COMPUTING CIRCUMSCRIPTION

Vladimir Lifschitz

Department of Computer Science
Stanford University
Stanford, CA 94305

Abstract

Circumscription is a transformation of predicate formulas proposed by John McCarthy for the purpose of formalizing non-monotonic aspects of commonsense reasoning. Circumscription is difficult to implement because its definition involves a second-order quantifier. This paper presents metamathematical results that allow us in some cases to replace circumscription by an equivalent first-order formula.

1. Introduction

Research in the theory of commonsense reasoning has revealed a fundamental difference between how universal assertions arc used in mathematics on the one hand, and in the area of commonsense knowledge on the other. In mathematics, when a proposition is claimed to be universally true, the assertion includes a complete list of conditions on the objects involved under which the proposition is asserted ("for all $x \neq 0$", "for all sufficiently large natural numbers", etc.) But in everyday life we often assert that a certain proposition is true "in general"; we know that there are exceptions, and we can list some of them, but the list of exceptions is not a part of the assertion. The "abnormality" of each item on the list is a separate piece of commonsense knowledge. We know, for instance, that birds, generally, can fly. We know, futhermore, that ostriches are exceptions. And penguins are exceptions. And dead birds arc exceptions. All these assertions appear to be separate commonsense facts.

The language of predicate logic has been created primarily for the purpose of formalizing mathematics, and it docs not provide any means for talking about what is "generally true" and what "exceptions" are. If we want to use that language for representing commonsense knowledge then methods for formalizing assertions about exceptions have to be developed.

The study of such methods belongs to the area of *non-inonotonic logic.* Extending an axiom set can force us to retract some of the conclusions we have derived from the axioms if the new axioms include additional information about abnormal objects. The set of theorems depends on the set of axioms in a non-monotonic way.

Consider a simple example which illustrates some of the difficulties involved. Let $BI\,x$, $OS\,x$, $FL\,x$ and $AB\,x$ represent the conditions "i is a bird", "2 is an ostrich", "x can fly" and "i is abnormal". We want to express these commonsense facts: birds, generally, can fly; ostriches are birds and cannot fly. Consider the formulas:

$$\forall x(BI\,x \wedge \neg AB\,x \supset FL\,x), \qquad (A_1)$$

$$\forall x(OS\,x \supset BI\,x),$$

$$(A_3)\ \forall x(OS\,x \supset \neg FL\,x).$$

These formulas represent a part of what has been said about the ability of birds to fly, but they do not say one important thing: objects are considered normal if there is no evidence to the contrary. The three available facts imply that ostriches are abnormal, and they give no information about the abnormality of any other objects. We want to be able to conclude then that ostriches are the only exceptions:

$$\forall x(AB\,x \equiv OS\,x). \qquad (1)$$

But (1) does not follow from the conjunction A of A_1,

We discuss here one of the approaches to this problem, the theory of circumscription (McCarthy 1980, 1984). The process of circumscription transforms A into a stronger formula A' which says essentially that AD has a minimal possible extension under the condition A. It turns out that A' is equivalent to the conjunction of A and (1).

A' depends on A non-monotonically. In this sense, circumscription provides an interpretation of non-monotonic reasoning in the usual (monotonic) logic.

In more complex cases, wc deal with several kinds of abnormality, and the extensions of several predicates AB_1, $AB_2, \ldots$ have to be minimized. These minimizations sometimes conflict with each other, and there may be a need to establish relative priorities between them (McCarthy 1984).

Currently there are no working systems of knowledge representation based on circumscription. Such a system would include a database A and a metamathematical statement describing how circumscription should be performed. (Such a description would specify, for instance, which predicates should be minimised, and what their priorities are). The system would also include a theorem prover capable of deriving logical consequences from the result A' of circumscribing A.

The design of such a system has to deal with a major difficulty: the definition of circumscription involves a second-order quantifier, so that A' is a formula of a second-order language. The purpose of this paper is to present mctamathematical theorems which, in some instances, enable us to replace the result of circumscription by an equivalent first-order formula. These methods can be successfully applied to some examples of circumscription that seem to be typical for applications to the formalization of commonsense knowledge, and, hopefully, they can be used as a basis for implementing circumscription. Our main tool is a theorem which establishes the equivalence of a special case of circumscription to a modification of Clark's predicate completion (Clark 1978). Connections between these two concepts were first studied in (Reiter 1982).

