynomial [Baader, Küsters, Molitor, 99]
acyclic TBox	coNP-complete [Nebel, 90]	polynomial [Baader, 03]
cyclic TBox	PSpace-complete [Baader, 90] [Kazakov, Nivelle, 03]	polynomial [Baader, 03]
general TBox	ExpTime-complete [Baader, Brandt, Lutz, 05]	polynomial [Brandt, 04]

Extension

to the more expressive DL $\mathcal{EL}^{++}$ [Baader, Brandt, Lutz; 05, 08]

Subsumption in the presence of GCIs remains polynomial if we add

- the bottom concept $\bot$, which stands for the empty set;

  $Clinical\_findig \sqcap Body\_part \sqsubseteq \bot$
- nominals, i.e., singleton concepts; $\{Denmark\}$
- restricted role-value-maps (RVMs), which can express transitivity and right-identities;
- domain and range restrictions for roles;

restrictions regarding their combined use

$$\begin{array}{rcl} \mathrm{domain}(has\_location) & \sqsubseteq & Clinical\_findig \\ \mathrm{range}(has\_location) & \sqsubseteq & Body\_part \end{array}$$

Dresden

Extension

to the more expressive DL $\mathcal{EL}^{++}$ [Baader, Brandt, Lutz; 05, 08]

Subsumption in the presence of GCIs remains polynomial if we add

- the bottom concept $\bot$, which stands for the empty set;
- nominals, i.e., singleton concepts;
- restricted role-value-maps (RVMs), which can express transitivity and right-identities;
- domain and range restrictions for roles;
- restricted concrete domains, which enable using datatypes such as numbers, strings, . . . in the definition of concepts.

$$>_{180}(\mathit{has\_diastolic\_bp\_mmHg}) \sqsubseteq \mathit{Hypertension}$$

Adding any of the other constructors available in OWL makes the subsumption problem intractable in the presence of GCIs.

Restricted RVMs

can express important properties of roles

$\epsilon \sqsubseteq part\_of$	reflexivity
$part\_of \circ part\_of \sqsubseteq part\_of$	transitivity
$proper\_part\_of \sqsubseteq part\_of$	role hierarchy
$has\_exact\_location \sqsubseteq has\_location$	role hierarchy
$has\_location \circ part\_of \sqsubseteq has\_location$	right identity

$$\begin{tikzcd} Hand \arrow[r, "part\_of"] & Arm \\ Hand\_injury \arrow[u, "has\_location"] \arrow[ur, dashed, "has\_location"'] & \end{tikzcd}$$

$$\begin{tikzcd} Hand \arrow[r, "part\_of"] & Arm \\ Hand\_amputation \arrow[u, "has\_exact\_location"] & \end{tikzcd}$$

Restricted RVMs

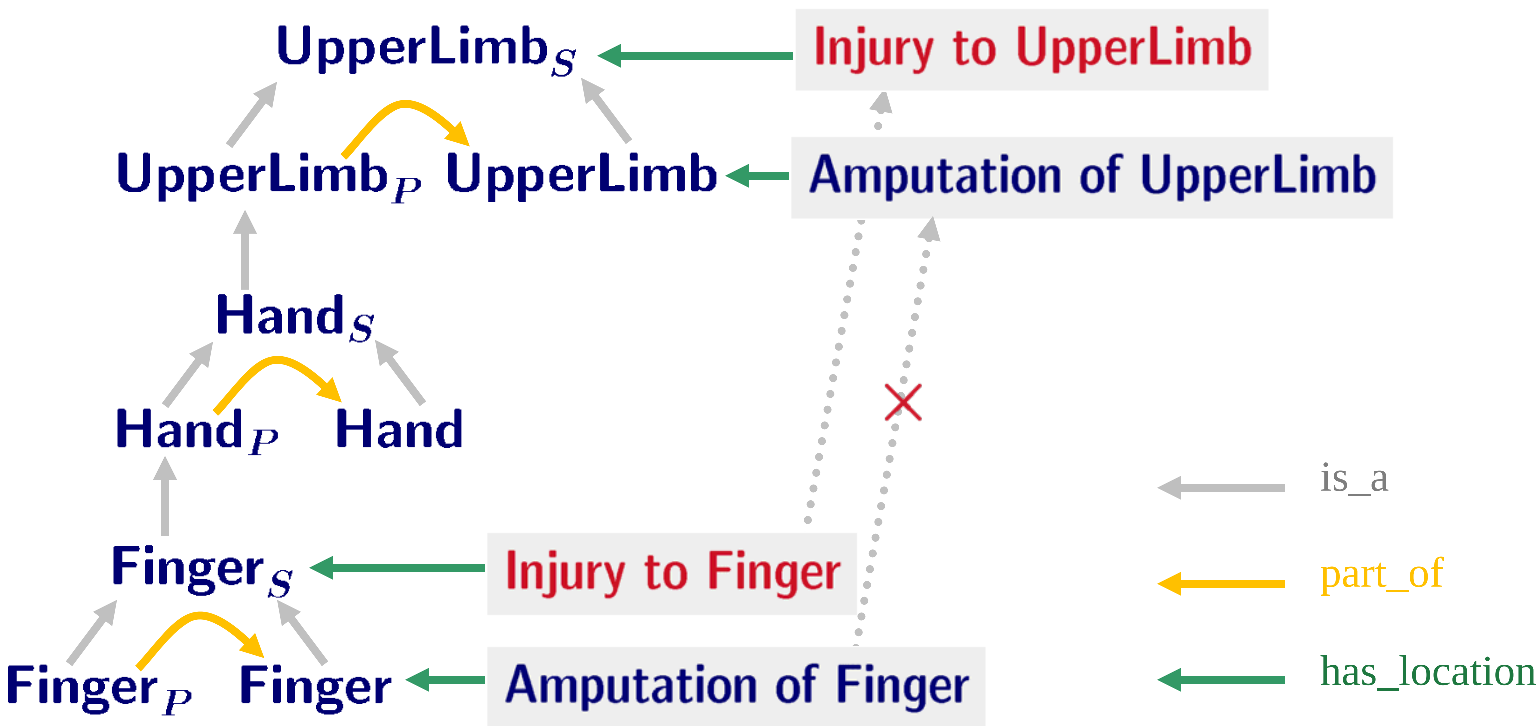
can express important properties of roles

