The Inconsistency of a Belief Revision System

By
George Tsiknis
Technical Report 88-4

January 1988
Revised June 1988

Abstract

In 1987 Chern Seet developed a belief revision algorithm and a deduction system by which, he claims, default reasoning can be accomplished. We show that his deduction system is inconsistent. Some obvious corrections are suggested but the resulting system is still inconsistent. Its behaviour is similar to that of a closed-world-assumption reasoner. We examine a case in which the modified system behaves like the predicate circumscription and also has a reasonable performance. Finally, we discuss some problems pertaining to Seet's revision strategy. A similar revision algorithm for normal default logic is outlined and the use of the SET model for handling exceptions —and default reasoning— is briefly discussed.

Key words: knowledge representation, default reasoning, belief revision, natural deduction, circumscription, SET data model.

1 Introduction

Nonmonotonic or default reasoning arises naturally in most enterprises dealing with knowledge. Such reasoning sanctions inferences based on assumptions shouldn't they contradict the current knowledge and beliefs. Deductions of this kind imitate the human reasoning in some respect, and are necessary when the reasoner has incomplete knowledge of the world in concern.

Several approaches to default reasoning have emerged in the last ten years. Most of them have the flavour of an extension of a first order logic or bear a resemblance to a modal logic. Among them we distinquish the nonmonotonic logic of McDermott and Doyle [McDermott 80], Reiter's default logic [Reiter 80], McCarthy's circumscription [McCarthy 80] and Delgrande's approach using conditional logic [Delgrande 87]. Gilmore [Gilmore 87d] also gives a brief discussion of the use of the SET data model for default reasoning.

A different approach is adopted by Seet in [Seet 87]. He claims that "default reasoning can be accomplished rather naturally if an appropriate strategy of belief revision is employed". For this purpose he defines a revision algorithm which, given a set of current beliefs B and a new belief β inconsistent with B, it produces a new consistent set B' (future beliefs) which incorporates β as an exception to the old beliefs through the use of a modal operator $\Box$. Subsequently, he defines a deduction system that is used to infer conclusions with respect to the current beliefs.

The revision algorithm—in spite of its complex definition in [Seet 87]—can be summerized as follows. The current beliefs are represented by a set B of clauses which may contain $\Box$ symbols while the new information $\beta \equiv b_1 \vee \cdots \vee b_{n-1} \vee \mathbf{b_n}$ is a clause without any $\Box$. β is assumed to have more than one disjuncts and it contains a *designated* literal (written in bold above). If β is a single literal clause it is transformed—by using identity axioms—into a multi-literal one; i.e. $P(a)$ can be written as $\neg x = a \vee P(x)$ (note that equivalence is not preserved but for this algorithm such transformation is justified). If β is consistent with B it is added to B. Otherwise, for any clause $l \equiv l_1 \vee \cdots \vee l_k$ in B

that contains a literal $l_j, 1 \leq j \leq k$ such that $l_j\sigma$ is identical to $\neg b_n\sigma$ for some general unifier σ , l becomes

$$l' \equiv l_1 \vee \cdots \vee l_k \vee \Box\neg(b_1 \vee \cdots \vee b_{n-1})\sigma$$

If l contains more literals unifiable with b_n the same procedure is applied to each one. At the end β is added to B.

Example. Taken from [Seet 87]. If B is

$$Bird(x) \rightarrow Fly(x)$$
$$Penguin(x) \rightarrow Bird(x)$$
$$Penguin(p)$$

and β is $Penguin(x) \rightarrow \neg Fly(x)$, then B' is

$$Bird(x) \wedge \neg\Box Penguin(x) \rightarrow Fly(x)$$
$$Penguin(x) \rightarrow Bird(x)$$
$$Penguin(p)$$
$$Penguin(x) \rightarrow \neg Fly(x).$$

It is worth noting that since $\Box$-formulas are introduced exclusively by the revision algorithm (initially B is empty), they exist only in positive context; i.e. no clause can have formulas of the form $\neg\Box\phi$ for some formula ϕ .

Also, we want to indicate that the $\Box$-symbols are not essential for the algorithm. The future beliefs in B' are correctly represented by the set $S(B')$ (for $Strict(B)$ in [Seet 87]) that has the same clauses as B' with the exception that $\Box$'s are deleted. After all, the first clause of B' in the example should represent the belief that all birds that are not penguins fly. The $\Box$-symbols are mainly used by the reasoning system as a means for default inferences.

The *raison d'être* of this paper is to show that Seet's reasoning system is inconsistent and his revision strategy is not so appropriate as he claims. At the end we discuss how other techniques can be used for the same purpose.

2 The Default Reasoning System

In Seet's system, beliefs are expressed in a language BL, containing the symbols of a first order language L together with the $\Box$-symbol. The well formed formulas BW of BL can be defined to be the least set satisfying the condition: If w is a well formed formula of L then w as well as $\Box$w are well formed formulas of BL. Although no formal semantics for the $\Box$-formulas is specified, two deduction systems (we call them SS and DS for strict and default system respectively) are defined by Seet as follows.

Definition of SS ($\vdash_s$): For any B, $B \subseteq BW$ and any first order formula ϕ of L,

$$B \vdash_s \phi \text{ iff } S(B) \vdash \phi$$

where $\vdash$ is a classical first orded provability relation.

Note: Seet does not make any distinction between $\vdash_s$ and $\vdash$. The fact that $\{\neg \Box \phi\} \vdash_s \neg\phi$ but $\{\neg \Box \phi\} \not\vdash \neg\phi$ for any ϕ , manifests their differences. Consequently. Seet's definition of DS has two distinct interpretations captured by the next definition.

Definition of DS ($\vdash_1, \vdash_2$): DS1 is the deduction system that contains the axioms and rules of a classical first order system together with the rule

$$\frac{\phi}{\Box \phi} \tag{1}$$

while DS2 is like DS1 with the exception that rule (1) is substituted by

$$\frac{B \vdash_s \phi}{B \vdash_2 \Box \phi} \tag{2}$$

where ϕ is a formula of L and $B \subseteq BL$.

In Seet's definition of DS, rule (2)—or (1)—is presented as

"From $B \vdash \phi$ infer $\Box \phi$ "

where his $\vdash$ is our $\vdash_s$ or $\vdash$. His rule seems peculiar mainly because B is not present in the conclusion of the rule. We believe that his intention is expressed by one rule of ours.