Proofs of some special cases of the results stated in this paper can be found in (Lifschitz 1984). Complete proofs will be published elsewhere.

2. Second-Order Formulas

A *second order language* is defined, just as a first order language, by sets of *function constants* and *predicate constants,* each of some arity n, $n \geq 0$. In the second order language we have, besides *object variables,* also n-ary *function variables* and n-ary *predicate variables.* (Object, variables and constants arc identified with function variables and constants of arity 0). Both function and predicate variables can be bound by quantifiers. A *sentence* is a formula without free (function or predicate) variables.

A structure M for a second order language L consists of a non-empty *universe* $|M|$, functions f r $|M|^n$ o M representing the function constants and subsets of $|M|^n$ representing the predicate constants. For any constant K, we denote the object (function or set) representing K in M by $M[K]$. Equality is interpreted as identity, function variables range over arbitrary functions from $|M|^n$ to M, and predicate variables range over arbitrary subsets of $|M|^n$. A *model* of a sentence A is any structure M such that A is true in M. A *implies* D if every model of A is a model of B; A is *equivalent* to B if they have the same models.

An n*-ary predicate* is an expression of the form $\lambda x A(x)$, where x is a tuple of n object variables, $A(x)$ a formula. As usual, if U is $\lambda x\, A(x)$, and t is a tuple of n terms, then Ut stands for $A(t)$. We identify a predicate constant P with the predicate $\lambda x Px$, and similarly for predicate variables.

If U, V are n-ary predicates, then $U \leq V$ stands for $\forall x(Ux \supset Vx)$. Thus $U \leq V$ expresses that the extension of U is a subset of the extension of V. We apply this notation to tuples $U = U_1, \ldots, U_m$ and $V = V_1, \ldots, V_m$ of predicates, assuming that they are similar (i.e., that U_i and V_i have the same arity): $U \leq V$ stands for

$$U_1 \leq V_1 \wedge \ldots \wedge U_m \leq V_m.$$

Furthermore, $U = V$ stands for $U \leq V \wedge V \leq U$, and $U < V$ stands for $U \leq V \wedge \neg(V \leq U)$. If $m = 1$ then $U < V$ means simply that the extension of U is a proper subset of the extension of V.

3. Parallel Circumscription

First we consider *parallel* circumscription, when no priorities between the minimized predicates are specified. Let P be a tuple of predicate constants, Z a tuple of function and/or predicate constants disjoint with P, and let $A(P,Z)$ be a sentence. The *circumscription of* P *in* $A(P,Z)$ *with variable* Z is the sentence

$$A(P,Z) \wedge \neg\exists pz(A(p,z) \wedge p < P). \qquad (2)$$

(Here p,z are tuples of variables similar to P,Z). This formula expresses that P has a minimal possible extension under the condition $A(P,Z)$ when Z is allowed to vary in the process of minimization. We denote (2) by $\mathrm{Circum}(A(P,Z);P;Z)$. When Z is empty, we write $\mathrm{Circum}(A(P);P)$.

In applications to the formalization of commonsense reasoning, $A(P,Z)$ is the conjunction of the axioms, P is the list of abnormality predicates, and Z is the list of symbols that we intend to characterize by means of circumscription, as FL in the example discussed in the introduction. The circumscription we intend to perform in that case is

$$\mathrm{Circum}(A(AB,FL);AB;FL). \qquad (3)$$

The model-theoretic meaning of circumscription can be expressed in terms of the following notation. Let P,

Z be as in the definition of circumscription. For any two structures M_1, M_2, we write $M_1 \leq^{P;Z} M_2$ if

(i) $|M_1| = |M_2|$,

(ii) $M_1[K] = M_2[K]$ for every constant K not in P, Z,

(iii) $M_1[P_i] \subset M_2[P_i]$ for every P_i in P.

Thus $M_1 \leq^{P;Z} M_2$ if M_1 and M_2 differ only in how they interpret the constants in P and Z, and the extension of each P_i in M_1 is a subset of its extension in M_2.

