Answers about validity and completeness of data: formal definitions, usefulness and computation technique

Robert Demolombe
ONERA-CERT-DTIM
2 Avenue E. Belin, B.P. 4025
31055 Toulouse Cedex
France
e-mail: Robert.Demolombe@cert.fr.

Abstract

We present here a continuation of our work presented in [Dem97] . We recall definitions of valid subsets or complete subsets of a database, and the modal logic that is used for reasoning about assumptions on validity and completeness, in order to characterise subsets of a standard answer that are either valid or complete. We formally define several forms of answers that are either extensional or intensional. Then, we analyse which kinds of anwers are really useful for users. Finally, we present an automated deduction method to compute these answers. This method is based on SOL-deduction, which has been designed for classical logic, and we show how it can be adapted to our modal logic.

1 Introduction

In this paper is presented a continuation of a work that has already been published in [Dem97] (see also [CDJ94, DJ94, Dem96]). We have considered situations where databases, or distributed database, or information sources, in general, may, or may not, be reliable in regard to stored data. Here we have restricted the scope to databases.

It is assumed that some parts of the database are valid or complete [Mot86, Mot89]. If we consider, for example, a database that contains data about bank agencies: their location, the fact that they are open or close, that accounts are negative, etc, it can be assumed, for example, that data about open agencies are valid, and data about downtown agencies are complete.

Property of validity intuitively means that if it is stored in the database that some bank is open, then this fact is "guaranteed" to be true. Property of completeness means that if it is "guaranteed" that some bank agency is located downtown, then this fact is stored in the database.

In this situation, if some user asks to the database what are agencies not located downtown, it is not obvious to see how assumptions about validity and completeness can be combined to characterise what parts of the answer are guaranteed to be either valid or complete.

For that purpose a formal modal logic has been defined in [Dem97] which is briefly recalled in section 2. In section 3 we analyse different forms of answers about validity and completeness, and circumstances where they are useful. Finally, in section 4 is presented an automated deduction technique that has been implemented to compute that kinds of answers.

2 Background on validity and completeness of data

In [Dem97] we have defined a modal logic (see [Che88]) that has been used to give a formal definition to the fact that a given subset of a database is reliable either in regard to validity or in regard to completeness of data. In this section we briefly recall these definitions.

In the axiomatic definition of our logic we have all the axiom schema of classical propositional calculus, and the inference rule Modus Ponens.

To characterise database content we use the modal operator B, and sentences of the kind Bp, where p is a sentence of a classical propositional calculus language, intuitively mean that p is a consequence of sentences stored in the database. The axiomatic of B is defined by the following axiom schemas:

$(K1)$ $B(p \rightarrow q) \rightarrow (Bp \rightarrow Bq)$

$(D1)$ $\neg((Bp) \wedge (B(\neg p)))$

and inference rule of necessitation (Nec) $\frac{\vdash p}{\vdash Bp}$.

To characterise what the system which manages the database believes, we have introduced another modal operator K, and sentences of the kind Kp intuitively mean that the system "strongly believes" p. We do not say that the system knows p because we consider that we are in a context where no information can be taken as true in an absolute sense. However, we want to be able to make a distinction between beliefs, for which we accept that they may be wrong beliefs, and strong beliefs for which we have the same behaviour as for knowledge. That is, strong beliefs are irrevocably considered as true beliefs. This intuitive meaning is formally expressed by axiom schema (T'). The axiomatic for K is defined by the following axiom schemas:

$(K2)\quad K(p \to q) \to (Kp \to Kq)$

$(D2)\quad \neg((Kp) \wedge (K(\neg p)))$

$(T')\quad K(Kp \to p)$

and inference rule of necessitation $(Nec)\ \frac{\vdash p}{\vdash Kp}$.

