Managing Inconsistent Ontologies

Tommie Meyer

Knowledge Systems Group
Meraka Institute
Pretoria
South Africa

10 June 2008

Outline

Outline

Reasoning with Ontologies

Ontologies

- Description of concepts and relationships between them
- Description Logics (DLs) are used to represent ontologies
- DLs provide the basis for reasoning with ontologies

Inconsistency management

- DL reasoners are surprisingly efficient at detecting inconsistencies
- Little guidance for resolving inconsistencies

Inconsistency Management for DLs

Propositional knowledge integration

- There is a vast body of literature on knowledge integration
- Applicable to a whole range of logics
- The solutions are typically propositional in nature

Application to DLs

- Application of existing propositional techniques to DLs
- Exploit the additional structure of DLs to refine these techniques
- Applicable to both inconsistency and concept unsatisfiability

Outline

meraka INSTITUTE
African Advanced Institute for Information & Communications Technology

Benferhat et al. 2004

Stratified knowledge bases

- $(S_1, \ldots, S_n)$ with each S_i a finite multi-set of sentences
- Lower indices represent higher reliability

Knowledge integration based on two principles

- Independence of sentences in the same stratum
- More reliable sentences should take precedence

Algorithms

- Benferhat et al: Disjunctive Maxi-adjustment
- Our version: Conjunctive Maxi-adjustment (CMA)

The CMA Algorithm

Input: $K = (S_1, \ldots, S_n)$
Output: A consistent classical knowledge base
$B := \emptyset$
FOR $i := 1$ to n
    $j := |S_i|$
    REPEAT
        $\phi := \bigvee$ of all $\bigwedge$s of size j of formulas of S_i
        $j := j - 1$
    UNTIL $B \cup \{\phi\}$ is consistent or $j = 0$
    IF $B \cup \{\phi\}$ is consistent
        $B := B \cup \{\phi\}$
    ENDIF
ENDFOR
return B

An example of CMA

1. $K = (S_1, S_2)$
2. $S_1 = \{\neg(p \wedge q), \neg(q \wedge r), \neg(p \wedge r)\}$, $S_2 = \{p, q, r\}$
3. S_1 is consistent: B is set to S_1
4. S_2 is inconsistent with B:
   ϕ is set to $(p \wedge q) \vee (p \wedge r) \vee (q \wedge r)$.
5. ϕ is inconsistent with B: ϕ is weakened to $p \vee q \vee r$
6. ϕ is consistent with B:
   B is set to $\{\neg(p \wedge q), \neg(q \wedge r), \neg(p \wedge r), p \vee q \vee r\}$

Outline

Disjunctive DL knowledge bases

Port CMA to DLs

- DLs are not expressive enough for a direct port
- E.g. $(C \sqsubseteq D) \sqcup C(a)$ is ill-formed
- This is not an issue for testing concept satisfiability
- Regardless: wish to retain the structure

Disjunctive DL knowledge bases

- Set of DL knowledge bases $\mathcal{B}$
- E.g. $\mathcal{B} = \{[C \sqsubseteq D], [C(a)]\}$
- Meaning: $\mathcal{B}$ is true iff at least one of its elements are true

The CMA algorithm for DLs

Input: $K = (S_1, \ldots, S_n)$; **Output**: A consistent DKB
$\mathcal{B} := \{\emptyset\}$
FOR $i := 1$ to n
    $\mathcal{C} := \mathcal{B}$
    FORALL $B \in \mathcal{C}$
        $j := |S_i|$
        REPEAT
            $\mathcal{X} := \{X \mid X \subseteq S_i \text{ and } |X| = j\}$
            $j := j - 1$
        UNTIL $B \cup X$ is consistent for some $X \in \mathcal{X}$
        $\mathcal{B} := (\mathcal{B} \setminus \{B\}) \cup$
         $\{B \cup X \mid (X \in \mathcal{X})\ \&(B \cup X) \text{ is consistent}\}$
    ENDFOR
ENDFOR
return $\mathcal{B}$

