Toward a uniform logical representation of different kinds of integrity constraints

Robert Demolombe
CERT/ONERA
Toulouse

Andrew J.I. Jones
Department of Philosophy and
Norwegian Research Centre for
Computers and Law
University of Oslo *

Jose Carmo
University of Lisbon

1 Introduction

There is no agreement in the litterature about the definition of integrity constraints in the context of data and/or knowledge bases. What kind of property people want to guarantee using integrity constraints? Is it the property of consistency of database content or the completeness of database content [6] or validity or completeness of database content with regard to the world [3] or some properties that should hold in the world of a given application domain [1] There is no unique answer to this question. It depends on the intention of users. However, even if we accept that integrity constraints are intended to impose constraints about different kinds of properties, it would be interesting to know what are the differences and common features, and what are the implicit assumptions for each kind of constraints.

The objective of this extended abstract is to present a first proposal in this direction. Since its objective is only to have a better understanding of integrity constraints it does not consider specific techniques that can be used to efficiently check whether constraints are violated [4] or to efficiently repair violations [5, 7].

To have a uniform representation of the different kinds of constraints we have adopted a logical framework. In this framework database content is represented by a set db of

*In sabbatical at University of Lisbon

formulas in a language of classical first order logic. The world, which is supposed to be represented by the database, is represented itself by a set w of formulas of the same language. In the following, we call the "world" a correct representation of the world in the language used for a given application domain.

To be able to make the distinction between a sentence p that represents part of database content, and the same sentence p that represents part of the world we use a doxastic logic, and the fact that p is part of the database content is represented by sentence B(p), which can be read: "the database believes p". The representation of database content in this doxastic logic is a set of sentences dbb defined from db in the following way:

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

We adopt for the modality B the modal system (KD) [2].

Since constraints are normative statements we also need to be able to make the distinction between the fact that p is the case and the fact that p should be the case. For this purpose we consider a deontic logic and sentences of the form O(p) can be read: "it should be the case that p" or "it oughts to be that p". There are many proposals for the formalisation of deontic modality O (see for example [1, 8], in this paper we do not elaborate about the choice of the appropriate deontic logic and, as a matter of simplification, we can adopt standard deontic logic SDL [2].

For each different kind of constraint we shall consider the definition of the property that should hold, the kind of sentences that are assumed to be true of the world, the characterisation of violation of constraints, and the secondary obligations, that is the kind of actions that should be performed to repair violations.

We shall not consider here constraints about the consistency of the set of database beliefs because it is not a matter of discussion that database content should be consistent.

2 Constraints about the world

To illustrate this kind of constraint let us consider the context of library management in a research laboratory. In this context managers want to control situations happening in the world. For instance, it may happen that members of the department leave the department without returning books they have borrowed. To prevent such situations library managers may explicitly characterize ideal situations by the constraint:

(1) Books can be borrowed only by members of the department.

which can be formally represented by:

(1') $\forall x \forall y O(\text{borrowed}(x, y) \rightarrow \text{dept}(x))$

A database could here be used as a means for detecting possible violations of this constraint. That is possible only if it is guaranteed that database content is a correct representation of the world, in the sense that if the database believes that some member a of the department has borrowed a book b then this fact is true in the world (inft. $B(\text{borrowed}(a, b)) \rightarrow \text{borrowed}(a, b)$), and if it is true in the world that a is member of the department then this fact is represented in the database (inft. $\text{dept}(a) \rightarrow B(\text{dept}(a))$) [1].

The property $B(\text{borrowed}(a, b)) \rightarrow \text{borrowed}(a, b)$ is called "validity" of database content in regard to the fact borrowed(a,b), and, in general, we call validity of database content in regard to p the property: $Bp \rightarrow p$. In a similar way, $\text{dept}(a) \rightarrow B(\text{dept}(a))$ is called "completeness" of database content in regard to the fact dept(a), and, in general, we call completeness of database content in regard to q the property: $q \rightarrow Bq$.

Now, on the basis of the assumption of validity of the database for the fact borrowed(a,b) and of completeness for the fact dept(a), if the database is in a state where it believes that a has borrowed book b and it does not believe that a is member of the department (inft. $B(\text{borrowed}(a, b)) \wedge \neg B(\text{dept}(a))$), we can infer that it is the case in the world that a has borrowed b and a is not member of the department (inft. $\text{borrowed}(a, b) \wedge \neg \text{dept}(a)$), which is a situation that violates constraint (1).

In general, if we have $Bp \wedge \neg Bq$, from assumptions $Bp \rightarrow p$ and $q \rightarrow Bq$, we can infer $p \wedge \neg q$, which characterizes a violation of a constraint of the form $O(p \rightarrow q)$.

This can be informally represented by Figure 1 (thin arrow represent implications that should hold, while thick arrows represent implications that are assumed to hold).