Clearly, $\leq^{P;Z}$ is a pre-order (i.e., a reflexive and transitive relation) on the class of all structures. It is possible, however, that $M_1 \leq^{P;Z} M_2$ and, at the same time, $M_2 \leq^{P;Z} M_1$ for two different structures M_1, M_2. This happens when M_1, M_2 differ by the interpretations of symbols in Z. In other words, $\leq^{P;Z}$ is, generally, not anti-symmetric and thus not a partial order. Still, we can speak of the minimality of structures with respect to $\leq^{P;Z}$: a structure M is *minimal* in a class S of structures if $M \in S$ and there is no structure $M' \in S$ such that $M' <^{P;Z} M$. (We write $M_1 <^{P;Z} M_2$ if $M_1 \leq^{P;Z} M_2$ but not $M_2 \leq^{P;Z} M_1$). This concept of minimality was introduced essentially in (McCarthy 1980).

The models of circumscription can be characterized now as follows:

Proposition 1. *A structure M is a model of* Circum$(A; P; Z)$ *iff M is minimal in the class of models of A with respect to* $\leq^{P;Z}$.

This equivalence follows from the fact that specifying values of p, z for which $A(p, z) \wedge p < P$ is true in M is equivalent to specifying a model M' of A such that $M' <^{P;Z} M$.

4. Examples

Before describing general methods for determining the result of circumscription, we discuss a few examples. In these examples Z is empty, and P is a single predicate constant.

Example 1. $A \equiv Pa$, a an object constant. Circumscription asserts that the extension of P is a minimal set satisfying this condition; in other words, a is its only element:

$$\text{Circum}(Pa; P) \equiv \forall x(Px \equiv x = a).$$

Example 2. $A \equiv \neg Pa$. This does not give any "positive" information about P and is true even when P is identically false. Hence

$$\text{Circum}(\neg Pa; P) \equiv \forall x \neg Px.$$

We get the same result whenever A contains no positive occurences of P.

Example 3. $A \equiv Pa \wedge Pb$. Circumscription asserts that a and b are the only elements of P:

$$\text{Circum}(Pa \wedge Pb; P) \equiv \forall x(Px \equiv x = a \vee x = b).$$

Example 4. $A \equiv Pa \vee Pb$. In Examples 1-3, circumscription provided an explicit definition and thus a unique possible value for P. In this case, there are two minimal values:

$$\text{Circum}(Pa \vee Pb; P) \equiv \forall x(Px \equiv x = a) \vee \forall x(Px \equiv x = b).$$

Example 5. $A \equiv Pa \vee (Pb \wedge Pc)$. The previous examples suggest that the answer might be

$$\forall x(Px \equiv x = a) \vee \forall x(Px \equiv x = b \vee x = c).$$

But we should take into consideration that a may be equal to one of b, c, and then the second disjunctive term does not give a minimal P. The correct answer is

$$\text{Circum}(Pa \vee (Pb \wedge Pc); P) \equiv \forall x(Px \equiv x = a) \\ \vee [\forall x(Px \equiv x = b \vee x = c) \wedge a \neq b \wedge a \neq c].$$

Example 6. $A \equiv \forall x(Qx \supset Px)$. Clearly, circumscription simply changes implication to equivalence:

$$\text{Circum}(\forall x(Qx \supset Px); P) \equiv \forall x(Qx \equiv Px).$$

More generally, for any predicate U (without parameters) which does not contain P,

$$\text{Circum}(U \leq P; P) \equiv U = P.$$

This can be also viewed as a generalization of Examples 1 and 3, because in those examples A can be written in the form $U \leq P$:

$$Pa \equiv \lambda x(x = a) \leq P,$$

$$Pa \wedge Pb \equiv \lambda x(x = a \vee x = b) \leq P.$$

Example 7. $A \equiv \exists x Px$. Circumscription asserts that the extension of P is a "minimal non-empty" set, i.e., a singleton:

$$\text{Circum}(\exists x Px; P) \equiv \exists x \forall y(Py \equiv x = y).$$

Example 8. In all examples above, we were able to express the result of circumscription without second-order quantifiers. Let now P, Q be binary predicate constants, and A be

$$\forall xy(Qxy \supset Pxy) \wedge \forall xyz(Pxy \wedge Pyz \supset Pxz).$$

Then Circum$(A; P)$ asserts that P is the transitive closure of Q and thus is not equivalent to a first-order sentence.