Before we proceed, we deem it necessary to clarify that the absence of formal semantics inhibits any formal discussion of the system. We do not intend to define any formal semantics for Seet's system; we just make the clarifications required for our proofs. In the rest of this paper whenever we use the first order provability relation $\vdash$ with a set of BL formulas we assume that $\Box$-formulas are treated as atomic formulas by $\vdash$. We also point out that while (1) is the usual necessitation rule, (2) is an unusual rule which sanctions inferences like $\{\neg\Box\neg\phi\} \vdash_2 \Box\phi$ and $\{\neg\Box\phi\} \vdash_2 \Box\neg\phi$ and its soundness should be disputable. Nevertheless, the following theorem holds.

Theorem 1 *Let B be a set of BL sentences and ϕ a BL sentence.*

a. If $B \vdash_1 \phi$ then $S(B) \vdash S(\phi)$.

b. If $B \vdash_2 \phi$ then $S(B) \vdash S(\phi)$.

Proof: By proof transformation from one system to the other.

a. Suppose $B \vdash_1 \phi$ and $a_1, a_2, \ldots, a_n$ is a proof of ϕ from B in DS1. We will show that $S(a_1), \ldots, S(a_n)$ is a proof of $S(\phi)$ from S(B) in the first order system embeded in DS1. For each i, $1 \leq i \leq n$:

1. If a_i is an axion then so is $S(a_i)$.
2. If $a_i \in B$ then $S(a_i) \in S(B)$.
3. If a_i was infered from $a_{i1}, \ldots, a_{ik}$, $1 \leq ij \leq i$, by a first order inference rule then, $S(a_i)$ is infered from $S(a_{i1}), \ldots, S(a_{ik})$ by the same rule.
4. If a_i was infered from a_k, $1 \leq k < i$, by (1) then $S(a_i)$ is a repetition of $S(a_k)$— it can as well be deleted.

It is easy to verify that $S(a_1), \ldots, S(a_1)$ is a first order proof of $S(\phi)$ from S(B).

b. Suppose $B \vdash_2 \phi$ and $a_1, a_2, \ldots, a_n$ is a proof of ϕ from B in DS2. We show that $[S(a_1)], \ldots, [S(a_n)]$ is a first order proof of $S(\phi\ )$ from S(B), where $[S(a_i)]$ is defined as follows.

Subcases (1),(2) and (3) of (a) are applicable here; in each one of them $[S(a_i)] = S(a_i)$. Case (4) becomes :

4. If a_i is $\Box b$ and is infered by rule (2) and $b_1, \ldots, b_k$ is a proof of b from S(B) in the underlying first order system then $[S(a_i)]$ is $b_1, \ldots, b_k$. The verification of $[S(a_1)], \ldots, [S(a_n)]$ being a first order proof is also trivial. $\Box$

Corollary 1 *Let B be a set of sentences of BL and ϕ be a first order sentence of L. If $B \vdash_1 \phi$ or $B \vdash_2 \phi$ then $S(B) \vdash \phi$.*

Proof : Direct consequence of theorem 1 and the fact that $S(\phi)$ is identical to ϕ. $\Box$

The last corollary will be used to show that both DS systems are inappropriate. We have already mentioned that the beliefs about the world in concern, are actually described by the set S(B) while $\Box$'s are used solely to facilitate the inference of default conclusions. But the conclusions we want to sanction are always first order sentences not $\Box$-sentences. The corollary indicates that all the conclusions we could infer from B, can be inferred from S(B) without the use of $\Box$'s. Consequently, if Seet's intention is to use one of the DS systems for any kind of default reasoning, he should rather use a—well understood and generally acceptable—first order system on first order beliefs instead.

3 The Resolution System

Surprisingly enough, Seet claims that his system (either DS1 or DS2) is equivalent to a resolution system that incorporates the following rules :

1. " $\Box$-terms are not unifiable with any literal.
2. The free variables of $\Box$-terms in a clause are affected by substitutions when that clause is used during resolution.
3. A $\Box$-term $\Box\phi$ that has no more free variables is removed from the clause if by "spawing" another (he actually means "any") resolution refutation process to prove ϕ from B (he actually means S(B) as the examples indicate) the proof fails."

The first clarification we want to make is that this resolution is *not* equivalent to any one of the previous systems. If B is $\{P(a)\}$ then $B \vdash_1 \Box\, P(a)$ and $B \vdash_2 \Box\, P(a)$ but, the resolution can not refute $\neg\Box P(a)$. Conversely, if B is $\{\neg\Box P(a) \rightarrow P(b)\}$ the resolution can prove P(b) which is not the case for DS1 or DS2. Moreover, theorem 2 shows that this system —called R herein— is inconsistent.

At this point—since we talk about consistency—the introduction of formal semantics is indispensable. We regard that the following semantics is the closest formalization of the informal semantics implied in [Seet 87].

Definitions. Let B be a set of BL sentences, S(B) as previously defined, M be a first order model of S(B) and DG(M) be the *diagram* or the *semantic closure* of M, that is all sentences of L that are true in M.

The *extension* of M is the set EX(M) defined as

$$EX(M) = DG(M) \cup \{\Box\phi | \phi \in WL \wedge \phi \in DG(M)\} \cup \{\neg\Box\phi | \phi \in WL \wedge \phi \notin DG(M)\}$$

where WL denotes the sentences of L.

