

ON CLOSED WORLD DATA BASES

Raymond Reiter

The University of British Columbia

Vancouver, British Columbia

ABSTRACT

Deductive question-answering systems generally evaluate queries under one of two possible assumptions which we in this paper refer to as the open and closed world assumptions. The open world assumption corresponds to the usual first order approach to query evaluation: Given a data base DB and a query Q, the only answers to Q are those which obtain from proofs of Q given DB as hypotheses. Under the closed world assumption, certain answers are admitted as a result of failure to find a proof. More specifically, if no proof of a positive ground literal exists, then the negation of that literal is assumed true.

In this paper, we show that closed world evaluation of an arbitrary query may be reduced to open world evaluation of so-called atomic queries. We then show that the closed world assumption can lead to inconsistencies, but for Horn data bases no such inconsistencies can arise. Finally, we show how for Horn data bases under the closed world assumption purely negative clauses are irrelevant for deductive retrieval and function instead as integrity constraints.

INTRODUCTION

Deductive question-answering systems generally evaluate queries under one of two possible assumptions which we in this paper refer to as the open and closed world assumptions. The open world assumption corresponds to the usual first order approach to query evaluation: Given a data base DB and a query Q, the only answers to Q

are those which obtain from proofs of Q given DB as hypotheses. Under the closed world assumption, certain answers are admitted as a result of failure to find a proof. More specifically, if no proof of a positive ground literal exists, then the negation of that literal is assumed true. This can be viewed as equivalent to implicitly augmenting the given data base with all such negated literals.

For many domains of application, closed world query evaluation is appropriate since, in such domains, it is natural to explicitly represent only positive knowledge and to assume the truth of negative facts by default. For example, in an airline data base, all flights and the cities which they connect will be explicitly represented. Failure to find an entry indicating that Air Canada flight 103 connects Vancouver with Toulouse permits one to conclude that it does not.

This paper is concerned with closed world query evaluation and its relationship to open world evaluation. In the section, Data Bases and Queries, we define a query language and the notion of an open world answer to a query. The section called The Closed World Assumption formally defines the notion of a closed world answer. The section, Query Evaluation Under the CWA, shows how closed world query evaluation may be decomposed into open world evaluation of so-called "atomic queries" in conjunction with the set operations of intersection, union and difference, and the relational algebra operation of projection. In the section, On Data Bases Consistent with the CWA, we show that the closed world assumption can lead to inconsistencies. We prove, moreover, that for Horn data bases no such inconsistencies can arise. Also, for Horn data bases, the occurrence of purely negative clauses is irrelevant to closed world query evaluation. By removing such negative clauses one is left with so-called definite data bases which are then consistent under both the open and closed world assumptions. Finally, in the section, The CWA and Data Base Integrity, we show that these purely negative clauses, although irrelevant to deductive retrieval, have a function in maintaining data base integrity.

In order to preserve continuity we have relegated all proofs of the results in the main body of this paper to an appendix.

DATA BASES AND QUERIES

The query language of this paper is set oriented, i.e. we seek all objects (or tuples of objects) having a given property. For example, in an airline data base the request "Give all flights and their carriers which fly from Boston to England" might be represented in our query language by:

$\langle x/\text{Flight}, y/\text{Airline} | (\exists z/\text{City}) \text{Connect } x, \text{Boston}, z \wedge \text{Owns } y, x \wedge \text{City-of } z, \text{England} \rangle$

which denotes the set of all ordered pairs (x, y) such that x is a flight, y is an airline and

$(\exists z/\text{City}) \text{Connect } x, \text{Boston}, z \wedge \text{Owns } y, x \wedge \text{City-of } z, \text{England}$

is true. The syntactic objects Flight, Airline and City are called types and serve to restrict the variables associated with them to range over objects of that type. Thus, $(\exists z/\text{City})$ may be read as "There is a z which is a city".

Formally, all queries have the form

$\langle x_1/\tau_1, \dots, x_n/\tau_n | (\exists y_1/\theta_1) \dots (\exists y_m/\theta_m) W(x_1, \dots, x_n, y_1, \dots, y_m) \rangle$

where $W(x_1, \dots, x_n, y_1, \dots, y_m)$ is a quantifier-free formula with free variables $x_1, \dots, x_n, y_1, \dots, y_m$ and moreover W contains no function signs. For brevity we shall often denote a typical such query by $\langle \vec{x}/\vec{\tau} | (\exists \vec{y}/\vec{\theta}) W \rangle$. The τ 's and θ 's are called types. We assume that with each type τ is associated a set of constant signs which we denote by $|\tau|$. For example, in an airline data base, $|\text{City}|$ might be $\{\text{Toronto}, \text{Boston}, \text{Paris}, \dots\}$. If $\vec{\tau} = \tau_1, \dots, \tau_n$ is a sequence of types we denote by $|\vec{\tau}|$ the set $|\tau_1| \times \dots \times |\tau_n|$.

A data base (DB) is a set of clauses containing no function signs. For an airline data base, DB might contain such information as:

"Air Canada flight 203 connects Toronto and Vancouver."

Connect AC203, Toronto, Vancouver

"All flights from Boston to Los Angeles serve meals."

$(\exists x/\text{Flight}) \text{Connect } x, \text{Boston}, \text{LA} \supset \text{Meal-serve } x$

Let $Q = \langle \vec{x}/\vec{\tau} | (\exists \vec{y}/\vec{\theta}) W(\vec{x}, \vec{y}) \rangle$ and let DB be a data base. A set of n -tuples of constant signs $\{\vec{c}^{(1)}, \dots, \vec{c}^{(r)}\}$ is an answer to Q (with respect to DB) iff

1. $\vec{c}^{(i)} \in |\vec{\tau}|$ $i = 1, \dots, r$ and

2. $\text{DB} \vdash \bigvee_{i \leq r} (\exists \vec{y}/\vec{\theta}) W(\vec{c}^{(i)}, \vec{y})$

Notice that if $\{\vec{c}^{(1)}, \dots, \vec{c}^{(r)}\}$ is an answer to Q , and \vec{c} is any

n-tuple of constant signs satisfying 1. then so also is $\{\vec{c}^{(1)}, \dots, \vec{c}^{(r)}, \vec{c}\}$ an answer to Q. This suggests the need for the following definitions:

An answer A to Q is minimal iff no proper subset of A is an answer to Q. If A is a minimal answer to Q, then if A consists of a single n-tuple, A is a definite answer to Q. Otherwise, A is an indefinite answer to Q. Finally define $\|Q\|_{OWA}$ to be the set of minimal answers to Q. (For reasons which will become apparent later, the subscript OWA stands for "Open World Assumption".) Notice the interpretation assigned to an indefinite answer $\{\vec{c}^{(1)}, \dots, \vec{c}^{(r)}\}$ to Q: \vec{x} is either $\vec{c}^{(1)}$ or $\vec{c}^{(2)}$ or...or $\vec{c}^{(r)}$ but there is no way, given the information in DB, of determining which. Instead of denoting an answer as a set of tuples $\{\vec{c}^{(1)}, \dots, \vec{c}^{(r)}\}$ we prefer the more suggestive notation $\vec{c}^{(1)} + \dots + \vec{c}^{(r)}$, a notation we shall use in the remainder of this paper.