The fact that strong beliefs are a particular kind of beliefs is expressed by the axiom schema:

$(KB)\quad Kp \to Bp$

Finally, we have two axiom schemas (OBS1) and (OBS2) to express the fact that the database system has a **complete** knowledge of database content. That is, he knows what is, and what is not, in the database.

$(OBS1)\quad Bp \to K(Bp)$

$(OBS2)\quad \neg Bp \to K(\neg Bp)$

Notice that these two axioms do not imply that database content is a complete representation of the world.

In the following we extend the language used to represent database content from propositional calculus to first order calculus. However, we accept the Domain Closure Axiom, as it is usually the case in database context, and quantified formulas can be seen as ground conjunctions or ground disjunctions. Then, from a theoretical point of view, we are in fact in the field of propositional calculus.

The two modal operators we have presented in previous paragraph can be used to define the notions of reliable data in regard to validity or in regard to completeness. Sentence $RV(p(x))$ means that the database system strongly believes that every sentence of the form $p(x)$ which is believed by the database is true in the world. Sentence $RC(p(x))$ means that the database system strongly believes that every sentence of the form $p(x)$ which is true in the world is believed by the database. In formal terms we have:

$$\mathrm{RV(p(x))} \overset{\mathrm{def}}{=} \mathrm{K(\forall x(Bp(x) \rightarrow p(x)))}$$

$$\mathrm{RC(p(x))} \overset{\mathrm{def}}{=} \mathrm{K(\forall x(p(x) \rightarrow Bp(x)))}$$

A given database state db is represented by a set of first order formulas, and a set of assumptions mdb about subsets of the database that are reliable for validity or completenes is represented by a set of sentences of the form $RV(p(x))$ or $RC(p(x))$. From db we define dbb that represents database content in terms of beliefs. We have:

$dbb = \{Bp \;:\; \vdash db \rightarrow p\} \cup \{\neg Bp \;:\; \nvdash db \rightarrow p\}$

3 Different sorts of answers

From a formal point view we can define several kinds of answers that inform users about parts of the answers that are either valid or complete. In this section we first present these formal definitions, and then we analyse which ones are really useful for users.

3.1 Formal definitions of answers

Standard answers

The standard answer s to query $q(x)$ is defined by:

$$s = \{a \;:\; \vdash dbb \rightarrow Bq(a)\}$$

The standard answer is the set of individuals such that the database believes that they satisfy $q(x)$.

Extensional answer about validity

The extensional answer about validity ev to query $q(x)$ is defined by:

$$ev = \{a \;:\; \vdash mdb \wedge dbb \rightarrow Kq(a)\}$$

The answer ev is the set of individuals such that the database strongly believes that they satisfy $q(x)$.

Intensional answer about validity

The intensional answer about validity iv to query $q(x)$ is defined by:

$$\begin{aligned} iv = \{q'(x) \ : \ & \vdash mdb \rightarrow RV(q'(x))\} \text{ and} \\ & \vdash \forall x(q'(x) \rightarrow q(x)) \text{ and} \\ & q'(x) \text{ is maximal for implication}\} \end{aligned}$$

The answer iv is a set of sentences that characterise valid parts of the standard answer in terms of properties instead of a characterisation in terms of a set of individuals.

The condition "$q'(x)$ is maximal for implication" could be reformulated in more formal terms, it means that for every $q'(x)$ in iv there is no other sentence $q''(x)$ in iv such that $q''(x)$ logically implies $q'(x)$.

If $q'(x)$ is in iv, by definition of RV we have (1) $\forall x K(Bq'(x) \rightarrow q'(x))$. Since we have $\vdash \forall x(q'(x) \rightarrow q(x))$ we also have $\vdash \forall x K(q'(x) \rightarrow q(x))$, and from (1) we can infer (2) $\forall x K(Bq'(x) \rightarrow q(x))$.

Then, if for some individual a we have $Bq'(a)$, by (OBS1) we have $KBq'(a)$, and from (2) we can infer $Kq(a)$, and from (KB) we infer $Bq(a)$. That means that a is in s and it is also in ev.

