Approximate (logical) Reasoning

Based on Work of:

Frank van Harmelen
Zhisheng Huang
Perry Groot
Heiner Stuckenschmidt
Annette ten Teije
Holger Wache

Overview

- Why Approximate Reasoning ?
- Scalability (Anytime Reasoning):
  - Concept Classification
  - Instance Retrieval
- Robustness:
  - Querying Heterogeneous Sources
  - Reasoning with inconsistency

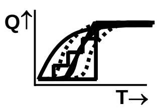
Logic =

perfect reasoning
under perfect conditions

- unlimited time
- Homogeneous knowledge
- Correct and consistent Knowledge

Robust Knowledge Representation

- Reliance on logic is a **strength**
  - Strong theoretical basis
  - Well known properties
  - Well known implementation techniques
- Reliance on logic is a **weakness**
  - Strict (no 'good enough' answers)
  - Abrupt (no intermediate answers)
  - Inefficient (no time/quality trade-off)

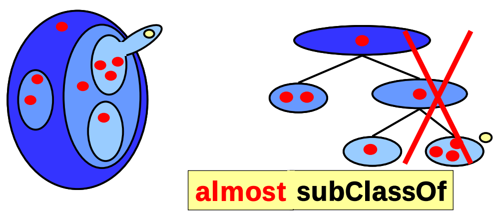


1: Fault tolerant classification

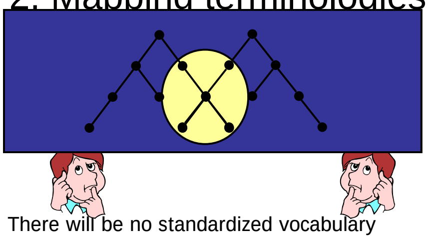
- Terminologies will be **sloppy**:
  - made by non-experts
  - made by machines:
    - scraping from
      - file-hierarchies,
      - mail-folders
      - todo-lists & phone-books on PDA's
    - machine learning from examples

1: Fault tolerant classification

- Sloppy terminologies need robust inference



2: Mapping terminologies



- There will be no standardized vocabulary
- Communication is possible only through shared vocabulary
- All concepts must be approximated in this shared vocabulary

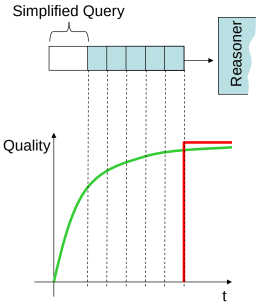
The overall dream….

Logo as a good model of **practical reasoning** (not just *idealised* reasoning)

Good model = mathematical+**computational**

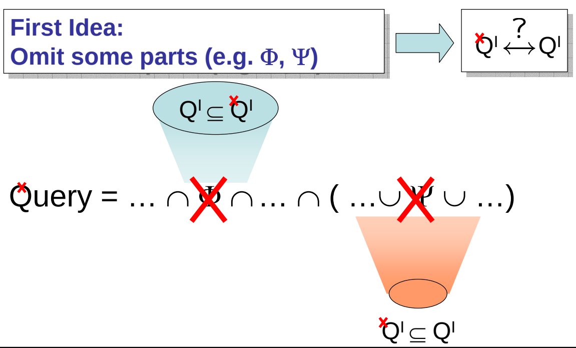
Approximate Classification in Description Logics

Approximate Entailment through Simplification

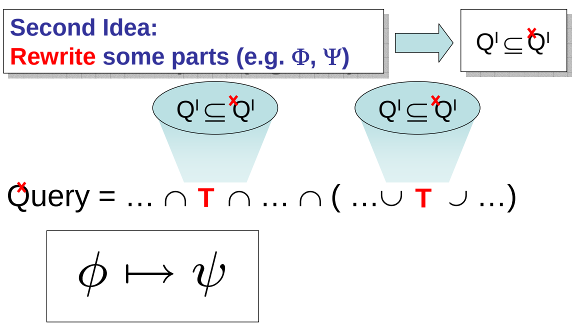


- Simplify query
- Simple query ⇒ fast query answering
- Simple query ⇒ approximated answers
- Continuously complete query
- Anytime behavior

How to simplify?

