ON INTEGRITY CONSTRAINTS

Raymond Reiter†
Department of Computer Science
University of Toronto
Toronto, Ontario M5S 1A4
Canada
e-mail: reiter@toronto.csnet

Abstract

We address the concept of a static integrity constraint as it arises in databases and Artificial Intelligence knowledge representation languages. Such constraints are meant to characterize the acceptable states of a knowledge base, and are used to enforce these legal states. We adopt the perspective that a knowledge base is a set of first order sentences, but argue, contrary to the prevailing view, that integrity constraints are epistemic in nature. Rather than being statements about the world, constraints are statements about what the knowledge base can be said to know.

We formalize this notion in the language KFOPCE due to Levesque and define the concept of a knowledge base satisfying its integrity constraints. We investigate constraint satisfaction for closed world knowledge bases. We also show that Levesque's axiomatization of KFOPCE provides the correct logic for reasoning about integrity constraints. Finally, we show how to determine whether a knowledge base satisfies its constraints for a restricted, but important class of knowledge bases and constraints.

† Fellow, Canadian Institute for Advanced Research.

1. INTRODUCTION

The concept of an integrity constraint arises in databases, and in AI knowledge representation languages. The basic idea is that only certain knowledge base states are considered acceptable, and an integrity constraint is meant to enforce these legal states.

Integrity constraints have two flavours - static and dynamic. The enforcement of a static constraint depends only on the current state of the knowledge base, independently of any of its prior states. The fact that every employee must have a social security number is an example of a static constraint. Dynamic constraints depend on two or more knowledge base states. For example, if employee salaries must never decrease, then in no future knowledge base state may an employee's salary be less than it is in the current state.

This paper is concerned exclusively with static constraints. We adopt the by now standard view that a knowledge base is a set of first order sentences. The conventional perspective on integrity constraints is that they too are first order sentences. We argue against this notion and propose instead that constraints are epistemic in nature; rather than being statements *about the world*, they are statements about what the knowledge base can be said to *know*. Thus the employee - social security number constraint says something like:

For each employee known to the knowledge base, there must also be known a social security number associated with that employee.

We formalize this notion in the language KFOPCE of Levesque (1981) and appeal to his ASK operator for defining the concept of a knowledge base satisfying its integrity constraints. We investigate this idea in the realm of closed world knowledge bases. We also show how Levesque's axiomatization of KFOPCE provides the correct logic for reasoning about integrity constraints. Finally, we provide a means for determining whether a knowledge base satisfies its constraints for a restricted, but important, class of knowledge bases and constraints.

2. WHAT IS AN INTEGRITY CONSTRAINT?

We adopt the prevailing view that a knowledge base is a set of a first order sentences.† The conventional perspective on integrity constraints is that they too are first order sentences (e.g. Lloyd & Topor (1985), Nicolas & Yazdanian (1978), Reiter (1984)). There are two definitions in the literature of a knowledge base *KB* satisfying an integrity constraint *IC*††:

Definition 1: Consistency (e.g. Kowalski (1978), Sadri and Kowalski (1987))

KB satisfies IC iff KB + IC is satisfiable.

Definition 2: Entailment (e.g. Lloyd and Topor (1985), Reiter (1984))

KB satisfies IC iff $KB \models IC$

† Later we shall be more specific about a choice of first order language.

†† There is a third definition specific to relational databases: a database *DB* satisfies *IC* iff *IC* is true in *DB* when *DB* is viewed as a model. We elaborate on this notion in Section 4, and relate it to the following two definitions.

Alas, neither definition correctly captures our intuitions. Consider the constraint about employees and their social security numbers:

$$(\forall x)\ emp\ (x) \supset (\exists\ y)\ ss\#\ (x,y) \tag{2.1}$$

1. Suppose KB = {emp (Mary)}. Then $KB + IC$ is satisfiable. But intuitively, we want the constraint to require KB to contain a ss# entry for Mary, so we want IC to be violated. Thus Definition 1 does not capture our intuitions.

2. Suppose KB = { }. Intuitively, this should satisfy IC, but $KB \not\models IC$. So Definition 2 is inappropriate.

An alternative definition comes to mind when one sees that constraints like (2.1) intuitively are interpreted as statements not about the world but about the *contents* of the knowledge base, or about what it *knows*. Thus, using the modal K for "knows", (2.1) should be rendered by:

$$(\forall x)\ K\ emp\ (x) \supset (\exists\ y)\ K\ ss\#\ (x,y)$$

Other Examples

1. To prevent a knowledge base from simultaneously assigning the properties *male* and *female* to the same individual, use the constraint

$$(\forall x)\ \neg K\ (male\ (x) \wedge female\ (x)).$$

2. To force a knowledge base to assign one of the properties *male* and *female* to each individual, use the constraint

$$(\forall x)\ K\ person\ (x) \supset K\ male\ (x) \vee K\ female\ (x).$$

3. To require that known instances of the relation *mother*(.,.) have first argument a *female person* and a second argument a *person*, use the constraint

$$(\forall x,y)\ K\ mother\ (x,y) \supset K\ (person\ (x) \wedge female\ (x) \wedge person\ (y)).$$