Intensional answer about completeness

The intensional answer about completeness $ic-$ to query $q(x)$ is defined by:

$$\begin{aligned} ic- = \{q'(x) \ : \ & \vdash mdb \rightarrow RC(q'(x))\} \text{ and} \\ & \vdash \forall x(q'(x) \rightarrow q(x)) \text{ and} \\ & q'(x) \text{ is maximal for implication}\} \end{aligned}$$

The answer $ic-$ gives an intensional characterisation of subsets of the standard answer that are complete.

If $q'(x)$ is $ic-$, by definition of RC we have (1) $\forall x K(q'(x) \rightarrow Bq'(x))$, and from $\vdash \forall x(q'(x) \rightarrow q(x))$ we have (2) $\vdash \forall x(Bq'(x) \rightarrow Bq(x))$ and (3) $\vdash \forall x K(Bq'(x) \rightarrow Bq(x))$. Then from (1) and (3) we have (4) $\forall x K(q'(x) \rightarrow Bq(x))$, and, by contraposition we have (5) $\forall x K(\neg Bq(x) \rightarrow \neg q'(x))$.

Then, if for a given individual a we have $\neg Bq(a)$, from (OBS2) we have $K\neg Bq(a)$, and by (5) we have $K\neg q'(a)$.

The intensional answer about completeness $ic+$ to query $q(x)$ is defined by:

$$\begin{aligned} ic{+} = \{q'(x) \ :\ & \vdash mdb \rightarrow RC(q'(x))\} \text{ and} \\ & \vdash \forall x(q(x) \rightarrow q'(x)) \text{ and} \\ & q'(x) \text{ is minimal for implication}\} \end{aligned}$$

The answer $ic+$ gives an intensional characterisation of supersets of the standard answer that are complete.

If $q'(x)$ is in $ic+$, by definition of RC we have (1) $\forall x K(q'(x) \rightarrow Bq'(x))$, and from $\vdash \forall x(q(x) \rightarrow q'(x))$ we have (2) $\vdash \forall x K(q(x) \rightarrow q'(x))$. Then, from (1) and (2) we have (3) $\forall x K(q(x) \rightarrow Bq'(x))$, and, by contraposition, we have (4) $\forall x K(\neg Bq'(x) \rightarrow \neg q(x))$.

Then, if for a given individual we have $\neg Bq'(a)$, from (OBS2) we have $K\neg Bq'(a)$, and by (4) we have $K\neg q(a)$.

Extensional answer about completeness

The extensional answers about completeness $ec-$ and $ec+$ give the extension of intensional answers about completeness. Their definitions are:

$$ec- = \{a \ :\ \vdash dbb \rightarrow Bq'(a)\} \text{ and } q'(x) \text{ is in } ic-\}$$

$$ec+ = \{a \ :\ \vdash dbb \rightarrow Bq'(a)\} \text{ and } q'(x) \text{ is in } ic+\ \}$$

3.2 Useful answers

Validity answers

There is no doubt about usefulness of extensional answers about validity.

For intensional answers we have to analyse the point more carefully. Let's consider an example where predicate "$agency(x)$" means that x is a bank agency, "$open(x)$" means that the agency x is open, and "$bank(x, y)$" means that x is an agency of bank y. We consider a user who is looking for an open agency and asks the query $q_1(x)$ below:

$q_1(x) = agency(x) \wedge open(x)$

which means: *what are open bank agencies?*. Suppose that in the intensional answer about validity iv we have $q'_1(x)$ below:

$q'_1(x) = agency(x) \wedge open(x) \wedge bank(x, BE)$

which means: *the answer is valid for all the agencies of Bank of Europe.* In fact, as we have shown in previous paragraph, we have $Bq'_1(a)$ implies

$Kq_1(a)$, which means that the answer is guaranteed valid not for agencies that **are** agencies of the Bank of Europe, but for agencies such that the database **believes that they are** agencies of the Bank of Europe. So, if a user ignores what are these latter agencies stored in the database he cannot make use of the intensional answer. Except, if the database is complete for Bank of Europe agencies, and the user knows what are these agencies.