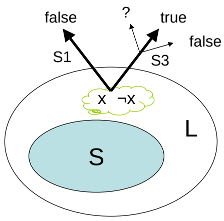


How to simplify? (II)

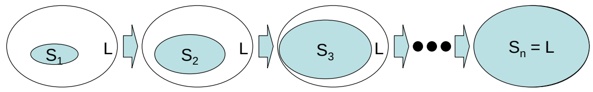


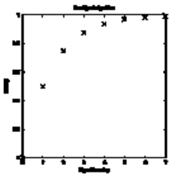
S1-/S3-entailment

- State of the Art:
  S1-/S3-entailment
  - sound and complete
  - semantic approach
- S1-entailment: interpret everything outside of S as false
- S3-entailment: interpret everything outside of S as true



S1/S3-Entailment & Anytime





$\models_0^\top, \models_1^\top, \models_2^\top, \dots \models_{n-1}^\top, \models$

- Anytime behavior when S_i will be enlarged continuously
- Interesting Feature:
  Reusing proof from previous level

Cadoli-Schaerf-Approximation for DLs

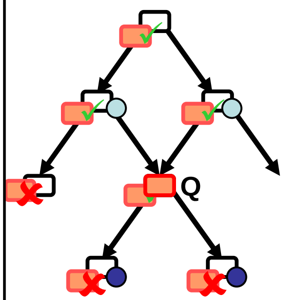
$$C_i^\top : \exists R.C \mapsto \top$$
$$C_i^\bot : \exists R.C \mapsto \bot$$

- 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 — Depth: 2 — Depth: 1

Application: Classification



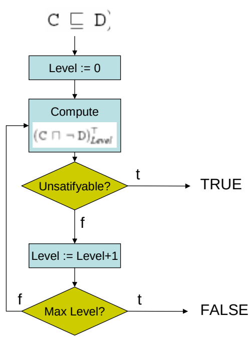
- Central process
  Classify Query Q
- Contained in
  - Generating the subsumption hierarchy
  - Instance Retrieval

1.

Classification algorithm

```
Algorithm 1 classification
Require: A classified concept hierarchy with root Root
Require: A query concept Q
  VISITED := ∅
  RESULT := ∅
  GOALS := {Root}
  while Goals ≠ ∅ do
    C ∈ Goals where {direct parents of C} ⊆ Visited
    GOALS := Goals \ {C}
    VISITED := Visited ∪ {C}
    if subsumed-by(Q,C) then
      GOALS := Goals ∪ {direct children of C}
    else
      PARENTS := direct parents of C
      RESULT := (Result ∪ Parents) \ {all ancestors of Parents}
    end if
  end while
  if |Result| = 1 ∧ subsumed-by(C,Q) then
    EQUAL := 'yes'
  else
    EQUAL := 'no'
  end if
  return Equal, Result
```

Approximating the Classification



- *C is subsumed by D* ⇔
  C⊓¬D is unsatisfiable
- Cadoli-Schaerf ensures:

$(C \sqcap \neg D)_{Level}^\top$ is unsatisfiable
$\Rightarrow (C \sqcap \neg D)$ is unsatisfiable

Approximation of subsumption

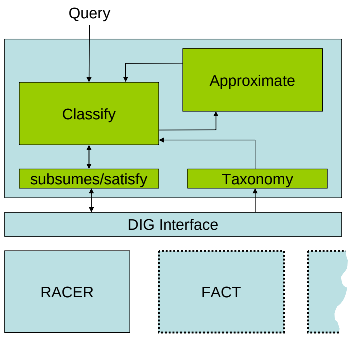
Algorithm 3 approx-$C^\top$-subsumption
Require: A complex concept expression C
Require: A Query Q
 I := 0
 repeat
  CURRENT := $(Q \sqcap \neg C)_I^\top$
  RESULT := unsatisfiable(Current)
  if Result = 'true' **then**
   break
  end if
  I := I+1
 until Current = C
 return Result

Algorithm 4 approx-$C^\perp$-subsumption
Require: A complex concept expression C
Require: A Query Q
 I := 0
 repeat
  CURRENT := $(Q \sqcap \neg C)_I^\perp$
  RESULT := unsatisfiable(Current)
  if Result = 'false' **then**
   break
  end if
  I := I+1
 until Current = C
 return Result

$\texttt{Query} \sqsubseteq \texttt{Concept} \Leftrightarrow (\texttt{Query} \sqcap \neg\texttt{Concept})$ is not satisfiable
$\Leftarrow (\texttt{Query} \sqcap \neg\texttt{Concept})_i^\top$ is not satisfiable.

$\texttt{Query} \not\sqsubseteq \texttt{Concept} \Leftrightarrow (\texttt{Query} \sqcap \neg\texttt{Concept})$ is satisfiable
$\Leftarrow (\texttt{Query} \sqcap \neg\texttt{Concept})_i^\perp$ is satisfiable.

