A POSSIBLE WORLD SEMANTICS FOR NON-HORN DATABASES

Edward P.F. Chan

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

Research Report CS-89-47

October, 1989

A Possible World Semantics for Non-Horn Databases

Edward P.F. Chan
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

October 30, 1989

Abstract

We investigate the fundamental problem of when a ground atom in a non-Horn database is assumed false. There are basically two different approaches for inferring negative information for non-Horn databases; they are Minker's Generalized Closed World Assumption (GCWA) and Ross and Topor's Disjunctive Database Rule (DDR). DDR is proposed to overcome some problems of GCWA. However, we argue that DDR may not correctly interpret information in a non-Horn database. A closed world assumption called PWS is proposed to overcome both the problems of GCWA and DDR. We also show that for databases with no negative clauses, the problem of determining if a negative ground literal is inferred under GCWA is NP-hard, while the same problem can be solved efficiently under DDR and PWS. However, in the general case, the problem becomes NP-hard for all three inference rules. DDR interprets disjunctions of atoms inclusively while GCWA interprets disjunctions of atoms unpredictably. PWS is more flexible by allowing *both* inclusive as well as exclusive interpretations of disjunctions of atoms. We also characterize the condition under which GCWA interprets disjunctions exclusively. Throughout this discussion, we assume both the head and body of a clause consist of atoms only.

1 Introduction

Query answering in general requires both positive as well as negative information. In database applications, negative information is numerous relative to positive information. To avoid storing the vast amount of negative data in a database, negative information is represented implicitly via some inference rules. For Horn deductive databases, there is a general consesus of how to derive negative information from a database [Clark78,Reit78]. However, for non-Horn databases, the situation is less satisfactory. Minker's *Generalized Closed World Assumption (GCWA)* [Mink82] is perhaps the most widely considered closed world assumption for non-Horn databases. For instance, see [GM86,HP88,YH85]. GCWA reduces to Reiter's CWA for definite databases [YH85]. GCWA has been extended in various ways. Yahya and Henschen proposed the *Extended Generalized Closed World Assumption (EGCWA)* for nonunit negative clauses [YH85]. Under EGCWA, a negative clause C is inferred if no minimal model contains all atoms in C. EGCWA reduces to GCWA for negative ground literals. EGCWA is further generalized by allowing the closed world assumption be applied to a subset of predicates in a database [GPP86]. Recent work on stratified databases allow negative subgoals in a clause [ABW88,BH86,Lif88, Przy88,RT88,Van88]. Having negative subgoals in a clause implies relative priorities are assigned to predicates in a database; and relations in the database are computed in the order assigned. The *perfect model semantics* [Przy88] is a generalization of GCWA when negative subgoals are allowed in a clause. It reduces to GCWA when no function symbol is allowed and both the body and head of a clause are positive. All the work described above are based on variants of the *minimal model semantics* [EK76].

Stable model semantics [GL88] and *well-founded semantics* [VRS88,Van89] are attempts to assign a natural meaning to normal logic programs. Well-founded semantics is further extended to non-Horn databases by the so-called *weak well-founded semantics* and *strong well-founded semantics* [Ross89]. Strong well-founded semantics infers a subset of perfect model semantics for non-Horn databases, and treats disjunctions of atoms exclusively. Weak well-founded semantics generalizes the DDR of Ross and Topor by allowing negative subgoals in a clause. Weak well-founded semantics coincides with DDR when no negative subgoal is allowed in a clause. Thus it suffers the same problem as will be described after Example 1. Lozinskii took a different approach and proposed PWA for inferring positive, negative as well as uncertain information in a non-Horn database [Loz89]. Under PWA, positive, negative and uncertain data are inferred depend on the frequency they appear in models of a state. Consequently, a ground atom could be inferred even if it is *not provable* from the state. For instance, A is inferred under PWA for the state $\{A \leftarrow B, A \leftarrow C\}$.

Although GCWA has been studied extensively, Ross and Topor observed that this assumption has some undesirable property. In particular, GCWA (and any aforementioned closed world assumption based on the minimal model semantics) is rigid in interpreting disjunctions of atoms. Consequently, the negative information inferred under GCWA may not be the ones that we wanted.

Example 1.1 *Let the following be predicates in a database.*

EXTREMELY_DANGEROUS(x): x is extremely dangerous,

PSYCHOPATH(x): x is a psychopath and

VIOLENT(x): x is violent.

Suppose we have the following (universally quantified) general rule in the database:

VIOLENT(x) & PSYCHOPATH(x) → EXTREMELY_DANGEROUS(x).

This rule states that if a suspect is violent and is a psychopath then the suspect is extremely dangerous.

Suppose the evidence in a crime scene suggests a suspect "Smith" is either a violent person or a psychopath (or both). So this uncertain information is recorded as a disjunction of tuples VIOLENT(Smith) ∨ PSYCHOPATH(Smith) in our database. The resulting state is a non-Horn database. Suppose the query "Is EXTREMELY_DANGEROUS(Smith) false?" is posed to the database, then what is the answer to this query?

Under GCWA, the answer to this query is "yes" since the only minimal models are { VIOLENT(Smith) } and {PSYCHOPATH(Smith)}. Informally, minimal models are possible worlds under GCWA and the two tuples VIOLENT(Smith) and PSYCHOPATH(Smith) could not simultaneously be true in any possible world. Consequently, the tuple EXTREMELY_DANGEROUS(Smith) is not considered possible under GCWA and therefore is assumed false.

On the other hand, if we interpret the clause VIOLENT(Smith) ∨ PSYCHOPATH(Smith) such that it could give rise to a possible world in which both VIOLENT(Smith) and PSYCHOPATH(Smith) are true. In this possible world, EXTREMELY_DANGEROUS(Smith) is a logical consequence. Under such an assumption, the query

"Is EXTREMELY_DANGEROUS(Smith) false?"

should have the answer "no". In fact, the possible world in which both tuples are true cannot be represented directly under GCWA. □

In the above example, the disjunction *VIOLENT(Smith)* ∨ *PSYCHOPATH(Smith) cannot* be interpreted inclusively under GCWA. To overcome the problem of GCWA, an alternative metarule, called *Disjunctive Database Rule (DDR)*, was proposed by Ross and Topor [RT88]. DDR has been shown to be equivalent to the so-called *Weak Generalized Closed World Assumption* in [RLM87]. DDR interprets the head of a clause inclusively. However, we argue that DDR may not correctly infer negative information in a non-Horn database. More specifically, DDR *ignores* negative clauses in the inference process. For instance, if we add the negative clause *¬(VIOLENT(Smith) & PSYCHOPATH(Smith))* to the database in Example 1, this clause will be ignored by DDR. That is, the negative literal *¬EXTREMELY_DANGEROUS(Smith)* is not inferred under DDR. In view of this, we propose a closed world assumption called PWS which overcomes problems in both GCWA and DDR. Independently, Sakama studied the same problem and proposed a *possible model semantics* [Saka89]. It turns out that PWS and Sakama's possible model semantics are equivalent. This is proved after PWS is introduced in Section 5.

In order to evaluate various closed world assumptions, Ross and Topor [RT88] proposed six criteria for comparison. One of them is *efficiency*. Efficiency is dealing with the question of how easy to determine if a negative datum is being inferred using an inference rule. They pointed out that the efficiency criterion cannot be determined absolutely. Rather it can only be used to compare two different closed world assumptions. They left open the question of whether one of GCWA and DDR is more "efficient". In this paper, we settle this question by showing that for databases with no negative clauses, the problem of determining if a negative ground literal is inferred under GCWA is NP-hard, while the same problem can be solved efficiently under DDR and PWS. However, in the general case, the problem becomes NP-hard for all three inference rules.

Section 2 will define the necessary notation used throughout this paper. Section 3 will review results on GCWA. Section 4 will study DDR. Section 5 will define PWS and study its property. Section 6 will highlight some important relationships among the three assumptions. Section 7 shows that PWS allows both exclusive as well as inclusive interpretations of disjunctions. A condition is also identified under which GCWA interprets disjunctions of atoms exclusively. Finally, conclusions will be drawn in Section 8.

2 Definitions and Notation

In this section, we briefly introduce notation that are necessary for the discussion in the following sections. We assume familiarity with basic terminology and theory of logic programming and relational databases as found, for example, in [Lloyd87,Reit84,Ull88]. We regard a database as a special kind of first-order theory with equality but with no function symbols [Reit84].