In more formal terms, if the intensional answer is of the form $q_1(x) \wedge cond(x)$, it is useful only if we have $RC(cond(x))$ and the user knows the extension of $Kcond(x)$.

Completeness answers

We first consider the extensional answers. If the standard answer to a query is $s = \{a, b, c, d, e, f\}$ and an extensional answer about completeness is $ec- = \{a, c, d\}$, what does it means that $ec-$ is complete? In fact the notion of completeness for a set is defined with respect to another set. Here, completeness means that for some intensional answer $q'(x)$ in $ic-$, all the elements that are guaranteed to satisfy $q'(x)$, in the sense $Kq'(x)$, are in $ec-$. So, if the user ignores what is the sentence $q'(x)$, he cannot know with respect to what $ec-$ is complete.

For instance, for query $q_1(x)$ we may have in $ic-$ the sentence $q_1''(x) = q_1(x) \wedge downtown(x)$. In that case, $ec-$ is useful only if the user knows that $ec-$ contains all the open agencies that are downtown.

The conclusion here is that extensional answers about completeness have to be completed with corresponding intensional answers about completeness. The same conclusion holds for answers of the kind $ec+$.

Let's analyse now usefulness of intensional answers.

We consider the same query $q_1(x)$ and we assume that $q_1'(x)$ is in $ic-$. From formal results shown in previous paragraph, if there is an agency, for instance g, which is not in the standard answer, that is such that we have $\neg Bq_1(g)$, then we have $K\neg q_1'(g)$. That means that for g user is guaranteed that it is not an open agency of the Bank of Europe. This information alone is not really useful, because it may be that g is not open, or that g is not an agency of the Bank of Europe. However, if in addition user knows that g is an agency of the Bank of Europe, then he can infer that g is not open.

In general, if $q_1'(x) = q_1(x) \wedge cond(x)$ is in $ic-$, for an individual a which is not in s (i.e. $\neg Bq_1(a)$), if the user knows that $cond(a)$ is true , he is guaranteed that a does not satisfy $q_1(x)$, in the sense $K\neg q_1(a)$.

Let's consider now the query $q_2(x) = agency(x) \wedge open(x) \wedge bank(x, BE)$. If we have in $ic+$ the sentence $q_2'(x) = agency(x) \wedge open(x)$, which guarantees that the database is complete for all the open agencies, from previous formal results, if some agency g is not in s, and the answer to the query $q_2'(g)$ is "no" (i.e. $\neg Bq_2'(g)$), then user is guaranteed that g does not satisfy $q_2(x)$ (i.e. $K\neg q_2(g)$). Here again this information is useful only if user knows that g is an agency of Bank of Europe.

Notice that the different kinds of answers about completeness allow to infer reliable negative information.

4 Automated deduction technique to compute answers

SOL-deduction

In this section is presented the automated deduction technique called SOL-deduction which is used to compute several sorts of answers. This technique has been defined for first order classical logic, but we shall see how it can be used for the modal logic presented in previous sections.

SOL-deduction is based on SOL-resolution, an inference rule designed by K. Inoue [Ino91, Ino92a, Ino92b] to generate logical consequences of a first order theory represented by a set of clauses Σ. One of the most important features of this inference rule is that it allows to concentrate consequence generation on clauses that satisfy some given property. For instance, to generate clauses formed only with positive literals, or clauses whose number of literals is less than or equal to a given fixed value.

We have no room here to give a formal presentation of SOL-resolution, so it will be presented through semi-formal definitions and examples (formal definitions can be found in [Ino92a]).

