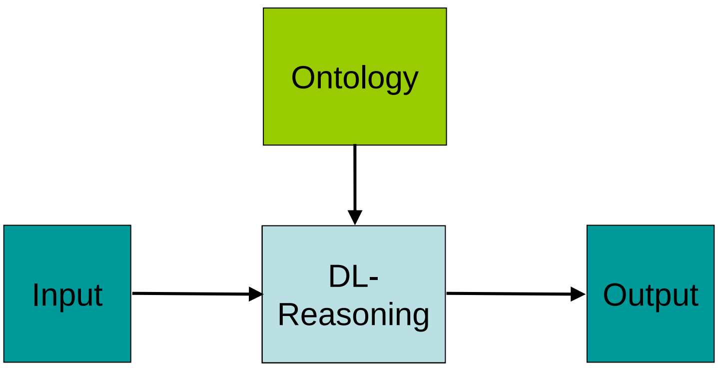
Approximation for the Semantic Web

The KnowledgeWeb point of view

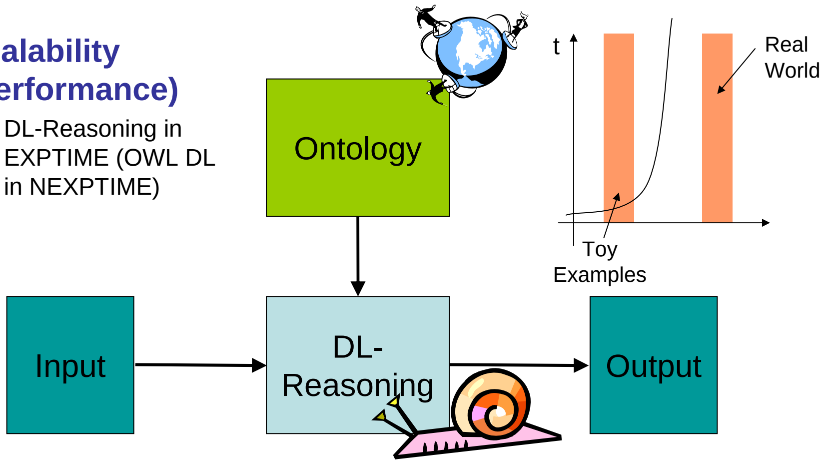
Holger Wache

Semantic Web Systems in General



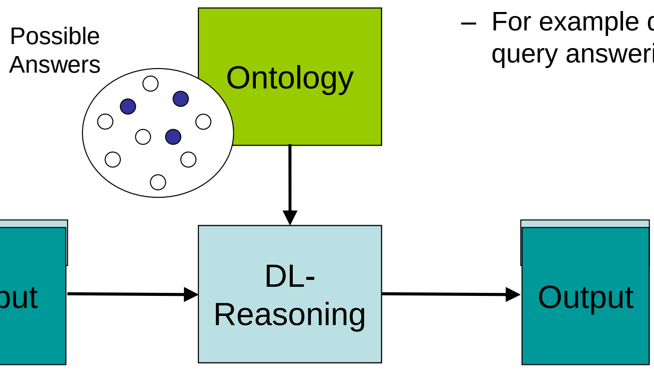
Problems tackled in KWEB

- **Scalability (Performance)**
  – DL-Reasoning in EXPTIME (OWL DL in NEXPTIME)

