Reasoning with beliefs provided by contradictory sources

Laurence Cholvy
ONERA Centre de Toulouse
Toulouse, France
cholvy@cert.fr

CONTEXT:

(1) Reasoning with information provided by several information sources that can be contradictory.

(2) we consider the case when information are beliefs

POTENTIAL APPLICATIONS:

intelligence, multi-databases querying, etc.

ASSUMPTION:

the information sources share the same information description language (propositional, first-order)

GENERAL IDEAS:

For dealing with contradictory beliefs provided by several sources:

(1) the first thing to do is to use the reliability of the sources (if it is known)

In order to resolve a contradiction, reject the (minimal) information provided by a less reliable source

(2) if we have no information about the reliability of the sources, then we can adopt a majority attitude

In order to resolve a contradiction, reject the information which is supported by less sources than its negation

IN THE FOLLOWING:

We present two logics (extensions of KD logics) corresponding to the two previous attitudes.

VOCABULARY, NOTATION, ASSUMPTION.

An information source can be :

- a "primitive" source (the ones to be merged, the ones we query) denoted $db_1, db_2 ... db_n$
- a source obtained by merging some primitive sources
  - when taking their reliability into account, denoted $db_1 > db_2 > ... > db_n$
  - by a majority attitude, denoted $db_1 * db_2 * ... * db_n$

ASSUMPTION : Primitive source are sets of literals. Furthermore, each db_i can be associated with a set of rules IDB_i so that $db_i \cup IDB_i$ is consistent and equivalent to a set of literals.

FIRST LOGIC: THE LANGUAGE

$B_{db}F$ means that there source db believes F

where db can be a primitive source db_i or a a source obtained from merging the primitive sources $db_1 ... dn_n$ assuming a reliability order. In this case, it is denoted $db_1 > ... > db_n$

ITS SEMANTICS

Kripke models

FIRST LOGIC: PROOF THEORY

(A0) Axioms of the propositional logic.

(A1) B_{db} ¬ F → ¬ B_{db} F

(A2) B_{db} F ∧ B_{db} (F → G) → B_{db} G

(A3) B_{db} l → $B_{db>db_i}$ l

(A4) B_{db_i} l ∧ ¬ B_{db} ¬ l → $B_{db>db_i}$ l

(A5) $B_{db>db_i}$ l → B_{db} l ∨ B_{db_i}l

(A6) B_{db} (l1 ∨ ... lp) → B_{db} l1 ∨ ... B_{db} lp where li's are non complementary literals.

INFERENCE RULES

(Nec) $\quad \vdash$ F $\quad \Longrightarrow \quad \vdash B_{db}$ F
(if F is a propositional formula)

(MP) $\quad \vdash$ F and $\vdash$ (F $\rightarrow$ G) $\Longrightarrow \vdash$ G

INTERESTING THEOREMS

$$\psi = \bigwedge_{i=1}^{n} (\bigwedge_{F \in db_i \cup IDB_i} B_{db_i} F \quad \bigwedge_{db_i \cup IDB_i \not\models l} \neg B_{db_i} l)$$

What does the source db believe :

$$\vdash \psi \rightarrow B_{db}???$$

and, what does not it believe :

$$\vdash \psi \rightarrow \neg B_{db}???$$

EXAMPLE :

$db_1 = \{highspeed\}$
$db_2 = \{lowaltitude\}$
$db_3 = \{\neg plane, helicoptere\}$

$IDB = \{highspeed \rightarrow plane,$
$lowaltitude \rightarrow helicoptere,$
$\neg(plane \wedge helicoptere)\}$

(common to the three information sources)

Some interesting theorems:

$\vdash \psi \rightarrow B_{db_1 > db_2 > db_3} highspeed$

$\vdash \psi \rightarrow B_{db_1 > db_2 > db_3} plane$

$\vdash \psi \rightarrow B_{db_1 > db_2 > db_3} \neg lowaltitude$

$\vdash \psi \rightarrow B_{db_1 > db_2 > db_3} \neg helicoptere)$

$\vdash \psi \rightarrow \neg B_{db_1 > db_2 > db_3} helicoptere$

$\vdash \psi \rightarrow B_{db_3 > db_2 > db_1} (\neg highspeed \wedge \neg plane \wedge lowaltitude \wedge helicoptere)$

$\vdash \psi \rightarrow B_{db_2 > db_1} \neg plane$

SECOND LOGIC: THE LANGUAGE

$B^j_{db}F$ means that there are j occurrences of F in the source denoted db

$B_{db}F$ means that the source db believes F

where db can be a primitive source db_i or a a source obtained from merging the primitive sources $db_1 ... dn_n$. In this case, it is denoted $db_1 * ... * db_n$

SEMANTICS

Kripke Models

SECOND LOGIC: PROOF THEORY

(A0) Axiom schemata of propositional logic

(A1) $B_{db}\neg F \rightarrow \neg B_{db}F$

(A2) $B_{db}F \wedge B_{db}(F \rightarrow G) \rightarrow B_{db}G$

(A3) $B^k_{db}l \leftrightarrow B^k_{db}\neg\neg l$

(A4) $B^k_{db}l \rightarrow \neg B^p_{db}l$ if $k \neq p$

(A5) $B^k_{db}l \wedge B^p_{db'}l \rightarrow B^m_{db*db'}l$ if $m = k + p$

(A6) $B^k_{db}l \wedge B^p_{db}\neg l \rightarrow B_{db}l$ if $k > p$

(A7) $B^k_{db}l \wedge B^k_{db}\neg l \rightarrow \neg B_{db}l$

(A8) $B_{db}(l_1 \vee ... \vee l_n) \rightarrow B_{db}l_1 \vee ... \vee B_{db}l_n$ where li's are non complementary literals.