The property that can be used to restrict consequence generation has to characterise a "stable production field". A "production field" P is a pair $< L, Cond >$, where L is a subset of the literals of a given first order language, and $Cond$ is a certain property to be satisfied by clauses. For instance L may be the set of literals formed with two given predicate symbols. A production field is stable if for two any clauses C and D such that C subsumes D, D belongs to P only if C belongs to P.

We can check that the production field mentionned before is stable.

Indeed, if D only contains the two fixed predicate symbols, and C subsumes D, then C also only contains these predicate symbols. At the opposite if the property would be that a clause contains at least one literal formed with a given predicate symbol, then the production field would not be stable.

To define SOL-derivations K. Inoue introduces the notion of "structured clause". A structured clause is a pair $< P, \vec{Q} >$, where P is a set of literals and $\vec{Q}$ is an ordered set of literals. SOL-derivations are lists of structured clauses that start with a clause of the form $< \Box, \vec{C} >$ and end with a clause of the form $< S, \Box >$, and such that $< P_{i+1}, \vec{Q}_{i+1} >$ can be generated from $< P_i, \vec{Q}_i >$ by applying one the of the following rules:

a) if l is the leftmost literal in $\vec{Q}_i$:
1) if $P_i \cup \{l\}$ is in the production field P, then P_{i+1} is $P_i \cup \{l\}$
and $\vec{R}_{i+1}$ is obtained by removing l from $\vec{Q}_i$,
2) if there is a clause B_i in $\Sigma \cup \{C\}$ that contains
a literal which can be resolved with l with mgu θ, then
P_{i+1} is $P_i\theta$, and $\vec{R}_{i+1}$ is obtained by concatenating $\vec{B}_i\theta$
to $\vec{Q}_i\theta$,removing the reduced literal in $B_i\theta$, and framing $l\theta$,
3) if
i) P_i or $\vec{Q}_i$ contains an unframed literal k different from l
or another occurence of l, or
ii) $\vec{Q}_i$ contains a framed literal $\neg k$,
and l and k are unifiable by mgu θ,
then $P_{i+1} = P_i\theta$ and $\vec{R}_{i+1}$ is obtained from
$\vec{Q}_i\theta$ by deleting $l\theta$,
b) $\vec{Q}_{i+1}$ is obtained from $\vec{R}_{i+1}$ by deleting every
framed literal not preceeded by an unframed literal.

The intuitive idea is to isolate in the part P of structured clauses the subset of the literals in a standard clause that satisfies property $Cond$. At the end of a derivation, since part $\vec{Q}$ is empty we obtain a consequence P that satisfies $Cond$.

Application of SOL-deduction to compute answers

We first consider computation of intensional answers about validity. We are looking for formulas $q'(x)$ such that $\vdash mdb \rightarrow RV(q'(x))$, and $q'(x)$ implies $q(x)$, and $q'(x)$ is maximal for implication.

Since, for every sentence in mdb and for $RV(q'(x))$, sentences are in the scope of the modal operator K, in virtue of properties of K we can remove these occurences of K. Let's call mdb' the set of sentences obtained from mdb after removing K. Then, the problem is equivalent to find $q'(x)$ such that:

$$\vdash mdb' \rightarrow \forall x(Bq'(x) \rightarrow q'(x))$$

where mdb' is a set of sentences of the form $\forall x(Bf(x) \rightarrow f(x))$ or $\forall x(g(x) \rightarrow Bg(x))$ [1].