4. To require that every known employee have a social security number, without necessarily knowing what that number is, use

$$(\forall x)\ K\ emp(x) \supset K\ (\exists\ y\ )ss\#(x,y).$$

5. Functional dependencies in relational database theory are integrity constraints of a particular form. On our notion of a constraint, the functional dependency that social security numbers be unique would be represented by:

$$(\ \forall x,y,z\ )\ K\ ss\#\ (x,y)\ \wedge\ K\ ss\#\ (x,z)\ \supset\ K\ y=z$$

Many other kinds of dependencies have been investigated for relational databases. Most of these can be represented as first order sentences (Fagin 1980, Nicolas & Gallaire 1978). The corresponding modalized forms of these first order sentences provide the correct reading of these dependencies, at least on our account of integrity constraints.

The view that integrity constraints are statements about the *content* of a knowledge base also serves to clarify a certain confusion in the literature about the different roles played by constraints and knowledge base formulas. According to the conventional account, constraints and knowledge bases are both first order sentences. Since constraints are external to the knowledge base, they do not enter into the query evaluation process. Yet, as first order sentences, they must express truths about the world, no less so than the knowledge base itself. Why then should they not contribute to answering queries? There is no clear answer to this in the literature. Nicolas and Gallaire (1978) propose various pragmatic criteria for treating a formula as a constraint rather than as a component of the knowledge base, but there appear to be no general principles. On our account, no such principles are necessary. Truths about the world, namely first order sentences, belong in the knowledge base. Truths about the knowledge base, namely modalized sentences, function as integrity constraints.

It remains to specify the semantics of the K operator, and to formally define the notion of a knowledge base satisfying its integrity constraints. To do so, we appeal to a simple, but somewhat nonstandard, first order language FOPCE and its modalized form KFOPCE, both due to Levesque (1981).†

KFOPCE is a first-order modal language with equality and with a single modal operator K (for "know"), constructed in the usual way from a set of predicate and variable symbols and a countably infinite set of symbols called *parameters*. Predicate symbols take variables and parameters as their arguments. Parameters can be thought of as constants. Their distinguishing feature is that they are pairwise distinct and they define the domain over which quantifiers range, i.e. the parameters represent a single universal domain of discourse. FOPCE is the language KFOPCE without the modal K.

A database KB of information about a world is a set of FOPCE sentences. We consider how Levesque defines the result of querying KB with a sentence of KFOPCE. This requires first specifying a semantics for KFOPCE. A *primitive sentence* (of KFOPCE) is any atom of the form $P(p_1, \ldots, p_n)$, where P is an n-ary predicate symbol and $p_1, \ldots, p_n$ are parameters. A *world structure* is any set of primitive sentences that includes $p=p$ for each parameter p, and that does not include $p_1=p_2$ for different parameters p_1 and p_2. The effect of this requirement on the equality predicate is that semantically the parameters are all pairwise distinct. A world structure is understood to be a set of true atomic facts. A *structure* is any set of world structures. The truth value of a sentence of KFOPCE in a world structure W and a structure Σ is defined as follows:

1. If p is a primitive sentence, p is true in W and Σ iff $p \in W$.

2. $\neg w$ is true in W and Σ iff w is not true in W and Σ.

3. $w_1 \vee w_2$ is true in W and Σ iff w_1 or w_2 is true in W and Σ.

4. $(\forall x)\ w(x)$ is true in W and Σ iff for every parameter p, $w(p)$ is true in W and Σ.

5. Kw is true in W and Σ iff for every $S \in \Sigma$, w is true in S and Σ.

Notice that condition 4 implies that, insofar as KFOPCE is concerned, the parameters constitute a single universal domain of discourse. The parameters are used to identify the known individuals. Notice also that when f is a FOPCE sentence (so that condition 5 need never be invoked in the truth recursion for f) then the truth value of f in W and Σ is independent of Σ, and we can speak of the truth value of f in W alone.

† FOPCE and KFOPCE are the function-free special cases of Levesque's (1984) languages L and KL respectively. For simplicity of exposition, we have chosen in this paper to treat the function-free case. Generalization of our results to L and KL remains a future research topic.

When W is a world structure and Σ a set of world structures, the pair (W, Σ) is a *model* of a set of KFOPCE sentences iff each sentence is true in W and Σ. When S is a set of KFOPCE sentences, we write $S \models_{KFOPCE} w$ whenever the KFOPCE sentence w is true in all models of S. Similarly, when S is a set of FOPCE sentences and w is a FOPCE sentence, we write $S \models_{FOPCE} w$ with the obvious meaning.

Given this semantics, Levesque defines the result of querying KB, a set of FOPCE sentences, with an arbitrary sentence of KFOPCE as follows:

Let $M(KB)$ be the set of models of KB. The result of querying KB with a sentence k of KFOPCE is defined to be

ASK(KB, k) = *yes* if for all $W \in M(KB)$, k is true in W and $M(KB)$.
= *no* if for all $W \in M(KB)$, k is false in W and $M(KB)$.
= *unknown* otherwise.