CMA for DLS

An example of CMA for DLs

1. $K = (S_1, S_2)$
2. $S_1 = [taxPR(Tommie), \neg PR(Tommie), \neg PR(Louise)]$
   $S_2 = [\neg PR \sqsubseteq \neg taxPR]$
3. S_1 is consistent: $\mathcal{B}$ is set to $\{S_1\}$
4. S_2 is inconsistent with S_1, the only element of $\mathcal{B}$
5. $\mathcal{B} = \{[\neg PR(Tommie), taxPR(Tommie), \neg PR(Louise)]\}$

CMA for DLs does not exploit DL structure

- Ought to conclude that Louise does not have taxPR
- Need to express: everyone without PR, except Tommie, do not have taxPR

Refined version of CMA for DLs

Enrich the language

- Replace subsumption with exception-based subsumption
- $C \sqsubseteq_n D$ means the subsumption may have n exceptions
- $C \sqsubseteq D$ expressed as $C \sqsubseteq_0 D$
- Can be expressed using nominals:
  $C \sqsubseteq_n D \equiv ((\leq n)\ C \sqcap \neg D)) \equiv (C \sqcap \neg D) \sqsubseteq \{a_1, \ldots a_n\}$

Additional structure allows for weakening statements

- $C \sqsubseteq D$ weakened to $C \sqsubseteq_1 D$
- In general: $C \sqsubseteq_n D$ weakened to $C \sqsubseteq_{n+1} D$

The refined CMA algorithm for DLs

Input: $K = (S_1, \ldots, S_n)$; **Output**: A consistent DKB

$\mathcal{B} := \{\emptyset\}$

FOR $i := 1$ to n

  $\mathcal{C} := \mathcal{B}$

  FORALL $B \in \mathcal{C}$

    $j := 0$

    REPEAT

      $\mathcal{X}$:= *appropriate weakening*

      $j := j + 1$

    UNTIL $B \cup X$ is consistent for some $X \in \mathcal{X}$

    $\mathcal{B} := (\mathcal{B} \setminus \{B\}) \cup$

     $\{B \cup X \mid (X \in \mathcal{X})\ \&(B \cup X)$ is consistent$\}$

  ENDFOR

ENDFOR

return $\mathcal{B}$

Example of refined CMA for DLs

1. $S_1 = [\neg PR(T), \neg PR(L)]$, and
   $S_2 = [TPR(T), TPR(L), \neg PR \sqsubseteq_0 \neg TPR]$
2. S_1 is consistent: $\mathcal{B}$ (and $\mathcal{C}$) = $\{S_1\}$
3. B = only element of $\mathcal{C}$: $[\neg PR(T), \neg PR(L)]$
4. S_2 is inconsistent with B: $\mathcal{X}$ =
   $\{[TPR(T), TPR(L), \neg PR \sqsubseteq_1 \neg TPR]$
   $[TPR(T), \neg PR \sqsubseteq_0 \neg TPR]$
   $[TPR(L), \neg PR \sqsubseteq_0 \neg TPR]$
5. Every element of $\mathcal{X}$ is inconsistent with B: $\mathcal{X}$ is set to:
   $\{[TPR(T), TPR(L), \neg PR \sqsubseteq_2 \neg TPR]$,
   $[TPR(T), \neg PR \sqsubseteq_1 \neg TPR]$,
   $[TPR(L), \neg PR \sqsubseteq_1 \neg TPR]$,
   $[\neg PR \sqsubseteq_0 \neg TPR]\}$

Example of refined CMA for DLs

Consistency is reached and exactly one of four cases hold

1. Tommie and Louise are the only two people without PR, but with taxPR
2. Tommie is the only person without PR, but with taxPR
3. Louise is the only person without PR, but with taxPR
4. All persons without PR, including Tommie and Louise, do not have taxPR

Outline

Maximally concept-satisfiable subsets

- Related to finding consistent subsets of maximal cardinality
- Closely related to finding maximally consistent subsets
- Current implementation: maximally concept-satisfiable subsets
- Specialised tableau algorithm for acyclic Tboxes
- Outperforms optimised reasoners using brute force
- Similar principles apply for pinpointing

The Specialised Tableau Algorithm

Algorithm for finding MSSs (relative to A_j)