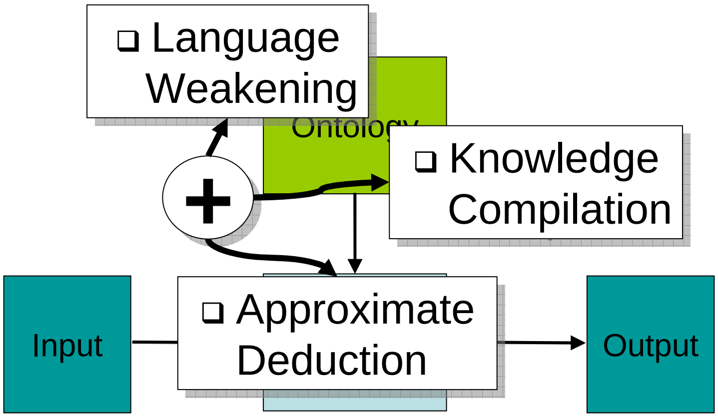


Problems tackled in KWEB

- **Robustness**
  – For example during query answering

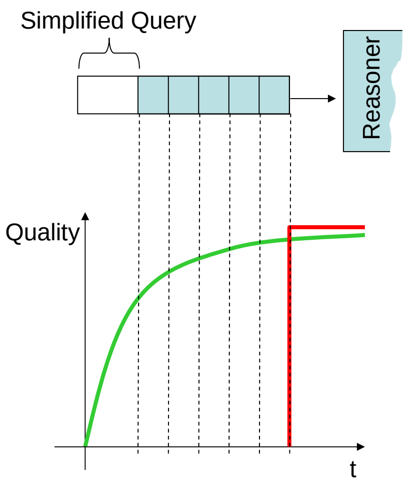


Approximation Approaches



5

Approximate Deduction through Simplification



- Simplify query
- Simple query ⇒ fast query answering
- Simple query ⇒ approximated answers
- Continuously complete query
- Anytime behavior

6

How to simplify?

First Idea:
Omit some parts (e.g. Φ, Ψ)

→ $Q^l \overset{?}{\longleftrightarrow} Q^l$

$Q^l \gg Q^l$

Query = … x Φ x … x (…w Ψ w …)

$Q^l \gg Q^l$

How to simplify? (II)

Second Idea:
Rewrite some parts (e.g. Φ, Ψ)

→ $Q^l \gg Q^l$

$Q^l \gg Q^l$ $Q^l \gg Q^l$

Query = … x A x … x (…w A w …)

$$\phi \mapsto \psi$$

Cadoli-Schaerf-Approximation for DLs

$$C_i^{\top} : \exists R.C \mapsto \top$$
$$C_i^{\bot} : \exists R.C \mapsto \bot$$

- Replacing some sub terms in concept expressions
- Well-founded theory with (theoretically) nice results

Cadoli-Schaerf-Approximation: Example

Depth of subconcept *D*:
number of universal quantifiers that have *D* in its scope.

$$(\exists \text{friend.tall}) \sqcap \forall \text{friend.}((\forall \text{friend.doctor}) \sqcap \exists \text{friend.}\neg \text{doctor})$$

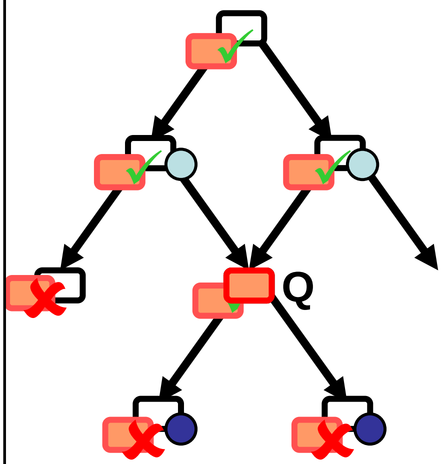
Depth: 0 — $(\exists \text{friend.tall})$; Depth: 2 — doctor in $(\forall \text{friend.doctor})$; Depth: 1 — $\exists \text{friend.}\neg \text{doctor}$

$S_0^{\top}$ $\quad \top \sqcap \forall \text{friend.}((\forall \text{friend.doctor}) \sqcap \top)$

$S_1^{\top}$ $\quad (\exists \text{friend.tall}) \sqcap \forall \text{friend.}((\forall \text{friend.doctor}) \sqcap \top)$

$S_2^{\top}$ $\quad (\exists \text{friend.tall}) \sqcap \forall \text{friend.}((\forall \text{friend.doctor}) \sqcap \exists \text{friend.}\neg \text{doctor}).$

Application: Classification



- Central process
  Classify Term Q
- Contained in
  - Generating the subsumption hierarchy
  - Instance Retrieval