We write $KB \Vdash k$ whenever ASK (KB, k) = yes.

Definitions

Henceforth, a *knowledge base* will be any set of sentences of FOPCE, and an *integrity constraint* will be any sentence of KFOPCE. We say that a knowledge base KB *satisfies* an integrity constraint IC iff $KB \Vdash IC$.

Levesque is concerned with querying a first order knowledge base using KFOPCE as a query language. In effect, our proposal is to understand integrity constraints as formally indistinguishable from queries, with the requirement that for any knowledge base state the answer to these queries must be "yes".

3. REASONING ABOUT INTEGRITY CONSTRAINTS

A natural question is this: What proof theory is appropriate for reasoning about integrity constraints? There are several reasons such a proof theory would be desirable. We might wish to determine whether certain constraints are redundant i.e. are entailed by the others. If the constraints are unsatisfiable, it would be important to know that. We might wish to simplify a given set of constraints in various ways, or explore their consequences.

The following is a simple consequence of the definitions of Section 2.

Theorem 3.1

If KB is a knowledge base and IC an integrity constraint then $KB \Vdash IC$ iff $KB \Vdash I$ for all sentences I of KFOPCE such that $IC \models_{KFOPCE} I$.

Thus, KB satisfies IC iff KB satisfies every KFOPCE consequence of IC. This means that KFOPCE is the appropriate logic for reasoning about integrity constraints. Levesque (1981) provides a sound and complete axiomatization for KFOPCE. Since our concern in this paper is with theoretical foundations for integrity constraints, we omit a description of Levesque's axiomatization. It is sufficient for our purposes to know that a suitable proof theory exists for reasoning about constraints.

4. CONSTRAINT SATISFACTION FOR CLOSED KNOWLEDGE BASES

Frequently, knowledge bases are treated as satisfying the closed world assumption (Reiter (1978), Lifschitz (1985)). The idea is that the knowledge base is viewed as completely representing all the positive information about some world. Any ground atomic fact not so represented is taken to be false. On one account of this assumption (Reiter 1978), we can define, for a set *KB* of FOPCE sentences,

$$\text{Closure}(KB) = KB \cup \{\neg P(p_1,\ldots,p_k) \mid P \text{ is a k-ary predicate symbol, the } p_i \text{ are parameters, and } KB \not\models_{FOPCE} P(p_1,\ldots,p_k)\}.$$

Under the closed world assumption about a given set of FOPCE sentences KB, it is closure (*KB*) which is taken to be the relevant knowledge base. Our concern in this section is how the closed world assumption affects integrity constraint satisfaction.

Theorem 4.1

If *KB* is a knowledge base and *IC* an integrity constraint, then

$$\text{Closure}(KB) \Vdash IC \text{ iff Closure}(KB) \models_{FOPCE} \overline{IC}$$

where $\overline{IC}$ is *IC* with all occurences of the *K* operator removed.

Proof:

It is sufficient to prove that for any FOPCE formula w with free variables **x**, Closure(*KB*) $\Vdash (\forall \mathbf{x})\, Kw \equiv w$. Now it is easy to see that for any knowledge base B, $B \Vdash (\forall \mathbf{x})\, Kw \supset w$. It remains to prove that Closure(*KB*) $\Vdash (\forall x)\, w \supset Kw$.

To that end, we show that Closure(*KB*) has at most one model. Assume, to the contrary that there are two models W_1 and W_2 of Closure(*KB*). Then there must be a primitive sentence $P(p_1,\ldots,p_n) \in W_1$ with $P(p_1,\ldots,p_n) \notin W_2$. Hence $KB \not\models P(p_1,\ldots,p_n)$ in which case $\neg P(p_1,\ldots,p_n) \in$ Closure(*KB*) so that W_1 cannot be a model of Closure(*KB*), contradiction. It now easily follows that Closure(*KB*) $\Vdash (\forall x)\, w \supset Kw$.

Thus, under the closed world assumption, integrity constraint satisfaction reduces to first order entailment. This is false for circumscriptive closure (Lifschitz 1985) and the generalized closed world assumption (Minker 1982), as the following example reveals.

Example

$$KB = \{p \vee q\} \qquad\qquad IC = \neg Kp$$

Both the circumscriptive and generalized closure of KB yield $\{p \wedge \neg q \;\vee\; \neg p \wedge q\}$ which $\Vdash \neg Kp$ but $\not\models_{FOPCE}$ $\neg p$

Theorem 4.2

If Closure (*KB*) is satisfiable, then the two definitions of what it means for Closure (*KB*) to satisfy an integrity constraint (definitions 1 and 2 of Section 2) are equivalent.

Proof:

Definitions 1 and 2 assume that an integrity constraint IC is a first order sentence.

1. We first prove that Definition 1 (consistency) implies Definition 2 (entailment). Suppose Closure(KB) $\not\models_{FOPCE} IC$. Since Closure(KB) is satisfiable, it has exactly one model. (See proof of Theorem 4.1). Hence, Closure(KB) $\models_{FOPCE} \neg IC$, i.e. Closure(KB) + IC is unsatisfiable.