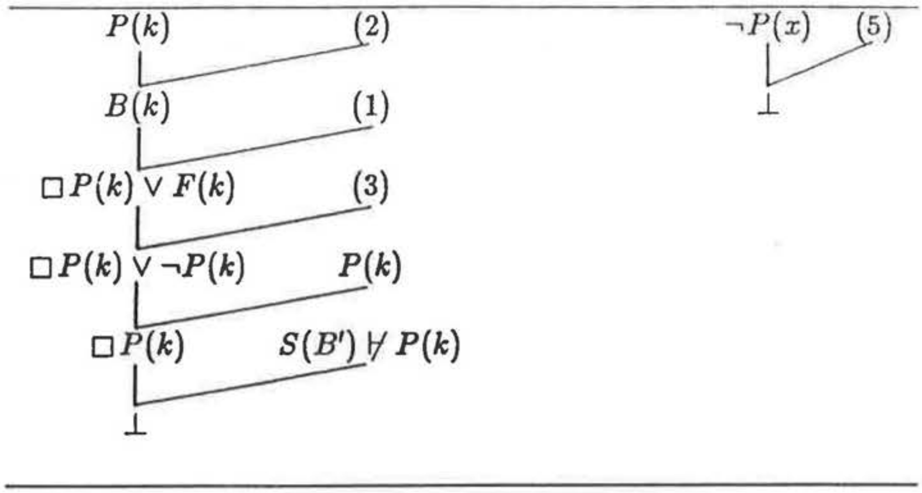
We now define the models of B. A first order structure (in the Tarskian sense) M is a model of B if M is a first order model of S(B) and for every $\phi \in B$, $\phi \in DG(EX(M))$.

Semantic entailment can also be defined as: $B \models \phi$ iff ϕ is true in all models of B which are minimal models of S(B).

Theorem 2 *The system R is inconsistent, in the sense that there is a consistent set B of sentences of BL and a sentence β such that $B \vdash_R \beta$ and $B \vdash_R \neg\beta$.*

Proof : Consider that we start with the following set of beliefs B about a world W (where B, P, F stand for Bird, Penguin, Fly):

$$(x).B(x) \rightarrow F(x)$$
$$(x).P(x) \rightarrow B(x)$$
$$B(t)$$
$$P(p)$$



(a) (b)

Figure 1: Deductions of $(x).\neg P(x)$ and $\exists x.P(x)$ from B'

Suppose we have aquired a new belief $(x).P(x) \rightarrow \neg F(x)$, which is inconsistent with B. The new set of beliefs B' becomes:

$$(x).B(x) \wedge \neg \Box P(x) \rightarrow F(x) \tag{1}$$

$$(x).P(x) \rightarrow B(x) \tag{2}$$

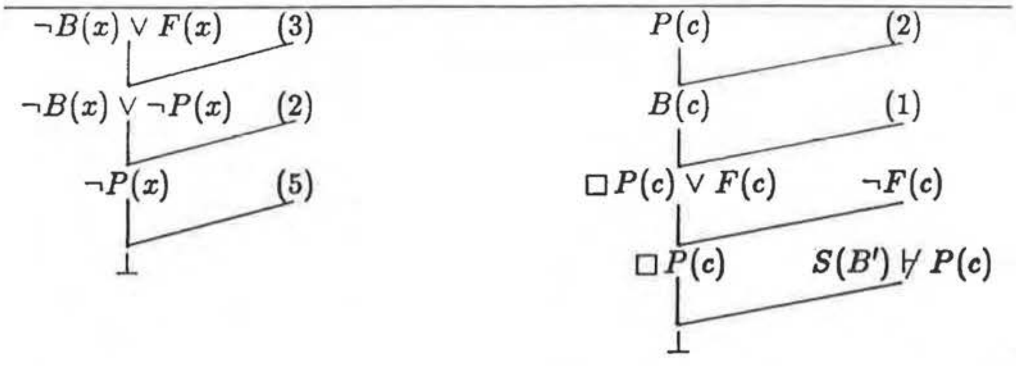
$$(x).P(x) \rightarrow \neg F(x) \tag{3}$$

$$B(t) \tag{4}$$

$$P(p) \tag{5}$$

Seet claims that this set is R-consistent, but $B' \vdash_R (x).\neg P(x)$ and $B' \vdash_R \exists x.P(x)$, as the derivations in figure 1 show. (Notation : In this report we use $\perp$ to denote contradiction.)

Three possible conclusions can be drawn from the last result : either the revision algorithm introduces inconsistencies or the R system is inconsistent or both the above. We tend to believe that—at least for this example— the first conclusion should be excluded, and B' should be considered to be consistent. An argument to support the last conjecture is the following.



(a) (b)

Figure 2: Refutations of $(x).B(x) \rightarrow F(x)$ and $P(c) \wedge \neg F(c)$

The set B' is consistent with respect to the given semantics. The interpretation

$$\{+B(t), +P(t), +B(p), -F(p), +F(t)\} \cup \{-P(trm), -B(trm) | trm \neq p \wedge trm \neq t\}$$

where trm is any term in the language, is a model of B.

Consequently we have established the claim that the resolution system is inconsistent.

$\Box$

Also Seet proudly shows that (the same set B') $B' \vdash_R (x).B(x) \rightarrow F(x)$. But

$$B' \cup \{(x).B(x) \rightarrow F(x)\} \vdash_R \perp$$

as figure 2a shows, while figure 2b exhibits a refutation of $P(c) \wedge \neg F(c)$ which witnesses that $B' \vdash_R P(c) \rightarrow F(c)$.

The problem with the refutation in figure 1a is at the last step. In order to refute $\Box P(k)$ we need to show that $S(B') \cup \{\neg P(k)\}$ is consistent. But k is a Skolem function. The full sentence that corresponds to $P(k)$ is $\exists x P(x)$. Its negation is $(x) \neg P(x)$ —quite different than $\neg P(k)$ —which is inconsistent with S(B') and $\Box P(k)$ can not be refuted. Consequently, the rule (3) of R should be changed to (3') :

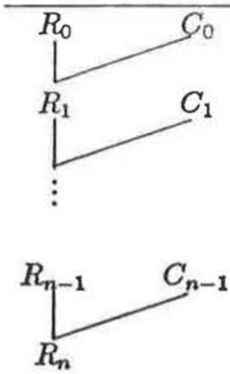


Figure 3: A linear resolution proof

3'. A $\Box$-formula $\Box\phi$ can be removed from the clause if $S(B) \cup \{\neg I(\phi)\}$ is consistent, where $I(\phi)$ is the formula resulted from ϕ by applying inverse Skolemization.

Inverse Skolemization does not always recover the original formula, and well known problems are associated with it. We definately do not intend to deal with such problems in this paper. Simply we assume that for the cases we consider it is feasible by tracing back the refutation procedure.

Note that any proof in this system is not constructive (can not be checked) mainly because of clause(3'). We modify it to (3").

3". A $\Box$-formula $\Box\phi$ is removed from the clause if there exists a Smullyan Open Tree [Smullyan 68], $SOT[S(B) \rightarrow \phi]$, which shows that $S(B) \cup \{\neg\phi\}$ is satisfiable.

