Circumscription

Vladimir Lifschitz
Department of Computer Sciences
and Department of Philosophy
University of Texas at Austin
Austin, TX 78712

1 Introduction

The idea of circumscription can be explained on a simple example. We would like to represent information about the locations of blocks in a blocks world, using the "default":

$$\text{Normally, a block is on the table.} \tag{1}$$

Then information about the location of a specific block will have to be included in the database only if that block is an exception to this default. If, for instance, B_1 is the only exception, then the database will contain, in addition to (1), the assertion:

$$B_1 \text{ is not on the table.} \tag{2}$$

If B_2 is another block, then we expect that a general logical mechanism of default reasoning will allow us to conclude:

$$B_2 \text{ is on the table.} \tag{3}$$

It is easy to express assertions (1)–(3) by first order formulas using three unary predicate constants: *Block*, *Ontable* and *Ab*. The role of the predicate *Ab* is to single out the blocks that are "abnormal" relative to the default (1). The assertions (1) and (2) can be represented by the formulas:

$$Block(x) \wedge \neg Ab(x) \supset Ontable(x), \tag{4}$$

$$\neg \mathit{Ontable}(B_1). \tag{5}$$

Also, we may use the fact that B_1 and B_2 are two different blocks:

$$\mathit{Block}(B_1),\ \mathit{Block}(B_2),\ B_1 \neq B_2. \tag{6}$$

The goal is to derive the formalization of (3):

$$\mathit{Ontable}(B_2). \tag{7}$$

Classical logic, by itself, does not achieve this goal, as (7) is not a logical consequence of the axioms (4)–(6). The following model M of this axiom set can serve as a counterexample. The universe of M consists of 2 objects, represented by the constants B_1 and B_2. The predicates *Block* and Ab are identically true in M, and *Ontable* is identically false. It is clear that all axioms (4)–(6) are true in M, while (7) is false.

The problem with the axiom set (4)–(6) is that the axioms say too little about the abnormality predicate Ab. Indeed, (4) is the only axiom in which Ab is mentioned at all. We did not say whether there are few or many abnormal objects; even the possibility that every object is abnormal, as is the case in the counterexample model M, is not eliminated. And if every object is abnormal, then, of course, the axiom (4) becomes trivial and says nothing about the locations of blocks.

The model M is a "bad," unintended model, because *it has too many abnormal objects.* There is no need to consider B_2 abnormal. Instead, we can reclassify B_2 as a block located on the table, and all axioms (4)–(6) will be still satisfied. We will get in this way another model M', in which the set of abnormal objects is $\{B_1\}$, rather than $\{B_1, B_2\}$. The model M' is "better" than M in the sense that the extent of Ab in M' is a proper subset of its extent in M.

On the other hand, M' cannot be further "improved" in this sense: The axioms (4)–(6) logically imply $Ab(B_1)$, so that B_1 has to belong to the extent of Ab in any model of the axioms. The extent of Ab in M' is *minimal.*

The main idea of circumscription is to consider, instead of *arbitrary* models of a given axiom set, only the models that satisfy a certain *minimality condition.* The minimality of Ab with respect to set inclusion is the most usual minimality condition in applications of circumscription to formalizing default reasoning.

In the example, M' is a minimal model of (4)–(6). There are other minimal models (containing more than 2 objects). But in every minimal model Ab satisfies the condition

$$Ab(x) \equiv x = B_1. \tag{8}$$

Since the desired conclusion (7) logically follows from (4)–(6) and (8), it holds in every minimal model of the axioms. We can say that (7) and (8) follow from the axioms "by circumscription": They hold in every model in which abnormality is "circumscribed."

Mathematically, circumscription is defined as a syntactic transformation of logical formulas. It transforms a sentence A into a stronger sentence A^*, such that the models of A^* are precisely the minimal models of A. In applications, A is the conjunction of the axioms ("facts included in the database"). The formulas that follow from A "by circumscription" are simply the formulas that are entailed[1] by A^*. Thus circumscription reduces default reasoning to classical logic by *reinterpreting the axioms*; the axiom set written as A is understood as A^*, and then classical logic is applied for deriving the "circumscriptive consequences" of A.

To be more precise, circumscription is a *family* of syntactic transformations, because several minimality conditions can be used in conjunction with the same axiom set A.

2 Definition and Examples

2.1 The Basic Case

Since the theory of circumscription concerns itself with comparing the extents of predicates, the following notation will be useful. For any predicate symbols P, Q of the same arity,

$$P = Q \text{ stands for } \forall x(P(x) \equiv Q(x)),$$
$$P \leq Q \text{ stands for } \forall x(P(x) \supset Q(x)),$$

[1]In this chapter, we understand entailment model-theoretically: A sentence is *entailed* by a set of axioms if it is true in all models of these axioms. If the axioms may include higher order quantifiers, then this condition is weaker than the possibility of deriving the sentence from the axioms using the postulates of any—inevitably incomplete—formalization of higher order logic.

where x is a tuple of distinct variables. Furthermore, $P < Q$ stands for

$$(P \leq Q) \wedge \neg(P = Q).$$

This formula expresses that the extent of P is a proper subset of the extent of Q.

Let $A(P)$ be a sentence containing a predicate constant P. The *circumscription of P in $A(P)$* is the following second order sentence $A^*(P)$:

$$A(P) \wedge \neg\exists p[A(p) \wedge p < P].$$

Here p is a predicate variable of the same arity as P.

This sentence has a simple meaning. The subformula $\neg\exists p[A(p) \wedge p < P]$ says that the extent of P is minimal, given the restriction $A(P)$. A model of A^* is a model of A that cannot be transformed into another model of A by making the extent of P smaller (relative to set inclusion).

Example 1 Let A be $P(a)$, where a is an object constant. Then A^* is

$$P(a) \wedge \neg\exists p[p(a) \wedge p < P].$$

This formula expresses that the object denoted by a belongs to the set denoted by P, but not to any proper subset of that set. In other words, the set is a singleton. We conclude that A^* is equivalent to a simple first order formula:[2]

$$A^* \equiv \forall x[P(x) \equiv x = a].$$

Example 2 A is $\neg P(a)$. This formula is true even when P is identically false (empty), so that

$$A^* \equiv \forall x \neg P(x).$$

Example 3 A is $P(a) \wedge P(b)$. Circumscription asserts that a and b are the only elements of P:

$$A^* \equiv \forall x[P(x) \equiv (x = a \vee x = b)].$$

[2]By *equivalent* we mean here that both formulas have the same models. These formulas are also equivalent in the stronger sense that the equivalence can be proved in the (incomplete) formal system of second order logic.