Notice that the formula of this example is "good" by all logical standards: it is universal and, moreover, Horn, and it contains no function symbols. What syntactic features make it difficult for circumscription? This question will be answered in the next section.

5. Separable Formulas

In Examples 2 and 6 above we saw that there are two classes of formulas for which the result of circumscription can be easily determined: formulas without positive occurences of P, and formulas of the form $U \leq P$, where U does not contain P. What about formulas constructed from subformulas of these two types using conjunctions and disjunctions?

First let us look at conjunctions of such formulas. Let P be again a tuple of predicate constants $P_1, \ldots P_m$. We say that *A(P)* is *solitary* with respect to *P* if it is a conjunction of

(i) formulas containing no positive occurences of $P_1, \ldots P_m$, and

(ii) formulas of the form $U \leq P_i$, where *U* is a predicate not containing $P_1, \ldots P_m$.

Using predicate calculus, we can write any solitary formula in the form

$$N(P) \wedge (U \leq P),$$

where *N(P)* contains no positive occurences of $P_1, \ldots,$ P_m, and *U* is a tuple of predicates not containing $P_1, \ldots P_m$. Then the result of circumscription is given by the formula

$$\text{Circum}(N(P) \wedge (U \leq P); P) \equiv N(U) \wedge (U = P). \quad (4)$$

Using this formula, we can do, in particular, the circumscriptions of Examples 1, 2, 3 and 6.

Next we want to be able to handle formulas with disjunctions, like Examples 4 and 5. If a formula is constructed from subformulas of forms (i) and (ii) using conjunctions and disjunctions, we call it separable. In a separable formula, positive occurences of $P_1, \ldots P_m$ are separated by conjunctions and disjunctions from negative occurences and from each other.

Any separable formula is equivalent to a disjunction of solitary formulas and consequently can be written in the form

$$\bigvee_i [N_i(P) \wedge (U^i \leq P)], \quad (5)$$

where $N_i(P)$ contains no positive occurences of $P_1, \ldots,$ P_m, and each U^i is a tuple of predicates not containing $P_1, \ldots P_m$.

The following theorem generalises (4) to separable formulas.

Theorem 1. *If A(P) is equivalent to (5) then* Circum(A(P); P) *is equivalent to*

$$\bigvee_i [D_i \wedge (U^i = P)], \quad (6)$$

where D_i is

$$N_i(U^i) \wedge \bigwedge_{j \neq i} \neg[N_j(U^j) \wedge (U^j < U^i)].$$

Thus the result of circumscription in a separable first-order sentence is a first-order sentence of about the same logical complexity. Circum(A(P); P) asserts that P may have one of the finite number of possible values U^1, U^2, ... Hence every model of Circum(A(P); P) belongs to one of a finite number of classes: there are models in which P is the same as U^1, models in which P is the same as U^2, etc. If M is a structure in which $P = U^i$ then D_i is the additional condition which, if true in M, guarantees that *M* is a model of Circum(A(P); P).

The transformation of $U^i \leq P$ into $U^i = P$ is based on the same idea as *predicate completion* of (Clark 1978): transforming sufficient conditions into necessary and sufficient. When applied to non-separable Horn formulas, like the formula of Example 8, predicate completion often gives conditions that arc weaker than the result of circumscription. On the other hand, the transformation of (5) into (G) is not restricted to Horn formulas and is in this respect more general.

Using Theorem 1, we can easily do all circumscriptions of Examples 1-6. Examples 7 and 8 arc not separable.

Example 7 shows that there arc non-separable formulas for which circumscription can be done in the first-order language. Here is one more example:

$$\text{Circum}(\forall x(P_1 x \vee P_2 x); P_1, P_2) \equiv \forall x(P_1 x \equiv \neg P_2 x). \quad (7)$$

The first argument can be easily written in the form separable with respect to P_1 and in the form separable with respect to P_2, but it is not equivalent to a formula separable with respect to the pair P_1, P_2.