Then, the problem is to find sentences $f_1, \ldots, f_i$ and $g_1, \ldots, g_j$ such that $\forall x(Bf_1(x) \rightarrow f_1(x))$, ... , $\forall x(Bf_i(x) \rightarrow f_i(x))$, $\forall x(g_1(x) \rightarrow Bg_1(x))$, ... , $\forall x(g_j(x) \rightarrow Bg_j(x))$ are in mdb', and we have:

$\vdash \forall x((f_1(x) \wedge \ldots \wedge f_i(x) \wedge \neg g_1(x) \wedge \ldots \wedge \neg g_j(x)) \rightarrow q(x))$

Indeed, in that case we have:

$\vdash \forall x(Bf_1(x) \wedge \ldots \wedge Bf_i(x) \wedge \neg Bg_1(x) \wedge \ldots \wedge \neg Bg_j(x) \rightarrow q(x))$

So, if we accept the Closed World Assumption, as people do in the context of Relational databases when answers are computed by using relational algebra, or SQL like languages, we have $\neg Bg$ equivalent to $B\neg g$, and therefore we have:

$\vdash \forall x(Bf_1(x) \wedge \ldots \wedge Bf_i(x) \wedge B\neg g_1(x) \wedge \ldots \wedge B\neg g_j(x) \rightarrow q(x))$

and

$\vdash \forall x(B(f_1(x) \wedge \ldots \wedge f_i(x) \wedge \neg g_1(x) \wedge \ldots \wedge \neg g_j(x)) \rightarrow q(x))$

That means that $q'(x) = f_1(x) \wedge \ldots \wedge f_i(x) \wedge \neg g_1(x) \wedge \ldots \wedge \neg g_j(x)$ is an intensional answer about validity.

To use SOL-deduction we first consider modal operator B as a first order predicate $believes(x)$, and we transform sentences into clauses. Let the clausal form of $f(x)$ and $\neg g(x)$ be defined by:

$\vdash \forall x(f(x) \leftrightarrow c_1(x) \wedge \ldots c_s(x))$ $\vdash \forall x(\neg g(x) \leftrightarrow d_1(x) \wedge \ldots \wedge d_t(x))$

where the c_is and d_js are clauses. We define Σ as the following set of clauses:

- if $\forall x(Bf(x) \rightarrow f(x))$ is in mdb', the set of clauses
$\neg believes(f(x)) \vee c_1(x)$, ... ,$\neg believes(f(x)) \vee c_s(x)$ is in Σ,

- if $\forall x(g(x) \rightarrow Bg(x))$ is in mdb', the set of clauses
$d_1(x) \vee believes(g(x))$, ... ,$d_t(x) \vee believes(g(x))$ is in Σ,

and there is no other clause in Σ.

---

[1] Notice that $\forall x(g(x) \rightarrow Bg(x))$ is equivalent to $\forall x(\neg Bg(x) \rightarrow \neg g(x))$.

We consider the production field where L is the set of literals formed with the predicate $believes(x)$, and the property $Cond$ is that every literal in a clause is formed with the predicate $believes(x)$. We can easily check that this production field is stable.

For the top clause $< \Box, \vec{C} >$ we have $C = \neg q(x_0)$, where x_0 is a Skolem constant.

If, at the end of a SOL-deduction we get the clause $< D, \Box >$, the clause D is of the form $\neg believes(f_1(x_0)) \vee \ldots \vee \neg believes(f_i(x_0)) \vee believes(g_1(x_0)) \vee \ldots \vee believes(g_i(x_0))$, and we have:

$\Sigma \vdash \forall x(believes(f_1(x_0)) \wedge \ldots \wedge believes(f_i(x_0)) \wedge \neg believes(g_1(x_0)) \wedge \ldots \wedge \neg believes(g_i(x_0)) \rightarrow q(x))$

Moreover, if consequences obtained by SOL-deduction which are subsumed by other consequences are removed, the intensional anwer we get is maximal for implication.

Now, if we consider the computation of extensional answers about validity, we just have to compute the answer to the following query $q'(x)$:

$q'(x) = f_1(x) \wedge \ldots \wedge f_i(x) \wedge \neg g_1(x) \wedge \ldots \wedge \neg g_j(x)$

For the computation of intensional answers about completeness $ci+$ that are a super set of the query, we can use SOL-deduction in a similar way.