Example 4 A is $P(a) \vee P(b)$. 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:

$$A^* \equiv \forall x[P(x) \equiv x = a] \vee \forall x[P(x) \equiv x = b].$$

Example 5 A is $P(a) \supset P(b)$. This is similar to Example 2, in that this formula is true even when P is empty, so that here again

$$A^* \equiv \forall x \neg P(x).$$

Example 6 A is $P(a) \vee [P(b) \wedge P(c)]$. The previous examples suggest that the answer might be given by

$$\forall x[P(x) \equiv x = a] \vee \forall x[P(x) \equiv (x = b \vee x = c)].$$

But we should take into consideration that a may be equal to b or c, and in this case the second disjunctive term does not give a minimal P. The correct answer is:

$$A^* \equiv \forall x[P(x) \equiv x = a] \vee \{\forall x[P(x) \equiv (x = b \vee x = c)] \wedge a \neq b \wedge a \neq c\}.$$

Example 7 A is $\forall x P(x)$. This condition gives a complete characterization of the extent of P; hence circumscription does not make it stronger:

$$A^* \equiv A \equiv \forall x P(x).$$

Example 8 A is $\forall x[Q(x) \supset P(x)]$. Circumscribing P transforms the implication into an equivalence:

$$A^* \equiv \forall x[Q(x) \equiv P(x)].$$

Example 9 A is $\exists x P(x)$. Circumscription asserts that the extension of P is a "minimal nonempty set," i.e., a singleton:

$$A^* \equiv \exists x \forall y [P(y) \equiv x = y].$$

Example 10 A is

$$P(a) \wedge \forall x [P(x) \supset P(f(x))].$$

A^* asserts that the extent of P consists of the values of the terms

$$a, f(a), f(f(a)), \ldots .$$

It is not equivalent to any first order sentence.

Let us look now at two examples in which P is a binary predicate.

Example 11 A is $\forall x P(x, x)$. Then

$$A^* \equiv \forall xy [P(x, y) \equiv x = y].$$

Example 12 A is

$$\forall xy [Q(x, y) \supset P(x, y)] \wedge \forall xyz [(P(x, y) \wedge P(y, z)) \supset P(x, z)].$$

A^* expresses that P is the transitive closure of Q. It cannot be reduced to a first order sentence.

Notice that, in almost all examples where we could reduce A^* to a first order formula, that formula included equality, even though there was no equality in the formula A itself. The circumscription operator can be applied to formulas containing equality, too, and this often needs to be done in applications of circumscription to formalizing default reasoning. (For instance, equality occurs in one of the axioms (6) from the blocks world example.) However, in the definition of circumscription, we do not allow P to be the equality predicate: It would not make much sense to substitute p for the equality symbol without requiring that p have the properties that make equality special, such as reflexivity and substitutivity.

2.2 Propositional Circumscription

In the special case when the arity of P is 0, $p < P$ stands for the propositional formula

$$(p \supset P) \wedge \neg(p \equiv P),$$

which is equivalent to $\neg p \wedge P$. Consequently, the circumscription formula $A^*(P)$ can be written as

$$A(P) \wedge \neg\exists p[A(p) \wedge \neg p \wedge P].$$

This is equivalent to

$$A(P) \wedge \neg[A(\mathit{false}) \wedge P],$$

or

$$A(P) \wedge [P \supset \neg A(\mathit{false})]. \tag{9}$$

The second term of (9) says that the value of P may be *true* only if changing it to *false* would make $A(P)$ false.

Example 13 A is $Q \supset P$. This is the propositional counterpart of Example 8. In this case (9) is

$$[Q \supset P] \wedge [P \supset \neg(Q \supset \mathit{false})],$$

which is equivalent to $Q \equiv P$.

2.3 Varying Constants

The basic case of circumscription is too special for most applications. In fact, it is too special even for formalizing the blocks world example from the introduction. We were interested there in the models of the axioms (4)–(6) that are minimal in the sense that the extent of Ab in them cannot be made smaller by changing the interpretations of two predicates, Ab and *Ontable*. In the definition of circumscription given above, minimality is understood as the impossibility of making the extent of the circumscribed predicate smaller *without changing anything else in the model*. We need to be able to specify that some of the object, function or predicate constants occurring in A are allowed to vary along with P in the process of minimizing its extent.

Let $A(P, Z_1, \ldots, Z_m)$ be a sentence containing a predicate constant P and object, function and/or predicate constants $Z_1, \ldots, Z_m$ (and possibly other object, function and predicate constants). The *circumscription of P in A with varied $Z_1, \dots, Z_m$* is the sentence

$$A(P, Z_1, \dots, Z_m) \wedge \neg \exists p z_1 \dots z_m [A(p, z_1, \dots, z_m) \wedge p < P]. \qquad (10)$$

Here p is a predicate variable of the same arity as P; if Z_i is an object constant, then z_i is an object variable, and if Z_i is a function/predicate constant, then z_i is a function/predicate variable of the same arity. The equality symbol is not allowed to appear in the list $Z_1, \dots, Z_m$ (see the end of Section 2.1).

If we denote the tuple $Z_1, \dots, Z_m$ by Z, and $z_1, \dots, z_m$ by z, then (10) can be written as

$$A(P, Z) \wedge \neg \exists p z [A(p, z) \wedge p < P]. \qquad (11)$$

This formula will be denoted by CIRC$[A; P; Z]$. The basic circumscription of P in A, defined earlier, corresponds to the empty Z, and we will write it as CIRC$[A; P]$.

The object, function and predicate constants that are not included in Z and are different from P are said to be *fixed* in CIRC$[A; P; Z]$.

The models of CIRC$[A; P; Z]$ are the models of A in which the extent of P cannot be made smaller even at the price of changing the interpretations of the constants Z.

Example 14 As we saw in Example 3,

$$\text{CIRC}[P(a) \wedge P(b); P] \equiv \forall x [P(x) \equiv (x = a \vee x = b)]. \qquad (12)$$

In this circumscription, a and b are fixed. The effect of circumscribing P in the same formula $P(a) \wedge P(b)$ with varied a and b is different. Consider, for instance, a model M of $P(a) \wedge P(b)$ in which a is represented by 0, b by 1, and P by $\{0, 1\}$. This model satisfies the formula in the right-hand side of (12). But the extent of P in M can be made smaller if we are allowed to change the interpretations of a and b simultaneously with changing P: We can represent a and b by 0, and P by $\{0\}$. In other words, M does not satisfy the stronger circumscription

$$\text{CIRC}[P(a) \wedge P(b); P; a, b]. \qquad (13)$$

By definition, this expression stands for

$$P(a) \wedge P(b) \wedge \neg \exists p z_1 z_2 [p(z_1) \wedge p(z_2) \wedge p < P].$$

It is equivalent to the first order formula

$$\forall x [P(x) \equiv x = a] \wedge a = b.$$

The same result can be obtained by allowing just one of the constants a, b to vary.