Theorem 1 cannot be applied directly to circumscriptions with a non-empty Z, like (3). Two observations often help in such cases.

First, every circumscription with a non-empty Z can be reduced to a circumscription with the empty Z:

Proposition 2. Circum(A(P, Z); P; Z) *is* equivalent to

$$A(P,Z) \wedge \mathrm{Circum}(\exists z A(P,z); P). \tag{8}$$

For example, (3) reduces to

$$\mathrm{Circum}(\exists fl(A_1(AB,fl) \wedge A_2 \wedge A_3(fl)); AB), \tag{9}$$

where fl is a binary predicate variable.

The problem with this trick is, of course, that the first argument of circumscription in (8), generally, contains new second-order quantifiers. In our example, we have to circumscribe *AB* in a formula with the quantifier $\exists fl$.

The second observation sometimes helps eliminate quantifiers like this. If *q* is a tuple of predicate variables, and *A(q)* is separable with respect to *q,* then we can write *A(q)* in the form

$$\bigvee_i [N_i(q) \wedge (U^i \leq q)]. \tag{5'}$$

(Separability with respect to a tuple of predicate variables is defined in the same way as separability with respect to a tuple of predicate constants). It can be easily seen that *eqA[q]* is equivalent then to $V_iN[_tU_i]$. In our example, the only positive occurence of *fl* is in the first conjunctive term, so we write the conjunction as

$$A_2 \wedge A_3(fl) \wedge \lambda x(BI\,x \wedge \neg AB\,x) \leq fl.$$

Then the first argument of (9) is equivalent to the conjunction of A_2 and

$$\forall x(OS\,x \supset \neg(BI\,x \wedge \neg AB\,x)).$$

In the presence of A_2 this simplifies to $\forall x(OS\,x \supset AB\,x)$. Thus it remains to circumscribe AB in $A_2 \wedge OS \leq AB$. By (4), the result is $A_2 \wedge OS = AB$. The second term here is equivalent to (1).

Many other examples of circumscription arising in connection with the formalization of commonsense reasoning can be handled in the same manner.

6. General Circumscription

In some applications of circumscription we have several abnormality predicates which are assigned different priorities. For example, we may wish to minimise AB_1 and AB_2, the former at higher priority than the latter. This means that, instead of minimizing (AB_1, AB_2) with respect to the relation $\leq$ defined by

$$(p_1,p_2) \leq (q_1,q_2) \equiv p_1 \leq q_1 \wedge p_2 \leq q_2,$$

we use the relation

$$(p_1,p_2) \preceq (q_1,q_2) \equiv p_1 \leq q_1 \wedge (p_1 = q_1 \supset p_2 \leq q_2). \tag{10}$$

To prepare for the general treatment of priorities in the next section, we define now the generalization of circumscription needed for that purpose.

Let p, q be disjoint similar tuples of predicate variables, and let $p \preceq q$ be a formula which has no parameters besides p, q. We say that $p \preceq q$ *defines a regular order in* a structure M if the sentences

$$\forall pq(p \leq q \supset p \preceq q),$$

$$\forall pqr(p \preceq q \wedge q \preceq r \supset p \preceq r),$$

$$\forall pq(p \preceq q \wedge q \preceq p \supset p = q)$$

are true in M. As the name suggests, such a formula defines a (partial) order on vectors of subsets of $|M|^n$. Clearly, $p \leq q$ defines a regular order in every structure. A more interesting example is given by (10). We write $p \prec q$ for $p \preceq q \wedge p \neq q$.

As before, let P be a tuple of predicate constants, Z a tuple of constants disjoint with P, $A(P,Z)$ a sentence. Let $p \preceq q$ be a formula which does not contain P, Z, and defines a regular order in every model of $A(P,Z)$. The *circumscription of* P *in* $A(P,Z)$ *with variable* Z *with respect to* $\preceq$, symbolically $\mathrm{Circum}_{\preceq}(A(P,Z); P; Z)$, is

$$A(P,Z) \wedge \neg\exists pz(A(p,z) \wedge p \prec P).$$

The results of Sections 3 and 5 can be extended to general circumscription as follows. We write

$$M_1 \leq^{P;Z;\preceq} M_2$$

if

(i) $|M_1| = |M_2|$,
(ii) $M_1[K] = M_2[K]$ for every constant K not in P, Z,
(iii) $p \preceq q$ is true in M_1 (or, equivalently, in M_2) for $M_1[P]$ as p and $M_2[P]$ as q.

