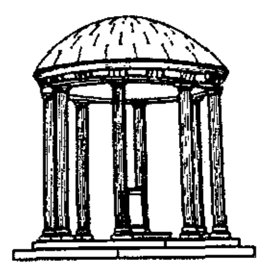
Generalized Closed World Assumption is Π_2^0-complete*

TR89-036

October, 1989

Jan Chomicki
V.S. Subrahmanian[1]



The University of North Carolina at Chapel Hill
Department of Computer Science
CB#3175, Sitterson Hall
Chapel Hill, NC 27599-3175

[1] *Dept. of Computer Science, University of Maryland, College Park, MD 20742.*

Generalized Closed World Assumption is Π_2^0-complete *

Jan Chomicki †

Dept. of Computer Science
University of North Carolina
Chapel Hill, NC 27599
chomicki@cs.unc.edu

V.S. Subrahmanian

Dept.of Computer Science
University of Maryland
College Park, MD 20742
vs@mimsy.umd.edu

October 3, 1989

Keywords: logic programming, non-Horn logic programs, negation, undecidability, arithmetical hierarchy.

1 Introduction

Minker [9] has defined an inference rule called the Generalized Closed World Assumption (GCWA). According to this rule, a negated ground atom $\neg A$ can be inferred from a *non-Horn* (called also *disjunctive* or *indefinite*) logic program P iff A is not in any minimal Herbrand model of P. GCWA, as opposed to CWA ([12]), does not lead to inconsistency and has been adopted as a standard rule for inferring negative information from a disjunctive logic program [16,3,15,7,11,6]).

In this note, we show that GCWA is Π_2^0-complete, i.e. at the second level of the arithmetical hierarchy [13]. The non-obvious part is Π_2^0-hardness. Therefore, GCWA is strictly harder than CWA which is Π_1^0-complete [2] for both Horn and non-Horn logic programs. Moreover, GCWA is strictly harder than a weaker inference rule called Weak GCWA in [10] and Disjunctive Database Rule in [14] which, like CWA, is Π_1^0-complete.

2 Preliminaries

Definite Horn logic programs [8] consist of universally-quantified clauses with exactly one positive literal. Non-Horn logic programs [9,5] consist of universally-quantified clauses with at least one positive literal.

Reiter [12] defined the *Closed World Assumption* (CWA) as the following inference rule:

---

*Submitted for publication.

†Research performed at the University of Maryland Institute for Advanced Computer Studies.

$\neg A \in CWA(P)$ iff $P \not\vdash A$

where P is a logic program and A is a ground atom belonging to the Herbrand base of P. For any definite Horn logic program P, $CWA(P)$ can be defined equivalently as:

$\neg A \in CWA(P)$ iff $M_P \not\models A$

where M_P is the least Herbrand model of P. In that case, $P \cup CWA(P)$ is consistent. Moreover, it is maximally consistent. However, in the case of non-Horn logic programs, $P \cup CWA(P)$ may be inconsistent.

Minker [9] defined the *Generalized Closed World Assumption* (GCWA) as the following inference rule:

$\neg A \in GCWA(P)$ iff $\forall K,\ P \vdash A \vee K \Rightarrow P \vdash K$

where P is a logic program, A is a ground atom, K is a disjunction of ground atoms, and A and all the disjuncts in K belong to the Herbrand base of P. He also showed that the above definition has a model-theoretic counterpart:

$\neg A \in GCWA(P)$ iff A does not belong to any minimal Herbrand model of P.

For any (Horn or non-Horn) logic program P, $P \cup GCWA(P)$ is consistent. Moreover, it is maximally consistent in the sense that adding more negative literals would result either in an inconsistency or in a new positive conclusion being derivable.

3 Main result

Theorem 3.1 *$GCWA(P)$ is a Π^0_2-complete set.*

Proof: We show first that $GCWA(P)$ is in Π^0_2. The complementary problem:

$\neg A \notin GCWA(P)$ iff $\exists K,\ P \vdash A \vee K and P \not\vdash K$

is recursively enumerable in Π^0_1, and therefore is in Σ^0_2.

We now show Π^0_2-hardness. Take an arbitrary Π^0_2 subset Q of some finitely generated Herbrand universe U that contains at least one constant and one function symbol. We will exhibit a non-Horn logic program P over the same Herbrand universe U such that

$\neg c(x) \in GCWA(P)$ iff $Q(x)$

for all x and some predicate symbol c.

By the definition of Π^0_2, there is a recursively enumerable relation R over U such that:

$Q(x)$ iff $\forall y,\ R(x,y)$.

For this relation R, there is a definite Horn logic program S such that:

$R(x,y)$ iff $S \vdash r(x,y)$

by a result of Andreka and Nemeti [1] (also [2, Theorem 7] and [4]).

We define the non-Horn logic program P postulated at the beginning of the proof in several steps.