Example 15 According to Example 8,

$$\text{CIRC}[\forall x (Q(x) \supset P(x)); P] \equiv \forall x (Q(x) \equiv P(x)).$$

The effect of circumscribing P with Q varied is stronger:

$$\text{CIRC}[\forall x (Q(x) \supset P(x)); P; Q] \equiv \forall x \neg P(x) \wedge \forall x \neg Q(x).$$

This is because any model of $\forall x (Q(x) \supset P(x))$ in which the extent of P is nonempty can be "improved" by making both P and Q empty.

Example 16 Let A be the conjunction of (the universal closures of) the axioms (4)–(6). We will show in Section 3.2 that the circumscription

$$\text{CIRC}[A; Ab; \mathit{Ontable}]$$

is equivalent to the conjunction of the axioms and with the definition (8) of Ab. If *Ontable* were fixed, we would have obtained a weaker (and less interesting) characterization of Ab:

$$Ab(x) \equiv \mathit{Block}(x) \wedge \neg \mathit{Ontable}(x).$$

The result of circumscription can only become stronger when the list of varied constants is extended: $\text{CIRC}[A; P; Z_1, \ldots, Z_m]$ is a logical consequence of $\text{CIRC}[A; P; Z_1, \ldots, Z_m, Z_{m+1}]$. Indeed, these symbols stand for

$$A(P, Z_1, \ldots, Z_m, Z_{m+1}) \wedge \neg \exists p z_1 \ldots z_m [A(p, z_1, \ldots, z_m, Z_{m+1}) \wedge p < P]$$

and

$$A(P, Z_1, \ldots, Z_m, Z_{m+1}) \wedge \neg \exists p z_1 \ldots z_m z_{m+1}[A(p, z_1, \ldots, z_m, z_{m+1}) \wedge p < P].$$

The second formula implies the first, because

$$A(p, z_1, \ldots, z_m, Z_{m+1}) \wedge p < P$$

implies

$$\exists z_{m+1}[A(p, z_1, \ldots, z_m, z_{m+1}) \wedge p < P].$$

2.4 Many-sorted and Higher Order Circumscription

The definition of circumscription (11) is applicable in a somewhat wider context than described above.

First, A can be a sentence of a *many-sorted* language. In many-sorted languages, a predicate constant is characterized not only by its arity, but also by the sorts of its arguments, and similarly for function constants and for predicate and function variables. To ensure then that (11) is syntactically correct, we should select p in such a way that the sort of each of its arguments be the same as the sort of the corresponding argument of P; a similar condition should be imposed on the choice of the variables z_i. The use of several sorts of variables is convenient for applications to knowledge representation, but, mathematically, the difference between one-sorted and many-sorted forms of circumscription is not very essential.

On the other hand, A can be a second order, or, generally, higher order sentence. Then (11) may have higher order quantifiers other than $\exists pz$. Moreover, the arguments of P and Z do not have to be individuals—they can be predicates or functions. Then, as in the case of many-sorted formulas, the variables p, z should be "similar" to the constants P, Z.

Higher order circumscription can be used, for instance, for defining when a set of blocks is a "tower." If object variables range over blocks, then sets of blocks can be represented by unary predicates. The following two formulas give a recursive definition of the relation $Tower(p, x)$, "p is a tower with the base x":

$$\begin{gathered} \forall y(p(y) \equiv y = x) \supset Tower(p, x), \\ Tower(p, x) \wedge On(x, y) \wedge \forall z(q(z) \equiv (p(z) \vee z = y)) \supset Tower(q, y). \end{gathered} \tag{14}$$

In addition, we need to say that $Tower(p, x)$ holds only if this follows from (14). One way to do that is to postulate the circumscription

$$\mathrm{CIRC}[A(Tower); Tower], \tag{15}$$

where $A(Tower)$ is the conjunction of the universal closures of the formulas (14). (Compare this use of circumscription with Examples 10 and 12.) The definition of circumscription, applied to (15), gives

$$A(Tower) \wedge \neg\exists tower[A(tower) \wedge tower < Tower].$$

Here *tower* is variable similar to the constant *Tower*, that is, a predicate variable whose arguments are a unary predicate and an individual.

We will encounter another example of higher order circumscription in Section 5.1.

2.5 Models of Circumscription

As was pointed out in Section 2.3, the models of $\mathrm{CIRC}[A; P; Z]$ are the models of A in which the extent of P cannot be made smaller without losing the property A, even at the price of changing the interpretations of the constants Z. We want to show now that this claim can be formulated as a precise mathematical proposition.

For simplicity, we restrict ourselves to the one-sorted case. Recall that a structure M for a one-sorted language is determined by its universe $|M|$ and by the interpretations $M[\![C]\!]$ of all individual, function and predicate constants C in the language. For two structures M_1, M_2 we write $M_1 \leq^{P;Z} M_2$ if

- $|M_1| = |M_2|$,
- $M_1[\![C]\!] = M_2[\![C]\!]$ for every constant C which is different from P and does not belong to Z,
- $M_1[\![P]\!] \subset M_2[\![P]\!]$.

In other words, $M_1 \leq^{P;Z} M_2$ means that M_1 and M_2 differ only in how they interpret P and Z, and the extent of P in M_1 is a subset of its extent in M_2.

The relation $\leq^{P;Z}$ is reflexive and transitive, so that we can talk about the elements of a given subclass of structures that are *minimal* relative to this relation. In particular, a model M of A is minimal relative to $\leq^{P;Z}$ if there is no model M' of A such that $M' <^{P;Z} M$. (We write $M_1 <^{P;Z} M_2$ if $M_1 \leq^{P;Z} M_2$ but not $M_2 \leq^{P;Z} M_1$.)

Proposition 1 *A structure M is a model of* CIRC$[A; P; Z]$ *if and only if M is minimal relative to $\leq^{P;Z}$.*

Proof 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 $M' <^{P;Z} M$.

We will see in Section 6.3 that CIRC$[A; P; Z]$ does not necessarily have models, even if A does.

The model-theoretic idea of minimization relative to $\leq^{P;Z}$ has a wider area of applicability than the definition of circumscription. Consider any classical theory Γ (that is, a set of sentences, called the *axioms* of Γ), whose language contains the constants P,Z. We can talk about the class of all models of Γ that are minimal relative to $\leq^{P;Z}$, and ask which sentences are true in all minimal models. If Γ has only finitely many axioms, then this amounts to considering the models of the circumscription

$$\mathrm{CIRC}\left[\bigwedge_{A \in \Gamma} A; P; Z\right],$$

and asking what the consequences of this circumscription are. If there are infinitely many axioms, then describing the minimal models of Γ in terms of circumscription would require a language with infinite conjunctions.