$\epsilon \sqsubseteq part\_of$	reflexivity
$part\_of \circ part\_of \sqsubseteq part\_of$	transitivity
$proper\_part\_of \sqsubseteq part\_of$	role hierarchy
$has\_exact\_location \sqsubseteq has\_location$	role hierarchy
$has\_location \circ part\_of \sqsubseteq has\_location$	right identity

Can be used to replace the SEP-triplet encoding of SNOMED CT.

SEP-triplets

[Schulz, Romacker, Hahn; 1998]

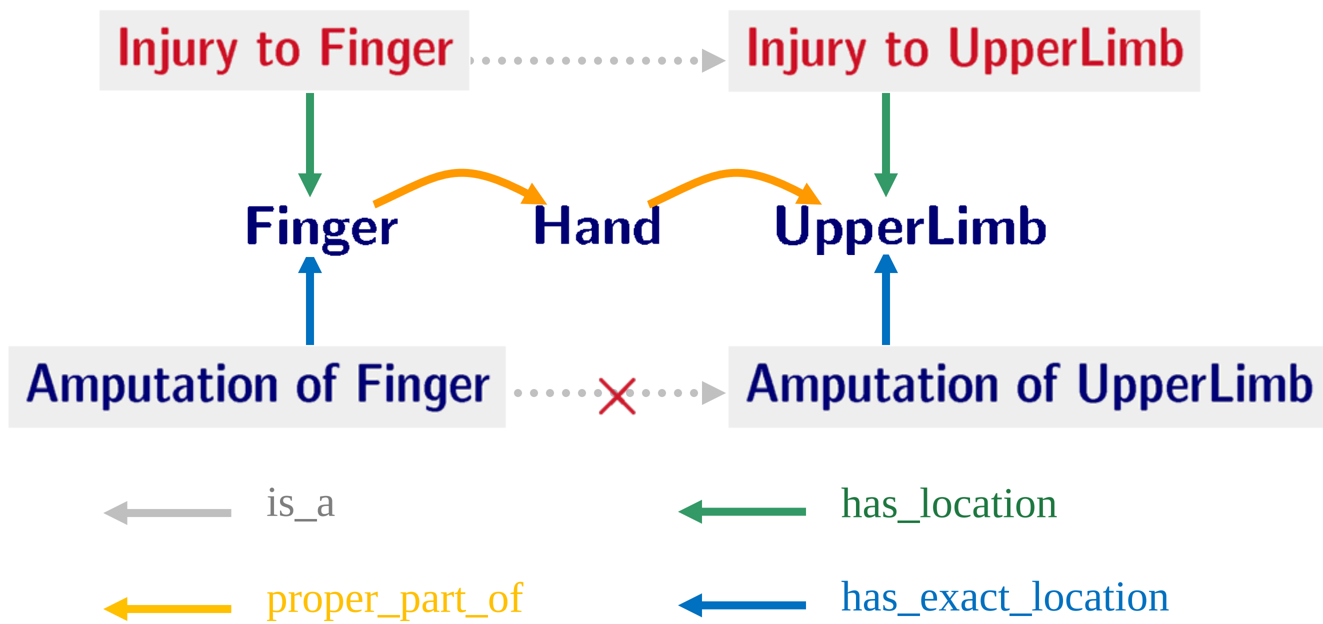


- \+ uses transitivity of is_a instead of making part_of transitive
- \+ can enable and block right-identity reasoning
- – increases the number of concepts considerably
- – indirect modelling makes it error-prone

Dresden

Re-engineered version without SEP-triplets

[Suntisrivaraporn, Baader, Schulz, Spackman; 2007]



cleaner modelling

fewer concepts

easier to use and less error prone

faster reasoning

Conclusion

Using DLs to define medical ontologies:

- formally well-understood semantics
- sound and complete reasoning support . . . not just for classification
- well-understood trade-off between expressivity and complexity of reasoning

Using $\mathcal{EL}$ to define medical ontologies:

- less expressive and thus easier to comprehend and use than OWL
- reasoning is tractable
- and stays so even if interesting means of expressivity (GCIs, restricted RVMs, domain and range restrictions, . . .) are added


Questions?