A *clause* is a formula of the form $A_1 \vee \cdots \vee A_m \leftarrow B_1 \ \& \ \cdots \& B_n$, where A_i and B_j are atoms. All variables in the clause are assumed to be universally quantified at the front of the clause. All clauses are assumed to be non-empty. $A_1 \vee \cdots \vee A_m$ is the *head* of the clause and $B_1 \ \& \ \cdots \& B_n$ the *body*. If the head of a clause contains a single atom, i.e. m=1, it is *Horn*. If $m \geq 2$, the clause is a *disjunctive* or *non-Horn* clause. Either the head or the body (but not both) of a clause may be empty. A clause is said to be *negative* if its head is empty. Negative clauses are considered as integrity constraints in a database. Negative clauses will be written as $\neg(B_1 \ \& \ \cdots \& B_n)$. Negative clauses are needed in a non-Horn database, since they are needed to represent exclusive disjunctions in general. As we will see in Section 7, negative clauses are added to a database to simulate the exclusive disjunctions under PWS. A clause is said to be *positive* if its body is empty. A clause is said to be *mixed* if both its head and body are non-empty. A *database* is a finite set of clauses. A database is *Horn* if it consists of Horn clauses only, otherwise it is *disjunctive* or *non-Horn*. Let $\mathcal{H}$ denote the *Herbrand base*, the set of all ground atoms for a database. Any subset of $\mathcal{H}$ is called a *Herbrand interpretation* or just an *interpretation*.

Let DB be a set of clauses and M a Herbrand interpretation of DB. M is said to be a *model* of DB if DB is true in I. M is said to be *minimal* if no proper subset of M is a model of DB. Let $MM(DB)$ be $\{M \mid M$ is a minimal models of $DB\}$. DB is said to be *consistent* if a model exists for DB, otherwise DB is said to be *inconsistent*. A clause C is *derivable*, denoted by $DB \vdash C$, if every model of DB is a model of C. A ground clause $C = A_1 \vee \cdots \vee A_n$ is *positive and minimally derivable* from DB if (i) C is positive, (ii) $DB \vdash C$ and (iii) $DB \not\vdash A_1 \vee \cdots \vee A_{i-1} \vee A_{i+1} \vee \cdots \vee A_n$, for every $1 \leq i \leq n$. Some atoms in a positive and minimally derivable ground clause are true. But given the current state, there is not enough information to determine which. Let $PMGC(DB)$ denote the set of all positive and minimally derivable ground clauses of DB.

3 GCWA

For Horn databases, Reiter's CWA states that a negative ground literal $\neg L$ is inferred if L is not derivable from the database [Reit78]. This CWA is logically equivalent to adding a new components $CWA(DB) = \{\neg A \mid A$ is a ground atom not derivable from $DB\}$ to a database DB, without storing $CWA(DB)$ explicitly. It is easy to show that Reiter's CWA does not work well for non-Horn databases. Let $DB = \{A \vee B\}$. Neither A nor B is implied by DB and hence $CWA(DB) = \{\neg A, \neg B\}$. $DB \cup CWA(DB)$ is inconsistent.

Minker suggested a generalized version of Reiter's CWA, called the *Generalized Closed World Assumption (GCWA)* [Mink82]. He described this assumption by defining a semantic as well as a syntactic version of GCWA and proved their equivalence. GCWA is based on the idea of minimal models.

A Semantic Definition of GCWA. Let DB be a consistent database and A a ground atom. $\neg A$ is inferred if A is not in any minimal model of DB.

Under GCWA, minimals models are used to denote possible worlds of a database.

Example 3.1 *Let* $DB = \{A \vee B, B \vee C \vee E, D \vee E, \neg(A \& D)\}$. *What are the possible worlds represented* *by DB under GCWA? The set of minimal models MM(DB) is* $\{\{A,E\}, \{B,D\}, \{B,E\}\}$. *Under GCWA, every atom except C is true in some possible world. Hence C is assumed false in DB.* □

Minker's GCWA has a close relationship with the class of positive and minimally derivable ground clauses. Let $\mathcal{H}$ be the Herbrand base and *ATOM(PMGC)* be the set $\{ A \mid A$ is a ground atom in $C \in$ *PMGC(DB)* $\}$. The syntactic definition of GCWA is $\mathcal{H} -$ *ATOM(PMGC)*.

A Syntactic Definition of GCWA. Let *DB* be a consistent database and *A* a ground atom. $\neg A$ is inferred if *A* is in $\mathcal{H} -$ *ATOM(PMGC)*.

The following theorem establishes the equivalence of the two versions of GCWA.

Theorem 3.1 *Let DB be a consistent database and A a ground atom. A is in* $\mathcal{H} -$ *ATOM(PMGC) iff A is not in any minimal model of DB.*

[Proof]: See [Mink82]. □

Let *GCWA(DB)* $= \{\neg A \mid A \in \mathcal{H} -$ *ATOM(PMGC)*$\}$. Under GCWA, a consistent database is augmented with *GCWA(DB)*. The following are some important properties of GCWA.

Theorem 3.2 *Let DB be consistent. Then DB* $\cup$ *GCWA(DB) is also consistent.*

[Proof]: See [Mink82]. □

Theorem 3.3 *Let DB be consistent and K a positive clause. DB* $\vdash K$ *iff DB* $\cup$ *GCWA(DB)* $\vdash K$.

[Proof]: See [Mink82] and [YH85]. □

The above theorem states that we cannot derive any more positive clauses from *DB* $\cup$ *GCWA(DB)* than from *DB*. However, there is some non-positive clauses that can be proven from *GCWA(DB)* but not from *DB*. This is due to the fact that a ground atom *A* not in any minimal model of *DB* can be inferred to be false. So some negative literal can be proven from *GCWA(DB)* but not from *DB*.

Theorem 3.4 *Let DB be consistent and A a ground atom. Then DB* $\cup$ *GCWA(DB)* $\vdash \neg A$ *iff* $\neg A \in$ *GCWA(DB).*

[Proof]: "If" Trivial.

"Only if" Suppose $\neg A \notin$ *GCWA(DB)*. By Theorem 3.1, *A* is in some minimal model *M* of *DB*. Clearly, *M* is a model of *GCWA(DB)*. Hence *DB* $\cup$ *GCWA(DB)* $\nvdash \neg A$. □

Under GCWA, it is essential to find algorithms to determine if $\neg A$ is derivable, where A is a ground atom. Some work has been done on this problem and these methods are based on finding resolutions for a set of clauses [YH85,HP88]. This problem is likely to be intractable. We show this by proving its complement is NP-hard, even when the set of clauses in a database is ground and contains *no* negative clauses. For a discussion on intractable problems, interested readers please refer to [GJ79].

Theorem 3.5 *Let DB be consistent and contains only positive and mixed clasues. Let A be a ground atom. Determining if $\neg A$ is inferred under GCWA is NP-hard.*

[Proof]: By Theorem 3.4, $\neg A$ is derivable iff A is not in any minimal model of DB. We will prove that determining if A is in some minimal model of DB is NP-hard.

To prove NP-hardness, we reduce instances of the hitting set problem to our problem. Hitting set was first demonstrated to be NP-complete in [Karp72].

Let $\{S_1, \ldots, S_n\}$ be a set of non-empty subsets of a finite set S. A *hitting set H* is a subset of S for which $|H \cap S_i| = 1, \forall i$. The hitting set problem is to determine if such a set exists.

For each instance of hitting set, it is transformed to a database DB as follows. Let the set of ground atoms in our DB be $S \cup \{B_1, \ldots, B_n, C, D\}$, where $\{B_1, \ldots, B_n, C, D\}$ and S are disjoint. Let $P = \{\{E, F\} \mid E$ and F are distinct members of S_i, for some $i\}$. Four sets of clauses are in DB:

(i) For each S_i, we have $A_{i1} \vee \cdots \vee A_{iq}$, where $S_i = \{A_{i1}, \ldots, A_{iq}\}$.

(ii) For each S_i, and for each $A_{ij} \in S_i$, we have $B_i \leftarrow A_{ij}$ in our DB.