2. Trivially, if Closure(KB) $\models_{FOPCE} IC$ and Closure(KB) is satisfiable, then Closure(KB) + IC is satisfiable, so Definition 2 implies Definition 1.

For closed knowledge bases Theorem 4.2 informs us that the two "classical" definitions (consistency vs. entailment) are equivalent for first order integrity constraints. By Theorem 4.1, our notion of constraint satisfaction reduces to first order entailment for closed knowledge bases. So under the closed world assumption all three definitions amount to the same thing, namely first order entailment.

All of this assumes a particularly simple form in the case of relational databases. We can view any instance DB of a relational database as a finite set of primitive non-equality sentences of the form $P(p_1, \ldots, p_n)$ for parameters p_i, together with the primitive sentences $p = p$ for each parameter p.† Query evaluation for DB is defined relative to Closure(DB). Clearly, Closure(DB) has a unique model which is DB itself, when viewed as a world structure. Thus, when IC is a first order integrity constraint, Closure(DB) satisfies IC iff Closure(DB) $\models_{FOPCE} IC$ iff IC is true in the world structure DB, which is the standard notion of constraint satisfaction in relational database theory.

5. TESTING CONSTRAINTS FOR GENERAL KNOWLEDGE BASES

Levesque (1981) provides a (noneffective) method for querying a finite knowledge base with a KFOPCE sentence. Since determining whether an integrity constraint is satisfied is formally identical to querying the knowledge base with that constraint, we can appeal to Levesque's query evaluation approach. For this reason, we briefly describe his method for query evaluation.

Let KB be a finite knowledge base. For formulas u, w of KFOPCE, define

$|w|_{KB} = w$ if w is a FOPCE formula,
$|\neg w|_{KB} = \neg |w|_{KB}$,
$|u \vee w|_{KB} = |u|_{KB} \vee |w|_{KB}$,
$|(\forall x)w|_{KB} = (\forall x)|w|_{KB}$,
$|Kw|_{KB} = \text{RES}(KB, |w|_{KB})$.

We shall have more to say about RES below.

Theorem 5.1 (Levesque 1981)

For KB a finite knowledge base and w a KFOPCE sentence, $|w|_{KB}$ is a FOPCE sentence and $KB \Vdash w$ iff $KB \models_{FOPCE} |w|_{KB}$.

† Notice we are here characterizing relational databases without null values. The proper treatment of null values considerably complicates this picture (Reiter 1986).

Thus, determining whether a constraint IC is satisfied requires finding $|IC|_{KB}$ followed by a first order theorem-proving task. In the case that w is an integrity constraint, we can improve on Theorem 5.1. Recall that it is the claim of this paper that constraints are really statements about a knowledge base, not about the world represented by that knowledge base. On this view, no FOPCE formulas should occur in a constraint outside the scope of a K operator. This indeed is the case for the examples of Section 2. Motivated by this observation and following Levesque (1984), we define the *pure* KFOPCE formulas to be the smallest set such that:

1. An equality atom is pure.

2. If w is a FOPCE formula, Kw is pure.

3. If w_1 and w_2 are pure, so are Kw_1, $(\forall x)\, w_1$, $\neg w_1, w_1 \vee w_2$.

Levesque's definition of RES, required to compute $|w|_{KB}$, follows:

RES (KB, w) = **If** w has no free variables
then if $KB \models_{FOPCE} w$
then $(\forall x)\, x = x$
else $(\forall x)\, x \neq x$
else (Assume that x is free in w and that the parameters appearing in KB or w are $i_1, \ldots, i_n$. Let i be any parameter not in KB or w.)
$(x = i_1 \wedge \text{RES}(KB,\ w^x_{i_1})) \vee \cdots \vee (x = i_n \wedge \text{RES}(KB,\ w^x_{i_n}))$
$\vee\ (x \neq i_1 \wedge \cdots \wedge x \neq i_n \wedge \text{RES}(KB,\ w^x_i)^i_x)$.

Lemma 5.2

If KB is a satisfiable knowledge base and w a FOPCE equality sentence, then $KB \models_{FOPCE} w$ iff $\models_{FOPCE} w$.

Proof:

The sufficiency is obvious. To prove the necessity, let M be a model of KB. Then M contains only the primitive equality sentences $p = p$ for each parameter p. But this is the case for any world structure, whether a model of KB or not. So if w is true in M, it is true in every world structure. Hence $\models_{FOPCE} w$.

Lemma 5.3

If w is a pure KFOPCE formula and KB a finite knowledge base, then $|w|_{KB}$ is a FOPCE equality formula with the same free variables as w.

Proof:

Induction on the shape of w.

1. If w is an equality atom, the result follows from the definition of $|w|_{KB}$.

2. If w has the form Ku where u is a FOPCE formula, then

$$|w|_{KB} = RES(KB,\ |u|_{KB})$$
$$= RES(KB, u) \text{ since } |u|_{KB} = u \text{ whenever } u \text{ is a FOPCE formula.}$$

This, by the definition of RES, is a FOPCE equality formula with the same free variables as u, hence as w.