Example 1.

Suppose DB knows of 4 humans and 2 cities:

$$|\text{Human}| = \{a, b, c, d\} \quad |\text{City}| = \{B, V\}$$

Suppose further that everyone is either in B or in V:

$$(x/\text{Human}) \text{Loc } x, B \vee \text{Loc } x, V$$

and moreover, a is in B and b is in V:

$$\text{Loc } a, B \quad \text{Loc } b, V$$

Then for the query "Where is everybody?"

$$Q = \langle x/\text{Human}, y/\text{City} \mid \text{Loc } x, y \rangle$$

we have

$$\|Q\|_{OWA} = \{(a, B), (b, V), (c, B) + (c, V), (d, B) + (d, V)\}$$

i.e. a is in B, b is in V, c is either in B or V and d is either in B or V.

Since it is beyond the scope of this paper, the reader is referred to Reiter [1977] or Reiter [1978] for an approach to query evaluation which returns $\|Q\|_{OWA}$ given any query Q.

THE CLOSED WORLD ASSUMPTION

In order to illustrate the central concept of this paper, we consider the following purely extensional data base (i.e., a data base consisting of ground literals only):

$$|\text{Teacher}| = \{\text{a}, \text{b}, \text{c}, \text{d}\}$$

$$|\text{Student}| = \{\text{A}, \text{B}, \text{C}\}$$

<u>Teach</u>	
a	A
b	B
c	C
a	B

Now consider the query: Who does not teach B?

$$Q = < x/\text{Teacher} | \text{Teach } x, \text{B} >$$

By the definition of the previous section, we conclude, counter-intuitively, that

$$\|Q\|_{\text{OWA}} = \emptyset.$$

Intuitively, we want $\{\text{c}, \text{d}\}$ i.e. $|\text{Teacher}| - \| < x/\text{Teacher} | \text{Teach } x, \text{B} > \|_{\text{OWA}}$. The reason for the counterintuitive result is that first order logic interprets the DB literally; all the logic knows for certain is what is explicitly represented in the DB. Just because Teach c, B is not present in the DB is no reason to conclude that Teach c, B is true. Rather, as far as the logic is concerned, the truth of Teach c, B is unknown! Thus, we would also have to include the following facts about Teach:

<u>Teach</u>	
a	C
b	A
b	C
c	A
c	B
d	A
d	B
d	C

Unfortunately, the number of negative facts about a given domain will, in general, far exceed the number of positive ones so that the requirement that all facts, both positive and negative, be explicitly represented may well be unfeasible. In the case of

purely extensional data bases there is a ready solution to this problem. Merely explicitly represent positive facts. A negative fact is implicitly present provided its positive counterpart is not explicitly present. Notice, however, that by adopting this convention, we are making an assumption about our knowledge about the domain, namely, that we know everything about each predicate of the domain. There are no gaps in our knowledge. For example, if we were ignorant as to whether or not a teaches C, we could not permit the above implicit representation of negative facts. This is an important point. The implicit representation of negative facts presumes total knowledge about the domain being represented. Fortunately, in most applications, such an assumption is warranted. We shall refer to this as the closed world assumption (CWA). Its opposite, the open world assumption (OWA), assumes only the information given in the data base and hence requires all facts, both positive and negative, to be explicitly represented. Under the OWA, "gaps" in one's knowledge about the domain are permitted.

Formally, we can define the notion of an answer to a query under the CWA as follows:

Let DB be an extensional data base and let $\overline{EDB} = \{\vec{Pc} \mid P \text{ is a predicate sign, } \vec{c} \text{ a tuple of constant signs and } \vec{Pc} \notin DB\}$. Then \vec{c} is a CWA answer to $\langle \vec{x}/\vec{\tau} \mid (\vec{Ey}/\vec{\theta})W(\vec{x}, \vec{y}) \rangle$ (with respect to DB) iff

1. $\vec{c} \in |\vec{\tau}|$ and
2. $DB \cup \overline{EDB} \vdash (\vec{Ey}/\vec{\theta})W(\vec{c}, \vec{y})$

For purely extensional data bases, the CWA poses no difficulties. One merely imagines the DB to contain all negative facts each of which has no positive version in the DB. This conceptual view of the DB fails in the presence of non ground clauses. For if $\vec{Pc} \notin DB$, it may nevertheless be possible to infer \vec{Pc} from the DB, so that we cannot, with impunity, imagine $\vec{Pc} \in DB$. The obvious generalization is to assume that the DB implicitly contains \vec{Pc} whenever it is not the case that $DB \vdash \vec{Pc}$.

Formally, we can define the notion of an answer to a query under the CWA for an arbitrary data base DB as follows:

Let

$\overline{EDB} = \{\vec{Pc} \mid P \text{ is a predicate sign, } \vec{c} \text{ a tuple of constant signs and } DB \nvdash \vec{Pc}\}$

Then $\vec{c}^{(1)} + \dots + \vec{c}^{(r)}$ is a CWA answer to

$\langle \vec{x}/\vec{\tau} \mid (\vec{Ey}/\vec{\theta})W(\vec{x}, \vec{y}) \rangle$ (with respect to DB) iff

1. $\vec{c}^{(i)} \in |\vec{\tau}|_{i=1, \dots, r}$ and
2. $DB \cup \overline{EDB} \vdash \forall_{\vec{i} \leq r} (\vec{E}y/\vec{\theta})W(\vec{c}^{(i)}, \vec{y})$

This definition should be compared with the definition of an answer in the previous section. We shall refer to this latter notion as an OWA answer. As under the OWA, we shall require the notions of minimal, indefinite and definite CWA answers. If Q is a query, we shall denote the set of minimal CWA answers to Q by $\|Q\|_{CWA}$.

Example 2.

We consider a fragment of an inventory data base.