A linear resolution proof of a first order sentence β from B has the form of figure 3, where :

1. R_0 is a clause of $\neg\beta$
2. For each i, $1 \leq i \leq n$, R_i is either
   (a) the usual resolvent of R_{i-1} and C_{i-1} if the resolved literals are $L(t)$ and $\neg L(t')$ or $\Box\phi(t)$ and $\neg\Box\phi(t')$, where L is a predicate symbol, ϕ a first order formula and t,t' unifiable terms, or

(b) the paramodulant of R_{i-1} and C_{i-1}, or

(c) $l_1 \vee \cdots \vee l_{i-1} \vee l_{i+1} \vee \cdots \vee l_n$ if R_{i-1} is $l_1 \vee \cdots \vee l_n$, C_{i-1} is $SOT[S(B) \rightarrow \phi]$ and l_i is $\Box\phi$.

3. For $0 \leq i \leq n-1$, $C_i \in B$ or C_i is a clause of $\neg\beta$ or C_i is R_j for some $j < i$ or C_i is a Smullian tree.

4. R_n is $\perp$, the empty clause.

The modified resolution system — denoted by R from now on — is equivalent to a deduction system that incorporates the axioms and rules of classical first order logic together with the (assumption) rule

$$S(B) \not\vdash \phi \text{ infer } B \vdash_R \neg\Box\phi$$

for any set B of BL formulas and any L formula ϕ .

The last rule is nothing more than the Closed World Assumption (CWA). The exemption is that in this case CWA is applied to $\Box$-formulas only. Because of its relation to CWA, the resolution system is expected to have most of the severe repercussions that his relative has.

It is worth noting that $\Box$-formulas are treated as atomic until case (3) of the definition of R is applied. Also note the absence of the $\Box$-introduction rule ($\neg\Box\phi$ refutation rule) in the definition of R. The latter implies further restrictions on the language, namely, formulas of the form $\Box\phi$ can only occur in positive context. Actually, the revision process introduces $\neg\Box\phi$ formulas only in the antecedent of an implication. In this restricted language the peculiar rule (3)—together with the reference to both sets B and S(B)—is equivalent to an axiom

$$\Box\phi \rightarrow \phi \qquad \text{(and its dangerous consequences)}$$

and a rule

$$\frac{\not\vdash \phi}{\neg\Box\phi}$$

Nevertheless, the following theores indicates that the system is inappropriate.

Theorem 3 *The modified resolution system R is inconsistent.*

Proof : Let B be the following set of sentences:

$$B(x) \wedge \neg \Box P(x) \wedge \neg \Box O(x) \rightarrow F(x) \tag{1}$$

$$P(x) \rightarrow B(x) \tag{2}$$

$$P(x) \rightarrow \neg F(x) \tag{3}$$

$$O(x) \rightarrow B(x) \tag{4}$$

$$O(x) \rightarrow \neg F(x) \tag{5}$$

$$P(t) \vee O(t) \tag{6}$$

B should be considered to be a consistent set. B has two models, one that includes $+O(t)$ and $-P(t)$, and another that has $-O(t)$ and $+P(t)$.

But $B \vdash_R F(t)$ and $B \vdash_R \neg F(t)$ as figure 4 shows.

$\Box$

We need the following definitions before we proceed :

Definitions . Given a finite set A of first order sentences and a predicate symbol $P(\overline{x})$ occuring in A, we denote by $C(A, P, Q)$ the *circumscription schema for P in A*

$$(A(Q) \wedge (\overline{x})(Q(\overline{x}) \rightarrow P(\overline{x})) \rightarrow (\overline{x}).P(\overline{x}) \rightarrow Q(\overline{x})$$

where $\overline{x}$ is $x_1, \ldots, x_n$, Q is a new symbol not in A (ranging over the first order formulas with n free variables) and A(Q) is the result of substituting Q for P everywhere in A.

Also, let $K(A, P, Q)$ be $A \cup \{C(A, P, Q)\}$.

We define P to be *implicitly c-definable in A by* Φ if there exists a first order formula $\Phi(\overline{x})$ with free variables $\overline{x}$ such that

$$K(A, P, \phi) \vdash (\overline{x}).P(\overline{x}) \leftrightarrow \Phi(\overline{x})$$

We call the last formula the *completion of P in A by* Φ and is denoted by $CF(A, P, \Phi)$.