Implementation



Mixed Results: Classifying in TAMBIS

- Application: Classification of Concepts

$\Rightarrow$ sequence of subsumption test: C $\subseteq$ 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		

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

$(C \sqsubseteq D)_i^\top \;\Rightarrow\; C \sqsubseteq D$

$(C \sqcap \neg D)_i^\perp$ 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

$C \sqsubseteq D \;\leftrightarrow C \sqcap \neg D$ is unsatisfiable

Query = … ∩ Φ ∩ … ∩ (… ∪ Ψ ∪ …)

- Term C
  - to be classified;
  - very often conjunction of subterms
  - e.g. conjunctive queries
- Term D
  - From the subsumption hierarchy
  - Very often also conjunction of subterms

Problem: Term Collapsing

$C \sqsubseteq D \;\leftrightarrow C \sqcap \neg D$ is unsatisfiable

Query = … ∩ **T** ∩ …

- Term C
  - to be classified;
  - very often conjunction of subterms
  - e.g. conjunctive queries

Problem: Term Collapsing

$C \sqsubseteq D \;\leftrightarrow C \sqcap \neg D$ is unsatisfiable

Query = $\bot$

Classifying in TAMBIS (IV)

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

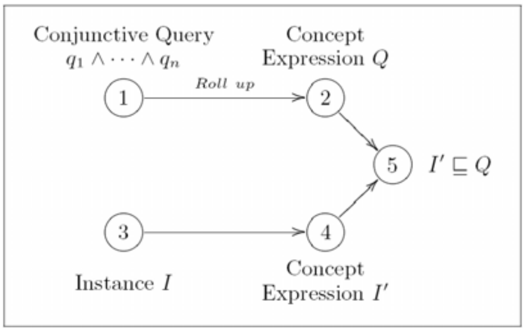
Term Collapsing: 157 = 100% 65 = 35,9% 190 = 62,1%

Lessons learned

$$\phi \mapsto \psi$$

- Avoid Term Collapsing
  - Replace ψ with something else than $\top$ or $\bot$
- Find better places to rewrite
  - Ontology-adapted ϕ?
- Look at special cases ←

Application: Instance Retrieval



What to approximate ?

- Approximating the whole expression suffers from the term collapsing problem
- Other options:
  - Approximate instance description
  - Approximate the ontology
  - Approximate only query

Approximating the Query

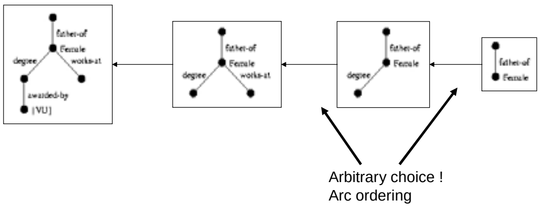
- Stepwise refinement of the query to exclude instances early on:

$$(I \not\sqsubseteq Q') \wedge (Q \sqsubseteq Q') \Longrightarrow I \not\sqsubseteq Q$$

- Resulting in a sequence $Q_1, \ldots, Q_n$ such that:
  - $Q \equiv Q_n$
  - $i > j \Rightarrow Q_i \subseteq Q_j$
- Addition of conjuncts from the original query produces the desired properties
- Basic question: order of addition ?

Variable Dependency

- Construct Query graph and use depth as a basis for selection:



Ordering Heuristics

- We estimate the size of the tableaux needed for checking satisfiability of the unfolded query
  - High penalties for disjunction and universal quantification, which are known to be the source of complexity
- Run-Time (ms):
  - Normal
  - S1
  - S3
  - alternative

	normal	$C^{\top}$	$C^{\perp}$	C^{Δ}
Q2	175	348	299	547
Q8	5373	8383	7753	9912
Q12	61410	93100	85764	56478
Q14	4372	6837	6017	7391
Q15	61560	90847	83714	114162
Q17	113289	158218	144689	93074

Analysis:

	normal			$C^{\top}$			$C^{\perp}$			C^{Δ}		
		true	false		true	false		true	false		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	597
	normal	10	597	normal	10	597	normal	10	597	normal	10	0
Q12				$L0$	0	7871	$L0$	7871	0	$L0$	15	7856
	normal	15	7856	normal	15	7856	normal	15	7856	normal	15	0
Q14										$L0$	408	0
										$L1$	5	403
				$L0$	0	407	$L0$	407	0	$L2$	5	0
	normal	5	403	normal	5	403	normal	5	403	normal	5	0
Q15				$L0$	0	6693	$L0$	6693	0	$L0$	6693	0
	normal	46	6647	normal	46	6647	normal	46	6647	normal	46	6647
Q17				$L0$	0	7873	$L0$	7873	0	$L0$	1	7872
	normal	1	7872	normal	1	7872	normal	1	7872	normal	1	0