First, we include the program S in P. Second, we introduce a new predicate symbol *term*, not appearing in S. The predicate $term(t)$ is true of any term t from U and can be defined by a finite set of definite Horn rules in an obvious way. Those rules are included in P.

Third, we introduce three new predicate symbols a,b and c, not appearing in S and defined by the following rules, the second of which is non-Horn:

$a(X,Y) \leftarrow r(X,Y).$
$a(X,Y) \vee b(X,Y) \leftarrow term(X), term(Y).$
$c(X) \leftarrow b(X,Y).$

We will show now that for all x:

$c(x)$ is not in any minimal Herbrand model of P iff $\forall y,\ S \vdash r(x,y)$

which is equivalent to:

$c(x)$ is in a minimal Herbrand model of P iff $\exists y,\ S \not\vdash r(x,y)$

and establishes the hardness result.

Assume first that:

$\forall y,\ S \vdash r(x,y).$

We will assume that $c(x)$ is in a minimal Herbrand model M_0 of P and derive a contradiction. If $c(x)$ is in a minimal Herbrand model M_0 of P, then there exists a y such that $M_0 \models b(x,y)$. Now because S is definite Horn, the original assumption implies:

$\forall y,\ M_S \models r(x,y)$

where M_S is the least Herbrand model of S. Thus also

$\forall y,\ M \models r(x,y)$

for any minimal Herbrand model M of P, because every such model has to contain M_S. Consequently,

$\forall y,\ M \models a(x,y)$

and because M is a minimal model:

$\forall y,\ M \models \neg b(x,y).$

This contradicts the fact that:

$M_0 \models b(x,y).$

We prove now the opposite direction. Assume:

$\exists y,\ S \not\vdash r(x,y)$.

We will prove that $c(x)$ is in a minimal Herbrand model of P. From the assumption follows that:

$P \not\vdash r(x,y)$

and there exists a Herbrand model M of P:

$M \not\models r(x,y)$.

Consequently, there is a minimal Herbrand model M_0 of P:

$M_0 \not\models r(x,y)$.

Then there is also a minimal Herbrand model M_1 of P such that:

$M_1 \models b(x,y)$

which implies that:

$M_1 \models c(x)$.

It is easy to see that the transformation leading from S to P is total and recursive. Therefore, given any Π^0_2 set, we have shown how to construct a non-Horn logic program P such that:

$\neg A \in GCWA(P)$ iff $A \in X$.

End of proof.

References

[1] H. Andreka and I. Nemeti. The Generalised Completeness of Horn Predicate Logic as a Programming Language. *Acta Cybernetica*, 4(1):3–10, 1978.

[2] K.R. Apt and H.A. Blair. Arithmetic Classification of Perfect Models of Stratified Programs. In R.A. Kowalski and K.A. Bowen, editors, *Fifth International Conference and Symposium on Logic Programming*, pages 766–779. MIT Press, 1988.

[3] N. Bidoit and R. Hull. Positivism vs. Minimalism in Deductive Databases. In *ACM SIGACT/SIGMOD Symposium on Principles of Database Systems*, 1986.

[4] H.A. Blair. Decidability in the Herbrand base. In *Workshop on Foundations of Deductive Databases and Logic Programming*, Washington,D.C., 1986. Manuscript.

[5] H. Gallaire, J. Minker, and J. Nicolas. Logic and Databases: a Deductive Approach. *ACM Computing Surveys*, 16(2):153–185, June 1984.

[6] M. Gelfond, H. Przymusinska, and T. Przymusinski. On the Relationship between Circumscription and Negation as Failure. *Artificial Intelligence*, 38:75–94, 1989.

[7] L.J. Henschen and H-S. Park. Compiling the GCWA in Indefinite Deductive Databases. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 193–216. Morgan Kaufmann, 1988.

[8] J.W. Lloyd. *Foundations of Logic Programming.* Springer Verlag, 2nd edition, 1987.

[9] J. Minker. On Indefinite Databases and the Closed World Assumption. In *Lecture Notes in Computer Science 138*, pages 292–308. Springer-Verlag, 1982.

[10] A. Rajasekar, J. Lobo, and J. Minker. Weak Generalized Closed World Assumption. *Journal of Automated Reasoning*, 1989. In print.

[11] A. Rajasekar and J. Minker. Procedural Interpretation of Non-Horn Logic Programs. In *International Conference on Automated Deduction.* Springer Verlag, Lecture Notes in Computer Science, 1988.

[12] R. Reiter. On closed world databases. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 55–76. Plenum Press, 1978.

[13] H. Rogers Jr. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, 1967.

[14] K.A. Ross and R.W. Topor. Inferring Negative Information from Disjunctive Databases. *Journal of Automated Reasoning*, 4(2):397–424, 1988.

[15] J.C. Shepherdson. Negation in Logic Programming. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 19–88. Morgan Kaufmann, 1988.

[16] A. Yahya and L. Henschen. Deduction in non-Horn databases. *Journal of Automated Reasoning*, 1(2):141–160, 1985.