Mixed Results: Classifying in TAMBIS

- Application: Classification of Concepts
  ⇒ sequence of subsumption test: C y D

≈0 ≈24 ≈279 ≈0

	normal			$C_i^\perp$			$C_i^\top$			$C_i^\perp \& C_i^\top$		
		true	false		true	false		true	false		true	false
Tambis (16)				$C_0^\perp$	157	32	$C_0^\top$	8	181	$C_0^\perp$	157	32
				$C_1^\perp$			$C_1^\top$			$C_0^\top$	8	149
	N	24	279	N			N			N		

$(C \not\sqsubseteq D)_i^\perp \;\Rightarrow\; C \not\sqsubseteq D$ $\qquad$ $(C \sqsubseteq D)_i^\top \;\Rightarrow\; C \sqsubseteq D$

$(C \sqcap \neg D)_i^\perp$ is satisfiable
$\Rightarrow (C \sqcap \neg D)$ is satisfiable

$(C \sqcap \neg D)_i^\top$ is unsatisfiable
$\Rightarrow (C \sqcap \neg D)$ is unsatisfiable

Further Results

		normal		$C_i^\perp$		$C_i^\top$		$C_i^\perp$&$C_i^\top$	
		true	false	true	false	true	false	true	false
Dolce (10)	$C_0^\perp$	-	-	0	0	-	-	0	0
	$C_0^\top$	-	-	-	-	0	0	0	0
	normal	10	113	10	113	10	113	10	113
Galen (10)	$C_0^\perp$	-	-	0	0	-	-	0	0
	$C_0^\top$	-	-	-	-	0	0	0	0
	normal	10	12190	10	12190	10	12190	10	12190
Monet (10)	$C_0^\perp$	-	-	0	0	-	-	0	0
	$C_0^\top$	-	-	-	-	0	0	0	0
	normal	20	656	20	656	20	656	20	656
MadCow (10)	$C_0^\perp$	-	-	145	0	-	-	145	0
	$C_0^\top$	-	-	-	-	5	140	5	140
	normal	66	152	66	152	61	152	61	152
Wine (10)	$C_0^\perp$	-	-	228	1	-	-	228	1
	$C_0^\top$	-	-	-	-	6	223	6	222
	normal	33	252	33	251	27	252	27	251

Problem: Term Collapsing

Subsumption Queries have this structure very often

Query =  ∀

- Term C
  - very often conjunction of subterms
  - e.g. conjunctive queries
- Term D
  - Very often also conjunction of subterms

Classifying in TAMBIS (IV)

	normal			$C_i^{\perp}$			$C_i^{\top}$			$C_i^{\perp}\&C_i^{\top}$		
		true	false		true	false		true	false		true	false
Tambis (16)				$C_0^{\perp}$	157	32	$C_0^{\top}$	8	181	$C_0^{\perp}$	157	32
				$C_1^{\perp}$	0	0	$C_1^{\top}$	0	0	$C_0^{\top}$	8	149
	N	24	279	N	24	247	N	16	279	N	16	247

Term Collapsing: 157 = 100% 65 = 35,9% 190 = 62,1%

Lessons learned

$$\phi \longmapsto \psi$$

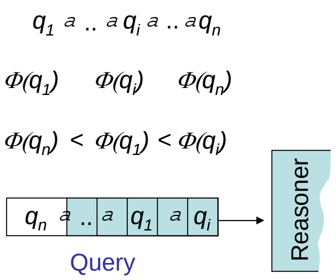
- Avoid Term Collapsing
  - Replace ψ with something else than `A` or `B`
- Find better places to rewrite
  - Ontology-adapted ϕ?