To repair a violation of constraint (1) secondary obligations are imposed. They may be, for instance, to send a letter to department member a, asking him to return borrowed books. At the moment books have been returned the fact borrowed(a,b) is no longer true, and since database content is supposed to be valid in regard to this fact, it has to be removed from the database. However, in the mean time, it is accepted to have $B(\text{borrowed}(a, b)) \wedge \neg B(\text{dept}(a))$.

[1] Along the paper we use "inft." as an abbreviation for "in formal terms".

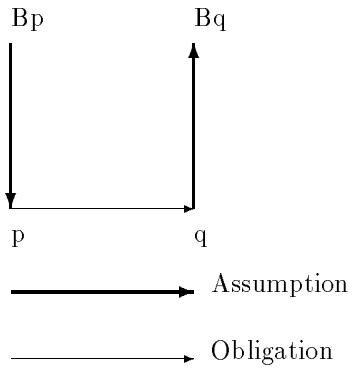

Figure 1: Constraints about the the world.

3 Constraints about the links between database content and the world

A constraint requiring validity of the database content must surely hold generally, for all facts, since it would be nonsense to consider acceptable a database whose content was false of the world - excluding here, of course, the possibility that a database might be designed to serve the function of deceiving its users.

Constraints about completeness, on the other hand, will ordinarily be optional, depending on the needs of the user. Thus a company might require that its database must be complete with respect to the age of each employee, but readily accept incompleteness with respect to a number of other facts about its employees.

To check whether constraints regarding validity and completeness have been violated, we need to assume that some database beliefs are true beliefs - that is, that they are true of the world. In many cases this assumption concerns generalizations of which it may reasonably be supposed that they have no counter-instances. For instance, it may be reasonable to assume that the following rule is always true, and in that sense cannot be violated:

(2) Only members of the department have a salary from the department.

which can be formally represented by:

(2') $\forall x((\exists y\ \text{salary}(x, y)) \rightarrow \text{dept}(x))$

Then, if we have constraints about validity of facts of type $\exists y\ \text{salary}(x, y)$, and about

completeness of facts of type dept(x), violations of these constraints may be detected by reference to (2'). For example, if the database believes that a has a salary, and it believes that a is not a member of the department (inft. $B(\exists y\ \text{salary}(a,y)) \wedge \neg B(\text{dept}(a))$), then we can infer that there is a violation either of the validity constraint: $O(B(\exists y\ \text{salary}(a,y)) \rightarrow (\exists y\ \text{salary}(a,y))$, or of the completeness constraint: $O(\text{dept}(a) \rightarrow B(\text{dept}(a))$. Indeed, from (2') we have: $\neg(\exists y\ \text{salary}(a,y)) \vee \text{dept}(a)$, and this entails: $(\neg(\exists y\ \text{salary}(a,y)) \wedge B(\exists y\ \text{salary}(a,y)) \vee (\text{dept}(a) \wedge \neg B(\text{dept}(a)))$.

In general, if we have constraints about the validity of p and about the completeness of q, that is: $O(Bp \rightarrow p)$ and $O(q \rightarrow Bq)$, and if it is assumed that $p \rightarrow q$ is always true in the world, a violation of at least one of these two constraints is characterised by a situation where we have: $Bp \wedge \neg Bq$.

This can be informally represented by Figure 2.

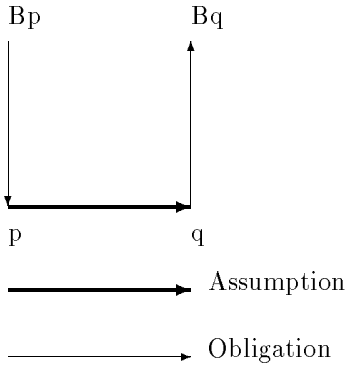

Figure 2: Constraints about the links between data base content and the world.

In that case, to repair a violation of $O(Bp \rightarrow p)$ or $O(q \rightarrow Bq)$, secondary obligations will require (respectively) either deletion of p from the database or insertion of q into the database. Here, in contrast to the case of constraints about the world, the situation where we have $Bp \wedge \neg Bq$ cannot be accepted, and secondary obligations require not change to the world, but change to the database.

4 Constraints about database content

A particular species of completeness constraint (cf. [6]) is worthy of special attention. Supposing, again, the library scenario, it may be that, in fact, the following generalization is true: every departmental member is either a professor or a student. And it may then well be the case that the library manager wants his classification of the departmental

members to be complete with respect to the categories of student and professor. (Perhaps lending entitlements are dependent on this classification, for instance.)

Thus the library database will be subject to the constraint:

(3) If the database believes that some x is a member of department, then the database should believe that x is a student or it should believe that x is a professor

which can be represented in formal terms by:

(3') $\forall x O(B(dept(x)) \rightarrow B(student(x)) v B(professor(x)))$

Another example of this type of constraint would be:

(4) If the database believes that some x is a member of department, then the database should be aware of the address of x

which can be represented in formal terms by:

(4') $\forall x O(B(dept(x) \rightarrow \exists y B(address(x, y)))$

The general forms of the constraints exhibited by (3') and (4') - cf. Reiter - are: $O(Bp \rightarrow Bq \vee Br)$ and $O(Bp \rightarrow \exists x Bq(x))$.