3. If w has the form $(\forall x)u$ where u is pure, then $|w|_{KB} = (\forall x)\ |u|_{KB}$ and the result follows easily by induction. Similarly, when w has the form $\neg u$ or $u \wedge v$ when u and v are pure. Finally, suppose w is Ku where u is pure. Then $|w|_{KB} = RES(KB, |u|_{KB})$. By induction, $|u|_{KB}$ is an equality formula with the same free variables as u, hence as w. By the definition of RES, $|w|_{KB}$ is an equality formula with the same free variables as w.

We can now present an improvement of Theorem 5.1 for pure integrity constraints.

Theorem 5.4

Suppose KB is a finite satisfiable knowledge base, and IC a pure integrity constraint. Then $|IC|_{KB}$ is a FOPCE equality sentence and KB satisfies IC iff $\models_{FOPCE} |IC|_{KB}$.

Proof:

By Lemma 5.3, $|IC|_{KB}$ is a FOPCE equality sentence. By Theorem 5.1, KB satisfies IC iff $KB \models_{FOPCE} |IC|_{KB}$ iff, by Lemma 5.2, $\models_{FOPCE} |IC|_{KB}$.

The function RES is rather complex. However, there is a special class of knowledge bases which admits a simple characterization of RES.

Definition

If $w(x_1, \ldots, x_n)$ is a FOPCE formula with distinct free individual variables $x_1, \ldots, x_n$, and KB is a knowledge base, then
Instances(KB, $w(x_1, \ldots, x_n)) = \{(p_1, \ldots, p_n) \mid p_i$ is a parameter and $KB \models_{FOPCE} w(p_1, \ldots, p_n)\}$.

Instances (KB, w) need not be finite. The finite case is important since it provides a simple characterization of RES, as follows.

Theorem 5.5

Suppose KB is a finite knowledge base and w a FOPCE formula with free variables $\mathbf{x}$. If Instances(KB, w) is finite, say Instances(KB,w) = $\{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$, then

$\models_{FOPCE} (\forall \mathbf{x})$ RES(KB, w) $\equiv \mathbf{x} = \mathbf{p}_1 \vee \cdots \vee \mathbf{x} = \mathbf{p}_n$.

Proof:

We appeal to the following version of a result of (Levesque 1984, Lemma 3.5):

For $\mathbf{p}$ a tuple of parameters, $\models_{FOPCE} RES(KB, w)^{\mathbf{x}}_{\mathbf{p}}$ iff $\mathbf{p} \in$ Instances (KB, w).

1. We first prove

$\models_{FOPCE} (\forall \mathbf{x})\ RES(KB, w) \supset \mathbf{x} = \mathbf{p}_1 \vee \cdots \vee \mathbf{x} = \mathbf{p}_n$

Suppose, for a parameter tuple $\mathbf{p}$, that $RES(KB, w)^{\mathbf{x}}_{\mathbf{p}}$ is true in a world structure Σ. By the definition of *RES*, $RES(KB, w)^{\mathbf{x}}_{\mathbf{p}}$ is an equality FOPCE sentence. Since all world structures agree on their interpretations of equality, $\models_{FOPCE} RES(KB, w)^{\mathbf{x}}_{\mathbf{p}}$. Therefore, by Levesque's result, $\mathbf{p} \in$ Instances $(KB, w) = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$, so $\mathbf{p} = \mathbf{p}_1 \vee \cdots \vee \mathbf{p} = \mathbf{p}_n$ is true in Σ.

2. Finally, we prove

$\models_{FOPCE}(\forall \mathbf{x})\, \mathbf{x} = \mathbf{p}_1 \vee \cdots \vee \mathbf{x} = \mathbf{p}_n \supset RES(KB, w)$.

Suppose, for a parameter tuple $\mathbf{p}$, that $\mathbf{p} = \mathbf{p}_1 \vee \cdots \vee \mathbf{p} = \mathbf{p}_n$ is true in a world structure Σ. Then $\mathbf{p} = \mathbf{p}_i$ is true in Σ for some i, so the tuple $\mathbf{p}$ is identical to $\mathbf{p}_i$. Hence, by Levesque's result, $\models_{FOPCE} RES(KB, w)^{\mathbf{x}}_{\mathbf{p}}$ so $RES(KB, w)^{\mathbf{x}}_{\mathbf{p}}$ is true in Σ.

Definition

A K_1 formula is a KFOPCE formula in which the scope of every K operator is a FOPCE formula.

All of the example constraints of Section 2 are K_1 formulas. Indeed, just as we expect constraints to be pure, we equally expect them to be K_1; it seems impossible to imagine the need for iterated modalities.