2.6 Circumscriptive Theories

The main motivation for the study of circumscription (and of nonmonotonic reasoning in general) comes from the theory of knowledge representation. Whenever a body of knowledge includes defaults, it cannot be directly represented as an axiomatic theory based on classical logic. The notion of a circumscriptive theory is a generalization of classical axiomatic theories, which attempts to overcome this difficulty using circumscription.

A classical axiomatic theory can be viewed simply as a set of sentences—its axioms. A circumscriptive theory will include, in addition, some declarations, describing how circumscription is to be applied to the conjunction of the axioms. These declarations characterize the *circumscription policy* of the theory. We will write the declaration which calls for circumscribing a predicate P with constants $Z_1, \ldots, Z_n$ varied as

$$\textbf{circ } P \textbf{ var } Z_1, \ldots, Z_n, \tag{16}$$

where $Z_1, \ldots, Z_n$ is a list of object, function and predicate constants not containing P. When $n = 0$, (16) will be abbreviated as **circ** P. By including (16) in a circumscriptive theory, we declare that we are only interested in the models of the axioms that are minimal with respect to the relation $\leq^{P;Z_1,\ldots,Z_n}$ defined above.

For instance, the blocks world example from the introduction corresponds to the circumscriptive theory whose axioms are the universal closures of the formulas (4)–(6), and which contains, in addition, the policy declaration

$$\textbf{circ } Ab \textbf{ var } Ontable.$$

Specifying a circumscriptive theory, like specifying a classical axiomatic theory, begins with choosing a language—one-sorted or many-sorted, first order or higher order. Then a circumscriptive theory is defined by a set Γ of sentences, called the *axioms* of the theory, and a set Δ of expressions of the form (16), called its *policy declarations*. A *model* of a circumscriptive theory (Γ, Δ) is any model of Γ which is minimal with respect to each of the relations $\leq^{P;Z_1,\ldots,Z_n}$ corresponding to the policy declarations (16) from Δ. A *theorem* of (Γ, Δ) is a sentence which is true in all its models.

How are these definitions related to the operator CIRC? Let Γ be finite and Δ nonempty. Then, in view of Proposition 1 (Section 2.5), the models of (Γ,Δ) are the same as the models of the classical axiomatic theory whose axioms are the circumscriptions

$$\mathrm{CIRC}\left[\bigwedge_{A\in\Gamma} A; P; Z_1, \ldots, Z_n\right] \tag{17}$$

for all declarations (16) from Δ.

Recall that circumscription is defined as the conjunction of two terms (11). The first conjunctive terms of all circumscriptions (17) corresponding to different policy declarations are identical, because each of them is $\bigwedge_{A\in\Gamma} A$.

What is essential in the contribution of each additional declaration is the second conjunctive term of the corresponding circumscription (17), expressing the minimality of P.

The formalism of circumscriptive theories is nonmonotonic, in the sense that some theorems may be lost when axioms are added to Γ. But it is clear that adding declarations to Δ can only make the theory stronger. Moreover, in view of the remark at the end of Section 2.3, extending a declaration (16) by adding more constants to the list $Z_1, \ldots, Z_n$ affects the theory monotonically also.

The semantics of circumscriptive theories can be characterized in terms of the operator CIRC even when the axiom set is infinite, if the following *local finiteness* condition holds: For any policy declaration (16) from Δ, the set of formulas from Γ containing any of the constants $P, Z_1, \ldots, Z_n$ is finite. Then the models of (Γ, Δ) are the same as the models of the classical axiomatic theory whose axiom set is obtained from Γ by adding, for each declaration (16) from Δ, the sentence

$$\mathrm{CIRC}\left[\bigwedge_{A \in \Gamma'} A; P; Z_1, \ldots, Z_n\right],$$

where Γ' is some finite part of Γ containing all occurrences of $P, Z_1, \ldots, Z_n$ in Γ.

Having further developed the theory of circumscription in the next section, we will discuss in detail some examples of circumscriptive theories, arising in connection with applications to inheritance reasoning (Section 4) and to reasoning about action and change (Section 5).

3 Rules for Computing Circumscription

In most of the examples above, we were able to simplify the circumscription formula in some way. More specifically, we could often rewrite the result of circumscription as a first order formula (Examples 1–9, 11, 13–16). In many cases, this first order formula had a particularly simple structure—it was an explicit definition of the circumscribed predicate (Examples 1–3, 5, 7, 8, 11, 13), or included such a definition as a conjunctive term (Examples 14–16). In this section we discuss computational methods for simplifying the circumscription formula. With appropriate changes, these methods can be used for simplifying higher order circumscription also.

3.1 Predicate Completion

In the following theorem, x is a tuple of variables, and $F(x)$ is a formula with all parameters explicitly shown.

Proposition 2 *If $F(x)$ does not contain P, then the circumscription*

$$\mathrm{CIRC}[\forall x(F(x) \supset P(x)); P]$$

is equivalent to

$$\forall x(F(x) \equiv P(x)).$$

Replacing the implication $F(x) \supset P(x)$ by the equivalence $F(x) \equiv P(x)$ is known as *predicate completion*.

Proposition 2 is a generalization of Example 8, where $F(x)$ is the atomic formula $Q(x)$. It can be applied to some of the other examples as well. For instance, the formula $P(a)$ is equivalent to

$$\forall x(x = a \supset P(x)).$$

Proposition 2 tells us that the result of circumscription is equivalent to the explicit definition of P given in Example 1. More generally, any ground atom $P(t_1, \ldots, t_n)$ can be written as $\forall x(F(x) \supset P(x))$, where $F(x)$ is $x_1 = t_1 \wedge \ldots \wedge x_n = t_n$, so that the circumscription

$$\mathrm{CIRC}[P(t_1, \ldots, t_n); P]$$

is equivalent to

$$\forall x_1 \ldots x_n[P(x_1, \ldots, x_n) \equiv x_1 = t_1 \wedge \ldots \wedge x_n = t_n].$$

The circumscriptions from Examples 7 and 11 can be computed using predicate completion also, because we can write $\forall x P(x)$ as

$$\forall x(\mathit{true} \supset P(x)),$$

and $\forall x P(x,x)$ as
$$\forall xy[x = y \supset P(x,y)].$$
As another example, consider the circumscription
$$\mathrm{CIRC}[\forall x P(f(x)); P]. \tag{18}$$
The formula $\forall x P(f(x))$ can be written as
$$\forall x[\exists y(x = f(y)) \supset P(x)].$$
Consequently, (18) is equivalent to
$$\forall x[P(x) \equiv \exists y(x = f(y))].$$

The conjunction of several formulas of the form $\forall x(F(x) \supset P(x))$ can be replaced by one such formula, because
$$\forall x(F_1(x) \supset P(x)) \wedge \ldots \wedge \forall x(F_k(x) \supset P(x))$$
is equivalent to
$$\forall x[(F_1(x) \vee \ldots \vee F_k(x)) \supset P(x)].$$
For instance, the formula $P(a) \wedge P(b)$ is equivalent to
$$\forall x[(x = a \vee x = b) \supset P(x)],$$
so that Proposition 2 allows us to justify Example 2 as well.