1. Every supplier of a part supplies all its subparts.

$(x/Supplier)(yz/Part)Supplies\ x,y \wedge Subpart\ z,y \supseteq Supplies\ x,z$

2. Foobar Inc. supplies all widgets.

$(x/Widget)Supplies\ Foobar,x$

3. The subpart relation is transitive.

$(xyz/Part)Subpart\ z,y \wedge Subpart\ y,x \supseteq Subpart\ z,x$

Assume the following type extensions:

$|Supplier| = \{Acme, Foobar, AAA\}$

$|Widget| = \{w_1, w_2, w_3, w_4\}$

$|Part| = \{p_1, p_2, p_3, w_1, w_2, w_3, w_4\}$

Finally, assume the following extensional data base:

Supplies	x	y	Subpart	x	y
Acme	p ₁			p ₂	p ₁
AAA	w ₃			p ₃	p ₂
AAA	w ₄			w ₁	p ₁

Then \overline{EDB} is:

<u>Supplies</u>	<u>x</u>	<u>y</u>	<u>Subpart</u>	<u>x</u>	<u>y</u>
	Acme	w ₃		P ₁	p ₁
	Acme	w ₄		P ₁	p ₂
	AAA	p ₁		P ₁	p ₃
	AAA	p ₂		P ₁	w ₁
	AAA	p ₃		P ₁	w ₂
	AAA	w ₁		P ₁	w ₃
	AAA	w ₂		P ₁	w ₄
	Foobar	p ₁		P ₂	p ₂
	Foobar	p ₂		P ₂	p ₃
	Foobar	p ₃		P ₂	w ₁
	P ₁	Acme		P ₂	w ₂
	P ₁	AAA		P ₂	w ₃
	P ₁	Foobar		P ₂	w ₄
	P ₁	p ₁		P ₃	p ₃
	P ₁	p ₂		P ₃	w ₁
	P ₁	p ₃		P ₃	w ₂
	P ₁	w ₁		P ₃	w ₃
etc.				P ₃	w ₄
				etc.	

The notion of a CWA answer is obviously intimately related to the negation operators of PLANNER (Hewitt [1972]) and PROLOG (Roussel [1975]) since in these languages, negation means "not provable" and the definition of \overline{EDB} critically depends upon this notion. Clark [1978] investigates the relation between this notion of negation as failure and its truth functional semantics. The need for the CWA in deductive question-answering systems has been articulated in Nicolas and Syre [1974].

Notice that under the CWA, there can be no "gaps" in our knowledge about the domain. More formally, for each predicate sign P and each tuple of constant signs \vec{c} , either $DB \vdash P\vec{c}$ or $\overline{EDB} \vdash \overline{P}\vec{c}$ and since, under the CWA the data base is taken to be $DB \cup \overline{EDB}$, we can always infer either $P\vec{c}$ or $\overline{P}\vec{c}$ from $DB \cup \overline{EDB}$. Since there are no "knowledge gaps" under the CWA, it should be intuitively clear that indefinite CWA answers cannot arise, i.e. each minimal CWA answer to a query is of the form \vec{c} . The following result confirms this intuition.

Theorem 1.

Let $Q = \langle \vec{x}/\vec{\tau} | (\vec{y}/\vec{\theta})W(\vec{x}, \vec{y}) \rangle$. Then every minimal CWA answer to Q is definite.

There is one obvious difficulty in directly applying the definition of a CWA answer to the evaluation of queries. The definition requires that we explicitly know \overline{EDB} and, as Example 2 demonstrates, the determination of \overline{EDB} is generally non trivial.

In any event, for non toy domains, $\overline{\text{EDB}}$ would be so large that its explicit representation would be totally unfeasible. Fortunately, as we shall see in the next section, there is no need to know the elements of $\overline{\text{EDB}}$ i.e. it is possible to determine the set of closed world answers to an arbitrary query Q by appealing only to the given data base DB.

QUERY EVALUATION UNDER THE CWA

It turns out that the CWA admits a number of significant simplifications in the query evaluation process. The simplest of these permits the elimination of the logical connectives \wedge and \vee in favour of set intersection and union respectively, as follows:

Theorem 2.

$$1. \quad \left\| \langle \vec{x}/\vec{\tau} \mid (\vec{E}\vec{y}/\vec{\theta})(W_1 \vee W_2) \rangle \right\|_{\text{CWA}} = \left\| \langle \vec{x}/\vec{\tau} \mid (\vec{E}\vec{y}/\vec{\theta})W_1 \rangle \right\|_{\text{CWA}} \cup \\ \left\| \langle \vec{x}/\vec{\tau} \mid (\vec{E}\vec{y}/\vec{\theta})W_2 \rangle \right\|_{\text{CWA}}$$

$$2. \quad \left\| \langle \vec{x}/\vec{\tau} \mid W_1 \wedge W_2 \rangle \right\|_{\text{CWA}} = \left\| \langle \vec{x}/\vec{\tau} \mid W_1 \rangle \right\|_{\text{CWA}} \cap \left\| \langle \vec{x}/\vec{\tau} \mid W_2 \rangle \right\|_{\text{CWA}}$$

Notice that in the identity 2, the query must be quantifier free. Notice also that the identities of Theorem 2 fail under the OWA. To see why, consider the following:

Example 3

$$|\tau| = \{a\}$$

DB: $P_a \vee R_a$

$$Q = \langle x/\tau \mid P_x \vee R_x \rangle$$

$$\|Q\|_{\text{OWA}} = \{a\}$$

but

$$\left\| \langle x/\tau \mid P_x \rangle \right\|_{\text{OWA}} = \left\| \langle x/\tau \mid R_x \rangle \right\|_{\text{OWA}} = \emptyset$$

Example 4

$$|\tau| = \{a, b\}$$

DB: $P_a \vee P_b, R_a, R_b$

$$Q = \langle x/\tau \mid P_x \wedge R_x \rangle$$

$$\|Q\|_{OWA} = \{a+b\}$$

but

$$\|\langle x/\tau | Px \rangle\|_{OWA} = \{a+b\}$$

$$\|\langle x/\tau | Rx \rangle\|_{OWA} = \{a,b\}$$

One might also expect that all occurrences of negation can be eliminated in favour of set difference for CWA query evaluation. This is indeed the case, but only for quantifier free queries and then only when $DB \cup \overline{EDB}$ is consistent.

Theorem 3.

If W , W_1 and W_2 are quantifier free, and $DB \cup \overline{EDB}$ is consistent, then

$$1. \|\langle \vec{x}/\vec{\tau} | \vec{W} \rangle\|_{CWA} = |\vec{\tau}| - \|\langle \vec{x}/\vec{\tau} | W \rangle\|_{CWA}$$

$$2. \|\langle \vec{x}/\vec{\tau} | W_1 \wedge \vec{W}_2 \rangle\|_{CWA} = \|\langle \vec{x}/\vec{\tau} | W_1 \rangle\|_{CWA} - \|\langle \vec{x}/\vec{\tau} | W_2 \rangle\|_{CWA}$$

To see why Theorem 3 fails for quantified queries, consider the following:

Example 5

$$|\tau| = \{a,b\}$$

DB: Pa, a

$$\text{Then } \overline{EDB} = \{\overline{Pa}, b, \overline{Pb}, a, \overline{Pb}, b\}$$

$$\text{Let } Q(P) = \langle x/\tau | (Ey/\tau) Px, y \rangle$$

$$Q(\overline{P}) = \langle x/\tau | (Ey/\tau) \overline{Px}, y \rangle$$

$$\text{Then } \|Q(P)\|_{CWA} = \{a\}$$

$$\|Q(\overline{P})\|_{CWA} = \{a,b\} \neq |\tau| - \|Q(P)\|_{CWA}$$

Notice also that Theorem 3 fails under the OWA.