The process of checking whether a constraint of these general kinds is violated does not itself involve an assumption to the effect that some beliefs are true beliefs; it is sufficient to consider just the set of database beliefs. For instance, considering the first of these two general forms, if the database content is such that we have: $Bp \wedge \neg Bq \wedge \neg Br$, then it is clear that the constraint $O(Bp \rightarrow Bq \vee Br)$ has been violated. Obviously, to repair the state of violation, the appropriate secondary obligation would require either the deletion of the sentence p, or the retention of p and the addition of either the sentence q or the sentence r. And which choice is made here will, of course, depend on what is true in the world.

In general, acceptance of a constraint of the form $O(Bp \rightarrow Bq \vee Br)$ implicitly assumes that the sentence $p \rightarrow q \vee r$ is true in the actual world.

This can be informally represented by Figure 3.

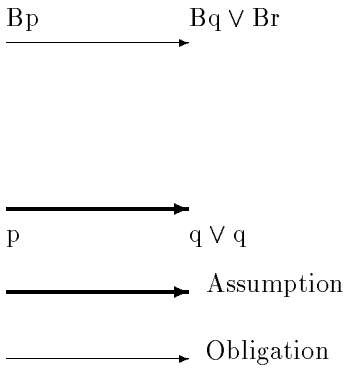

Figure 3: Constraints database containt.

5 Conclusion

We have outlined a logical framework for the representation of some species of database integrity constraints. In each case, the constraints were represented as obligation sentences. For examples of type (1'), the obligation sentence pertains to how things ought to be in the actual world, whereas in examples of what we have called validity constraints and completeness constraints (including those constraints exhibited by (3') and (4')) the obligation sentences pertain to how things ought to be in the database.

We have outlined a logical framework for the representation of some species of database integrity constraints. In each case, the constraints were represented as obligation sentences. For examples of type (1'), the obligation sentence pertains to how things ought to be in the actual world, whereas in examples of what we have called validity constraints and completeness constraints (including those constraints exhibited by (3') and (4')) the obligation sentences pertain to how things ought to be in the database.

Since all of these integrity constraints are represented as obligation sentences, they are all are violable; but there is an important distinction between the kind of response taken to violation of constraints of type (1'), and the kind of response taken to violation of the other types. For when a constraint of type (1') is violated, we tolerate the fact that this state of violation is represented in the database, but recovery procedures will be designed to change the state of the world so that the violation is repaired (and then the database must also be changed accordingly: the borrowed books, say, are finally returned, and so we delete from the database the sentence saying that books are on loan to a non-member). But when a violation of the other kinds of integrity constraints occurs, it is the resulting database state which is not tolerable, and must be changed, either because the information registered in the database is not true, or because it is

incomplete.

The discussion in [1] of what they (following Sergot) called "soft" integrity constraints, was primarily focussed on examples of type (1'). What they called "hard" constraints are not deontic constraints of the kind considered in this abstract, but rather necessary constraints which no database state will ever be allowed to violate. For instance, a constraint to the effect that no person is both male and female may be a "hard" constraint for a given database in the sense that the database will never be allowed to register some person x as both male and female.

In future work we plan to develop further this attempted classification of types of database constraints, reflecting (as we have done here) some differences between types of violable constraints, but incorporating in addition those constraints which - as far as the database is concerned - are deemed inviolable, necessary truths.

References

[1] J. Carmo and A.J.I. Jones. Deontic Database Constraints and the Characterisation of Recovery. In A.J.I.Jones and M.Sergot, editors, *2nd Int. Workshop on Deontic Logic in Computer Science*, pages 56–85. Tano A.S., 1994.

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

[3] R. Demolombe and A.J.I. Jones. Integrity Constraints Revisited. *Journal of the Interest Group in Pure and Applied Logics*, 4(3), 1996.

[4] J-M. Nicolas and K. Yazdanian. Integrity checking in Deductive Databases. In H. Gallaire and J. Minker, editors, *Logic and Databases*. Plenum, 1982.

[5] A. Olivé. Integrity checking in Deductive Databases. In *17th Int. Conf. on Very Large Data Bases*, 1991.

[6] R. Reiter. What Should a Database Know? *Journal of Logic Programming*, 14(2,3), 1992.

[7] E. Teniente and A. Olivé. The Events Method for View Updating in Deductive Databases. In *Int. Conf. on Extending Data Base Technology*, 1992.

[8] R.J. Wieringa and J-J. Meyer. Applications of Deontic Logic in Computer Science: a concise overview. In J-J. Meyer and R.J. Wieringa, editors, *Proc. First Int. Workshop on Deontic Logic in Computer Science*, 1991.