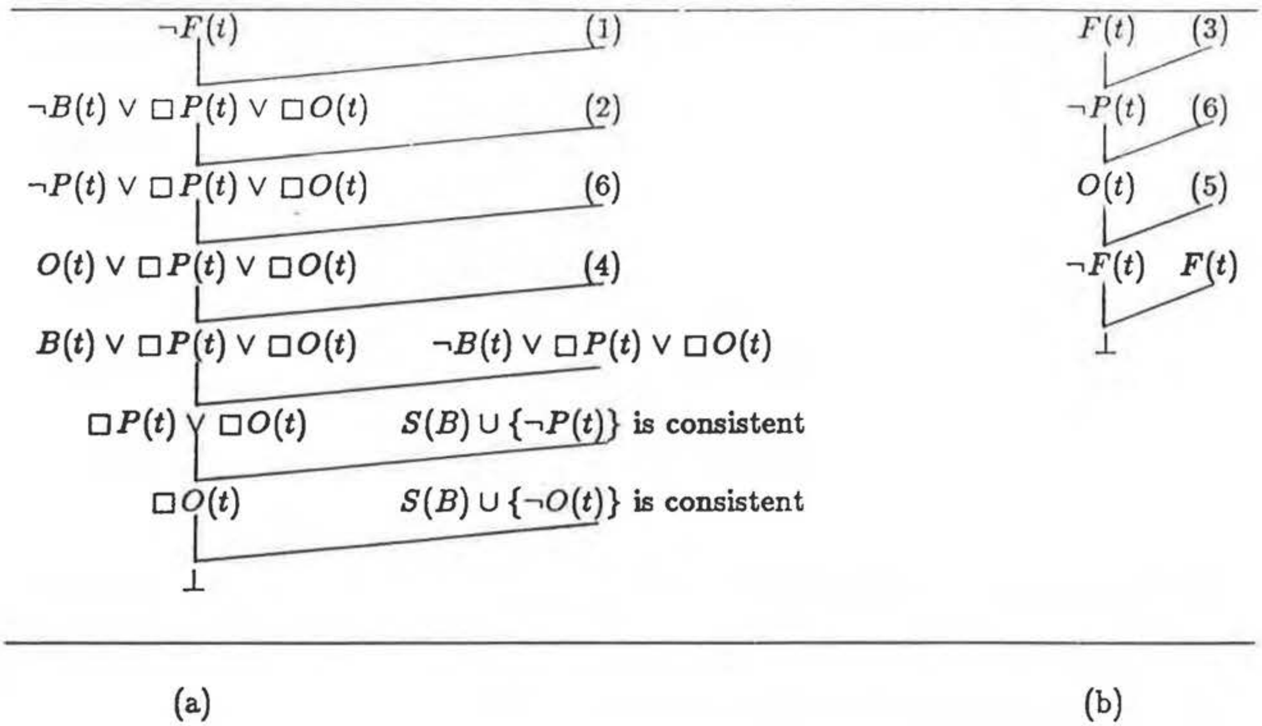


Figure 4: Deductions of $F(t)$ and $\neg F(t)$ from B

P is said to be *c-decidable in A by* Φ if P is implicitly c-definable in A by Φ and Φ is decidable in A; that is, for any variable free term $\bar{t}$, $A \vdash \Phi(\bar{t})$ or $A \vdash \neg\Phi(\bar{t})$. $\Box$

Note that by using [Davis 80] results, it can be shown that if P is c-decidable, A has a unique minimal model with respect to P. Also, if A is consistent and P is c-decidable in A by Φ, $A \cup \{CF(A, P, \Phi)\}$ is consistent.

Lemma 1 *Let A be a set of first order sentences. If $P_1, \ldots, P_n$ are all c-decidable in A, any sentence ϕ whose all predicate symbols are among the P_i's is decidable in $\cup\{CF(P_i, \Phi_i)\} \cup A$.*

Proof : By simple induction on the length of ϕ. $\Box$

Theorem 4 *Let B be a set of BL clauses, and $P_1, \ldots, P_n$ all the predicate symbols that occur inside the scope of a $\Box$-symbol in B. Suppose there exist first order formulas $\Phi_1, \ldots, \Phi_n$ such that P_i is c-decidable by Φ_i in S(B), and let*

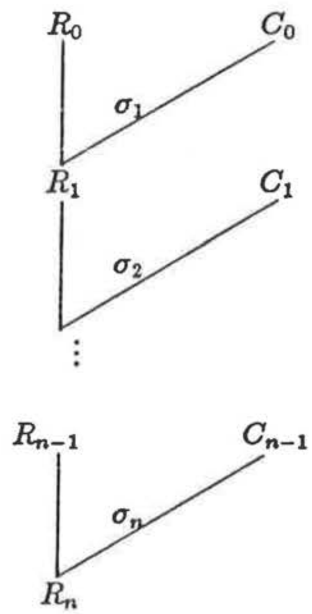
$$SC(B) = \cup CF(S(B), P_i, \Phi_i) \cup S(B)$$

For any sentence ϕ of BL if $B \vdash_R \phi$ then $SC(B) \vdash S(\phi)$.

Proof. If SC(B) is inconsistent, the theorem is vacuously true. We assume that SC(B) is consistent and $\vdash$ is the provability relation of the Gentzen classical logic LK with identity. If l is a clause of B , by ()l we denote the universal closure of l and by ()B the set that results from B if every l in B is replaced by ()l.

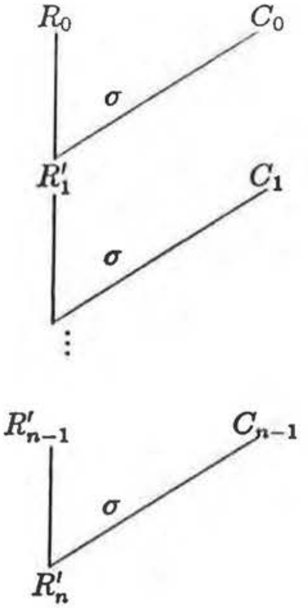
We will show, by induction on the length of the resolution proof that if $B \vdash_R \phi$ then $()S(B) \vdash ()S(\phi)$. Note that every clause $()l$ of ()S(B) is represented as $\Rightarrow ()l$ in Gentzen syntax.

Let



be a linear resolution proof of ϕ from B where σ_i is the unifier used at the i-th step (it is empty if the step corresponds to (2.3) of the definition of R.

Let $\sigma_0 = \sigma_n \sigma_{n-1} \cdots \sigma_1$ be the composition of the substitutions $\sigma_n, \ldots, \sigma_1$, and σ be the ground substitution that results from σ_0 by substituting any free variable in σ_0 by a constant c not occurring in B. Then



is a (ground) R-proof of ϕ from B (R'_i is $\sigma(R_i)$).

Now consider the i-th step of this proof. We show how it can be transformed into a Gentzen proof step from ()S(B).

a. Step i corresponds to step (2a) or (2b) of the definition of R. Then $R_{i-1} \equiv \sigma(L \vee l(t))$ where $L \equiv l_1 \vee \cdots \vee l_n$, $C_{i-1} \equiv C \vee \neg l(t')$ where $C \equiv c_1 \vee \cdots \vee c_m$, and $R_i \equiv \sigma(L \vee C)$, and the step is replaced by the derivation:

$$\dfrac{\dfrac{\Rightarrow (L' \vee l'(t)) \quad \dfrac{\sigma(L' \vee l'(t)) \Rightarrow \sigma(L' \vee l'(t))}{()(L' \vee l'(t)) \Rightarrow \sigma(L' \vee l'(t))}}{\Rightarrow \sigma(L') \vee \sigma(l'(t))} \qquad \dfrac{\dfrac{\dfrac{\sigma(C' \vee \neg l'(t')) \Rightarrow \sigma(C' \vee \neg l'(t'))}{()(C' \vee \neg l'(t')) \Rightarrow \sigma(C' \vee \neg l'(t'))} \quad \Rightarrow ()(C' \vee \neg l'(t'))}{\Rightarrow \sigma(C' \vee \neg l'(t'))}}{\sigma(l'(t')) \Rightarrow \sigma(C')}}{\Rightarrow \sigma(L') \vee \sigma(C')}$$

where, $L' \equiv S(L)$, $C' \equiv S(C)$ and $l' \equiv S(l)$. Moreover, the part in the left box is omitted if $i \neq 1$ and the right box is substituted by the proof of R'_k if C_{i-1} is R'_k, $k < i - 1$.