- Γ has n axioms of the form $A_i \doteq C_i$
- The algorithm generates a tree in classical tableau style
- Nodes are labelled with indexed concept assertions
- $(a : C, I)$: axioms indexed by I are responsible for $a : C$
- Root node contains $(a : A_j, \emptyset)$
- The tree is generated from a number of expansion rules
- Each node x maintains an exclusion-set $E(x)$
- Performs conversion to NNF on the fly

Expansion Rules

1. D^+-rule

- If $(a : A_i, I)$ is in $L(x)$ and has not been tagged then
  - Tag $(a : A_i, I)$ and $L(x) := L(x) \cup \{(a : C_i, I \cup \{i\})\}$
- Lazy unfolding for positive concepts
- Observe the updating of the index set

2. D^--rule

- If $(a : \neg A_i, I)$ is in $L(x)$ and has not been tagged then
  - Tag $(a : \neg A_i, I)$ and $L(x) := L(x) \cup \{(a : \neg C_i, I \cup \{i\})\}$
- Lazy unfolding for negative concepts
- Observe the updating of the index set

Expansion Rules

3. $\sqcap$-rule

- If $(a : C \sqcap D, l) \in L(x)$ then
  - $L(x) := L(x) \setminus \{(a : C \sqcap D, l)\} \cup \{(a : C, l), (a : D, l)\}$
- Classical $\sqcap$-rule

4. $\sqcup$-rule

- If $(a : C \sqcup D, l) \in L(x)$ then
  - Create two children y and z of x
  - $L(y) := L(x) \setminus \{(a : C \sqcup D, l)\} \cup \{(a : C, l)\}; E(y) := E(x)$
  - $L(z) := L(x) \setminus \{(a : C \sqcup D, l)\} \cup \{(a : D, l)\}; E(z) := E(x)$
- Classical $\sqcup$-rule

Expansion Rules

5. $\exists$-rule

- If $(a : \exists R.C, I) \in L(x)$ and rules 1-4 can't be applied then
  - $X := \{(b : C, I)\} \cup \{(b : D, I \cup J) \mid (a : \forall R.D, J) \in L(x)\}$
  - (b is a new unique individual name not used before)
  - $L(x) := (L(x) \setminus \{(a : \exists R.C, I)\}) \cup X$
- Combines $\exists$- and $\forall$-rules for classical tableau systems
- Observe the updating of the index sets

Expansion Rules

6. $\bot$-rule

- If $(a : A, I) \in L(x)$ and $(a : \neg A, J) \in L(x)$ then
  - For every $i \in I \cup J$ do:
  - Create a new child y of x
  - $L(y) := L(x) \setminus \{(b : D, K) \mid i \in K\}; E(y) := E(x) \cup \{i\}$
- This rule breaks clashes one by one
- Observe the updating of the exclusion-sets

Outline

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4, A_2 \doteq \forall R.D, A_3 \doteq \exists R.C, A_4 \doteq \forall R.\neg D\}$

Initialise root node

$L(r) = \{(a : A_1, \emptyset)\}$

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4, A_2 \doteq \forall R.D, A_3 \doteq \exists R.C, A_4 \doteq \forall R.\neg D\}$

Apply the D^+-rule to A_1

$L(r) = \{(a : A_1, \emptyset)\}$

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4, A_2 \doteq \forall R.D, A_3 \doteq \exists R.C, A_4 \doteq \forall R.\neg D\}$

Apply the D^+-rule to A_1

$$L(r) = \{(a : A_1, \emptyset), (a : (A_2 \sqcap A_3 \sqcap A_4, \{1\}))\}$$

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4, A_2 \doteq \forall R.D, A_3 \doteq \exists R.C, A_4 \doteq \forall R.\neg D\}$

Apply the $\sqcap$-rule twice:

$$L(r) = \{(a : A_1, \emptyset), (a : (A_2 \sqcap A_3 \sqcap A_4, \{1\})\}$$

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4, A_2 \doteq \forall R.D, A_3 \doteq \exists R.C, A_4 \doteq \forall R.\neg D\}$