We see that the circumscription $\mathrm{CIRC}[A(P); P]$ can be computed by predicate completion whenever $A(P)$ is the conjunction of several formulas of the form $\forall x(F(x) \supset P(x))$. It may happen that $A(P)$ includes, in addition, several conjunctive terms that do not contain P at all. The presence of such terms does not make the task of computing circumscription any more difficult, because of the following general property of circumscription: For any sentence B not containing P, Z,
$$\mathrm{CIRC}[A(P,Z) \wedge B; P; Z] \equiv \mathrm{CIRC}[A(P,Z); P; Z] \wedge B. \tag{19}$$
This is immediate from the definition (11). For instance,
$$\mathrm{CIRC}[P(a) \wedge Q(a); P] \equiv \forall x(P(x) \equiv x = a) \wedge Q(a).$$

As another application of (19), we can compute the result of circumscribing P relative to a sentence which does not contain P:

$$\mathrm{CIRC}[B; P] \equiv B \wedge \forall x \neg P(x). \tag{20}$$

(This fact will be generalized in Section 3.3.) Proof:

$$\begin{aligned}\mathrm{CIRC}[B; P] &\equiv \mathrm{CIRC}[\forall x(\mathit{false} \supset P(x)) \wedge B; P] \\ &\equiv \mathrm{CIRC}[\forall x(\mathit{false} \supset P(x)); P] \wedge B \\ &\equiv \forall x(\mathit{false} \equiv P(x)) \wedge B \\ &\equiv B \wedge \forall x \neg P(x).\end{aligned}$$

In particular,

$$\mathrm{CIRC}[\mathit{true}; P] \equiv \forall x \neg P(x). \tag{21}$$

Proposition 2 can be stated more concisely using *predicate expressions* of the form

$$\lambda x_1 \dots x_n F(x_1, \dots, x_n), \tag{22}$$

where $F(x_1, \dots, x_n)$ is a formula. This formula may have free variables other than $x_1, \dots, x_n$; they are called the *parameters* of (22). For instance, $\lambda x(x = y)$ is a unary predicate expression with the parameter y. Like a predicate constant, a predicate expression without parameters represents a set of tuples of elements of the universe.

Predicate expressions are convenient for describing formulas obtained by substitution. If E is (22), and $t_1, \dots, t_n$ are terms, then $E(t_1, \dots, t_n)$ stands for the formula $F(t_1, \dots, t_n)$. For instance, $\lambda x(x = y)(a)$ stands for $a = y$. If $A(P)$ is a formula containing a predicate constant P, and E is a predicate expression of the same arity as P, then $A(E)$ stands for the result of replacing each atomic part $P(t_1, \dots, t_n)$ in $A(P)$ by $E(t_1, \dots, t_n)$ (after renaming the bound variables in A in the usual way, if necessary). For instance, if $A(P)$ is $P(a) \wedge P(b)$, then $A(\lambda x(x = y))$ is $a = y \wedge b = y$. For any predicate expression E, the formulas

$$\forall p A(p) \supset A(E), \quad A(E) \supset \exists p A(p)$$

are theorems of second order logic.

The notation $P = Q$, $P \leq Q$, $P < Q$, introduced in Section 2.1, can be applied when one or each of P, Q is a predicate expression. For instance,

$\forall x[x = a \supset P(x)]$ can be abbreviated as $\lambda x(x = a) \leq P$. In this notation, Proposition 2 can be stated as follows:

$$\mathrm{CIRC}[E \leq P; P] \equiv E = P \tag{23}$$

for any predicate expression E without parameters which does not contain P.

Proof of Proposition 2 The left-hand side of (23) stands for

$$E \leq P \wedge \neg \exists p(E \leq p \wedge p < P). \tag{24}$$

Notice that

$$\exists p(E \leq p \wedge p < P) \equiv E < P.$$

(To prove this right to left, take p to be E.) Consequently, (24) can be written as

$$E \leq P \wedge \neg(E < P),$$

which is equivalent to $E = P$.

The assumption that $F(x)$ in Proposition 2 does not contain P is essential. For instance,

$$\mathrm{CIRC}[\forall x(P(x) \supset P(x)); P]$$

is *not* equivalent to $\forall x(P(x) \equiv P(x))$, because, according to (21), this circumscription gives $\forall x \neg P(x)$. Circumscription relative to a formula of the form $\forall x(F(x) \supset P(x))$ can be computed by predicate completion only when this formula is "nonrecursive."

3.2 Elimination of Varied Constants

The following fact shows how circumscription with varied constants can be reduced to the basic case.

Proposition 3 *The circumscription*

$$\mathrm{CIRC}[A(P, Z); P; Z]$$

is equivalent to

$$A(P, Z) \wedge \mathrm{CIRC}[\exists z A(P, z); P].$$

The proof is given by the chain of equivalences:

$$\begin{aligned}
&A(P,Z) \wedge \mathrm{CIRC}[\exists z A(P,z); P] \\
&\qquad \equiv A(P,Z) \wedge \exists z A(P,z) \wedge \neg\exists p[\exists z A(p,z) \wedge p < P] \\
&\qquad \equiv A(P,Z) \wedge \neg\exists pz[A(p,z) \wedge p < P] \\
&\qquad \equiv \mathrm{CIRC}[A(P,Z); P; Z].
\end{aligned}$$

For instance, according to Proposition 3, (13) is equivalent to

$$P(a) \wedge P(b) \wedge \mathrm{CIRC}[\exists z_1 z_2 (P(z_1) \wedge P(z_2)); P]. \tag{25}$$

Since the formula $\exists z_1 z_2(P(z_1) \wedge P(z_2))$ is equivalent to $\exists z P(z)$, the circumscription in (25) is known to us from Example 9, and we can rewrite (25) as

$$P(a) \wedge P(b) \wedge \exists x \forall y [P(y) \equiv x = y].$$

This is equivalent to the answer given in Example 14.

If the list Z contains function constants or predicate constants, then $\mathrm{CIRC}[\exists z A(P,z); P]$ is a higher order circumscription, and we need to reduce $\exists z A(P,z)$ to a first order formula before simplfying this circumscription can be attempted. The following simple facts of second order logic are often useful. The formulas

$$\exists p \forall x [F(x) \supset p(x)], \quad \exists p \forall x [p(x) \supset F(x)] \tag{26}$$

are identically true whenever $F(x)$ does not contain p. The formula

$$\exists p \{\forall x [F(x) \supset p(x)] \wedge \forall x [p(x) \supset G(x)]\} \tag{27}$$

is equivalent to $\forall x[F(x) \supset G(x)]$, if $F(x)$ and $G(x)$ do not contain p. (To prove each of these assertions, take $p(x)$ to be $F(x)$.)