(iii) For each $\{A_{jq}, A_{jp}\} \in P$, we have $D \leftarrow A_{jq} \& A_{jp}$.

(iv) $C \vee D \leftarrow B_1 \& \cdots \& B_n$ is in DB.

DB is consistent since the set of ground atoms in DB is a model of DB. We first show some important properties about the DB.

Fact 1: Let M be a minimal model of DB. If $D \in M$ then $C \notin M$.

[Proof]: Assume otherwise. Let M be a minimal model that contains both C and D. Let $M' = M - \{C\}$. We claim M' is a model of DB. Since C occurs only in (iv) and (iv) contains D, DB is true under M'. Hence M' is a model. A contradiction to M is a minimal model of DB. □

Fact 2: If M is a model of DB, then for all B_i, $B_i \in M$.

[Proof]: Assume otherwise. Then there is a $B_i \notin M$. Since M is a model of DB, all clauses in DB are true under M. But since B_i is false and $B_i \leftarrow A_{ij}$ is true under M, $A_{ij} \notin M$, $\forall j$. But this implies $A_{i1} \vee \cdots \vee A_{iq}$ is false under M. A contradiction. □

Fact 3: If M is a model of DB, then for each i, there is at least a $A_{ij} \in S_i$ such that $A_{ij} \in M$.

[Proof]: Trivially true because of the clauses in (i). □

Fact 4: Let M be a minimal model of DB containing C. For each S_i, there is at most one A_{ij} such that $A_{ij} \in M$.

[Proof]: Suppose the two distinct ground atoms A_{ip} and A_{iq} of S_i are both in M. Then $D \leftarrow A_{ip} \& A_{iq}$ in (iii) would imply D must be true in M. By Fact 1, this is not possible since M contains C. □

We are now ready to show that a hitting set of S exists iff C is in some minimal model of DB.

"If" Let M be a minimal model of DB containing C. We claim $M \cap S$ is a hitting set. By Facts 3 and 4, our claim follows.

"Only if" If H is a hitting set, we claim $M = H \cup \{B_1, \ldots, B_n, C\}$ is a minimal model of DB. For all clauses in (i), they are true under M since by assumption H is a hitting set and therefore contains exactly one element from each S_i. For all clauses in (ii) and (iv), each of them contains either one B_i or C and hence is true under M. For each clause $D \leftarrow A_{ip} \& A_{iq}$ in (iii), H and therefore M, will contain at most one of A_{ip} and A_{iq}, hence either $\neg A_{ip}$ or $\neg A_{iq}$ is true. Therefore M is a model of DB.

We now prove M is minimal. We cannot remove any element from H or else one of the clauses in (i) will be falsified. By Fact 2, all B_i's must be in M. If C is deleted from M, then the clause $C \vee D \leftarrow B_1 \& \cdots \& B_n$ in (iv) will be falsified since all B_i's are true under M and D is not in M. Therefore M is a minimal model of DB and M contains C. □

4 Disjunctive Database Rule

In this section, we study the Disjunctive Database Rule (DDR) introduced by Ross and Topor [RT88]. DDR was proposed to overcome a problem of GCWA described in the Introduction. An equivalent definition called *Weak Generalized Closed World Assumption* is presented in [RLM87]. We first define the syntactic and fixpoint definitions of DDR. We then show that the problem of determining if a ground negative literal is derivable under DDR can be solved efficiently if a database contains no negative clauses, and is NP-hard in general. First, we require the concept of closed set of a database.

Let DB be a database and S a subset of $\mathcal{H}$. Then S is a *closed set* of DB if, for every element A of S, and for every ground instance C of a clause in DB such that A is in the head of C, there exists an atom B in the body of C such that B is in S. It is easy to see that the *greatest closed set* exists. It is obtained by the union all the closed sets of DB [RT88]. The greatest closed set of DB is denoted as *gcs(DB)*.

Example 4.1 *Let* $DB = \{A \vee B,\ C \leftarrow A \& B\}$. *By the definition of closed set, A and B cannot be in any closed set. Consequently, C cannot be in any closed set. Hence the gcs(DB) =* $\emptyset$. □

A Syntactic Definition of the DDR. Let *DB* be a database and *A* a ground atom. $\neg A$ is inferred if $A \in gcs(DB)$.

To define the fixpoint definition of DDR, we require a mapping T_{DB} from Herbrand interpretations for *DB* to itself.

Let *DB* be a database and *I* a Herbrand interpretation for *DB*. Then $T_{DB}(I) = \{ A \in \mathcal{H} \mid C$ is a ground instance of a clause in *DB*, *A* is in the head of *C*, and for all *B* in the body of *C*, $B \in I\}$. We also define $T_{DB} \uparrow 0 = \emptyset$, $T_{DB} \uparrow n+1 = T_{DB}(T_{DB} \uparrow n)$ and $T_{DB} \uparrow \omega = \cup_{n=0}^{\infty} T_{DB} \uparrow n$.

Example 4.2 *Let* $DB = \{A \vee B,\ C \vee D \vee E \leftarrow A,\ F \leftarrow C \& D,\ \neg(C \& D)\}$. *Then* $T_{DB} \uparrow 1 = \{A, B\}$. $T_{DB} \uparrow 2 = \{A, B, C, D, E\}$. $T_{DB} \uparrow 3 = \{A, B, C, D, E, F\}$. *Therefore* $T_{DB} \uparrow \omega = \{A, B, C, D, E, F\}$. *Notice that, with and without the negative clause* $\neg(C \& D)$, $T_{DB} \uparrow \omega$ *is the same.* □

A Fixpoint Definition of the DDR. Let *DB* be a consistent database and *A* is a ground atom. $\neg A$ is inferred if $A \in \mathcal{H} - T_{DB} \uparrow \omega$.

Let us denote $DDR(DB) = \{\neg A \mid A \in \mathcal{H} - T_{DB} \uparrow \omega\}$. Under DDR, a consistent database is augmented with *DDR(DB)*. DDR has properties very similar to those under GCWA.

Theorem 4.1 *The two definitions of DDR are equivalent. That is,* $gcs(DB) = \mathcal{H} - T_{DB} \uparrow \omega$.

[Proof]: See [RT88]. □

Theorem 4.2 *Let DB be consistent. Then* $DB \cup DDR(DB)$ *is also consistent.*

[Proof]: See [RT88]. □

Theorem 4.3 *Let DB be consistent and K a positive clause.* $DB \vdash K$ *iff* $DB \cup DDR(DB) \vdash K$.

[Proof]: See [RT88]. □

Lemma 4.4 *If M is a model of DB, then* $M \cap T_{DB} \uparrow \omega$ *is also a model of DB.*

[Proof]: See [RT88]. □

Theorem 4.5 *Let DB be consistent. If* $K = B_1 \vee \cdots \vee B_m \leftarrow A_1 \,\&\, \cdots \,\&A_n$ *is a non-positive clause such that* $DB \cup DDR(DB) \vdash K$ *but* $DB \not\vdash K$*, then for some* i, $\neg A_i \in DDR(DB)$.

[Proof]: Since $B_1 \vee \cdots \vee B_m \leftarrow A_1 \,\&\, \cdots \,\&A_n$ is non-positive, $n \geq 1$ and $m \geq 0$. As $DB \not\vdash K$, there is a model M of DB containing every A_i but none of B_j. Let N be $M \cap T_{DB} \uparrow \omega$. By Lemma 4.4, N is also a model of DB. Since $N \subseteq T_{DB} \uparrow \omega$, N is a model of $DB \cup DDR(DB)$. By assumption, $DB \cup DDR(DB) \vdash K$. Thus K is true under N. Since N contains no B_j, there is some A_i such that $A_i \notin N$. This implies $A_i \notin T_{DB} \uparrow \omega$. Thus, $\neg A_i \in DDR(DB)$. □

An important property of DDR that is different from GCWA is that it is *syntax-dependent*. That is, given two logically equivalent databases, the set of negative ground atoms inferred using DDR may be different.

Example 4.3 *Let* $DB_1 = \{A \vee B, A\}$ *and* $DB_2 = \{A\}$. DB_1 *and* DB_2 *are logically equivalent but* $\neg B$ *is not inferred when* DB_1 *is considered under DDR. Intuitively,* DB_1 *represents two possible worlds* $\{A\}$ *and* $\{A, B\}$ *under DDR. Since B is possibly true, it cannot be assumed false. Conversely, B is not possibly true in* DB_2 *and thus is assumed false under DDR.* □

A fundamental question is to determine if a negative ground literal is inferred under a closed world assumption. The following results show that under DDR, the problem can be solved efficiently if a database contains no negative clauses, but is NP-hard in the general case.