Apply the $\sqcap$-rule twice:

$$L(r) = \{(a : A_1, \emptyset), (a : A_2, \{1\}), (a : A_3, \{1\}), (a : A_4, \{1\})\}$$

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4,\ A_2 \doteq \forall R.D,\ A_3 \doteq \exists R.C,\ A_4 \doteq \forall R.\neg D\}$

Apply the D^+-rule to A_2

$$L(r) = \{(a : A_1, \emptyset), (a : A_2, \{1\}), (a : A_3, \{1\}), (a : A_4, \{1\})\}$$

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4, A_2 \doteq \forall R.D, A_3 \doteq \exists R.C, A_4 \doteq \forall R.\neg D\}$

Apply the D^+-rule to A_2

$L(r) = \{(a : A_1, \emptyset), (a : A_2, \{1\}), (a : A_3, \{1\}), (a : A_4, \{1\}),$

$(a : \forall R.D, \{1, 2\})\}$

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4, A_2 \doteq \forall R.D, A_3 \doteq \exists R.C, A_4 \doteq \forall R.\neg D\}$

Apply the D^+-rule to A_3

$L(r) = \{(a : A_1, \emptyset), (a : A_2, \{1\}), (a : A_3, \{1\}), (a : A_4, \{1\}),$

$(a : \forall R.D, \{1, 2\})\}$

meraka INSTITUTE
African Advanced Institute for Information & Communications Technology

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4, A_2 \doteq \forall R.D, A_3 \doteq \exists R.C, A_4 \doteq \forall R.\neg D\}$

Apply the D^+-rule to A_3

$L(r) = \{(a : A_1, \emptyset), (a : A_2, \{1\}), (a : A_3, \{1\}), (a : A_4, \{1\}),$

$(a : \forall R.D, \{1, 2\}), (a : \exists R.C, \{1, 3\}))\}$

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4,\ A_2 \doteq \forall R.D,\ A_3 \doteq \exists R.C,\ A_4 \doteq \forall R.\neg D\}$

Apply the D^+-rule to A_4

$L(r) = \{(a : A_1, \emptyset), (a : A_2, \{1\}), (a : A_3, \{1\}), (a : A_4, \{1\}),$

$(a : \forall R.D, \{1, 2\}), (a : \exists R.C, \{1, 3\}))\}$

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4, A_2 \doteq \forall R.D, A_3 \doteq \exists R.C, A_4 \doteq \forall R.\neg D\}$

Apply the D^+-rule to A_4

$L(r) = \{(a : A_1, \emptyset), (a : A_2, \{1\}), (a : A_3, \{1\}), (a : A_4, \{1\}),$

$(a : \forall R.D, \{1, 2\}), (a : \exists R.C, \{1, 3\})), (a : \forall R.\neg D, \{1, 4\})\}$

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4, A_2 \doteq \forall R.D, A_3 \doteq \exists R.C, A_4 \doteq \forall R.\neg D\}$

Apply the ∃-rule

$$L(r) = \{(a : A_1, \emptyset), (a : A_2, \{1\}), (a : A_3, \{1\}), (a : A_4, \{1\}),$$

$$(a : \forall R.D, \{1, 2\}), (a : \exists R.C, \{1, 3\})), (a : \forall R.\neg D, \{1, 4\})\}$$

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4,\ A_2 \doteq \forall R.D,\ A_3 \doteq \exists R.C,\ A_4 \doteq \forall R.\neg D\}$

Apply the $\exists$-rule

$L(r) = \{(a : A_1, \emptyset), (a : A_2, \{1\}), (a : A_3, \{1\}), (a : A_4, \{1\}),$

$(a : \forall R.D, \{1, 2\}), (a : \exists R.C, \{1, 3\})), (a : \forall R.\neg D, \{1, 4\}),$

$(b : D, \{1, 2, 3\})), (b : C, \{1, 3\}), (b : \neg D, \{1, 3, 4\})\}$

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4, A_2 \doteq \forall R.D, A_3 \doteq \exists R.C, A_4 \doteq \forall R.\neg D\}$