For instance, computing the circumscription from Example 15 can be reduced to (21):

$$\begin{aligned}
&\mathrm{CIRC}[\forall x(Q(x) \supset P(x)); P; Q] \\
&\qquad \equiv \forall x(Q(x) \supset P(x)) \wedge \mathrm{CIRC}[\exists q \forall x(q(x) \supset P(x)); P] \\
&\qquad \equiv \forall x(Q(x) \supset P(x)) \wedge \mathrm{CIRC}[true; P] \\
&\qquad \equiv \forall x(Q(x) \supset P(x)) \wedge \forall x \neg P(x) \\
&\qquad \equiv \forall x \neg P(x) \wedge \forall x \neg Q(x).
\end{aligned}$$

In Example 16, A is the conjunction $A_1 \wedge A_2 \wedge A_3$, where

$$A_1 \equiv \forall x[Block(x) \wedge \neg Ab(x) \supset Ontable(x)],$$

$$A_2 \equiv \neg Ontable(B_1),$$

$$A_3 \equiv Block(B_1) \wedge Block(B_2) \wedge B_1 \neq B_2.$$

To determine the effect of circumscription in this case, we first notice that, by Proposition 3,

$$\mathrm{CIRC}[A; Ab; Ontable] \equiv A \wedge \mathrm{CIRC}[\exists Ontable.A; Ab].$$

(We take the liberty of using *Ontable* both as a predicate constant and as a bound predicate variable.) The quantifier $\exists Ontable$ can be eliminated in the following way:

$$\begin{aligned}
&\exists Ontable.A \\
&\quad \equiv \exists Ontable(A_1 \wedge A_2) \wedge A_3 \\
&\quad \equiv \exists Ontable\{\forall x[Block(x) \wedge \neg Ab(x) \supset Ontable(x)] \\
&\qquad\qquad\qquad\qquad\qquad \wedge \forall x[Ontable(x) \supset x \neq B_1]\} \wedge A_3 \\
&\quad \equiv \forall x[Block(x) \wedge \neg Ab(x) \supset x \neq B_1] \wedge A_3 \\
&\quad \equiv \forall x[Block(x) \wedge x = B_1 \supset Ab(x)] \wedge A_3.
\end{aligned}$$

The circumscription of Ab in this formula can be done by predicate completion. We continue the computation as follows:

$$\begin{aligned}
&\mathrm{CIRC}[A; Ab; Ontable] \\
&\qquad \equiv A \wedge \forall x[Ab(x) \equiv (Block(x) \wedge x = B_1)] \wedge A_3 \\
&\qquad \equiv A \wedge \forall x[Ab(x) \equiv (Block(x) \wedge x = B_1)] \\
&\qquad \equiv A \wedge \forall x[Ab(x) \equiv x = B_1],
\end{aligned}$$

since A includes the axiom $Block(B_1)$, and in its presence $x = B_1$ implies $Block(x)$.

To further illustrate this method, consider the modification of Example 16 which includes the additional axiom

$$\neg Ontable(B_2).$$

We include this axiom in A_2, so that now A_2 stands for

$$\neg Ontable(B_1) \wedge \neg Ontable(B_2).$$

In the process of eliminating $\exists Ontable$, we rewrite this formula as

$$\forall x[Ontable(x) \supset x \neq B_1 \wedge x \neq B_2],$$

so that $\exists Ontable.A$ becomes

$$\forall x[Block(x) \wedge (x = B_1 \vee x = B_2) \supset Ab(x)] \wedge A_3,$$

and the final answer is

$$A \wedge \forall x[Ab(x) \equiv (x = B_1 \vee x = B_2)].$$

If z consists of several predicate variables $z_1, \ldots, z_n$, then, using transformations of this kind, we may be able to eliminate all quantifiers $\exists z_1, \ldots, \exists z_n$ one by one.

3.3 Positive and Negative Formulas

Recall that an occurrence of a predicate symbol in a formula is *positive*, if it is in the range of an even number of negations, and *negative* otherwise (assuming that $\supset$ and $\equiv$ have been eliminated in favor of other connectives). A formula $A(P)$ is *positive* (relative to P) if all occurrences of P in it are positive, and *negative* if all occurrences of P are negative. Obviously, a formula which does not contain P at all is both positive and negative.

Several useful properties of circumscription can be stated in terms of positive and negative formulas. They are based on the following two facts of second order logic: If $A(P)$ is positive, then

$$\forall pq[A(p) \wedge p \leq q \supset A(q)]$$

is universally valid; if $A(P)$ is negative, then

$$\forall pq[A(p) \wedge q \leq p \supset A(q)]$$

is universally valid. These facts are easy to prove simultaneously by induction on the size of $A(P)$.

Proposition 4 *If $B(P)$ is negative, then the circumscription*

$$\mathrm{CIRC}[A(P) \wedge B(P); P]$$

is equivalent to

$$\mathrm{CIRC}[A(P); P] \wedge B(P).$$

This is similar to the equivalence (19), except that there B was supposed to have no occurrences of P at all, and now negative occurrences are permitted. On the other hand, Proposition 4 deals with the basic case of circumscription only (no varied constants).

Proof $\mathrm{CIRC}[A(P) \wedge B(P); P]$ stands for

$$A(P) \wedge B(P) \wedge \neg\exists p[A(p) \wedge B(p) \wedge p < P].$$

Since $B(P)$ is negative, $B(P)$ and $p < P$ imply $B(p)$. Hence the part in the brackets can be replaced by $A(p) \wedge p < P$. It remains to notice that

$$A(P) \wedge \neg\exists p \neg[A(p) \wedge p < P]$$

is $\mathrm{CIRC}[A(P); P]$.

As a corollary, we get that (20) holds whenever B is negative, and not only when B does not contain P at all, as we required in Section 3.1. For instance, (20) can be used for determining the effect of circumscription in Example 2:

$$\mathrm{CIRC}[\neg P(a); P] \equiv \neg P(a) \wedge \forall x \neg P(x) \equiv \forall x \neg P(x).$$

If $A(P, Z)$ is positive relative to P, then the second order quantifier $\exists p$ in $\mathrm{CIRC}[A(P, Z); P; Z]$ can be always eliminated:

Proposition 5 *If $A(P, Z)$ is positive relative to P, then the circumscription*

$$\mathrm{CIRC}[A(P, Z); P; Z]$$

is equivalent to

$$A(P, Z) \wedge \neg\exists xz[P(x) \wedge A(\lambda y(P(y) \wedge x \neq y), z)]. \tag{28}$$

The second conjunctive term of (28) expresses that it is impossible to get a model of A by removing *exactly one element* from the set representing P (and possibly changing Z).