We now have a (conceptually) simple way to determine whether *KB* satisfies *IC* whenever *IC* is a K_1 sentence with the property that Instances (*KB*,*w*) is finite for each subformula *w* of *IC* which is the scope of a *K* operator:

1. Determine $|IC|_{KB}$ using the defintion of $|\;|_{KB}$ and Theorem 5.5.

2.1 When *IC* is pure, *KB* satisfies *IC* iff $\models_{FOPCE} |IC|_{KB}$ by Theorem 5.4.

2.2 When *IC* is not pure, *KB* satisfies *IC* iff $KB \models_{FOPCE} |IC|_{KB}$ by Theorem 5.1.

Our final task is to characterize a useful class of knowledge bases and integrity constraints which admits this approach to testing constraint satisfaction.

Definitions

The *positive existential* (p.e.) FOPCE formulas are defined by the smallest set such that:

1. An atomic formula other than an equality atom is p.e.

2. If w is p.e., so is $(\exists x)w$.

3. If w_1 and w_2 are p.e. so are $w_1 \wedge w_2$ and $w_1 \vee w_2$.

A *rule* is a sentence of the form $(\forall \mathbf{x})\ A \supset B$ where A is a conjunction of atomic formulas other than equality atoms, B is a p.e. formula, and every variable of $\mathbf{x}$ occurs free in A.

A knowledge base is *elementary* iff it is a finite set of p.e. sentences and rules. Notice that elementary knowledge bases make no mention of equality.

Elementary knowledge bases are analogous to the deductive databases widely studied in the logic programming community (e.g. (Lloyd & Topor 1985)). They are more general than deductive databases by admitting disjunctions and existential quantification. They are slightly less general by requiring that for rules ($\forall$ **x**) $A \supset B$, the variables of **x** must all occur free in A, although this is a minor restriction in practice.

Lemma 5.6

Suppose *KB* is an elementary knowledge base. Then *KB* has a model with the property that each primitive non-equality sentence in that model mentions only parameters occurring in *KB*.

Proof:

Without loss of generality, assume *KB* mentions at least one parameter. (If not, augment *KB* with the rule $P(p) \supset P(p)$ for some unary predicate P and parameter p.) For p.e. sentences w define

$M_{KB}(w) = \{w\}$ if w is a primitive sentence,
$= M_{KB}(w_1) \cup M_{KB}(w_2)$ if w has the form $w_1 \vee w_2$ or $w_1 \wedge w_2$,
$= M_{KB}(u_p^x)$ where p is a parameter mentioned in *KB*, when w has the form $(\exists x)u$.

Define

$\Sigma_0(KB) = \cup M_{KB}(w)$ where the union is over all p.e. sentences w of *KB*.

If w is a *KB* rule of the form $(\forall \mathbf{x}) A \supset B$, define, for $i \geq 0$,

$C_{i,KB}(w) = \cup M_{KB}(B_{\mathbf{p}}^{\mathbf{x}})$ where the union is over all tuples **p** of parameters for which each primitive sentence of the conjunct $A_{\mathbf{p}}^{\mathbf{x}}$ is in $\Sigma_i(KB)$.

Define

$\Sigma_{i+1}(KB) = \Sigma_i(KB) \cup C_{i,KB}(w)$ where the union is over all rules w of *KB*.

Finally, define

$\Sigma(KB) = \bigcup_{i=0}^{\infty} \Sigma_i(KB) \cup \{p = p \mid p \text{ is a parameter}\}$.

By construction, $\Sigma(KB)$ consists of primitive sentences only. Moreover, since *KB* is elementary, it makes no mention of the equality predicate. Hence no $\Sigma_i(KB)$ contains a primitive equality sentence. Therefore, $\Sigma(KB)$ is a world structure. We show, by induction on i, that $\Sigma_i(KB)$ mentions only parameters occurring in *KB*. Clearly, $\Sigma_0(KB)$ has this property. Assume that $\Sigma_i(KB)$ has this property. To show that $\Sigma_{i+1}(KB)$ does, we must prove that $C_{i,KB}(w)$ does for each rule w of *KB*. Suppose $(\forall \mathbf{x}) A \supset B$ is such a rule. It is sufficient to prove that $B_{\mathbf{p}}^{\mathbf{x}}$ mentions only parameters occurring in *KB* whenever each primitive sentence of the conjunct $A_{\mathbf{p}}^{\mathbf{x}}$ is in $\Sigma_i(KB)$. But all variables of **x** are mentioned in A so that all the parameters **p** are mentioned in $\Sigma_i(KB)$. By induction, the parameters **p** occur in *KB*, so that $B_{\mathbf{p}}^{\mathbf{x}}$ mentions only parameters occurring in *KB*.

We have proved that $\Sigma(KB)$ is a world structure, and that each primitive non-equality sentence in $\Sigma(KB)$ mentions only parameters occurring in *KB*. It remains only to show that $\Sigma(KB)$ is a model of *KB*. To that end, notice that by the construction of $\Sigma_0(KB)$ every p.e. sentence of *KB* is true in $\Sigma(KB)$. To complete the proof, we need only show that every rule of *KB* is true in $\Sigma(KB)$. Suppose $(\forall \mathbf{x}) A \supset B$ is such a rule, and **p** is a tuple of parameters such that $A_{\mathbf{p}}^{\mathbf{x}}$ is true in $\Sigma(KB)$. Since A does not mention the equality predicate, $A_{\mathbf{p}}^{\mathbf{x}}$ is true in $\bigcup_{i=0}^{\infty}$

Σ_i (*KB*), hence in Σ_j (*KB*) for some *j*. Then by definition, M_{KB} ($B^{\mathbf{x}}_{\mathbf{p}}$) $\subseteq \Sigma_{j+1}$ (*KB*), and since $B^{\mathbf{x}}_{\mathbf{p}}$ is true in M_{KB} ($B^{\mathbf{x}}_{\mathbf{p}}$), it is true in Σ_{j+1} (*KB*), hence in Σ(*KB*). Therefore $(\forall \mathbf{x})\, A \supset B$ is true in Σ (*KB*).