Theorem 4.6 *Let DB be consistent and A a ground atom.* $DB \cup DDR(DB) \vdash \neg A$ *iff* $DB \vdash \neg A$ *or* $A \in \mathcal{H} - T_{DB} \uparrow \omega$.

[Proof]: "If" Trivial.

"Only if" Follows from Theorem 4.5. □

Lemma 4.7 *Let DB be consistent and contains no negative clauses. Then* $DB \not\vdash \neg A$*, for all* $A \in \mathcal{H}$.

[Proof]: If DB contains no negative clauses, then $\mathcal{H}$ is a model of DB. □

Theorem 4.8 *Let DB be consistent and contains no negative clauses. Then determining if* $\neg A$ *is derivable under DDR can be done in time polynomial to the size of DB.*

[Proof]: By Theorem 4.6 and Lemma 4.7, $DB \cup DDR(DB) \vdash \neg A$ iff $A \in \mathcal{H} - T_{DB} \uparrow \omega$. $T_{DB} \uparrow \omega$ can clearly be computed efficiently. Hence the theorem follows. □

Theorem 4.9 *Let DB be a consistent database and A a ground atom. Then determining if $\neg A$ is derivable under DDR is NP-hard.*

[Proof]: To prove NP-hardness, we reduce instances of the hitting set problem to our problem.

Let $\{S_1, \ldots, S_n\}$ be a set of non-empty subsets of a finite set S. For each instance of hitting set, it is transformed to a database DB as follows. Let the set of ground atoms in DB be $S \cup \{B, C\}$, where $\{B, C\}$ and S are disjoint. Let $P = \{\{E, F\} \mid E$ and F are distinct members of S_i, for some $i\}$. Three sets of clauses are in DB:

(i) We have the clause $B \vee C$.

(ii) For each S_i, we have $A_{i1} \vee \cdots \vee A_{iq} \leftarrow B$, where $S_i = \{A_{i1}, \ldots, A_{iq}\}$.

(iii) For each $\{A_{jq}, A_{jp}\} \in P$, we have $\neg(A_{jq} \& A_{jp})$.

DB is consistent since DB is true under the interpretation $\{C\}$.

We first observe that $T_{DB} \uparrow \omega$ is the set of all ground atoms in our DB. By Theorem 4.6, $DB \cup DDR(DB) \vdash \neg B$ iff $DB \vdash \neg B$ or $B \in \mathcal{H} - T_{DB} \uparrow \omega$. Since $B \notin \mathcal{H} - T_{DB} \uparrow \omega$, $DB \cup DDR(DB) \vdash \neg B$ iff $DB \vdash \neg B$. We claim that $DB \vdash \neg B$ iff no hitting set exists for $\{S_1, \ldots, S_n\}$.

"If" If $DB \nvdash \neg B$, then there is a model M of DB such that $B \in M$. Since $B \in M$ and clauses in (ii) are true under M, $|M \cap S_i| \geq 1, \forall i$. If $|M \cap S_i| > 1$, then clauses in (iii) are not satisfiable, $\forall i$. Hence a hitting set exists for $\{S_1, \ldots, S_n\}$.

"Only if" If a hitting set H exists for $\{S_1, \ldots, S_n\}$, then $H \cup \{B\}$ is a model of DB. Hence $DB \nvdash \neg B$.
□

5 A Possible World Semantics PWS

DDR was proposed to overcome some problem of GCWA. However, as shown in Example 4.2 and the example below, negative clauses are *not* taken into consideration under DDR. Consequently, DDR may not correctly infer information in a non-Horn database.

Example 5.1 *Let $DB = \{D, A \vee B \leftarrow D, C \leftarrow A \& B, \neg(A \& B)\}$. Under DDR, $T_{DB} \uparrow \omega = \{A, B, C, D\}$. That is, under DDR, the set of negative ground literals added to DB is independent of the negative clause $\neg(A \& B)$. Intuitively, the negative information added to the state with $\neg(A \& B)$ should be different from the state without it.* □

In this section, we propose a closed world assumption called PWS which overcomes the problems of GCWA and DDR. In Section 5.1, we give a possible world definition and a fixpoint definition and show their equivalence. In Section 5.2, we prove that Sakama's possible model semantics [Saka89] and PWS are equivalent. Finally we study properties of PWS in Section 5.3.

5.1 A Possible World Definition and a Fixpoint Definition of PWS

In the remainder of this paper, we are interested in the semantics of a set of clauses. To simplify the discussion, we assume our databases consist of ground clauses only. That is, we assume general rules are instantiated with constants in a database. Let $DB = PC \cup MC \cup NC$, where PC, MC and NC are sets of positive ground clauses, mixed ground clauses and negative ground clauses, respectively. To understand what is the correct closed world semantics, it is essential to know what possible worlds are represented by a set of clauses.

Example 5.2 *Let* $DB = \{A \vee B, B \vee C \vee E, D \vee E, \neg(A \& D)\}$. *What are the possible worlds represented by DB? We first observe that a possible world is a set of assumed true atoms. Moreover, the set forms a model of DB.* $\{A, C, E\}$ *is a possible world, so is* $\{B, D\}$. *However,* $\{A, C, D\}$ *is not a possible world since* $\neg(A \& D)$ *is not satisfying under the interpretation* $\{A, C, D\}$. *So a possible world necessarily be a model restricted to atoms in the database.* □

Example 5.3 *Let* $DB = \{D, A \vee B \leftarrow D, C \leftarrow A \& B, \neg(A \& B)\}$. *What are the possible worlds represented by DB? The atom D should be in any possible world. If D is in a possible world, so is A or B (but not both). Since A and B cannot be true in any possible world, C should be in any possible world. Hence, the set of possible worlds is* $\{\{A, D\}, \{B, D\}\}$. □

To summarize, a possible world is a set of positive and assumed true facts and its logical consequences subject to restrictions imposed by the negative clauses in a database. We now give a formal definition of possible worlds for a DB that captures the essence that it must be a model and it contains exactly those atoms that are logical consequences of some assumed true facts.

Let $C = B_1 \vee \cdots \vee B_m \leftarrow A_1 \& \cdots \& A_n$ be a mixed clause. Then $rhs(C) = \{A_1, \ldots, A_n\}$ and $lhs(C) = \{B_1, \ldots, B_m\}$. Let $subset(lhs(C))$ denote a non-empty subset of $lhs(C)$.

Given a $DB = PC \cup MC \cup NC$, a *possible consequence (pc)* of a set of ground atoms G with respect to DB is a finite (possibly empty) sequence $s_1 : rhs(C_1) \rightarrow subset(lhs(C_1))$, ..., $s_n : rhs(C_n) \rightarrow subset(lhs(C_n))$ satisfying the following conditions:

(i) $C_i \in MC, \forall i$.

(ii) $C_i \neq C_j$, if $i \neq j$.

(iii) $rhs(C_i) \subseteq (G \cup subset(lhs(C_1)) \cup \cdots \cup subset(lhs(C_{i-1})))$, for all $1 \leq i \leq n$.

A pc $<s_1, \ldots, s_n>$ of G *covering* X if $(G \cup subset(lhs(C_1)) \cup \cdots \cup subset(lhs(C_n))) = X$, for some pc of G with respect to DB. Let $ATOM(S)$ be $\{A \mid A$ is a ground atom in some clause in $S\}$. M is a *possible world* of DB if there is a subset G of $ATOM(PC)$ and there is a pc $\xi = <s_1, \ldots, s_n>$ of G such that (i) ξ covering M and (ii) M is a model of DB. It is possible that more than one pc give rise to the same possible world.

In other words, any subset of $ATOM(PC)$ that satisfies the negative clauses could give rise to a possible world. Those atoms in a possible world that are not in $ATOM(PC)$ must be proven to be possible consequences before they are considered to be true in the possible world. Referring to the Example 5.3, C is an atom not in $ATOM(PC)$ and C cannot be proven to be true given the possible world $\{D\}$. Hence C is not part of the possible world.