b. If the i-th step corresponds to (2c) then $R'_{i-1} \equiv \sigma(L) \vee \sigma(\Box\phi)$, $R_i \equiv \sigma(L)$, and C_{i-1} is a SOT showing that $S(B) \cup \{\neg\phi\}$ is consistent. But, by the hypothesis and lemma 2 $\neg\sigma(\phi)$ is provable from SC(B). Consequently, the step is replaced by the derivation:

$$\dfrac{\Rightarrow \sigma(L) \vee \sigma(\phi) \qquad \dfrac{\dfrac{T}{\Rightarrow \neg\sigma(\phi)}}{\sigma(\phi) \Rightarrow}}{\Rightarrow \sigma(L)}$$

where T is a proof tree of $\Rightarrow \neg\sigma(\phi)$.

□

Lemma 2 *Under the same assumptions of theorem 3, if SC(B) is consistent B is R-consistent.*

Proof. Suppose B is R-inconsistent. Then $B \vdash_R \phi$ and $B \vdash_R \neg\phi$ for some BL-formula ϕ. By theorem 3, $SC(B) \vdash S(\phi)$ and $SC(B) \vdash \neg S(\phi)$.

□

The last result shows that the resolution system behaves properly when all the predicates in the scope of a □ have an explicit definition when circumscribed. If this is not the case, the system might be inconsistent. A witness of the last claim is the example in theorem 3 where neither P nor Q have an explicit definition if they get circumscribed in S(B).

In the end, we would like to point out that with the semantics we described earlier in this section, we can not envision any modification that results to a sound system. By changing the semantics and assuming that $\Box\phi$ is an abbreviation of $\neg M \neg\phi$, with M being the consistency operator of [McDermott 80], the logic system becomes a special case of McDermott's non-monotonic logic, but no proof procedure is known to exist for the general case.

4 The Revision Strategy

In this section we discuss the revision algorithm of [Seet 87] but we drop the □-symbols all together. □-symbols were used solely (though unsuccessfully) to enable default reasoning; they do not contribute anything in eliminating inconsistencies. Consequently, the desired function of the algorithm can be described as : Given a consistent set B of first order sentencies and a sentence β , inconsistent with B, revise B into B' such that $B' \cup \{\beta\}$is consistent and for any formula ϕ , irrelevant to β, if $B \vdash \phi$ then $B' \vdash \phi$.

In general, the problem of revision is known to be undecidable. Even for simple database theories the similar problem of updates is hard especially in the presence of incomplete information. The impression [Seet 87] gives to the reader that the strategy presented therein solves this problem is highly misleading. He avoids any discussion about the classes of formulas the algorithm works appropriately with. Note that well before the revision algorithm is triggered, the consistency of $B \cup \{\beta\}$ has to be decided. If $B \cup \{\beta\}$ does not belong to some decidable class of formulas its satisfiability is an unsolvable problem.

Nevertheless,we now proceed to present some of the most obvious problems pertaining to Seet's revision algorithm. More problems might be revealed should more complicated cases be examined.

a. *Designated Literals.* The revision algorithm requires that any clause β that represents a newly acquired conflicting knowledge should contain a designated literal u. u indicates the predicate which the rest of the clause is an exception to. This designated literal can not always be distinquishable or—even worst—may not be in β. Consider B to be the following set of sentences (where "O" stands for Ostrich)

$$(x).B(x) \wedge \neg P(x) \wedge \neg O(x) \rightarrow F(x) \tag{1}$$

$$(x).P(x) \rightarrow B(x) \tag{2}$$

$$(x).O(x) \rightarrow B(x) \tag{3}$$

$$(x).P(x) \rightarrow \neg F(x) \tag{4}$$

$$(x).O(x) \rightarrow \neg F(x) \tag{5}$$

$$B(t) \tag{6}$$

$$F(t) \tag{7}$$

Let β be $O(t) \vee P(t)$. If we let u be $O(t)$ then the algorithm changes the clauses (3) and (6) to

$$(x).O(x) \wedge P(t) \rightarrow B(x)$$
$$(x).O(x) \wedge P(t) \rightarrow \neg F(x)$$

but still the new set is inconsistent. Similar results are obtained by leting u to be $P(t)$. The problem here is that β is an exception to $F(t)$ which should be the designated literal! Also note that a simple deletion of (7) is enough to restore consistency. The peculiarity described in this papagraph is also a subcase of the next problem.

b. *Implicit Exceptions.* When the exceptions are explicit i.e. when a specific bird namely tweety, or a whole class of birds, like ostriches, does not fly, the algorithm can easily cope with it. The problem arises when the exception is indirect (implicit). The previous example illustrates one kind of such exception. In the presence of function symbols, a simple example of implicit exception is the following. Let B be

$$(x).R(x) \rightarrow Q(f(x))$$
$$(x).R(x) \rightarrow \neg Q(g(x)).$$

Now let the new information be

$$(x).f(x) = h(g(x)).$$

Consistency can not be restored without the explicit definitions of f, g, h.