INFERENCE RULES

(Nec) $\vdash$ F $\Longrightarrow$ $\vdash B_{db}$ F
(if F is a propositional formula)

(MP) $\vdash$ F and $\vdash$ (F $\rightarrow$ G) $\Longrightarrow$ $\vdash$ G

INTERESTING THEOREMS

$$\psi = \bigwedge_{i=1}^{n} \left(\bigwedge_{db_i \cup IDB_i \models l} B^1_{db_i} l \wedge \bigwedge_{db_i \cup IDB \not\models l} B^0_{db_i} l \right)$$

What does the source db believes :

$$\vdash \psi \rightarrow B_{db}???$$

and, what does not it believe :

$$\vdash \psi \rightarrow \neg B_{db}???$$

EXAMPLE

$db_1 = \{highspeed\}$
$db_2 = \{lowaltitude\}$
$db_3 = \{\neg plane, helicoptere\}$

$IDB = \{highspeed \rightarrow plane,$
$lowaltitude \rightarrow helicoptere,$
$\neg(plane \wedge helicoptere)\}$

(common to the three information sources)

Some interesting theorems:

$\vdash \psi \rightarrow B_{db_1 * db_2 * db_3} \quad \neg highspeed$

$\vdash \psi \rightarrow B_{db_1 * db_2 * db_3} \quad \neg plane$

$\vdash \psi \rightarrow B_{db_1 * db_2 * db_3} \quad helicoptere$

$\vdash \psi \rightarrow \neg B_{db_1 * db_2 * db_3} \quad lowaltitude$

$\vdash \psi \rightarrow \neg B_{db_1 * db_2 * db_3} \quad \neg lowaltitude$

AUTOMATED DEDUCTION IN THESE LOGICS

For each logic, we have defined a PROLOG program, META, such that :

PROLOG proves $B(db, l)$ in META iff
$\vdash \psi \rightarrow B_{db}F$

PROLOG fails to prove $B(db, l)$ in META iff
$\vdash \psi \rightarrow \neg B_{db}F$

(where db can be a primitive or merged source)

In these programs, expliciting ψ is not needed: we only have to list the contents of the sources. Negation as failure is used to deduce what the source does not believe.

APPLICATION TO FIRST-ORDER DATABASES :

Assumption:
(1) databases are "equivalent to a set of literals" (no pure disjunction)
(2) No recursion.

Closed queries :

answer(db, F) = YES $\Longleftrightarrow$
PROLOG proves B(db, F) in META

answer(db, F) = NO $\Longleftrightarrow$
PROLOG proves B(db, ¬ F) in META

answer(db, F) = ? else

Open queries :

The answer to the query "in the merged source db, what are the X which are supposed to satisfy F", is defined by :

answer(db, F(X)) = { A : PROLOG proves B(db, F(A)) in META }

$$IDB$$
$$\forall x\ St(x) \rightarrow Self(x)$$
$$\forall x\ Emp(x) \rightarrow Restau(x)$$
$$\forall x\ \neg Self(x) \vee \neg Restau(x)$$

.

.

$db1$	$db2$	$db3$
$Stud(john)$	$Restau(john)$	$Stud(john)$
$Emp(henry)$	$\neg Self(henry)$	$Stud(henry)$
$\neg Stud(philip)$	$Restau(henry)$	$Restau(philip)$
$Restau(philip)$	$\neg Emp(philip)$	$\neg Stud(philip)$
	$Self(philip)$	

- $answer(db_1 > db_2 > db_3, Stud(x)) = \{john\}$
- $answer(db_2 > db_1 > db_3, Stud(x)) = \emptyset$
- $answer(db_1 > db_2 > db_3, Restau(x)) = \{philip, henry\}$
- $answer(db_2 > db_1, Self(philip)) = YES$
- $answer(db_1 * db_2 * db_3, Stud(x)) = \{john\}$
- $answer(db_1 * db_2 * db_3, Emp(x)) = \emptyset$
- $answer(db_1 * db_2 * db_3, Restau(x)) = \{philip, henry\}$
- $answer(db_1 * db_2, Self(philip)) = ?$
- $answer(db_1 * db_2 * db_3, Self(philip)) = NO$

CONCLUDING REMARKS

- Different versions of these logics :
  - Explicit beliefs
  - Reliability depending on topics of information
  - partial order of reliability
- Models theory
- completeness of the logics when sets of literals
- extension to disjunction: open problem
- extension forseen: mixing the two attitudes