Example 5.4 *Let* $DB=\{A \vee B, C \vee D \vee E \leftarrow A, F \leftarrow C \& D\}$. *Then the following are some possible worlds of DB:* $\{B\}$, $\{A,C\}$, $\{A,D\}$, $\{A,E\}$, $\{A,C,D,F\}$, $\{A,C,E\}$, $\{A,D,E\}$, $\{A,C,D,E,F\}$. $\{B\}$ *is a possible world since the empty pc gives rise to* $\{B\}$ *and* $\{B\}$ *is a model of DB.* $\{A,C,D,F\}$ *is a possible world since* $\{A\} \rightarrow \{C, D\}$ *and* $\{C, D\} \rightarrow \{F\}$ *is a pc of* $\{A\}$ *and the set is a model of DB. Notice that this set is not a minimal model of DB.* $\{A,C,D,E,F\}$ *is a possible world since a pc of* $\{A\}$ *deriving this set is* $\{A\} \rightarrow \{C, D, E\}$ *and* $\{C, D\} \rightarrow \{F\}$. *However,* $\{B,X\}$, *where X is any symbol other than B is not a possible world. For instance,* $\{A, B\}$ *is not a possible world since it is not a model for* $C \vee D \vee E \leftarrow A$. $\{A,C,D\}$ *is not a possible world since* $\{A,C,D\}$ *is not a model of DB.* $\{A,D,E,F\}$ *is not a possible world since F cannot be generated without C.* □

Example 5.5 *Let* $DB = \{A \vee F, B \vee C \leftarrow A, D \vee E \leftarrow A \& B, \neg(A \& D)\}$. *What are the possible worlds represented by DB? If A is in a possible world, so is B or C (or both). Since D cannot be in any possible world, E must be true when B and C are true. So* $\{A, B, C, E\}$ *is a possible world. Clearly* $\{F\}$ *is also a possible world.* □

A ground atom A is *true* if A is in all possible worlds. A is said to be *assumed-false* if A is not in any possible world. A is said to be *possibly-true* if A is in some but not all possible worlds of DB.

Let $PW(DB)$ = $\{W \mid W$ is a possible world of $DB\}$. $True(DB)$ =$\{ A \mid A$ is a ground atom in every possible world of $DB\}$ and $Possibly\_true(DB)$ = $\{A \mid A$ is a ground atom and A is possibly-true in $DB\}$.

Theorem 5.1 *$\cup PW(DB) = True(DB) \cup Possibly\_true(DB)$.*

[Proof]: Follows directly from the definitions of $PW(DB)$, $True(DB)$ and $Possibly\_true(DB)$. □

A Possible World Definition of the PWS. Let DB be a consistent database and A a ground atom. $\neg A$ is inferred if $A \in \mathcal{H} - \cup PW(DB)$.

So under PWS, a database is augmented with negative ground literals whose positive counterparts are not possibly true.

To define the fixpoint semantics, we need the following. A *kernel* is any subset S of $ATOM(PC)$ which is a model of PC. Let S be a set of clauses. S is said to be *r-closed*, if $\neg(A_1 \& \cdots \& A_m) \in S$ whenever $\neg B_1, \ldots, \neg B_n$ and $B_1 \vee \cdots \vee B_n \leftarrow A_1 \& \cdots \& A_m$ are in S. Given S, we could always generate its r-closed set. The r-closed set of S is denoted as S^*. We should point out that all clauses in S^* are derivable from S. Let $DNC = \{\neg W \mid \neg W$ is a negative clause in $(MC \cup NC)^*\}$.

Let $C = B_1 \vee \cdots \vee B_n \leftarrow A_1 \& \cdots \& A_m$ be a non-negative clause. The *Horn transformation of* C is the set of Horn clauses $\{B_1 \leftarrow A_1 \& \cdots \& A_m, \ldots, B_n \leftarrow A_1 \& \cdots \& A_m\}$. Suppose P is a set of clauses. The *Horn transformation of* P, denoted **Horn**(P), is the set of Horn transformations of each clause in P. A subset S of **Horn**($PC \cup MC$) is a *maximal Horn representation* of DB if (i)S contains a kernel of DB, (ii) S is consistent with DNC, (iii) $S \cup \{C\}$ is inconsistent with DNC, where $C \in$ **Horn**(MC) $- S$.

Since DB is finite, there is a finite set of maximal Horn representations. Let it be $\{MHR_1, \ldots, MHR_m\}$.

A Fixpoint Definition of the PWS. Let DB be a consistent database and A a ground atom. $\neg A$ is inferred if $A \in \mathcal{H} - \cup_{i=1}^{m} T_{MHR_i} \uparrow \omega$, where $\{MHR_1, \ldots, MHR_m\}$ is the set of maximal Horn representations of DB.

Theorem 5.2 *The two definitions of PWS are equivalent. That is,* $\cup PW(DB) = \cup_{i=1}^{m} T_{MHR_i} \uparrow \omega$.

[Proof]: $\cup PW(DB) \subseteq \cup_{i=1}^{n} T_{MHR_i} \uparrow \omega$. Let W be a possible world. $W \cap ATOM(PC)$ gives rise to W via a pc χ. Without loss of generality, we assume each $subset(lhs(C_i))$ in the sequence is a single attribute. Clearly, $W \cap ATOM(PC)$ and the clauses in χ are a subset of some maximal Horn representation MHR_i. Thus W is a subset of $T_{MHR_i} \uparrow \omega$.

$\cup PW(DB) \supseteq \cup_{i=1}^{n} T_{MHR_i} \uparrow \omega$. Clearly there is a possible consequence of the kernel of MHR_i covering $T_{MHR_i} \uparrow \omega$. What remains to be shown is that $T_{MHR_i} \uparrow \omega$ is a model of *DB*. Since the maximal Horn representation contains a kernel, $T_{MHR_i} \uparrow \omega$ satisfies the positive clauses trivially. Similarly since all negative clauses are included in *DNC* and the representation is consistent with *DNC*, all negative clauses in *DB* are satisfiable under $T_{MHR_i} \uparrow \omega$. If $T_{MHR_i} \uparrow \omega$ does not satisfy some mixed clause $B_1 \vee \cdots \vee B_n \leftarrow A_1 \& \cdots \& A_m$, then all A_i are elements of $T_{MHR_i} \uparrow \omega$, and all $\neg B_j$ are in *DNC*. But this implies that $\neg(A_1 \& \cdots \& A_m) \in DNC$. This contradicts that $T_{MHR_i} \uparrow \omega$ satisfies *DNC*. Thus $T_{MHR_i} \uparrow \omega$ is a model and hence a possible world of *DB*. □

5.2 Sakama's Possible Model Semantics and PWS are equivalent

Independently, Sakama proposed a possible model semantics for non-Horn databases [Saka89]. It turns out that Sakama's closed world semantics is equivalent to PWS. First we define Sakama's possible model semantics.

Let C: $A_1 \vee \cdots \vee A_m \leftarrow B_1 \& \cdots \& B_n$ be a disjunctive clause. Recall that $rhs(C) = \{B_1, \ldots, B_n\}$ and $lhs(C) = \{A_1, \ldots, A_m\}$. A *split clause* of C is a non-empty set of Horn clauses $\{ A \leftarrow B_1 \& \cdots \& B_n \mid A \in S$ where S is a non-empty subset of $\{ A_1, \ldots, A_m\}\}$. A *split database* of *DB* is a database obtained from *DB* by replacing each disjunctive clause $C \in DB$ by a split clause of C. A model M of *DB* is a *possible model* if M is a minimal model of a split database of *DB*.

Sakama's Possible Model Semantics. Let *DB* be a consistent database and A a ground atom. $\neg A$ is inferred if A is not in any possible model of *DB*.

We next show that Sakama's semantics is equivalent to PWS by proving possible models are equivalent to possible worlds.