The last example in this category involves attributes with value domains of cardinality greater than 2. So far we consider boolean (or binary) attributes. For instance, *Fly* assigns to each entity (in Bird) one of the values true or false (with respect to flying). Now consider the attribute *Color* that has more choices. Suppose we reasoning about blackboards and B is

$$(x)(y)(z).Board(x) \wedge Color(x,y) \wedge Color(x,z) \rightarrow y = z$$
$$(x).Board(x) \rightarrow Color(x, black)$$
$$Board(b)$$

If we find that $Color(b, green)$ the algorithm in [Seet 87] can not resolve the contradiction (even if u is $Color(b, green)$) since there are no literals of "opposite sign" to resolve upon.

c. *Minimal Inconsistent Subsets.* The algorithm revises considerably more clauses than necessary. A new belief $\beta \equiv l_1 \vee \cdots \vee l_n$ conflicting with B and having l_n as its designator, causes a modification to any clause C in B that contains a literal c that is unifiable and complementary to l_n, even if C is irrelevant to β. As a result many irrelevant "exceptions" accumulate in the clauses of C. In the case that β is an explicit exception, a clause $C \equiv c_1 \vee \cdots \vee c_k$, where $\sigma(c_k)$ is identical to $\sigma(\neg l_n)$ for some σ, needs to be revised only if $B \vdash \sigma(c_1 \vee \cdots \vee c_{k-1} \rightarrow l_1 \vee \cdots \vee l_{n-1})$ or $B \vdash \sigma(l_1 \vee \cdots \vee l_{n-1} \rightarrow c_1 \vee \cdots \vee c_{k-1})$. In the case of an implicit exception, minimal inconsistent subsets should be found and revised successively until the set becomes consistent (hard problem).

d. *Order of Revision.* The algorithm accepts new information in clausal form only and also one clause at a time. The order at which these clauses are presented to the system is significant since different orders may result to defferent set of beliefs. For instance, suppose that the set of current beliefs is

$$B(t) \tag{1}$$

and we have been informed that

$$(x){:}B(x) \rightarrow F(x) \tag{2}$$

and $$\neg F(t). \tag{3}$$

If the clauses are processed in the order (2),(3) we obtain

$$(x).B(x) \wedge \neg x = t \rightarrow F(x)$$
$$B(t)$$
$$\neg F(t)$$

and t can not fly, while the order (3),(2) yields

$$B(t)$$
$$B(t) \vee \neg F(t)$$
$$(x).B(x) \rightarrow F(x)$$

but now t can fly!

The final issue we would like to address is that [Seet 87] does not make any distinction between knowledge and beliefs; both are treated as beliefs. We strongly regard, when a world is represented in a formal language, certain axioms, postulates and facts about it should be differentiated from the set of beliefs; after all, the former should not be revisable.

5 Revision and Default Logic

For the simple case where all the beliefs are Horn clauses and the exceptions are explicit, revision can be easily achieved using normal default logic [Reiter 80] as follows.

We call *base predicates* the predicates that represent "natural kind" sets, like Bird, Penguin etc. and *attributes* the predicates that assign attributes to entities of the base ones, i.e. Fly, Color, etc. We also assume that the base predicates form a hierarchical structure similar to an inheritance hierarchy. More information about this structure can be found in [Gilmore 87a] and [Gilmore 87b]. We say that a base predicate P is *more general* than Q if P is higher in the hierarchy tree, while attributes are considered to be less general than any base predicate.

We regard that a world is represented by a default theoty K=(C,D), where C is a set of first order formulas and D is a set of normal defaults (initially D may be empty).

Suppose new knowledge is acquired that is inconsistent with C and is represented by the clause l. l is translated into the clause

$$l_1 \wedge \cdots \wedge l_n \rightarrow f(t)$$

where n can be 0 and f is the least general predicate(ties are broken by random choice). For each clause c in C that contains $\neg f(t')$ and t, t' are unifiable by σ, we transform it into

$$c_1 \wedge \cdots \wedge c_k \rightarrow \neg f(t')$$

where k can be 0, and:

1. If c is identical to $\neg l$ then c is deleted.

2. If $C \vdash \sigma(l_1 \wedge \cdots \wedge l_n \rightarrow c_1 \wedge \cdots \wedge c_k)$ then c is deleted from C, $\frac{c_1 \wedge \cdots \wedge c_k : Mf}{f}$ is added to D and l is added to C.

3. If $C \vdash \sigma(c_1 \wedge \cdots \wedge c_k \rightarrow l_1 \wedge \cdots \wedge l_n)$ then $\frac{l_1 \wedge \cdots \wedge l_n : M \neg f}{\neg f}$ is added to D.

4. When a clause c of C is deleted from C the following revision is applied:

   Any default $d \equiv \frac{a:Mb}{b}$ such that $a \rightarrow b$ is consistent with C and there is no default $d' \equiv \frac{a':Mb'}{b'}$ in D for which

$$\sigma(b) \equiv \sigma(\neg b') \text{ and } C \vdash \sigma(a') \rightarrow \sigma(a)$$

   for some unifier σ, then d is deleted from D and the clause $a \rightarrow b$ is added to C.

Example. Let K=(C,D) where

$$C = \{B(t), \neg F(t)\} \text{ and } D = \emptyset$$

If l is $B(x) \rightarrow F(x)$ then K becomes

$$\begin{array}{c} B(t) \\ \neg F(t) \end{array} \qquad\qquad \frac{B(x) : MF(x)}{F(x)}$$

Note that from now on any exception to $B(x) \rightarrow F(x)$ will not initiate any revision of the kind Seet's algorithm would do.

If $F(t)$ is discovered later, K becomes

$$\{B(t), F(t), B(x) \rightarrow F(x)\}$$

But if $B(x) \rightarrow P(x)$ were in C and $\frac{P(x):M\neg F(x)}{\neg F(x)}$ were in D, the previous default would have stayed in D.

Naturally, rhe revision procedure we have just presented can not handle implicit exceptions and many other cases as the reader might have already noticed. Our intention is not to describe a complete revision procedure, since we believe that no one exists, but to outline how explicit exeptions can be handled by using the normal default logic of [Reiter 80] for which a sound proof procedure exists after all.

6 SET Model and Exeptions

The SET conceptual model developed by Gilmore and described in [Gilmore 87a], [Gilmore 87b], [Gilmore 87c], is based on a natural deduction set theory presented in [Gilmore 86]. The knowledge about a word W is presented in SET mainly by a sequence of definitions of the sets discovered to exist in W (base, defined, system defined sets), the relations among them (usually expressed as domain and degree constraints) and the membership (or extensions) of the base sets. It is worth noting that this model is generally enough to include inheritance hierarchy as a special case.

So far, in this report we have tacitly followed Seet's paradigm in which all we know about a world is what we believe about it; in general, this is not the case. Certain properties that constitute the "universal laws" of the world being modelled, are beyond any dispute or revision. In case the intensions and the relations of the existing sets enjoy such revision immunity, default reasoning and belief revision in the SET model can be achieved easily by means of defined sets and regular data base updates. A brief discussion on this issue can be found in [Gilmore 87d].