Focused Case: Instance Retrieval

- Find all instances *a* which belongs to a query *Q*: *a:Q*
- Tool *InstanceStore:*
  - Try to replace DL reasoning as much as possible with (scalable) DB retrieval
  - Only applyable to role-free A-Boxes
    $a{:}Q \leftrightarrow I_a \sqsubseteq Q$; I_a concept description of instance a
- Boolean Conjunctive Queries
  - $q_1 \wedge \cdots \wedge q_n$, where $q_1, \cdots, q_n$ are of the form *x:C* or $\langle x,y \rangle{:}R$
  - Restrict to those which can be converted to a concept expression *C*

17

New Approximation Method: Heuristic Ordering of Conjuncts



- Compute a ranking value for each conjunct
  $$\Phi(q_i) : C \mapsto \mathbb{R}$$
- Order the conjuncts $q_1, \cdots, q_n$ according to their value
- Complete approximated query with more and more expensive conjuncts

18

How to order conjuncts?

- According to the needed computation time for each conjunction
  – Estimate the computation time a priori
- According to the possible search space reduction
  – Prefer conjuncts which eliminate a lot of instances

How to estimate the computation costs?

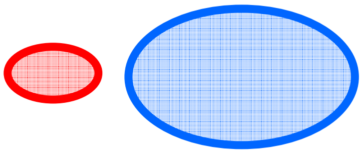
$$
\begin{aligned}
\Phi(A) &= 1 \\
\Phi(\neg A) &= 0 \\
\Phi(C \sqcap D) &= 2 + \Phi(C) + \Phi(D) \\
\Phi(C \sqcup D) &= \phi + 2 + \Phi(C) + \Phi(D) \\
\Phi(\exists\, R.C) &= 2 + \Phi(C) \\
\Phi(\forall\, R.C) &= n + n \cdot \Phi(C)
\end{aligned}
$$

where ϕ is the current value of $\Phi(E)$

where n is the number of existential quantifiers in E

Effects of Cadoli-Schaerf for Subsumption

C y D

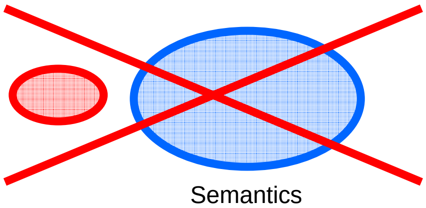


(C y D)^B "↦B"

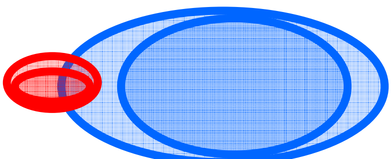
C y D ↔ ¶ C x =D

Effects of Cadoli-Schaerf for Subsumption

C y D



(C y D)^B "↦B"