The property $\vdash mdb \rightarrow RC(q'(x))$ is reformulated into $\vdash mdb' \rightarrow \forall x(q'(x) \rightarrow Bq'(x))$, which is equivalent to $\vdash mdb' \rightarrow \forall x(\neg Bq'(x) \rightarrow \neg q'(x))$. We also have property $\vdash \forall x(q(x) \rightarrow q'(x))$ which is equivalent to $\vdash \forall x(\neg q'(x) \rightarrow \neg q(x))$. Then, we have to find sentences $q'(x)$ such that $\vdash mdb' \rightarrow \forall x(\neg Bq'(x) \rightarrow \neg q(x))$. Here again, if we accept that $\neg Bg$ is equivalent to $B\neg g$, the property is equivalent to $\vdash mdb' \rightarrow \forall x(B\neg q'(x) \rightarrow \neg q(x))$, which is of the same form as property that characterises intensional answers about validity and can be computed in the same way from the top clause $C = q(x_0)$.

5 Conclusion

We have shown that not every kind of answer is useful for users. Extensional answers about validity ev are always useful. Usefulness of intensional answers depends on what user knows, and extensional answers about completeness are useless if they are not completed by their intensional correspondants.

A prototype of SOL-deduction has been implemented by Laurence Cholvy [CD97a, CD97b] which we have used to test our method on several examples. Response times range from 0.130s to 2.450 s. For intensional answer of the kind $ic-$ we cannot use the method presented in section 4 and we have to investigate this issue in future works.

Acknowledgements. This work has been supported by *Centre National d'Etudes des Télécommunications (CNET)* in the context of contract CNET/ONERA-CERT 96 1B 317.

References

[CD97a] L. Cholvy and R. Demolombe. Génération de réponses non-standard dans le contexte d'un dialogue coopératif. Phase 1.2 Définition et caractérisation des procédures élémentaires et génériques permettant de calculer les réponses coopératives. Technical Report 2, ONERA-CERT-DTIM, 1997.

[CD97b] L. Cholvy and R. Demolombe. Génération de réponses non-standard dans le contexte d'un dialogue coopératif. Phase 1.3 Mise en oeuvre des procédures élaborées lors de la phase 1.2. Technical Report 3, ONERA-CERT-DTIM, 1997.

[CDJ94] L. Cholvy, R. Demolombe, and A.J.I. Jones. Reasoning about the safety of information: from logical formalization to operational definition. In *Proc. of 8th International Symposium on Methodologies for Intelligent Systems*, 1994.

[Che88] B. F. Chellas. *Modal Logic: An introduction.* Cambridge University Press, 1988.

[Dem96] R. Demolombe. Validity Queries and Completeness Queries. In *Proc. of 9th International Symposium on Methodologies for Intelligent Systems*, 1996.

[Dem97] R. Demolombe. Answering queries about validity and completeness of data: from modal logic to relational algebra. In T. Andreasen, H. Christiansen, and H. L. Larsen, editors, *Flexible Query Answering Systems.* Kluwer Academic Publishers, 1997.

[DJ94] R. Demolombe and A.J.I. Jones. Deriving answers to safety queries. In R. Demolombe and T. Imielinski, editors, *Nonstandard queries and nonstandard answers*. Oxford University Press, 1994.

[Ino91] K. Inoue. Consequence-Finding Based on Oredered Linear Resolution. In *Proc. of International Joint Conference on Artificial Intelligence*, Sydney, 1991.

[Ino92a] K. Inoue. Linear Resolution for Consequence Finding. *Artificial intelligence, an International Journal*, 56, 1992.

[Ino92b] K. Inoue. *Studies on Abductive and Nonmonotonic Reasoning*. PhD thesis, Kyoto University, 1992.

[Mot86] A. Motro. Completeness information and its application to query processing. In *Proc. of 12th International Conference on Very Large Data Bases*, 1986.

[Mot89] A. Motro. Integrity = validity + completeness. *ACM TODS*, 14(4), 1989.