If the world being modelled, perpetually evolves or the acquired knowledge is insufficient to ensure the previous condition, more sofisticated techniques (if any) have to be employed for default attribution and revision. In the last case, we believe that the SET model can be extended to perform adequately. We complete this section with an example that illustrates some of the problems that arise in the latter case.

Example. Suppose we know that most of Mollusks are shell-bearers, all Cephalopods are Mollusks but they do not have shells, Moll is a Mollusk and Fred a Cephalopod. The SET description of this world would be :

$$VS \equiv \{x : STRING | x =' Y' \lor x =' N'\}$$
$$MOLLUSK \equiv \{x | x =' Moll' \lor x =' Fred'\}$$
$$SHELL \equiv \{x : MOLLUSK, v : VS | x =' Fred' \land v =' N'\}$$
$$SHELL - FREE \equiv \{x : MOLLUSK | < x,' N' >: SHELL\}$$

$$CEPHALOPOD \equiv \{x : SHELL - FREE | x =' Fred'\}$$
$$DSHELL \equiv \{x : MOLLUSK, v : VS | < x, v >: SHELL \vee$$
$$([\forall u : VS] \neg < x, u >: SHELL \wedge v =' Y'\}$$

As a result, Moll is a shell bearer as $<' Moll',' Y' >: DSHELL$.

Suppose we have found that Nauty is a Cephalopod that bears shell (i.e. is a Nautilus). To incorporate the new fact, the model should be revised into :

$$VS \equiv \{x : STRING | x =' Y' \vee x =' N'\}$$
$$MOLLUSK \equiv \{x | x =' Moll' \vee x =' Fred' \vee x =' Nauty'\}$$
$$SHELL \equiv \{x : MOLLUSK, v : VS | (x =' Fred' \wedge v =' N') \vee (x =' Nauty' \wedge v =' Y')\}$$
$$CEPHALOPOD \equiv \{x : MOLLUSK | x =' Fred' \vee x =' Nauty'\}$$
$$DSHELL \equiv \{x : MOLLUSK, v : VS | < x, v >: SHELL \vee$$
$$([\forall u : VS] \neg < x, u >: SHELL \wedge x : CEPHALOPOD \wedge v =' N') \vee$$
$$([\forall u : VS] \neg < x, u >: SHELL \wedge \neg x : CEPHALOPOD \wedge v =' Y')\}$$

Note that the constraint that Cephalopods are not shell bearers has been eliminated and the set SHELL-FREE is not needed any more.

7 Conclusion

We have discussed a variety of problems pertaining to Seet's reasoning and revision system. We have primarily shown that his reasoning system is inconsistent and similar to a CWA-reasoning. Also we have proved that in the case that a circumscription of all the exceptions ($\Box$-formulas) implicitly defines them, the reasoning system behaves appropriately. Moreover, if the exceptions are decidable a first order system can be used instead.

Seet's "modal" system, if modified, becomes a special case of McDermott and Doyle non-monotonic logic if $\Box\phi$ is interpreted as an abbreviation of $\neg M \neg \phi$. No complete proof procedure is known to exist for this logic in general but, if the modal formulas are restricted to be of the form $a \wedge Mb \rightarrow b$ for some first order formulas a, b, a proof procedure might be possible. If so, an algorithm similar to the one we have described for default logic can be used for revision.

Subsequently, we were concerned with Seet's revision algorithm and its problems. In general, any sentence contradictory to the current beliefs can be viewed as an exception and should demand a revision to restore consistency. In this framework, the revision algorithm is inadequate but, it can adequately handle explicit exceptions which are represented by Horn clauses. For the same class of exceptions a similar algorithm was given for normal default theories.

Finally, some reflections on the adequacy of the SET model for default reasoning and belief revision were stated. The advantages of SET as a conceptual modelling technique, presented in [Gilmore 87b], suggest its potential on knowledge representation which remains to be seen and is among the topics of our future research.

ACKNOWLEDGEMENT

I would like to thank Paul Gilmore and Wolfgang Bibel for their suggestions as well as for their moral and financial support.

REFERENCES

DAVIS, M. 1980. The Mathematics of Non-Monotonic Reasoning. Artificial Intelligence, 13, pp. 73-80.

DELGRANDE, J. 1987. An Approach to Default Reasoning Based on a First Order Conditional Logic. Proceedings of the Sixth AAAI Conference on Artificial Intelligence, Seattle Washington, pp. 340-345.

DOYLE, J. 1984. Circumscription and Implicit Definability. Carnegie-Mellon University, Technical Report CMU-CS-84-154.

GILMORE, P. 1986. Natural Deduction Based Set Theories: a New Resolution of the Old Paradoxes. Journal of Symbolic Logic, 51(2), pp. 393-411.

GILMORE, P. 1987a. The SET Conseptual Model and the Domain Graph Method of Table Design. Department of Computer Science, University of British Columbia, Vancouver, B.C., Technical Report 87-7.

GILMORE, P. 1987b. Justifications and Applications of the SET Conseptual Model. Department of Computer Science, University of British Columbia, Vancouver, B.C., Technical Report 87-9.

GILMORE, P. 1987c. A Foundation for the Entity-Relationship Model: Why and How? Department of Computer Science, University of British Columbia, Vancouver, B.C., Technical Report 87-11. Also in Proceedings of the 6th Entity-Relationship Conference. (Nov 1987).

GILMORE, P. 1987d. Formalizing Attribution by Default. Department of Computer Science, University of British Columbia, Vancouver, B.C., Technical Report 87-26.

McCARTHY, J. 1980. Circumscription-A Form of Non-Monotonic Reasoning. Artificial Intelligence, 13, pp. 27-39.

McDERMOTT, D. and DOYLE, J. 1980. Non-Monotonic Logic I. Artificial Intelligence, 13, pp. 41-72.

REITER, R. 1980. A Logic for Default Reasoning. Artificial Intelligence, 13, pp. 81-132.

SEET, C. 1987. Default Reasoning Through Belief Revision Strategy. Proceedings of the Sixth AAAI Conference on Artificial Intelligence, Seattle Washington, pp. 380-384.

SMULLYAN, R. 1968. First Order Logic. Springer-Verlag, Berlin.

GENTZEN, G. 1969. The Collected Papers of Gerhard Gentzen. *Edited by* Szabo, M., North-Holland, Amsterdam.