Theorem 5.3 *Let DB be a consistent database. Then W is a possible model of DB iff W is a possible world of DB.*

[Proof]: "If" Let W be a possible world of *DB*. By definition, W is a model of *DB*. Next we show that W is a minimal model of a split database. The split database is constructed as follows. Since W is a possible world, $W \cap ATOM(PC)$ gives rise to W via a pc χ. Without loss of generality, we assume for each $C \in MC$ not in χ, $rhs(C)$ is not contained in W. For each clause C of PC, $ATOM(C) \cap W$ is non-empty. Replace each such C by the set of atoms$ATOM(C) \cap W$. For each $\{A_{i1}, \ldots, A_{im}\} \rightarrow \{B_{i1}, \ldots, B_{in}\}$ in the pc, replace the corresponding clause C_i by Horn clauses in $\{B_{ij} \leftarrow A_{i1} \& \cdots \& A_{im} \mid$

$B_{ij} \in \{B_{i1}, \ldots, B_{in}\}\}$. For each mixed clause *C* not in the pc, replace it by a split clause of *C*. Let *DB'* be the resulting *DB* from the above transformation. Then *DB'* is a split database of *DB*. It is easy to see that *W* is a minimal model of this split database.

"Only if" Let *M* be the minimal model of a split database *DB'* of *DB*. Let *G* be the set of ground atoms in *DB'*. Let χ be the sequence of mixed clauses of *DB'* involved in computing the minimal model. Then χ is a pc of *G* covering *M*. Since *M* is a model of *DB*, *M* is a possible world of *DB*. □

5.3 Properties of the PWS

In this subsection, we study the possible world semantics described above. We first point out that, like DDR, PWS is *syntax-dependent.* We should also point out that DDR and PWS coincide for databases containing no negative clauses. Next, we show a close relationship between models and possible worlds. By definition, a possible world is a model. The next theorem shows that a model contains a possible world. This result follows directly from the correctness of Algorithm 1.

Algorithm 1: Given a model *M* of *DB*, find a subset of *M* which is a possible world of *DB*.

Input: $DB = PC \cup MC \cup NC$ and *M*.

Output: $P \subseteq M$ and is a possible world of *DB*.

Method:

(1) Let $i=1, P = M \cap ATOM(PC)$.

(2) **While** (there is a $C \in MC$ such that $rhs(C) \subseteq P$) **do**

(3) Let C_i be C.

(4) $P = P \cup (lhs(C_i) \cap M)$.

(5) $MC = MC - \{C_i\}$.

(6) $i=i+1$.

(7) **end.**

(8) Output *P*.

We observe the following.

Fact 1: Suppose the **while** loop is executed *n* times, $n \geq 0$. Then $rhs(C_1) \rightarrow (lhs(C_1) \cap M), \ldots, rhs(C_n) \rightarrow (lhs(C_n) \cap M)$ is a pc of $M \cap ATOM(PC)$ covering *P*.

[Proof]: This follows from the condition in statement (2) and the fact that $(lhs(C_i) \cap M \neq \emptyset, \forall i$. □

Fact 2: The final *P* output is a possible world of *DB*.

[Proof]: To show this, we have to show that P is a model of DB. Since P contains $M \cap ATOM(PC)$, P is a model of PC. Suppose the **while** loop has been executed $n \geq 0$ times and let MC be partitioned in $\{C_1, \ldots, C_n\}$ and $\{D_1, \ldots, D_k\}$, where C_i's are those mixed clauses that are processed and removed from the original MC in the **while** loop. By statements (2) and (4) and the fact that M is a model, P satisfies C_i's, $\forall i$. For each D_j, $rhs(D_j)$ is not a subset of P. This means D_j is satisfying with respect to P. Therefore P is a model of MC. NC is satisfying with respect to M and since P is a subset of M, NC is satisfying with respect to P. Hence P is a model of DB. Together with Fact 1, P is a possible world of DB. □

Theorem 5.4 *Let DB be consistent and M a model of DB. Then there is a possible world P such that P* $\subseteq M$.

[Proof]: Follows from Fact 2 above. □

Corollary 5.5 *If M is a minimal model, then M is a possible world of DB.*

[Proof]: Follows from the definition of minimal model and from Theorem 5.4. □

Theorem 5.6 *Let DB be consistent and A a ground atom.* $DB \vdash A$ *iff A is in every possible world of DB.*

[Proof]: "If" If A is not derivable, then A is not in some model M of DB. By Theorem 5.4, there is a possible world P which is a subset of M and P does not contain A. A contradiction.

"Only if" If A is derivable, then A is true in every model. Hence A is true in every possible world of DB. □

Theorem 5.7 *Let DB be consistent and A a ground atom.* $DB \vdash \neg A$ *implies A is not in any possible world of DB.*

[Proof]: Follows trivially from the fact that every possible world is a model of DB. □

The following lemma characterizes when an atom in some positive clause is not in any possible world.

Lemma 5.8 *Let DB be consistent and A a ground atom in ATOM(PC). A is a ground atom not in any possible world of DB iff* $\neg A$ *is derivable from DB.*

[Proof]: "If" Follows from Theorem 5.7.

"Only if" Prove by contradiction. If $\neg A$ is not derivable from DB, then there is some model M that contains A. Since A is an element in $ATOM(PC)$, input M to Algorithm 1 will produce a possible world that contains A. A contradiction. □

$\cup PW(DB)$ denotes the set of true and possibly-true atomic facts in our database. Under PWS, a consistent database is augmented with $PWS(DB) = \{\neg A \mid A \text{ is an atom in } \mathcal{H}\text{-}\cup PW(DB)\}$.

Theorem 5.9 *Let DB be a consistent database. $DB \cup PWS(DB)$ is consistent.*

[Proof]: Since DB is consistent, there is a model M of DB. By Theorem 5.4, a subset M' of M is a possible world of DB. Hence every atom A in $\mathcal{H} - \cup PW(DB)$ is not in M'. Therefore $PWS(DB) = \{\neg A \mid A \text{ is an atom in } \mathcal{H} - \cup PW(DB)\}$ is true under M'. Hence M' is a model of $DB \cup PWS(DB)$. □

Lemma 5.10 *A positive clause K is true in every model of DB iff K is true in every minimal model of DB.*

[Proof]: See [YH85]. □

Lemma 5.11 *Let DB be consistent. M is a minimal model of DB iff M is a minimal model of $DB \cup PWS(DB)$.*

[Proof]: "Only if" Let M be a minimal model of DB. Let $\neg A \in PWS(DB)$. By Corollary 5.5, any minimal model of DB is a possible world. Hence A is not in any minimal model of DB. Hence $\neg A$ is true under M and therefore M is a model of $PWS(DB)$. Since DB is a subset of $DB \cup PWS(DB)$, M is a minimal model of $DB \cup PWS(DB)$.

"If" Suppose M is a minimal model of $DB \cup PWS(DB)$ but not a minimal model of DB. If M is not a minimal model of DB, then there is a proper subset M' of M which is a minimal model of DB. Using an arguement similar to the one in "Only if" part, M' is a minimal model of $DB \cup PWS(DB)$. A contradiction. □

Theorem 5.12 *Let DB be a consistent database and K a positive clause. $DB \cup PWS(DB) \vdash K$ iff $DB \vdash K$.*

[Proof]: By Lemma 5.11, $MM(DB) = MM(DB \cup PWS(DB))$. By Lemma 5.10, the theorem follows. □

Theorem 5.13 *Let DB be a consistent database and A a ground atom.* $DB \cup PWS(DB) \vdash \neg A$ *iff* $A \in \mathcal{H} - \cup PW(DB)$.

[Proof]: "If" Trivial.

"Only if" Assume $DB \cup PWS(DB) \vdash \neg A$. If $A \in \cup PW(DB)$, then $A \in W$, for some possible world W. By definition of possible worlds and $PWS(DB)$, W is a model of $DB \cup PWS(DB)$. A contradiction. Hence $A \notin \cup PW(DB)$. □

The following result shows that under PWS, the problem of determining if a negative ground literal is inferred can be solved efficiently if a database contains no negative clauses and is co-NP-complete in general.

Theorem 5.14 *Let DB be consistent and contains no negative clauses. Let A be a ground atom. The problem of determining if the* $\neg A$ *is inferred under PWS can be solved efficiently.*

[Proof]: By Theorem 5.13, $\neg A$ is inferred under PWS iff A is not in any possible world. If DB contains no negative clauses, then **Horn**(DB) is the only maximal Horn representation and hence $\cup PW(DB)$ can be computed efficiently. □

Theorem 5.15 *Let DB be consistent and A be a ground atom. The problem of determining if* $\neg A$ *is inferred under PWS is co-NP-complete.*

[Proof]: By Theorem 5.13, $\neg A$ is inferred iff A is not in any possible world. We will show that determining if A is in some possible world is an NP-complete problem. To show the problem is in NP, nondeterministically select a subset S of $ATOM(PC)$ and a pc ξ of S covering X, for some subset X containing A. Verify that ξ is a valid pc of S. If ξ is valid, then test if X is a model of DB. All these verifications can be done in polynomial time.