By an atomic query we mean any query of the form $\langle \vec{x}/\vec{\tau} | (Ey/\theta) Pt_1, \dots, t_n \rangle$ where P is a predicate sign and each t is a constant sign, an x , or a y .

Theorems 2 and 3 assure us that for quantifier free queries, CWA query evaluation can be reduced to the Boolean operations of

set intersection union and difference applied to atomic queries. However, we can deal with quantified queries by introducing the following projection operator (Codd [1972]):

Let $Q = \langle \vec{x}/\vec{\tau}, z/\psi | W \rangle$ where W is a possibly existentially quantified formula, and \vec{x} is the n -tuple x_1, \dots, x_n . Then $\|Q\|_{CWA}$ is a set of $(n+1)$ -tuples, and the projection of $\|Q\|_{CWA}$ with respect to z , $\pi_z \|Q\|_{CWA}$, is the set of n -tuples obtained from $\|Q\|_{CWA}$ by deleting the $(n+1)$ st component from each $(n+1)$ -tuple of $\|Q\|_{CWA}$. For example, if $Q = \langle x_1/\tau_1, x_2/\tau_2, z/\psi | W \rangle$ and if

$$\|Q\|_{CWA} = \{(a, b, c), (a, b, d), (c, a, b)\}$$

then

$$\pi_z \|Q\|_{CWA} = \{(a, b), (c, a)\}$$

Theorem 4.

$$\|\langle \vec{x}/\vec{\tau} | (\vec{E}\vec{y}/\vec{\theta}) W \rangle\|_{CWA} = \pi_{\vec{y}} \|\langle \vec{x}/\vec{\tau}, \vec{y}/\vec{\theta} | W \rangle\|_{CWA}$$

where $\pi_{\vec{y}}$ denotes $\pi_{y_1} \pi_{y_2} \dots \pi_{y_m}$

Corollary 4.1

1. $\|\langle \vec{x}/\vec{\tau} | (\vec{E}\vec{y}/\vec{\theta}) \bar{W} \rangle\|_{CWA} = \pi_{\vec{y}} \|\langle \vec{x}/\vec{\tau}, \vec{y}/\vec{\theta} | \bar{W} \rangle\|_{CWA}$
 $= \pi_{\vec{y}}(|\vec{\tau}| \times |\vec{\theta}| - \|\langle \vec{x}/\vec{\tau}, \vec{y}/\vec{\theta} | W \rangle\|_{CWA})$
2. $\|\langle \vec{x}/\vec{\tau} | (\vec{E}\vec{y}/\vec{\theta}) W_1 \wedge W_2 \rangle\|_{CWA} = \pi_{\vec{y}}(\|\langle \vec{x}/\vec{\tau}, \vec{y}/\vec{\theta} | W_1 \rangle\|_{CWA}$
 $\cap \|\langle \vec{x}/\vec{\tau}, \vec{y}/\vec{\theta} | W_2 \rangle\|_{CWA})$

Thus, in all cases, an existentially quantified query may be decomposed into atomic queries each of which is evaluated under the CWA. The resulting sets of answers are combined under set union, intersection and difference, but only after the projection operator is applied, if necessary.

Example 6.

$$\begin{aligned} & \|\langle x/\tau | (E y/\theta) P x, y \vee Q x, y R x, y \rangle\|_{CWA} \\ &= \|\langle x/\tau | (E y/\theta) P x, y \rangle\|_{CWA} \cup \pi_y (\|\langle x/\tau, y/\theta | Q x, y \rangle\|_{CWA} \\ & \quad \cap \|\langle x/\tau, y/\theta | R x, y \rangle\|_{CWA}) \end{aligned}$$

$$\begin{aligned}
 & \| \langle x/\tau | PxQx \vee \bar{Rx} \rangle \|_{CWA} = \| \langle x/\tau | Px \rangle \|_{CWA} \\
 & \cap \| \langle x/\tau | Qx \rangle \|_{CWA} \cup [|\tau| - \| \langle x/\tau | Rx \rangle \|_{CWA}] \\
 & \| \langle x/\tau | (Ey/\theta)Px, y \vee Qx, y \bar{Rx}, y \rangle \|_{CWA} \\
 & = \| \langle x/\tau | (Ey/\theta)Px, y \rangle \|_{CWA} \cup \pi_y(\langle x/\tau, y/\theta | Qx, y \rangle \|_{CWA}) \\
 & - \| \langle x/\tau, y/\theta | Rx, y \rangle \|_{CWA}
 \end{aligned}$$

In view of the above results, we need consider CWA query evaluation only for atomic queries.

We shall say that DB is consistent with the CWA iff $DB \cup \overline{EDB}$ is consistent.

Theorem 5.

Let Q be an atomic query. Then if DB is consistent with the CWA, $\|Q\|_{CWA} = \|Q\|_{OWA}$.

Theorem 5 is the principal result of this section. When coupled with Theorems 2 and 3 and the remarks following Corollary 4.1 it provides us with a complete characterization of the CWA answers to an arbitrary existential query Q in terms of the application of the operations of projection, set union, intersection and difference as applied to the OWA answers to atomic queries. In other words, CWA query evaluation has been reduced to OWA atomic query evaluation. A consequence of this result is that we need never know the elements of \overline{EDB} . CWA query evaluation appeals only to the given data base DB.

Example 7.