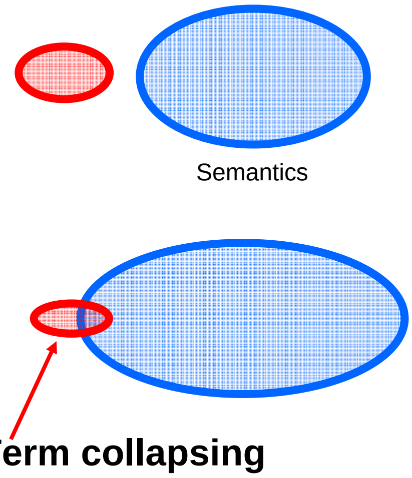
(C y D)^B ↔ ¶ (C x =D)^B

↓C^B =D^B↕

Effects of CS for Subsumption: Term Collapsing

$C \not\sqsubseteq D$

$(C \not\sqsubseteq D)^{B}$ “↦B“

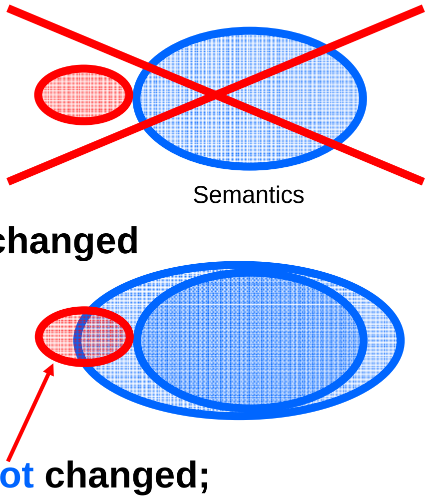


Effects of new Approximation

$(C_a \not\sqsubseteq Q)$

only Q changed

$(C_a \not\sqsubseteq Q)^{\Delta}$



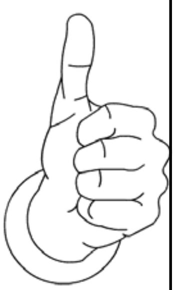
Results: Subsumption tests

	normal			$C^{\top}$			[illegible]			C^{Δ}		
		true	false		true	false		[illegible]	[illegible]		true	false
Q2										$L0$	20	0
										$L1$	20	0
				$L0$	0	19	$L0$	19	0	$L2$	9	11
	normal	9	11	normal	9	11	normal	9	11	normal	9	0
Q8										$L0$	607	0
				$L0$	0	606	$L0$	606	0	$L1$	10	[illegible]97
	normal	10	597	normal	10	597	normal	10	59[illegible]	normal	10	0
Q12				$L0$	0	7871	$L0$	7871	0	$L0$	15	[illegible]856
	normal	15	7856	normal	15	7856	normal	15	7856	normal	15	0
Q14										$L0$	408	0
										$L1$	5	[illegible]03
				$L0$	0	407	$L0$	407	0	$L2$	5	0
	normal	5	403	normal	5	403	normal	5	40[illegible]	normal	5	0
Q15				$L0$	0	6693	$L0$	6693	0	$L0$	6693	0
	normal	46	6647	normal	46	6647	normal	46	6647	normal	46	[illegible]647
Q17				$L0$	0	7873	$L0$	7873	0	$L0$	1	[illegible]872
	normal	1	7872	normal	1	7872	normal	1	7872	normal	1	0

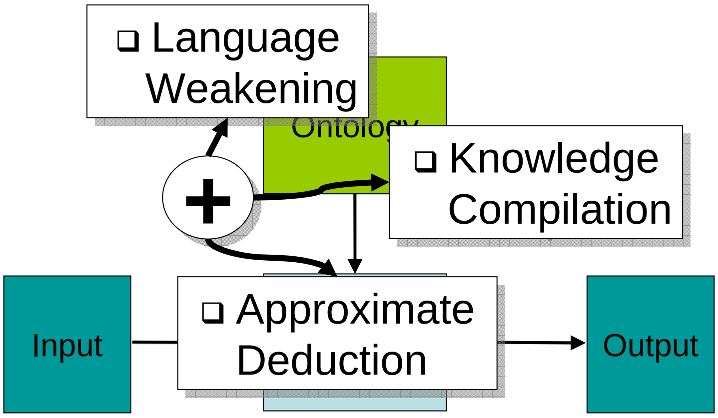
More Levels

Results: Time

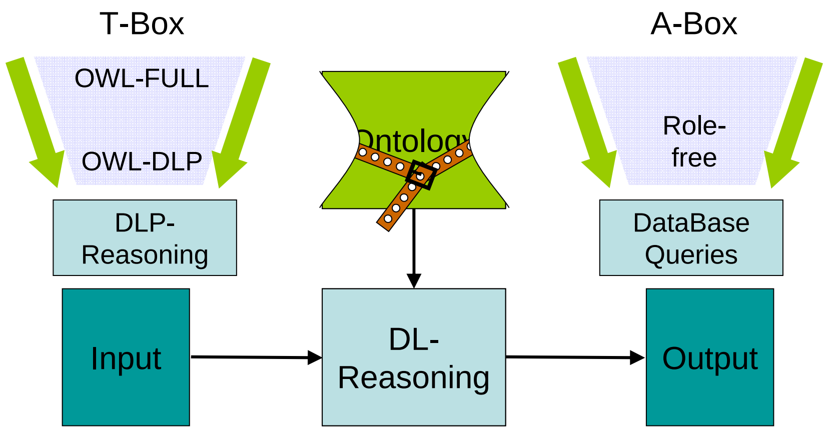
	normal	$C^{\top}$	$C^{\perp}$	C^{Δ}
Q2	175	348	299	547
Q8	5373	8383	7753	9912
Q12	61[illegible]	[illegible]	[illegible]	56478
Q14	43[illegible]	[illegible]	[illegible]17	7391
Q15	61560	90847	83714	114162
Q17	113289	[illegible]	[illegible]	93074