For instance, Proposition 5 allows us to do the circumscription from Example 1 as follows:

$$\begin{aligned}\mathrm{CIRC}[P(a); P] &\equiv P(a) \wedge \neg\exists x[P(a) \wedge x \neq a] \\ &\equiv P(a) \wedge \forall x[P(x) \supset x = a] \\ &\equiv \forall x[P(x) \equiv x = a].\end{aligned}$$

Example 9 can be done in a similar way:

$$\begin{aligned}\mathrm{CIRC}[\exists x P(x); P] &\equiv \exists x P(x) \wedge \neg\exists x[P(x) \wedge \exists x'[P(x') \wedge x \neq x']] \\ &\equiv \exists x P(x) \wedge \forall x x'[P(x) \wedge P(x') \supset x = x'].\end{aligned}$$

The last formula is equivalent to the answer given in Section 2.1.

Proof of Proposition 5 It is sufficient to prove the equivalence

$$\exists p[A(p, z) \wedge p < P] \equiv \exists x[P(x) \wedge A(\lambda y(P(y) \wedge x \neq y), z)].$$

Left to right: Take p such that $A(p, z)$ and $p < P$, and let x be such that $P(x)$ and $\neg p(x)$. Clearly, $p \leq \lambda y(P(y) \wedge x \neq y)$. Since $A(P, Z)$ is positive relative to P, it follows that $A(\lambda y(P(y) \wedge x \neq y), z)$. Right to left: Take x such that $P(x)$ and $A(\lambda y(P(y) \wedge x \neq y), z)$, and take p to be $\lambda y(P(y) \wedge x \neq y)$.

Proposition 5 shows that $\mathrm{CIRC}[A(P); P]$ can be written as the first order sentence

$$A(P) \wedge \neg\exists x[P(x) \wedge A(\lambda y(P(y) \wedge x \neq y))]$$

whenever $A(P)$ is first order and positive. Using Proposition 4, we conclude from this fact:

Corollary 1 *If $A(P)$ is a positive first order sentence, and $B(P)$ is a negative first order sentence, then the circumscription $\mathrm{CIRC}[A(P) \wedge B(P); P]$ is equivalent to a first order sentence.*

Thus a circumscription is first order whenever the negative occurrences of the circumscribed predicate are "separated" from the positive occurrences by $\wedge$. Of course, the formulas from Examples 10 and 12 (Section 2.1), where the result of circumscription cannot be written without second order quantifiers, do not have this form.

Unfortunately, some very simple cases when circumscription is first order are not covered by Corollary 1. For instance, the formula from Example 5 can be written as $\neg P(a) \vee P(b)$; the positive and negative parts are separated by $\vee$, rather than $\wedge$. The following is known about applying circumscription to disjunctions:

Proposition 6 *Let $F(P)$ be a sentence of the form*

$$\bigvee_{i=1}^{k} [A_i(P) \wedge P \leq E_i] \;\vee\; \bigvee_{j=1}^{l} [E'_j \leq P \wedge B_j(P)], \tag{29}$$

where $A_1(P), \ldots, A_k(P)$ are positive sentences, $B_1(P), \ldots, B_l(P)$ are negative sentences, and $E_1, \ldots, E_k, E'_1, \ldots, E'_l$ are predicate expressions not containing P. Then the circumscription $\mathrm{CIRC}[F(P); P]$ is equivalent to

$$\begin{aligned} F(P) \wedge \textstyle\bigwedge_{i=1}^{k} \{\neg A_i(\lambda x(P(x) \wedge E_i(x))) \vee \mathrm{CIRC}[A_i(P); P]\} \\ \wedge \textstyle\bigwedge_{j=1}^{l} \{\neg (E'_j < P) \vee \neg B_j(E'_j)\}. \end{aligned}$$

The circumscriptions $\mathrm{CIRC}[A_i(P); P]$ can be computed using Proposition 5. Notice that Proposition 2 is a special case of this theorem (take $k = 0$, $l = 1$, $B_1(P) \equiv true$). Every disjunction $A(P) \vee B(P)$, where $A(P)$ is positive and $B(P)$ is negative, can be written in the form (29) (take $k = l = 1$, $E_1 = \lambda x.true$, $E'_1 = \lambda x.false$). Any quantifier-free sentence can be converted to this form, with $k = 0$ (write it in disjunctive normal form, and, using the methods of Section 3.1, write the positive part of the j-th conjunction as $E'_j \leq P$).

As the last example of the use of positivity and negativity, we will mention that these concepts allow us to generalize the method for simplifying second order formulas of the forms (26) and (27) used in Section 3.2. Namely, it is easy to see that, for any positive formula $A(P)$ and negative formula $B(P)$,

- $\exists p A(p)$ is equivalent to $A(\lambda x.true)$,
- $\exists p B(p)$ is equivalent to $B(\lambda x.false)$,
- $\exists p[A(p) \wedge p \leq E]$ is equivalent to $A(E)$,

- $\exists p[E \leq p \land B(p)]$ is equivalent to $B(E)$.

(Here E is a predicate expression which does not have p among its parameters).

The last two assertions suggest the question whether, more generally, any formula of the form

$$\exists p[A(p) \land B(p)],$$

where $A(P)$ is first order positive and $B(P)$ first order negative, is always equivalent to a first order formula. The answer to this question is no. Let $F(P)$ be the formula

$$\forall y \exists x P(x,y) \land \forall xyz[P(x,y) \land P(x,z) \supset y = z] \land \exists x \neg \exists y P(x,y).$$

It expresses that P is a mapping of a proper subset of the universe onto the whole universe, so that $\exists p F(p)$ is true if and only if the universe is infinite. It follows that this sentence is not equivalent to a first order formula.

3.4 Confirming Conjectures about Circumscription

When circumscription is applied to a formula with a complicated syntactic structure, not covered by any of the results from Sections 3.1–3.3, there is still hope that our understanding of the intuitive meaning of the formula may suggest a reasonable conjecture about the extent of the circumscribed predicate. How can we prove such a conjecture? Using directly the definition of circumscription is cumbersome, and this can be often avoided.