We consider the inventory data base of Example 2. Suppose the following query:

$$\begin{aligned}
 Q = & \langle x/Supplier | (Ey/Widget)Supplies x, y \wedge Subpart y, p_1 \\
 & \wedge \overline{Supplies} x, p_3 \rangle
 \end{aligned}$$

Then

$$\|Q\|_{CWA} = \pi_y(\|Q_1\|_{OWA} \cap \|Q_2\|_{OWA}) \cap (|\text{Supplier}| - \|Q_3\|_{OWA})$$

where

$$Q_1 = \langle x/Supplier, y/Widget | Supplies x, y \rangle$$

$$Q_2 = \langle x/\text{Supplier}, y/\text{Widget} | \text{Subpart } y, p_1 \rangle$$

$$Q_3 = \langle x/\text{Supplier} | \text{Supplies } x, p_3 \rangle$$

It is easy to see that

$$\|Q_1\|_{\text{OWA}} = \{(Foobar, w_1), (Foobar, w_2), (Foobar, w_3), (Foobar, w_4), (AAA, w_3), (AAA, w_4), (Acme, w_1), (Acme, w_2)\}$$

$$\|Q_2\|_{\text{OWA}} = \{(Acme, w_1), (Acme, w_2), (AAA, w_1), (AAA, w_2), (Foobar, w_1), (Foobar, w_2)\}$$

$$\|Q_3\|_{\text{OWA}} = \{\text{Acme}\}$$

whence

$$\pi_y(\|Q_1\|_{\text{OWA}} \cap \|Q_2\|_{\text{OWA}}) = \{\text{Foobar}, \text{Acme}\}$$

and

$$|\text{Supplier}| - \|Q_3\|_{\text{OWA}} = \{\text{Foobar}, \text{AAA}\}$$

Hence

$$\|Q\|_{\text{CWA}} = \{\text{Foobar}\}.$$

ON DATA BASES CONSISTENT WITH THE CWA

Not every consistent data base remains consistent under the CWA.

Example 8.

DB: $\text{Pa} \vee \text{Pb}$

Then, since $\text{DB} \not\models \text{Pa}$ and $\text{DB} \not\models \text{Pb}$, $\overline{\text{EDB}} = \{\overline{\text{Pa}}, \overline{\text{Pb}}\}$ so that $\text{DB} \cup \overline{\text{EDB}}$ is inconsistent.

Given this observation, it is natural to seek a characterization of those data bases which remain consistent under the CWA. Although we know of no such characterization, it is possible to give a sufficient condition for CWA consistency which encompasses a large natural class of data bases, namely the Horn data bases. (A data base is Horn iff every clause is Horn i.e. contains at most one positive literal. The data base of Example 2 is Horn.)

Theorem 6

Suppose DB is Horn, and consistent. Then $DB \cup \overline{EDB}$ is consistent i.e., DB is consistent with the CWA.

Following van Emden [1977] we shall refer to a Horn clause with exactly one positive literal as a definite clause. If DB is Horn, let $\Delta(DB)$ be obtained from DB by removing all non definite clauses i.e., all negative clauses. The following Theorem demonstrates the central importance of these concepts:

Theorem 7

If $Q = \langle \vec{x}/\vec{\tau} | (\vec{E}y/\vec{\theta})W \rangle$ and DB is Horn and consistent, then $\|Q\|_{CWA}$ when evaluated with respect to DB yields the same set of answers as when evaluated with respect to $\Delta(DB)$. In other words, negative clauses in DB have no influence on CWA query evaluation.

Theorem 7 allows us, when given a consistent Horn DB, to discard all its negative clauses without affecting CWA query evaluation. Theorem 7 fails for non Horn DBs, as the following example demonstrates:

Example 9

DB: $\overline{Pa} \vee \overline{Ra}, Ra \vee Sa, Pa$

Then $DB \vdash Sa$

But $\Delta(DB) = \{Ra \vee Sa, Pa\}$ and $\Delta(DB) \not\vdash Sa$.

Let us call a data base for which all clauses are definite a definite data base.

Theorem 8

If DB is definite then DB is consistent.

Corollary 8.1

If DB is definite then

(i) DB is consistent

(ii) DB is consistent with the CWA.

Corollary 8.1 is a central result. It guarantees data base and CWA consistency for a large and natural class of data bases. Since the data base of Example 2 is definite we are assured that it is consistent with the CWA.

In van Emden [1977], he addresses, from a semantic point of view, the issues of data base consistency under the CWA. He defines the notion of a "minimal model" for a data base as the intersection of all its models. If this minimal model is itself a model of the data base, then the data base is consistent with the CWA. Van Emden goes on to point out some intriguing connections between minimal models and Scott's minimal fixpoint approach to the theory of computation, results which are elaborated in van Emden and Kowalski [1976].

THE CWA AND DATA BASE INTEGRITY

Theorem 7 has an interesting consequence with respect to data base integrity. In a first order data base, both intensional and extensional facts may serve a dual purpose. They can be used for deductive retrieval, or they can function as integrity constraints. In this latter capacity they are used to detect inconsistencies whenever the data base is modified. For example, if the data base is updated with a new fact then logical consequences of this fact can be derived using the entire data base. If these consequences lead to an inconsistency, the update will be rejected.

In general, it is not clear whether a given fact in a data base functions exclusively as an integrity constraint, or for deductive retrieval, or both (Nicolas and Gallaire [1978]). However, if the data base is both Horn and closed world, Theorem 7 tells us that purely negative clauses can function only as integrity constraints. Thus the CWA induces a partition of a Horn data base into negative and non-negative clauses. The latter are used only for deductive retrieval. Both are used for enforcing integrity.

SUMMARY

We have introduced the notion of the closed world assumption for deductive question-answering. This says, in effect, "Every positive statement that you don't know to be true may be assumed false". We have then shown how query evaluation under the closed world assumption reduces to the usual first order proof theoretic approach to query evaluation as applied to atomic queries. Finally, we have shown that consistent Horn data bases remain consistent under the closed world assumption and that definite data bases are consistent with the closed world assumption.

ACKNOWLEDGMENT

This paper was written with the financial support of the National Research Council of Canada under grant A7642. Much of this research was done while the author was visiting at Bolt, Beranek and Newman, Inc., Cambridge, Mass. I wish to thank Craig Bishop for his careful criticism of an earlier draft of this paper.

APPENDIX

Proofs of Theorems

Theorem 1.

Let $Q = \langle \vec{x}/\vec{\tau} | (\vec{y}/\vec{\theta})W(\vec{x}, \vec{y}) \rangle$. Then every minimal CWA to Q is definite.

The proof requires the following two lemmas:

Lemma 1

Let w_1, \dots, w_r be propositional formulae. Then

$$DB \cup \overline{EDB} \vdash w_1 \vee \dots \vee w_r$$

$$\text{iff } DB \cup \overline{EDB} \vdash w_i \text{ for some } i.$$

Proof: The "only if" half is immediate.

With no loss in generality, assume that the set of W's is minimal, i.e., for no i do we have

$$DB \cup \overline{EDB} \vdash w_1 \vee \dots \vee w_{i-1} \vee w_{i+1} \vee \dots \vee w_r.$$

Suppose w_1 is represented in conjunctive normal form, i.e. as a conjunct of clauses. Let $C = L_1 \vee \dots \vee L_m$ be a typical such clause. Then $DB \cup \overline{EDB} \vdash L_i$ or $DB \cup \overline{EDB} \vdash \overline{L}_i$, $i=1, \dots, m$. Suppose the latter is the case for each i , $1 \leq i \leq m$. Then $DB \cup \overline{EDB} \vdash \overline{C}$ so that $DB \cup \overline{EDB} \vdash w_1$. Since also $DB \cup \overline{EDB} \vdash w_1 \vee \dots \vee w_r$, then $DB \cup \overline{EDB} \vdash w_2 \vee \dots \vee w_r$ contradicting the assumption that the set of W's is minimal. Hence, for some i , $1 \leq i \leq m$, $DB \cup \overline{EDB} \vdash L_i$ so that $DB \cup \overline{EDB} \vdash C$. Since C was an arbitrary clause of w_1 , $DB \cup \overline{EDB} \vdash w_1$ which establishes the lemma.