Definition

Suppose *w* is a FOPCE formula with free variables $x_1, \ldots, x_n$. Then *w* has *disjunctively linked variables* iff for each of its subformulas of the form $w_1 \vee w_2$, those free variables of w_1 which are among $x_1, \ldots, x_n$ are precisely the same as those of w_2 which are among $x_1, \ldots, x_n$.

Example

The following have disjunctively linked variables:

$P(a,b) \vee Q(a,c)$

$(\forall x)\,(U(x) \vee W(x))$

$P(a,x) \vee Q(x,x)$

$(\exists\ y\ z)\,(P(y,x) \vee R(y,z,x)) \vee (\exists\ u)\,(P(u,a) \wedge Q(u,x))$

The following do not:

$(\forall x)\,(U(x) \vee W(y))$

$P(x,y) \vee Q(y,z)$

Theorem 5.7

If *KB* is an elementary knowledge base and *w* a p.e. FOPCE formula with disjunctively linked variables, then Instances (*KB*, *w*) is finite.

Proof:

By Lemma 5.6, *KB* has a model *M* with the property that each primitive non-equality sentence in *M* mentions only parameters occurring in *KB*. We prove, by induction on the shape of *w*, that whenever *w* has free variables **x** and **p** is a tuple of parameters, if $w^{\mathbf{x}}_{\mathbf{p}}$ is true in *M* then the parameters of **p** are all mentioned in *KB*. Since *KB* is finite, it will follow that Instances (*KB*,*w*) must be finite. The inductive proof follows:

1. Suppose *w* is an atomic non-equality formula and $w^{\mathbf{x}}_{\mathbf{p}}$ is true in *M*. Then, because $w^{\mathbf{x}}_{\mathbf{p}}$ is a primitive non-equality sentence, $w^{\mathbf{x}}_{\mathbf{p}} \in M$ and the result follows from Lemma 5.6.

2. If *w* is $u \wedge v$, the result follows trivially by induction.

3. Suppose *w* is $u \vee v$, and $w^{\mathbf{x}}_{\mathbf{p}}$ is true in *M*. Then $u^{\mathbf{x}}_{\mathbf{p}}$ is true in *M*, or $v^{\mathbf{x}}_{\mathbf{p}}$ is true in *M*. Moreover, since *w* has disjunctively linked variables, every variable of **x** occurs free in both *u* and *v*, so the result follows by induction.

4. Suppose *w* is $(\exists\ y)u$ where, without loss of generality, we can assume *y* is distinct from any of the variables of **x**. If $w^{\mathbf{x}}_{\mathbf{p}}$ is true in *M*, then for some parameter π, $u^{\mathbf{x},y}_{\mathbf{p},\pi}$ is true in *M*. By induction, the parameters **p**, π are mentioned in *KB*, so in particular so are the parameters **p**.

Suppose *KB* is elementary and *IC* is a K_1 integrity constraint each of whose *K* operators has a p.e. scope *w* with existentially linked variables. Then Theorem 5.7 gurantees that we can adopt the approach to determining constraint satisfaction described above immediately following the proof of Theorem 5.5.

One last point: In order to use the above approach, *IC's K* operators must have p.e. scopes with existentially linked variables. This is the case for the first four examples of Section 2, but not for the fifth, which is a functional dependency. The reason is that this functional dependency has an equality atom as the scope of a *K*, and p.e. formulas cannot mention equality. Fortunately, it is a simple result that

$$\models_{KFOPCE} (\forall x,y)\, K(x{=}y) \equiv (x{=}y).$$

Hence, by Theorem 3.1 the given functional dependency is equivalent to

$$(\forall x,y,z)\, Kss\#\,(x,y) \wedge Kss\#\,(x,z) \supset y = z.$$

This has p.e. scopes for each *K* operator. Notice that this is also a pure constraint. Since many of the dependencies which arise in relational databases appeal to equality, it is important that any equality atom occurring as a scope of a *K* operator may be extracted from that *K*. Moreover, if the original sentence was pure, so also is the resulting sentence.