Proposition 1′. *A structure M is a model of* $\mathrm{Circum}_{\preceq}(A; P; Z)$ *iff M is minimal in the class of models of A with respect to* $\leq^{P;Z;\preceq}$.

Theorem 1′. *If* $A(P)$ *is equivalent to (5) then* $\mathrm{Circum}_{\preceq}(A(P); P)$ *is equivalent to*

$$\bigvee_i [D_i \wedge (U^i = P)],$$

where D_i *is*

$$N_i(U^i) \wedge \bigwedge_{j \neq i} \neg[N_j(U^j) \wedge (U^j \prec U^i)].$$

Notice that this formula for D_i differs from the formula of Theorem 1 only when there are at least two disjunctive terms. Consequently, the effect of general circumscription on a *solitary* formula can be computed using the same formula (4) that we used for parallel circumscription. Thus we have:

Corollary. *If A is solitary then* $\mathrm{Circum}_{\preceq}(A; P)$ *does not depend on* $\preceq$.

Proposition 2′. $\mathrm{Circum}_{\preceq}(A(P,Z); P; Z)$ *is equivalent to*

$$A(P,Z) \wedge \mathrm{Circum}_{\preceq}(\exists z A(P,z); P).$$

In the next section we show how this generalisation of circumscription works in applications to the formalization of commonsense reasoning.

7. Priorities

The database B defined below contains these commonsense facts: things, in general, do not fly; airplanes and birds, in general, do; but ostriches and dead birds, generally, do not. B is the conjunction of these formulas:

$$\forall x(OS\,x \supset BI\,x), \qquad (B_1)$$

$$\forall x \neg(BI\,x \wedge PL\,x), \qquad (B_2)$$

$$\forall x(\neg AB_1\,x \supset \neg FL\,x), \qquad (B_3)$$

$$\forall x(PL\,x \wedge \neg AB_2\,x \supset FL\,x), \qquad (B_4)$$

$$\forall x(BI\,x \wedge \neg AB_3\,x \supset FL\,x), \qquad (B_5)$$

$$\forall x(OS\,x \wedge \neg AB_4\,x \supset \neg FL\,x), \qquad (B_6)$$

$$\forall x(BI\,x \wedge DE\,x \wedge \neg AB_5\,x \supset \neg FL\,x). \qquad (B_7)$$

We expect that circumscribing $AB_1, \ldots, AB_5$ in B should give the following result: AB_2, AB_4 and AB_5 are identically false (since there is no evidence that they are not); ostriches and dead birds are the only objects satisfying AB_3; airplanes and the birds that are alive and not ostriches are the only objects satisfying AB_1.

However, the circumscription

$$\mathrm{Circum}(B; AB_1, \ldots, AB_5; FL)$$

does not lead to these conclusions. The reason is that the goals of minimising our five abnormality predicates conflict with each other. For instance, minimizing the extensions of AB_2 and AB_3 conflicts with the goal of minimising AB_1.

The solution proposed in (McCarthy 1984) is to establish priorities between different kinds of abnormality. Let a tuple P of predicate variables be broken into disjoint parts P^1, P^2, ..., P^k. We want to express the idea that the predicates in P^1 should be minimized at higher priority than the predicates in P^2, P^2 at higher priority than P^3, etc. Let p^i, q^i be tuples of predicate variables similar to P^i, and let p, q stand for $p^1, \ldots, p^k$ and $q^1, \ldots, q^k$. Define

$$p \preceq q \equiv \bigwedge_{i=1}^{k} \left(\bigwedge_{j=1}^{i-1} p^j = q^j \ \supset\ p^i \leq q^i \right). \qquad (11)$$

If $k = 1$ then (11) defines simply $p \leq q$. If $k = 2$, P^1 consists of only one predicate P_1, and P^2 consists of one predicate P_2, then (11) becomes (10).