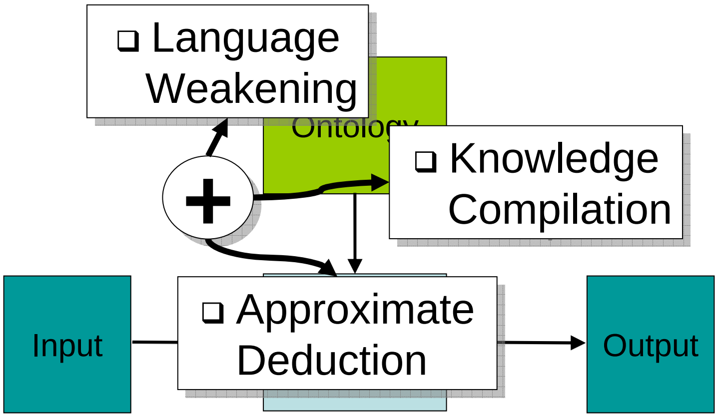
Approximation Approaches



Approximation through Language Weakening

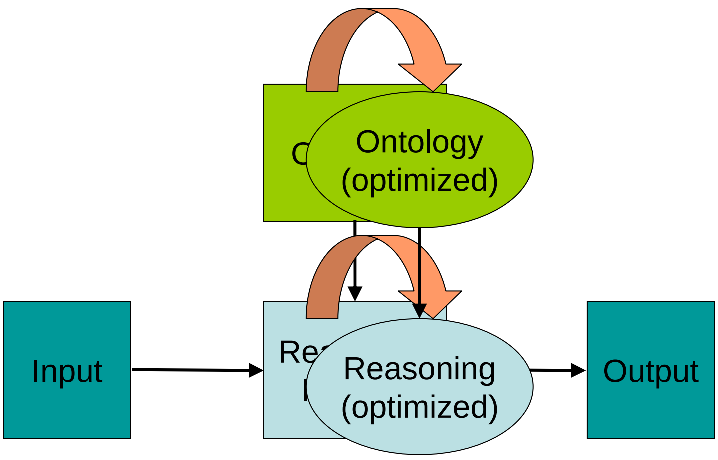


Approximation Approaches



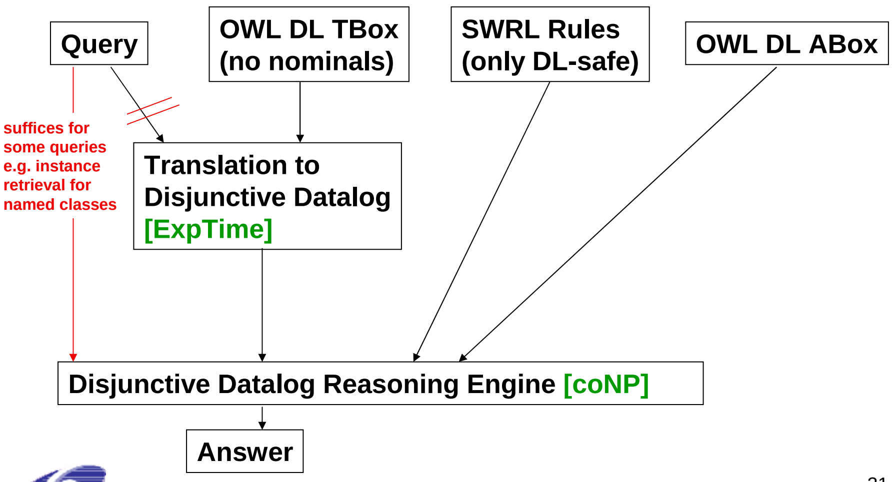
29

Approximation through Knowledge Compilation



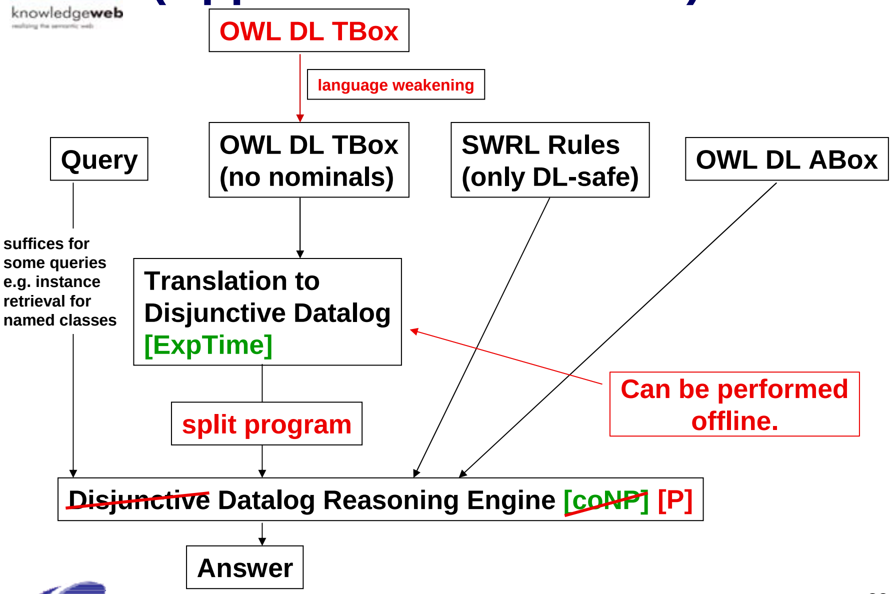
30

Standard: KAON2



31

(Approximated: KAON2) = Screech



32

Screech simple example

serbian w croatian y european
eucitizen y european
german w french w beneluxian y eucitizen
beneluxian ≡ luxembourgian w dutch w belgian

serbian(ljiljana).	serbian(nenad).	german(pascal).
french(julien).	croatian(boris).	german(markus).
german(stephan).	croatian(denny).	indian(sudhir).
belgian(saartje).	german(rudi).	german(york).

33

Screech simple example

beneluxian ≡ luxembourgian w dutch w belgian

KAON2 translates into the following four clauses:

~~luxembourgian(x) b dutch(x) b belgian(x) ˝ beneluxian(x)~~
beneluxian(x) ˝ luxemburgian(x)
beneluxian(x) ˝ dutch(x)
beneluxian(x) ˝ belgian(x)

Screech split first clause:

luxembourgian(x) ˝ beneluxian(x)
dutch(x) ˝ beneluxian(x)
belgian(x) ˝ beneluxian(x)

c **luxembourgian(saartje)**
c **dutch(saartje)**
c **belgian(saartje)**

34

Screech reasoning

- data complexity is **P**

- complete
- but unsound

- inference can be described in terms of standard notions from *non-monotonic reasoning*

Screech Performance (not optimized yet)

- Galen ontology
  - 673 axioms, 175 classes
  - randomly populated with 500 individuals

- After KAON2: 267 disjunctions in 133 rules eliminated

- Complete run:
  - queried for the extensions of all 175 Galen classes
  - resulting in 5809 classifications (Screech)
    - 5353 (i.e. **92.2%**) **correct**
  - For 138 out of 175 classes: computed extension correct
  - Average **time saved: 39.0%**

Summary

- Approximation approaches start to improve performance
  - Cadoli-Schaerf Approximation seems to not to work in practical settings
  - Heuristic approximation but performance improvements (only) in restricted cases?!
  - Screech 40% speed-up with only 8% wrong answers but only in one use-case
- Open questions:
  - Try to understand (theoretically) why they work
  - Benchmarking (more use-cases)
  - What about Robustness?

37

Thank you for your attention!