To prove NP-hardness, we reduce instances of the hitting set problem to our problem. For each instance of hitting set, a database DB is constructed as follows. Let $\{S_1, \ldots, S_n\}$ be a set of non-empty subsets of a finite set S. Let the set of ground atoms in our DB be $S \cup \{B_1, \ldots, B_n, C_1, \ldots, C_n, D\}$, where $\{B_1, \ldots, B_n, C_1, \ldots, C_n, D\}$ and S are disjoint. Let $P = \{\{E, F\} \mid E$ and F are distinct members of S_i, for some $i\}$. Four sets of clauses are in our DB:

(i) For each S_i, we have a clause $A_{i1} \vee \cdots \vee A_{iq} \vee B_i$, where $S_i = \{A_{i1}, \ldots, A_{iq}\}$.

(ii) For each S_i and for each $A_{ij} \in S_i$, we have $C_i \leftarrow A_{ij}$ in our DB.

(iii) For each $\{A_{jq}, A_{jp}\} \in P$, we have $\neg(A_{jq} \& A_{jp})$.

(iv) $D \leftarrow C_1 \ \& \cdots \& \ C_n$ is in DB.

We first show that our DB is consistent. Let an interpretation I be $\{B_1, \ldots, B_n, C_1, \ldots, C_n, D\}$. Since every clause in (i), (ii) and (iv) of our DB has at least one ground atom in the interpretation, every clause is true under I. Clauese in (iii) is true under I trivially. Hence DB is consistent.

We now ready to show that a hitting set of S exists iff D is in some possible world Q of DB.

"If" If D is in some possible world Q of DB then there is a subset T of $S \cup \{B_1, \ldots, B_n\}$ such that a pc of T covering Q. Since D is in Q and there is a unique clause in our DB containing D, namely the clause in (iv), $\{C_1, \ldots, C_n\} \rightarrow \{D\}$ is an element in the pc. Since each C_i is only derived from clauses in (ii) and nowhere else, this implies Q contains at least one element from each S_i. If there is some S_i such that Q contains more than one element, say Q contains both A_{i1} and A_{i2} from S_i, by clauses in (iii), $\neg(A_{i1} \& A_{i2})$ is a clause in our DB and cannot be satisfied by Q. Therefore Q cannot be a model of DB. Hence we can conclude that $Q \cap S$ is a hitting set of S.

"Only if" If H is a hitting set of S, then let $H \cap S_i = \{A_i\}$, $\forall i$. Then $\{A_1\} \rightarrow \{C_1\}, \ldots, \{A_n\} \rightarrow \{C_n\}$, $\{C_1, \ldots, C_n\} \rightarrow \{D\}$ is a pc of H. It can be verified trivially that $H \cup \{A_1, \ldots, A_n, C_1, \ldots, C_n, D\}$ is a model of DB and therefore is a possible world of DB. This completes our proof. □

6 GCWA, DDR and PWS

In this section, we investigate the relationships among GCWA, DDR and PWS.

Theorem 6.1 *Let DB be consistent and K a positive clause.* $DB \vdash K$ *iff* $DB \cup GCWA(DB) \vdash K$ *iff* $DB \cup DDR(DB) \vdash K$ *iff* $DB \cup PWS(DB) \vdash K$.

[Proof]: Follows from Theorems 3.3, 4.3 and 5.12. □

We have shown that under all three inference rules, the sets of positive clauses implied by a consistent database are the same. However, this is not true for non-positive clauses.

Let $NON\text{-}POS_X$ denote the set $\{ W \mid W$ is a non-positive clause that is derivable under $X\}$.

Theorem 6.2 $NON\text{-}POS_{DDR} \subseteq NON\text{-}POS_{PWS} \subseteq NON\text{-}POS_{GCWA}$.

[Proof]: Since $\cup MM(DB) \subseteq \cup PW(DB) \subseteq T_{DB} \uparrow \omega$, $DB \cup DDR(DB) \subseteq DB \cup PWS(DB) \subseteq DB \cup GCWA(DB)$. By the monotonicity property of first-order theory, $NON\text{-}POS_{DDR} \subseteq NON\text{-}POS_{PWS} \subseteq NON\text{-}POS_{GCWA}$ follows. □

In general, the inclusions are proper.

7 Inclusive and Exclusive Interpretations of Disjunctions under PWS

In real-life applications, disjunctions of atoms in a database are interpreted exclusively as well as inclusively. DDR always interprets the head of a disjunctive clause inclusively while GCWA is rigid in interpreting the head of a disjunctive clause, as was illustrated in the Introduction. PWS has the advantage over GCWA and DDR in that PWS allows both inclusive as well as exclusive interpretations of disjunctive clauses. This is accomplished by augmenting databases with appropriate negative clauses.

We first show that GCWA does not always interpret disjunctions exclusively, as was claimed in [RT88,Saka89].

Example 7.1 *Let* $DB = \{A \vee B,\ A \vee C,\ B \vee D,\ E \leftarrow A \& B\}$. $MM(DB) = \{\{A,B,E\},\{A,D\},\{B,C\}\}$. *If we interpret each disjunctions of atoms exclusively, we do not expect A and B are simultaneously true in a minimal model.* □

Let $C : A_1 \vee \cdots \vee A_m \leftarrow B_1 \ \& \ \cdots \& B_n$ be a disjunctive clause. If the head is interpreted exclusively, then exactly one A_i is true if the body of C is true in an interpretation. For each such disjunctive clause C in a database DB that is interpreted exclusively, DB is augmented with a set of negative clauses S_C, where $S_C = \{ \neg(A_i \& A_j) \mid A_i$ and A_j are distinct atoms in the head of the clause $C\}$. Let DB be a database and let $AUG(DB)$ be the database obtained by augmenting negative clauses for each disjunctive clause that is interpreted exclusively as described in above. The head of a clause is interpreted inclusively if it is not interpreted exclusively. Notice that $AUG(DB)$ may not be consistent. Suppose $AUG(DB)$ is consistent. Then an intepretation I is an *extended possible world* of DB if I is a possible world of $AUG(DB)$.

Example 7.2 *Let* $DB = \{D,\ A \vee B \leftarrow D,\ C \leftarrow A \& B\}$. *Suppose the user specifies that the clause* $A \vee B \leftarrow D$ *is interpreted exclusively. What this means is that if D is true then either A is true or B is true, but not both. Then the augmented database AUG(DB) is* $\{D,\ A \vee B \leftarrow D,\ C \leftarrow A \& B,\ \neg(A \& B)\}$. □

Next we identify a condition under which GCWA interprets heads of disjunctive clauses exclusively. An interpretation W of a DB is said to satisfy the *exclusive interpretation* condition if for every clause C in DB, if $rhs(C) \subseteq W$, then $| \ lhs(C) \cap W| = 1$.

Theorem 7.1 *Let AUG(DB) be consistent and assume every disjunctive clause is interpreted exclusively, then the following are equivalent.*

(i) W is an extended possible world of DB.

(ii) W is a minimal model of AUG(DB).

(iii) W is a minimal model of DB and W satisfies the exclusive interpretation condition.

[Proof]: By definition, W is an extended possible world of DB iff W is a possible world of $AUG(DB)$.

(i) $\Rightarrow$ (ii). If W is a possible world of $AUG(DB)$, then W is a model of $AUG(DB)$. By Algorithm 1 in Section 5.3, there is a pc $s_1{:}rhs(C_1) \rightarrow subset(lhs(C_1))$, ..., $s_n{:}rhs(C_n) \rightarrow subset(lhs(C_n))$ of $W \cap ATOM(PC)$ covering W. Because every disjunctive clause is interpreted exclusively, $|W \cap C| = 1$, for each $C \in PC$. Also for each clause C_i in the pc, $|W \cap subset(lhs(C_i))| = 1$. This implies that W is a minimal model of $AUG(DB)$.