6. *Discussion*

We have argued that integrity constraints are statements about the contents of a knowledge base, not about the world. They are thus metatheoretic in character. We have appealed to Levesque's logics FOPCE and KFOPCE to formalize the concept of a knowledge base satisfying its constraints. There are many issues which we have not explored, but which deserve attention:

1. Our results should be extended to Levesque's (1984) languages *L* and *KL* which admit function symbols.

2. Notice that we appealed to Levesque's ASK operator to define the concept of constraint satisfaction. Now ASK is an extra-logical notion, so constraint satisfaction is not defined *within* KFOPCE i.e. this concept is not defined in terms of KFOPCE *validity*. Recently, Levesque (1987) has defined a logic within which one can define this concept. Briefly, this logic has two modalities: O (for *only know*) and B (for *believe*). We can then say that *KB* satisfies *IC* iff O*KB* $\supset$ B*IC* is a valid sentence of this logic. The consequences of this notion of constraint satisfaction remain to be explored.

3. We have not explored mechanisms for incremental modifications to a knowledge base. Usually a knowledge base will be known to satisfy its constraints. When a (normally) small change is made to it, it should not be necessary to verify all its constraints ab initio. Rather, only enough computation should be devoted to verify the change in its state, given that its prior state was acceptable. Nicolas (1982) provides such mechanisms for relational databases, as do Lloyd and Topor (1985) for deductive databases. Similar mechanisms must be devised for our concept of integrity checking.

4. Many knowledge representation languages (e.g. Chung et al (1987)) provide mechanisms for *procedural attachment* which are invoked whenever a change is made to the knowledge base state. Such procedures normally check to see whether certain conditions hold in the current state and if so, may change this state in various ways. Such changes may trigger other procedures, and so on. A simple example is a procedure triggered by an update of an employee record. It might then search for a social security entry for that employee and, failing in this, request this entry from the user. Clearly, this is a procedural version of the integrity constraint

$$(\forall x)\, K\, emp(x) \supset (\exists\, y)\, Kss\#(x,\, y).$$

In general, there is an intimate connection between procedural attachment and integrity constraints. It would be worthwhile exploring this relationship, perhaps with two objectives in mind:

(i) Since there is a logic of integrity constraints, we can explore the consequences of the constraints, hence of their procedural incarnations.

(ii) Correctness proofs should be possible for the procedures relative to their logically specified constraints.

Acknowledgements

I am grateful to Hector Levesque for a number of helpful discussions and suggestions regarding this paper and to Gerhard Lakemeyer for his comments on an earlier draft. This research was supported by grant A9044 of the National Science and Engineering Research Council of Canada.

REFERENCES

Chung, L., Rios-Zertuche, D., Nixon, B. and Mylopoulos, J. (1987). Process management and assertion enforcement for a semantic data model, Technical Note, Dept. of Computer Science, University of Toronto.

Fagin, R. (1980). Horn clauses and database dependencies, *ACM Symp. on Theory of Computing*, 123-134.

Kowalski (1978). Logic for data description, in: H. Gallaire and J. Minker (eds.), *Logic and Data Bases*, Plenum Press, New York, pp. 77-103.

Levesque, H.J. (1981). A formal treatment of incomplete knowledge bases, Ph.D. thesis, Dept. of Computer Science, University of Toronto; also available as Technical Report No. 3, Fairchild Laboratory for Artificial Intelligence Research, Palo Alto, California.

Levesque, H.J. (1984). Foundations of a functional approach to knowledge representation, *Artificial Intelligence 23*, pp. 155-212.

Levesque, H.J. (1987). All I know: a study in autoepistemic logic, Tech. report, Dept. of Computer Science, University of Toronto.

Lifschitz, V. (1985). Closed-world databases and circumscription, *Artificial Intelligence 27*, pp. 229-235.

Lloyd, J.W., and Topor, R.W. (1985). A basis for deductive database systems, *J. Logic Programming 2:* 93-109.

Minker, J. (1982). On indefinite databases and the closed world assumption, *Proc. 6th Conf. Automat. Deduct.*, New York, pp. 292-308.

Nicolas, J.M. (1982). Logic for improving integrity checking in relational databases, *Acta Inform. 18 (3)*, 227-253.

Nicolas, J.M. and Gallaire, H. (1978). Data base: theory vs. interpretation, in: H. Gallaire and J. Minker (eds.), *Logic and Data Bases*, Plenum Press, New York, pp. 33-54.

Nicolas, J.M. and Yazdanian, K. (1978). Integrity checking in deductive data bases, in: H. Gallaire and J. Minker (eds.), *Logic and Data Bases*, Plenum Press, New York, pp. 325-344.

Reiter, R. (1978). On closed world data bases, in: H. Gallaire and J. Minker (eds.), *Logic and Data Bases*, Plenum Press, New York, pp. 55-76.

Reiter, R. (1984). Towards a logical reconstruction of relational database theory, in: M.L. Brodie, J. Mylopoulos and J.W. Schmidt (eds.), *On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases and Programming Languages*, Springer, New York, pp. 191-233.

Reiter, R. (1986). A sound and sometimes complete query evaluation algorithm for relational databases with null values, *J.ACM 33 (2)*, 349-370.

Sadri, F. and Kowalski, R. (1987). An application of general purpose theorem-proving to database integrity, in J. Minker (ed.), *Foundations of Deductive Databases and Logic Programming*, Morgan Kaufmann Publishers, Palo Alto.