Lemma 2

$DB \cup \overline{EDB} \vdash (\vec{y}/\vec{\theta})W(\vec{y})$ iff there is a tuple $\vec{d} \in |\vec{\theta}|$ such that
 $DB \cup \overline{EDB} \vdash W(\vec{d})$.

Proof: The "only if" half is immediate.

Since $DB \cup \overline{EDB} \vdash (\vec{y}/\vec{\theta})W(\vec{y})$ then for tuples $\vec{d}^{(1)}, \dots, \vec{d}^{(r)} \in |\vec{\theta}|$

$$DB \cup \overline{EDB} \vdash \bigvee_{i \leq r} W(\vec{d}^{(i)})$$

The result now follows by Lemma 1.

Proof of Theorem 1:

Suppose, to the contrary, that for $m \geq 2$, $\vec{c}^{(1)} + \dots + \vec{c}^{(m)}$ is a minimal CWA answer to Q. Then

$$DB \cup \overline{EDB} \vdash_{i \leq m} (\vec{E}y/\vec{\theta}) W(\vec{c}^{(i)}, \vec{y})$$

i.e.,

$$DB \cup \overline{EDB} \vdash (\vec{E}y/\vec{\theta}) \vee_{i \leq m} W(\vec{c}^{(i)}, \vec{y})$$

so by Lemma 2 there is a tuple $\vec{d} \in |\vec{\theta}|$ such that

$$DB \cup \overline{EDB} \vdash_{i \leq m} W(\vec{c}^{(i)}, \vec{d})$$

By Lemma 1, $DB \cup \overline{EDB} \vdash W(\vec{c}^{(i)}, \vec{d})$ for some i whence $\vec{c}^{(i)}$ is an answer to Q, contradicting the assumed indefiniteness of $\vec{c}^{(1)} + \dots + \vec{c}^{(m)}$.

Theorem 2.

$$\begin{aligned} 1. \quad \| < \vec{x}/\vec{\tau} | (\vec{E}y/\vec{\theta})(W_1 \vee W_2) > \|_{CWA} &= \| < \vec{x}/\vec{\tau} | (\vec{E}y/\vec{\theta})W_1 > \|_{CWA} \\ &\quad \cup \| < \vec{x}/\vec{\tau} | (\vec{E}y/\vec{\theta})W_2 > \|_{CWA} \end{aligned}$$

$$2. \quad \| < \vec{x}/\vec{\tau} | W_1 \wedge W_2 > \|_{CWA} = \| < \vec{x}/\vec{\tau} | W_1 > \|_{CWA} \cap \| < \vec{x}/\vec{\tau} | W_2 > \|_{CWA}$$

Proof: 1. follows from Lemmas 1 and 2 and Theorem 1. The proof of 2. is immediate from Theorem 1.

Theorem 3.

If W, W_1 and W_2 are quantifier free, and $DB \cup \overline{EDB}$ is consistent, then

$$1. \quad \| < \vec{x}/\vec{\tau} | \vec{W} > \|_{CWA} = |\vec{\tau}| - \| < \vec{x}/\vec{\tau} | W > \|_{CWA}$$

$$2. \quad \| < \vec{x}/\vec{\tau} | W_1 \wedge \vec{W}_2 > \|_{CWA} = \| < \vec{x}/\vec{\tau} | W_1 > \|_{CWA} - \| < \vec{x}/\vec{\tau} | W_2 > \|_{CWA}$$

Proof: 1. The proof is by structural induction on W. Denote $\| < \vec{x}/\vec{\tau} | W > \|_{CWA}$ by $Q(W)$.

We must prove

$$Q(\vec{W}) = |\vec{\tau}| - Q(W).$$

Case 1: W is Pt_1, \dots, t_m where P is a predicate sign and t_1, \dots, t_m are terms.

Suppose $\vec{c} \in Q(\bar{W})$. Let $\Pi(\vec{c})$ be Pt_1, \dots, t_m with all occurrences of x_i replaced by c_i . Then $DB \cup \overline{EDB} \vdash \Pi(\vec{c})$. Since $DB \cup \overline{EDB}$ is consistent, $DB \cup \overline{EDB} \not\vdash \Pi(\vec{c})$, i.e. $\vec{c} \notin Q(W)$. Since $\vec{c} \in |\vec{\tau}|$, then $\vec{c} \in |\vec{\tau}| - Q(W)$, so that $Q(\bar{W}) \subseteq |\vec{\tau}| - Q(W)$. Now suppose $\vec{c} \in |\vec{\tau}| - Q(W)$. Then $\vec{c} \notin Q(W)$ so $DB \cup \overline{EDB} \not\vdash \Pi(\vec{c})$. But then $DB \cup \overline{EDB} \vdash \Pi(\vec{c})$, and since $\vec{c} \in |\vec{\tau}|$, then $\vec{c} \in Q(\bar{W})$, so that $|\vec{\tau}| - Q(W) \subseteq Q(\bar{W})$.

Case 2: W is $U_1 \wedge U_2$.

$$\begin{aligned} \text{Assume, for } i=1,2 \text{ that } Q(\bar{U}_i) = |\vec{\tau}| - Q(U_i). \\ \text{Then } Q(\bar{W}) &= Q(\overline{U_1 \wedge U_2}) \\ &= Q(\overline{U_1} \vee \overline{U_2}) \\ &= Q(\overline{U_1}) \cup Q(\overline{U_2}) \quad \text{by Theorem 2} \\ &= [|\vec{\tau}| - Q(U_1)] \cup [|\vec{\tau}| - Q(U_2)] \\ &= |\vec{\tau}| - [Q(U_1) \cap Q(U_2)] \\ &= |\vec{\tau}| - Q(U_1 \wedge U_2) \quad \text{by Theorem 2} \\ &= |\vec{\tau}| - Q(W) \end{aligned}$$

Case 3: W is $U_1 \vee U_2$.

The proof is the dual of Case 2.

Case 4: W is \bar{U} .