Conclusions

- Theoretical approaches have problems
  - Web Ontologies are too simple
  - Approximation often becomes meaningless
- Works for the limited case of conjunctive queries:
  - No term collapsing
  - Leads to improvements for hard cases

Short Break

After the Break:
Approximation for Robustness

See you back in 5 Minutes

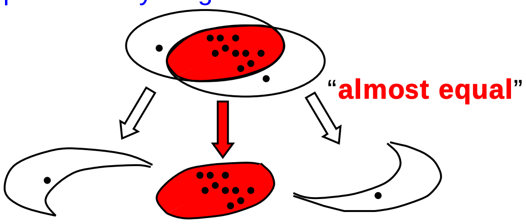
Approximation for Robustness

Aligning Terminologies
Reasoning with Inconsistency

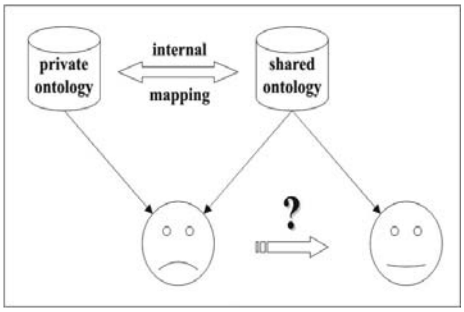
Combined ontologies need sloppy inference

Mapping ontologies is almost always messy:

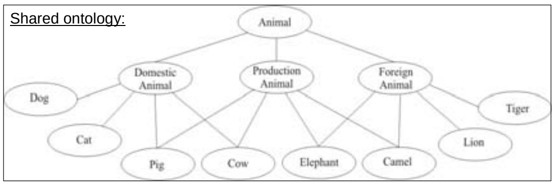
post-doc ≈ young-researcher

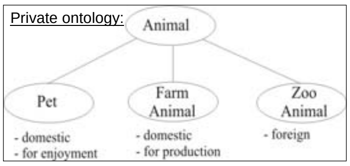


A Communication Problem



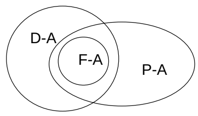
A Toy Example of Ontology Mismatch



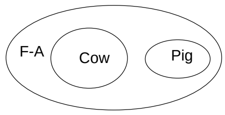


Approximation of Concepts

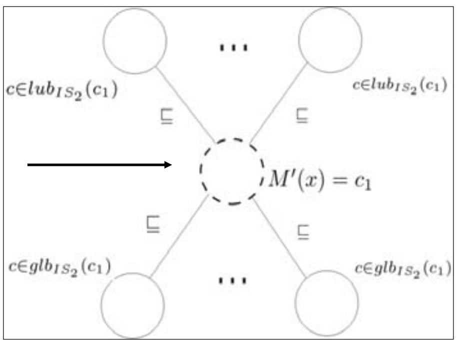
- Upper bounds (lub):
  - Pet → (Domestic-Animal)
  - Farm-Animal → (Domestic-Animal & Production-Animal)
  - Zoo-Animal → (Foreign-Animal)



- Lower bounds (glb):
  - Pet → (Cat v Dog)
  - Farm Animal → (Cow v Pig)
  - Zoo Animal → (Camel v Elephant v Tiger v Lion)



Determining Bounds

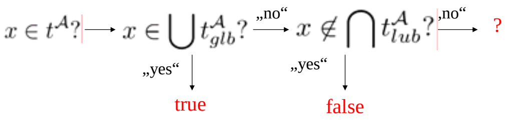


Deciding Membership

- We can define an approximate classifier that:
  - Returns ‚Yes' if the object definitely is a member of the approximated concept, i.e. its lower boundary
  - Returns ‚No' in the object does definitly not belong to the concept, i.e. its upper boundary
  - Returns ‚?' if the object is between the bounds

Theory Approximation

- answers are computed as follows:



[Selman & Kautz 1996]

Re-writing Queries

- The General Idea:
  - replace non-negated concepts by their lower bound
  - Replace negated concepts by their upper bound
- Example Re-Writing:
  (Animal & ¬(Pet v Farm-Animal))
  ➢ (Animal & ¬Pet & ¬Farm-Animal)
  ➢ (Animal & ¬(Cat v Dog) & ¬(Cow v Pig))