(ii) $\Rightarrow$ (iii) W is a minimal model of $AUG(DB)$ implies W is a model of DB. If W is not a minimal model of DB, then it can be easily shown that W is not a minimal model of $AUG(DB)$. W satisfies the exclusive interpretation condition because of the negative clauses in $AUG(DB) - DB$.

(iii) $\Rightarrow$ (i) Let W be a minimal model of DB satisfying the exclusive interpretation condition. By Corollary 5.5, W is a possible world of DB. Since W satisfies the exclusive interpretation condition, W satisfies all negative clauses in $AUG(DB)$ - DB. Hence W is a possible world of $AUG(DB)$. □

Theorem 7.2 *Let AUG(DB) be consistent. If every disjunctive clause is interpreted inclusively, then W is an extended possible world of DB iff W is a possible world of DB.*

[Proof]: Since every disjunctive clause is interpreted inclusively, the augmented database $AUG(DB)$ is the same as DB. Then the theorem follows trivially from the definition of extended possible world. □

8 Conclusions

GCWA and DDR are two popular inference rules for inferring negative information in non-Horn databases. A problem with GCWA (in fact, for all closed world assumptions based on the minimal model semantics) is that inclusive disjunctions of atoms cannot always be represented. DDR tried to overcome this problem by allowing inclusive interpretation for the head of a clause. However, we argued on the semantics ground that DDR may not correctly infer negative information represented by a non-Horn database. A closed world semantics PWS was proposed to overcome problems in both GCWA and

DDR. We also showed how PWS is extended to allow *both* exclusive as well as inclusive interpretations of disjunctions of atoms.

A related fundamental question is how to answer queries under these assumptions efficiently. Work has been done on this problem [HP88,GM86,YH85]. We studied GCWA, DDR and PWS and showed that, without negative clauses, the problem of determining if a negative ground literal is inferred under DDR and PWS can be solved efficiently, but is NP-hard for GCWA. However the problem becomes NP-hard in general for DDR and PWS.

Throughout this discussion, we assume no negative literal is allowed in the body of a clause. However, our closed world semantics can be extended easily to stratified disjunctive databases in the same way as shown in [Saka89].

Acknowledgement

The author thanks H. Katsuno, Professors A.O. Mendelzon and D. Wood for various stimulating discussions. This work was supported by the Natural Sciences and Engineering Research Council of Canada.

References

[ABW88] Apt, K., Blair, H., Walker, A., "Towards a Theory of Declarative Knowledge," *Foundation of Deductive Databases and Logic Programming*, (J. Minker, Ed.), Morgan Kaufmann Publishers, Los Altos, CA, 1988, pp. 89-148.

[BH86] Bidoit, N., Hull, R., "Positivism Versus Minimalism in Deductive Databases," *Proceedings of the 5th ACM Symposium on Principles of Database Systems*, Cambridge, 1986, pp. 123-132.

[Clark78] Clark, K.L., "Negation as Failure," *Logic and Databases*, (H. Gallaire and J. Minker, Ed.), Plenum Press, New York, 1978, pp. 293-322.

[EK76] Emdem, M.H. van, Kowalski, R.A., "The Semantics of Predicate Logic as a Programming Language," *JACM, (23:4)*, 1976, pp. 733-742.

[GJ79] Garey, M.R., Johnson, D.S., *Computers and Intractability-A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.

[GL88] Gelfond, M., Lifschitz, V., "The Stable Model Semantics for Logic Programming," *Proceedings of the 5th Symposium on Logic Programming*, Seattle, 1988, pp. 1070-1080.

[GPP86] Gelfond, M., Przymusinski, H., Przymusinski, T., "The Extended Closed World Assumption and Its Relationship to Parallel Circumscription," *Proceedings of the 5th ACM Symposium on Principles of Database Systems*, Cambridge, 1986, pp. 133-139.

[GM86] Grant, J. , Minker, J., "Answering Queries in Indefinite Databases and the Null Value Problem," *Advances in Computing Theory, Vol. 3, The Theory of Databases*, (P. Kanellakis, Ed.), JAI Press, Greenwich, CT, 1986, pp. 247-267.

[HP88] Henschen, L.J., Park, H.S., "Compiling the GCWA," *Foundation of Deductive Databases and Logic Programming*, (J. Minker, Ed.), Morgan Kaufmann Publishers, Los Altos, CA, 1988, pp. 395-438.

[Karp72] Karp, R.M., "Reducibility among Combinatorial Problems," *Complexity of Computer Computations*, (R.E. Miller and J.W. Thatcher, Ed.), Plenum Press, New York, 1972, pp. 85-104.

[Lloyd87] Lloyd, J.W., *Foundations of Logic Programming*, Symbolic Computation Series, Springer-Verlag, Berlin, 1987.

[Lif88] Lifschitz, V., "On the Declarative Semantics of Logic Programs with Negation," *Foundation of Deductive Databases and Logic Programming*, (J. Minker, Ed.), Morgan Kaufmann Publishers, Los Altos, CA, 1988, pp. 177-192.

[Loz89] Lozinskii, E.L., "Plausible World Assumption," *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning*, (R. Brachman, H. Levesque, R. Reiter, Ed.), Morgan Kaufmann Publishers, Los Altos, CA, 1989, pp. 266-275.

[Mink82] Minker, J., "On Indefinite Databases and the Closed World Assumption," *Lecture Notes in Computer Science 128*, Springer-Verlag, 1982, pp. 292-308.

[Przy88] Przymusinski, T., "On the Semantics of Stratified Deductive Databases," *Foundation of Deductive Databases and Logic Programming*, (J. Minker, Ed.), Morgan Kaufmann Publishers, Los Altos, CA, 1988, pp. 193-216.

[RLM87] Rajasekar, A., Lobo, J., Minker, J., "Weak Generalized Closed World Assumption," *CS-TR-1992*, Department of Computer Science, University of Maryland, College Park, November 1987. To appear in *Journal of Automated Reasoning*.

[Reit78] Reiter, R., "On Closed World Data Bases," *Logic and Databases*, (H. Gallaire and J. Minker, Ed.), Plenum Press, New York, 1978, pp. 55-76.

[Reit84] Reiter, R., "Towards a Logical Reconstruction of Relational Database Theory," *On Conceptual Modelling*, (M. Brodie, M.L. Mylopoulos and J.W. Schmit, Ed.), Springer-Verlag, New York, 1984, pp. 163-189.

[Ross89] Ross, K.A. "The Well Founded Semantics for Disjunctive Logic Programs," to appear in *Proceedings of the 1st International Conference on Deductive and Object-Oriented Databases*, Kyoto, Japan, 1989.

[RT88] Ross, K.A., Topor, R.W., "Inferring Negative Information from Disjunctive Databases," *Journal of Automated Reasoning (4:4)*, 1988, pp. 397-424.

[Saka89] Sakama, C. "Possible Model Semantics for Disjunctive Databases," to appear in *Proceedings of the 1st International Conference on Deductive and Object-Oriented Databases*, Kyoto, Japan, 1989.

[Shep84] Shepherdson, J. C., "Negation as Failure: A Comparison of Clark's Completed Data Base and Reiter's Closed World Assumption," *Journal of Logic Programming (1:1)*, 1984, pp. 51-81.

[Ull88] Ullman, J.D., *Principles of Database and Knowledge-base Systems, Vol. I*, Computer Science Press, Rockville, Maryland, 1988.

[Van88] Van Gelder, A., "Negation as Failure Using Tight Derivation for General Logic Programs," *Foundation of Deductive Databases and Logic Programming*, (J. Minker, Ed.), Morgan Kaufmann Publishers, Los Altos, CA, 1988, pp. 193-216.

[VRS88] Van Gelder, A., Ross, K., Schlipf, J.S., "Unfounded Sets and Well-founded Semantics for General Logic Programs," *Proceedings of the 7th ACM Symposium on Principles of Database Systems*, Austin, Texas, 1988, pp. 221-230.

[Van89] Van Gelder, A. "The Alternating Fixpoint of Logic Programs with Negation," *Proceedings of the 8th ACM Symposium on Principles of Database Systems*, Philadelphia, 1989, pp. 1-10.

[YH85] Yahya, A., Henschen, L.J., "Deduction in Non-Horn Databases," *Journal of Automated Reasoning (1:2)* 1985, pp. 141-160.