$\perp$-rule: New child for each element of $\{1, 2, 3\} \cup \{1, 3, 4\}$

$L(r) = \{(a : A_1, \emptyset), (a : A_2, \{1\}), (a : A_3, \{1\}), (a : A_4, \{1\}),$

$(a : \forall R.D, \{1, 2\}), (a : \exists R.C, \{1, 3\})), (a : \forall R.\neg D, \{1, 4\}),$

$(b : D, \{1, 2, 3\})), (b : C, \{1, 3\}), (b : \neg D, \{1, 3, 4\})\}$

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4,\ A_2 \doteq \forall R.D,\ A_3 \doteq \exists R.C,\ A_4 \doteq \forall R.\neg D\}$

$\bot$-rule: New child for 1

$L(x_1) = \{(a : A_1, \emptyset),$ ~~$(a : A_2, \{1\})$~~, ~~$(a : A_3, \{1\})$~~, ~~$(a : A_4, \{1\})$~~,

~~$(a : \forall R.D, \{1, 2\})$~~,~~$(a : \exists R.C, \{1, 3\}))$~~, ~~$(a : \forall R.\neg D, \{1, 4\})$~~,

~~$(b : D, \{1, 2, 3\}))$~~, ~~$(b : C, \{1, 3\})$~~, ~~$(b : \neg D, \{1, 3, 4\})$~~$\}$

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4, A_2 \doteq \forall R.D, A_3 \doteq \exists R.C, A_4 \doteq \forall R.\neg D\}$

$\bot$-rule: New child for 2

$L(x_2) = \{(a : A_1, \emptyset), (a : A_2, \{1\}), (a : A_3, \{1\}), (a : A_4, \{1\}),$

~~$(a : \forall R.D, \{1,2\})$~~$, (a : \exists R.C, \{1,3\})), (a : \forall R.\neg D, \{1,4\}),$

~~$(b : D, \{1,2,3\}))$~~$, (b : C, \{1,3\}), (b : \neg D, \{1,3,4\})\}$

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4,\ A_2 \doteq \forall R.D,\ A_3 \doteq \exists R.C,\ A_4 \doteq \forall R.\neg D\}$

$\perp$-rule: New child for 3

$L(x_3) = \{(a : A_1, \emptyset), (a : A_2, \{1\}), (a : A_3, \{1\}), (a : A_4, \{1\}),$

$(a : \forall R.D, \{1,2\})$, ~~$(a : \exists R.C, \{1,3\}))$~~, $(a : \forall R.\neg D, \{1,4\})$,

~~$(b : D, \{1,2,3\}))$~~, ~~$(b : C, \{1,3\})$~~, ~~$(b : \neg D, \{1,3,4\}))$~~$\}$

Example

$\Gamma = \{A_1 \doteq A_2 \sqcap A_3 \sqcap A_4, A_2 \doteq \forall R.D, A_3 \doteq \exists R.C, A_4 \doteq \forall R.\neg D\}$

$\perp$-rule: New child for 4

$L(x_4) = \{(a : A_1, \emptyset), (a : A_2, \{1\}), (a : A_3, \{1\}), (a : A_4, \{1\}),$

$(a : \forall R.D, \{1,2\}), (a : \exists R.C, \{1,3\})), ~~(a : \forall R.\neg D, \{1,4\})~~,$

$(b : D, \{1,2,3\})), (b : C, \{1,3\}), ~~(b : \neg D, \{1,3,4\})~~\}$

Algorithm Properties

- Sound, Complete, Terminating
- Checking A_j-concept satisfiability is PSPACE-complete
- Maintenance of index-sets yields EXPTIME
- Preliminary experiments give faster runtimes than a black box approach using FACT++
- Easily modified to find MSSs for all concept names

Outline

Future Work

Belief Change

- How to reconcile new information with existing beliefs
  - Revision: Add a new statement to your current beliefs
  - Contraction: Remove a statement from your current beliefs

Ontology Change

- During ontology construction the subsumption hierarchy can be computed automatically
- In practice ontology engineers want to modify this
- Add a subsumption link explicitly: revision
- Remove a subsumption link explicitly: contraction