Assume that $Q(\bar{U}) = |\vec{\tau}| - Q(U)$. Since $Q(U) \subseteq |\vec{\tau}|$, it follows that $Q(U) = |\vec{\tau}| - Q(\bar{U})$. i.e. $Q(\bar{W}) = |\vec{\tau}| - Q(W)$.

$$\begin{aligned} Q(W_1 \wedge \bar{W}_2) &= Q(W_1) \cap Q(\bar{W}_2) \quad \text{by Theorem 2} \\ &= Q(W_1) \cap [|\vec{\tau}| - Q(W_2)] \text{ by 1.} \\ &= Q(W_1) - Q(W_2) \quad \text{since } Q(W_1) \subseteq |\vec{\tau}|. \end{aligned}$$

Theorem 4.

$$\| \langle \vec{x}/\vec{\tau} | (E\vec{y}/\vec{\theta}) W(\vec{x}, \vec{y}) \rangle \|_{CWA} = \pi_{\vec{y}} \| \langle \vec{x}/\vec{\tau}, \vec{y}/\vec{\theta} | W(\vec{x}, \vec{y}) \rangle \|_{CWA}$$

where $\pi_{\vec{y}}$ denotes $\pi_{y_1} \pi_{y_2} \dots \pi_{y_m}$

Proof:

$$\text{Suppose } \vec{c} \in \llbracket \langle \vec{x}/\vec{\tau} | (\vec{E}\vec{y}/\vec{\theta})W(\vec{x}, \vec{y}) \rangle \rrbracket_{\text{CWA}}$$

Then by definition

$$\text{DB} \cup \overline{\text{EDB}} \vdash (\vec{E}\vec{y}/\vec{\theta})W(\vec{c}, \vec{y})$$

whence by Lemma 2 there is a tuple $\vec{d} \in |\vec{\theta}|$ such that

$$\text{DB} \cup \overline{\text{EDB}} \vdash W(\vec{c}, \vec{d})$$

i.e., $\vec{c}, \vec{d} \in \llbracket \langle \vec{x}/\vec{\tau}, \vec{y}/\vec{\theta} | W(\vec{x}, \vec{y}) \rangle \rrbracket_{\text{CWA}}$

i.e., $\vec{c} \in \pi_y \llbracket \langle \vec{x}/\vec{\tau}, \vec{y}/\vec{\theta} | W(\vec{x}, \vec{y}) \rangle \rrbracket_{\text{CWA}}$

Now Suppose $\vec{c} \in \pi_y \llbracket \langle \vec{x}/\vec{\tau}, \vec{y}/\vec{\theta} | W(\vec{x}, \vec{y}) \rangle \rrbracket_{\text{CWA}}$

Then for some tuple $\vec{d} \in |\vec{\theta}|$

$$\vec{c}, \vec{d} \in \llbracket \langle \vec{x}/\vec{\tau}, \vec{y}/\vec{\theta} | W(\vec{x}, \vec{y}) \rangle \rrbracket_{\text{CWA}}$$

so that $\text{DB} \cup \overline{\text{EDB}} \vdash W(\vec{c}, \vec{d})$

$$\text{i.e., } \text{DB} \cup \overline{\text{EDB}} \vdash (\vec{E}\vec{y}/\vec{\theta})W(\vec{c}, \vec{y})$$

$$\text{i.e. } \vec{c} \in \llbracket \langle \vec{x}/\vec{\tau} | (\vec{E}\vec{y}/\vec{\theta})W(\vec{x}, \vec{y}) \rangle \rrbracket_{\text{CWA}}$$

Theorem 5.

Let Q be an atomic query. Then if DB is consistent with the CWA, $\llbracket Q \rrbracket_{\text{CWA}} = \llbracket Q \rrbracket_{\text{OWA}}$

Proof: The proof requires the following:

Lemma 3

If DB is consistent with the CWA then every atomic query has only definite OWA answers.

Proof:

Let $Q = \langle \vec{x}/\vec{\tau} | (\vec{E}\vec{y}/\vec{\theta})P(\vec{x}, \vec{y}) \rangle$ be an atomic query where $P(\vec{x}, \vec{y})$ is a positive literal. Suppose, on the contrary, that Q has an indefinite OWA answer $\vec{c}^{(1)} + \dots + \vec{c}^{(m)}$ for $m \geq 2$. Then

$$\text{DB} \vdash \bigvee_{i \leq m} (\vec{E}\vec{y}/\vec{\theta})P(\vec{c}^{(i)}, \vec{y}) \quad (1)$$

and for no i , $1 \leq i \leq m$, is it the case that $\text{DB} \vdash (\vec{E}\vec{y}/\vec{\theta})P(\vec{c}^{(i)}, \vec{y})$.

Hence, for all $\vec{d} \in |\vec{\theta}|$, $DB \not\vdash P(\vec{c}^{(i)}, \vec{d})$ $i=1, \dots, m$.

Thus $\overline{P}(\vec{c}^{(i)}, \vec{d}) \in \overline{EDB}$ for all $\vec{d} \in |\vec{\theta}|$, $i=1, \dots, m$.

Hence, $DB \cup \overline{EDB} \vdash \overline{P}(\vec{c}^{(i)}, \vec{d})$ for all $\vec{d} \in |\vec{\theta}|$, $i=1, \dots, m$ and from

(1), $DB \cup \overline{EDB} \vdash \bigvee_{i \leq m} (Ey/\vec{\theta})P(\vec{c}^{(i)}, \vec{y})$

i.e. $DB \cup \overline{EDB}$ is inconsistent, contradiction.

Proof of Theorem 5:

Let $Q = \langle \vec{x}/\vec{\tau} | (Ey/\vec{\theta})P(\vec{x}, \vec{y}) \rangle$ where $P(\vec{x}, \vec{y})$ is a positive literal. By Lemma 3 $\|Q\|_{OWA}$ consists only of definite answers. Now

$$\begin{aligned}\vec{c} \in \|Q\|_{CWA} &\text{ iff } \vec{c} \in |\vec{\tau}| \text{ and } DB \vdash (Ey/\vec{\theta})P(\vec{c}, \vec{y}) \\ \vec{c} \in \|Q\|_{CWA} &\text{ iff } \vec{c} \in |\vec{\tau}| \text{ and } DB \cup \overline{EDB} \vdash (Ey/\vec{\theta})P(\vec{c}, \vec{y})\end{aligned}$$

Hence $\|Q\|_{OWA} \subseteq \|Q\|_{CWA}$.

We prove $\|Q\|_{CWA} \subseteq \|Q\|_{OWA}$. To that end, let $\vec{c} \in \|Q\|_{CWA}$. Then $DB \cup \overline{EDB} \vdash P(\vec{c}, \vec{d})$ for some $\vec{d} \in |\vec{\theta}|$.

If $DB \vdash P(\vec{c}, \vec{d})$, then $\vec{c} \in \|Q\|_{CWA}$ and we are done.