Re-Writing Algorithm

Algorithm 1 Translate-Message

Require: The Message to be translated: C
Require: A list of shared concepts: S
Require: A terminological knowledge base T
  racer.in-tbox(T)
  for all t is an concept term in C **do**
    if t is negated **then**
      $B[t] :=$ racer.directSupers(t)
      $B'[t] := B[t] \cap S$
      $Q(t) := (c_1 \wedge \cdots \wedge c_n)$ for $c_i \in B'[t]$
    else
      $B[t] :=$ racer.directSubs(t)
      $B'[t] := B[t] \cap S$
      $C(t) := (c_1 \vee \cdots \vee c_n)$ for $c_i \in B'[t]$
    end if
    C' := proc Replace t in C' by $C(t)$
  end for
  return C'

Approximation for handling Inconsistency

Inconsistency and Explosion

- The classical entailment is explosive:

  P, ¬ P |= Q (Any formula is a logical consequence of a contradiction.)

- The conclusions derived from an inconsistent ontology using the standard reasoning may be completely meaningless.

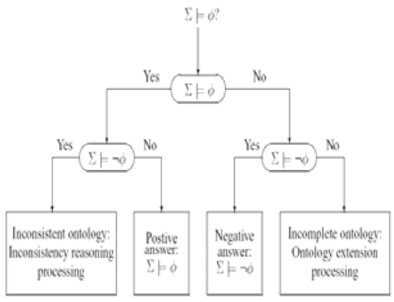
Why DL reasoning cannot escape the explosion

- The derivation checking is usually achieved by the satisfiability checking.
- Σ |=φ ⇔ Σ ∪{¬φ} is not satisfiable.
- Tableau algorithms are approaches based on the satisfiability checking
- Σ is inconsistent => Σ is not satisfiable => Σ ∪{¬φ} is not satisfiable.

Two main approaches to deal with inconsistent Ontologies

- Reasoning with Inconsistent Ontologies (RIO) by using non-standard (in particula approximate) reasoning
- Inconsistency Diagnose and Repair (will be discussed tomorrow)

Reasoning with Inconsistency: pre-processing



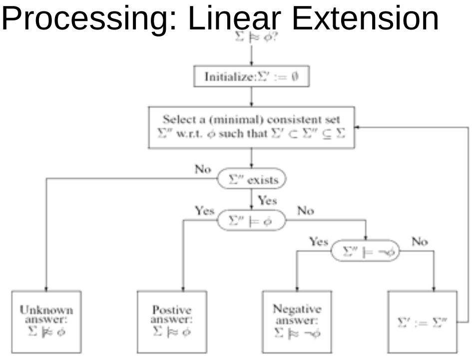
I will ask someone else.

I do not know

Reasoning with inconsistent ontologies: Main Idea

- select some consistent sub-theory from an inconsistent ontology
  - using a selection function, which can be defined on the syntactic or semantic relevance
- apply standard reasoning on the selected sub-theory to find meaningful answers

Inconsistency Reasoning Processing: Linear Extension



Inconsistency by Default rules

- Bird $\subseteq$ Animal
- Bird $\subseteq$ Fly
- Eagle $\subseteq$ Bird
- Penguin $\subseteq$ Bird
- Penguin $\subseteq$ ¬Fly

Selections for the Bird Ontology

- Bird $\subseteq$ Animal
- Bird $\subseteq$ Fly
- Eagle $\subseteq$ Bird
- Penguin $\subseteq$ Bird
- Penguin $\subseteq$ ¬Fly

selected set for queries on penguins

Selected set for other queries

Inconsistency by Modelling errors

- Brain $\subseteq$ CentralNervousSystem
- Brain $\subseteq$ BodyPart
- CentralNervousSystem $\subseteq$ NervousSystem
- BodyPart $\subseteq$ $\neg$NervousSystem

Selections on the Brain Example

- Brain $\subseteq$ CentralNervousSystem
- Brain $\subseteq$ BodyPart
- CentralNervousSystem $\subseteq$ NervousSystem
- BodyPart $\subseteq$ $\neg$NervousSystem

Selected set for queries on bodyparts

Selections on the Brain Example

- Brain $\subseteq$ CentralNervousSystem
- Brain $\subseteq$ BodyPart
- CentralNervousSystem $\subseteq$ NervousSystem
- BodyPart $\subseteq$ $\neg$NervousSystem

Selected set for queries on nervous systems

Selection Functions

- Measured by relevance
- Syntactic relevance vs. semantic relevance
- Semantic distance, semantic relateness, semantic similarity in the computational linguistics

The RIO System (available online)