Formula (11) defines a regular order in every structure. We denote the circumscription $\mathrm{Circum}_{\preceq}(A; P; Z)$ with respect to this order by

$$\mathrm{Circum}(A; P^1 > \ldots > P^k; Z)$$

and call it *prioritised circumscription.*

To see how establishing priorities affects the result of circumscription, compare (7) with the result of prioritised circumscription:

$$\mathrm{Circum}(\forall x(P_1 x \vee P_2 x); P_1 > P_2) \equiv \forall x \neg P_1 x \wedge \forall x P_2 x. \qquad (12)$$

Minimizing P_1 at higher priority means that we minimise (P_1, P_2) with respect to (10). Without priorities, circumscription only leads to the conclusion that P_1 and P_2 do not overlap. In (12) we make the extension of P_1 as small as possible, even if it leads to making the extension of P_2 larger; that makes P_1 identically false and P_2 identically true.

In applications it is reasonable to assign higher priorities to the abnormality predicates representing exceptions to "more specific" commonsense facts. In the example above, we use the circumscription

$$\text{Circum}(B; AB_4, AB_5 > AB_2, AB_3 > AB_1; FL). \quad (13)$$

How to compute the result of a prioritized circumscription? We can try to use Theorem 1' and Proposition 2'. It turns out, however, that in cases when priorities are essential, the axiom set is usually not separable with respect to the collection of all abnormality predicates; at best, we have separability with respect to individual *ABs* or small groups of *ABs.* Even in simple cases, doing prioritized circumscription requires an additional tool.

Such a tool is given by the fact that any prioritized circumscription can be written as a conjunction of parallel circumscriptions, as follows:

Theorem 2. Circum$(A; P^1 > \ldots > P^k; Z)$ *is equivalent to*

$$\bigwedge_{i=1}^{k} \text{Circum}(A; P^i; P^{i+1}, \ldots, P^k, Z).$$

According to this theorem, we can do circumscription (12) by taking the conjunction of

$$\text{Circum}(\forall x(P_1 x \vee P_2 x); P_1; P_2)$$

and

$$\text{Circum}(\forall x(P_1 x \vee P_2 x); P_2).$$

Each of the two circumscriptions can be easily evaluated using the methods of Section 5. The first of them gives $\forall x \neg P_1 x$, the second $\forall x(P_2 x \equiv \neg P_1 x)$. The conjunction' of these formulas is equivalent to the right-hand side of (12).

The result of circumscription (13) can be determined along the same lines. We come up with the conclusion that (13) is equivalent to the universal formula

$$\begin{aligned} & B_1 \wedge B_2 \\ & \wedge FL = AB_1 = \lambda x(PL\,x \vee (BI\,x \wedge \neg OS\,x \wedge \neg DE\,x)) \\ & \wedge AB_3 = \lambda x(OS\,x \vee (BI\,x \wedge DE\,x)) \\ & \wedge AB_2 = AB_4 = AB_5 = \lambda x.false. \end{aligned}$$

Acknowledgements

I am indebted to John McCarthy for introducing me to problems of non-monotonic reasoning and circumscription. I have also benefited from discussing this work with David Etherington, Michael Gelfond, Benjamin Grosof, Kurt Konolige, Nils Nilsson, Donald Perlis and Raymond Reiter.

References

Clark, K. (1978) "Negation as Failure", In: *Logic and Databases* (Gallaire, H. and Minker, J., Eds.), Plenum Press, N. Y., 1978, 293-322.

Lifschitz, V. (1984) "Some Results on Circumscription", Technical Report No. STAN-CS-84-1019, Department of Computer Science, Stanford University, 1984 (= *AAAI Workshop on Non-Monotonic Reasoning,* 1984, 151-164).

McCarthy, J. (1980) "Circumscription - A Form of Non-Monotonic Reasoning", *Artificial Intelligence 13,* 1980, 295-323.

McCarthy, J. (1984) "Applications of Circumscription to Formalizing Commonsense Knowledge", *AAAI Workshop on Non-Monotonic Reasoning,* 1984, 295-323, and to appear in *Artificial Intelligence,*

Reiter, R. (1982) "Circumscription Implies Predicate Completion (Sometimes)", Proc. *AAAI-82,* 1982, 418-420.