Otherwise, $DB \not\vdash P(\vec{c}, \vec{d})$ so that $\overline{P}(\vec{c}, \vec{d}) \in \overline{EDB}$

i.e. $DB \cup \overline{EDB} \vdash P(\vec{c}, \vec{d})$ and $DB \cup \overline{EDB} \vdash \overline{P}(\vec{c}, \vec{d})$

i.e. DB is inconsistent with the CWA, contradiction.

Theorem 6.

Suppose DB is Horn, and consistent. Then $DB \cup \overline{EDB}$ is consistent, i.e. DB is consistent with the CWA.

Proof: Suppose, on the contrary, that $DB \cup \overline{EDB}$ is inconsistent. Now a theorem of Henschen and Wos [1974] assures us that any inconsistent set of Horn clauses has a positive unit refutation by binary resolution in which one parent of each resolution operation is a positive unit. We shall assume this result, without proof, for typed resolution*. Then since $DB \cup \overline{EDB}$ is an inconsistent

*Because all variables are typed, the usual unification algorithm (Robinson [1965]) must be modified to enforce consistency of types. Resolvents are then formed using typed unification. For details, see (Reiter [1977]).

Horn set, it has such a (typed) positive unit refutation. Since all clauses of \overline{EDB} are negative units, the only occurrence of a negative unit of \overline{EDB} in this refutation can be as one of the parents in the final resolution operation yielding the empty clause. There must be such an occurrence of some $\overline{U} \in \overline{EDB}$, for otherwise \overline{EDB} does not enter into the refutation in which case DB must be inconsistent. Hence, $DB \cup \{\overline{U}\}$ is unsatisfiable, i.e. $DB \vdash \overline{U}$. But then \overline{U} cannot be a member of \overline{EDB} , contradiction.

Theorem 7.

If $Q = <\vec{x}/\vec{t}|(\vec{Ey}/\vec{\theta})W>$ and DB is Horn and consistent, then $\|Q\|_{CWA}$ when evaluated with respect to DB yields the same set of answers as when evaluated with respect to $\Delta(DB)$. In other words, negative clauses in PB have no influence on CWA query evaluation.

Proof: By Theorems 2, 3, and 4 CWA query evaluation is reducible to OWA evaluation of atomic queries whenever DB is consistent. Hence, with no loss in generality, we can take Q to be an atomic query. Suppose then that $Q = <\vec{x}/\vec{t}|(\vec{Ey}/\vec{\theta})P(x,y)>$, where $P(x,y)$ is a positive literal. Denote the value of $\|Q\|_{CWA}$ with respect to DB by $\|Q\|_{CWA}^{DB}$. Similarly, $\|Q\|_{CWA}^{\Delta(DB)}$, $\|Q\|_{OWA}^{DB}$, $\|Q\|_{OWA}^{\Delta(DB)}$. We must prove $\|Q\|_{CWA}^{DB} = \|Q\|_{CWA}^{\Delta(DB)}$. Since DB is consistent and Horn, so also is $\Delta(DB)$ so by Theorem 6, both DB and $\Delta(DB)$ are consistent with the CWA. Hence, by Theorem 5, it is sufficient to prove $\|Q\|_{OWA}^{DB} = \|Q\|_{OWA}^{\Delta(DB)}$. Clearly $\|Q\|_{OWA}^{\Delta(DB)} \subseteq \|Q\|_{OWA}^{DB}$ since $\Delta(DB) \subseteq DB$. We prove $\|Q\|_{OWA}^{DB} \subseteq \|Q\|_{OWA}^{\Delta(DB)}$. To that end, let $\vec{c} \in \|Q\|_{OWA}^{DB}$. Then $DB \vdash (\vec{Ey}/\vec{\theta})P(\vec{c},\vec{y})$. Hence, as in the proof of Theorem 6, there is a (typed) positive unit refutation of $DB \cup \{P(\vec{c},\vec{y})\}$. Since DB is Horn and consistent, $\overline{F}(\vec{c},\vec{y})$ enters into this refutation, and then only in the final resolution operation which yields the empty clause. Clearly, no negative clause other than $\overline{F}(\vec{c},\vec{y})$ can take part in this refutation i.e. only definite clauses of DB enter into the refutation. Hence we can construct the same refutation from $\Delta(DB) \cup \{\overline{F}(\vec{x},\vec{y})\}$ so that $\Delta(DB) \vdash P(\vec{c},\vec{y})$ i.e. $\vec{c} \in \|Q\|_{OWA}^{\Delta(DB)}$.

Theorem 8.

If DB is definite, then DB is consistent.

Proof: Every inconsistent set of clauses contains at least one negative clause.

REFERENCES

1. Clark, K.L. [1978] Negation as Failure, In *Logic and Data Bases* (H. Gallaire and J. Minker, Eds.), Plenum Press, New York, N.Y., 1978, 293-322.
2. Codd, E.F. [1972] Relational Completeness of Data Base Sub-languages, In *Data Base Systems* (R. Rustin, Ed.), Prentice-Hall, Englewood Cliffs, N.J., 1972, 65-98.
3. Henschen, L. and Wos, L. [1974] Unit Refutations and Horn Sets, *JACM* 21, 4 (October 1974), 590-605.
4. Hewitt, C. [1972] Description and Theoretical Analysis (Using Schemata) of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot, *AI Memo No. 251*, MIT Project MAC, Cambridge, Mass., April 1972.
5. Nicolas, J.M. and Gallaire, H. [1978] Data Bases: Theory vs. Interpretation, In *Logic and Data Bases* (H. Gallaire and J. Minker, Eds.), Plenum Press, New York, 1978, 33-54.
6. Nicolas, J. M. and Syre, J.C. [1974] Natural Question Answering and Automatic Deduction in the System Syntex, *Proceedings IFIP Congress 1974*, Stockholm, Sweden, August, 1974.
7. Reiter, R. [1977] An Approach to Deductive Question-Answering, *BBN Report No. 3649*, Bolt, Beranek and Newman, Inc., Cambridge, Mass., Sept. 1977.
8. Reiter, R. [1978] Deductive Question-Answering on Relational Data Bases, In *Logic and Data Bases* (H. Gallaire and J. Minker, Eds.), Plenum Press, New York, N.Y., 1978, 149-177.
9. Robinson, J. A. [1965] A Machine Oriented Logic Based on the Resolution Principle, *JACM* 12, (January 1965), 25-41.
10. Roussel, P. [1975] PROLOG: Manuel de Reference et d'Utilisation, Groupe d'Intelligence Artificielle, U.E.R. de Luminy, Universite d'Aix-Marseille, Sept. 1975.
11. van Emden, M. H. [1977] Computation and Deductive Information Retrieval, Dept. of Computer Science, University of Waterloo, Ont., Research Report CS-77-16, May 1977.
12. van Emden, M.H. and Kowalski, R.A. [1976] The Semantics of Predicate Logic as a Programming Language, *JACM* 23, (Oct. 1976), 733-742.