Let us begin with the basic case. We would like to prove that the effect of the circumscription $\mathrm{CIRC}[A(P);P]$ is to make P equal to E, where E is a predicate expression without parameters, not containing P. More precisely, we would like to prove the equivalence

$$\mathrm{CIRC}[A(P);P] \equiv A(P) \land P = E.$$

We will show that, in order to do this, it is sufficient to establish two facts:

$$A(P) \supset A(E) \quad \text{and} \quad A(P) \supset E \leq P. \tag{30}$$

As an example, take the formula $P(a) \land \neg P(b)$. We want to confirm the conjecture that circumscribing P relative to this formula makes P equal to

$\lambda x(x = a)$. For $P(a) \wedge \neg P(b)$ as $A(P)$ and $\lambda x(x = a)$ as E, the conditions (30) turn into

$$P(a) \wedge \neg P(b) \supset a = a \wedge b \neq a$$

and

$$P(a) \wedge \neg P(b) \supset \forall x(x = a \supset P(x)).$$

Both formulas are indeed universally valid, so that we can conclude:

$$\begin{aligned} \mathrm{CIRC}[P(a) \wedge \neg P(b); P] &\equiv P(a) \wedge \neg P(b) \wedge \forall x(P(x) \equiv x = a) \\ &\equiv \forall x(P(x) \equiv x = a) \wedge \neg P(b). \end{aligned}$$

What we are going to prove is an extension of the principle used in this example to circumscription with varied predicates:

Proposition 7 *Let E be a predicate expression without parameters, containing neither P nor Z. If the sentences*

$$A(P, Z) \supset \exists z A(E, z) \tag{31}$$

and

$$A(P, Z) \supset E \leq P \tag{32}$$

are universally valid, then so is the sentence

$$\mathrm{CIRC}[A(P, Z); P; Z] \equiv A(P, Z) \wedge P = E. \tag{33}$$

Proof To prove (33) left to right, assume $\mathrm{CIRC}[A(P, Z); P; Z]$, that is,

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

From the first conjunctive term we conclude, using (32), that $E \leq P$, and, using (31), that $\exists z A(E, z)$. From the second conjunctive term,

$$\neg \exists z[A(E, z) \wedge E < P].$$

The last two formulas imply $\neg(E < P)$. In combination with $E \leq P$, this gives $P = E$. Right to left: assume

$$A(P, Z) \wedge P = E. \tag{34}$$

Since (32) is universally valid, so is

$$\forall pz[A(p,z) \supset E \leq p] \tag{35}$$

(constants in a universally valid formula can be replaced by universally quantified variables). Assume $A(p,z) \wedge p < P$. Then, by (35), $E \leq p < P$, contrary to the second term of (34). This contradiction proves the second term of the circumscription in (33).

By essentially the same argument we can prove the relativized form of Proposition 7: For any set Γ of sentences containing neither P nor Z, if Γ implies (31) and (32) (that is, if these sentences are true in every model of Γ), then Γ implies (33).

To illustrate the use of Proposition 7 with a nonempty Z, let us go back to Example 15 (Section 2.3). We want to show that the circumscription

$$\mathrm{CIRC}[\forall x(Q(x) \supset P(x)); P; Q]$$

makes P equal to $\lambda x.\mathit{false}$. The conditions (31) and (32) in this case are

$$\forall x(Q(x) \supset P(x)) \supset \exists q \forall x(q(x) \supset \mathit{false})$$

and

$$\forall x(Q(x) \supset P(x)) \supset \forall x(\mathit{false} \supset P(x)).$$

It is clear that they are universally valid (in the first of them, take q to be $\lambda x.\mathit{false}$). Then, by Proposition 7, the circumscription is equivalent to

$$\forall x(Q(x) \supset P(x)) \wedge \forall x \neg P(x),$$

which is obviously equivalent to the answer given in Example 15.

3.5 Definite Formulas

A formula is *definite* in the predicates $P_1, \ldots, P_n$ if it is the conjunction of implications of the form

$$F(P_1, \ldots, P_n) \supset P_i(t_1, \ldots, t_k), \tag{36}$$

where $t_1, \ldots, t_k$ are terms and $F(P_1, \ldots, P_n)$ is a formula positive relative to each of $P_1, \ldots, P_n$. For instance, a conjunction of definite clauses[3] can be written as a formula definite in the set of all its predicates.

The following theorem shows that circumscribing a predicate in a definite formula always produces an explicit definition. Unlike Proposition 2 (Section 3.1), it only gives a *higher order definition*, but the class of formulas to which it is applicable is much wider. For instance, Examples 10 and 12 from Section 2.1, as well as the tower example from Section 2.4 are definite.

Proposition 8 *If $A(P_1, \ldots, P_n)$ is the universal closure of a formula definite in $P_1, \ldots, P_n$, then the circumscription*

$$\mathrm{CIRC}[A(P_1, \ldots, P_n); P_1; P_2, \ldots, P_n]$$

is equivalent to

$$A(P_1, \ldots, P_n) \wedge \forall x [P_1(x) \equiv \forall p_1 \ldots p_n (A(p_1, \ldots, p_n) \supset p_1(x))]. \qquad (37)$$

In the proof of Proposition 8, we denote the predicate expression

$$\lambda x \forall p_1 \ldots p_n (A(p_1, \ldots, p_n) \supset p_i(x))$$

by E_i $(i = 1, \ldots, n)$.

Lemma 1 *The sentence $A(E_1, \ldots, E_n)$ is universally valid.*

Proof The sentence $A(E_1, \ldots, E_n)$ is the universal closure of the conjunction of some formulas of the form

$$F(E_1, \ldots, E_n) \supset E_i(t_1, \ldots, t_k),$$

that is,

$$F(E_1, \ldots, E_n) \supset \forall p_1 \ldots p_n (A(p_1, \ldots, p_n) \supset p_i(t_1, \ldots, t_k)).$$

Take any of these formulas, and assume $F(E_1, \ldots, E_n)$ and $A(p_1, \ldots, p_n)$. From the last assumption and the definition of E_i we conclude that $E_i(x)$

[3]A clause is *definite* if exactly one of its literals is positive.

implies $p_i(x)$, so that $E_i \leq p_i$ $(i = 1, \ldots, n)$. Since all occurrences of the predicate symbols $P_1, \ldots, P_n$ in $F(P_1, \ldots, P_n)$ are positive, we get $F(p_1, \ldots, p_n)$. Since $A(p_1, \ldots, p_n)$ contains the universal closure of

$$F(p_1, \ldots, p_n) \supset p_i(t_1, \ldots, t_k)$$

as a conjunctive term, we conclude that $p_i(t_1, \ldots, t_k)$.

Proof of Proposition 8 According to Proposition 7 (Section 3.4), it is sufficient to verify that the sentences

$$A(P_1, \ldots, P_n) \supset \exists p_2 \ldots p_n A(E_1, p_2, \ldots, p_n)$$

and

$$A(P_1, \ldots, P_n) \supset E_1 \leq P_1$$

are universally valid. The validity of the former follows from the lemma (take $p_2, \ldots, p_n$ to be $E_2, \ldots, E_n$). The latter can be written as

$$A(P_1, \ldots, P_n) \supset \forall x[\forall p_1 \ldots p_n(A(p_1, \ldots, p_n) \supset p_1(x)) \supset P_1(x)],$$

and it is sufficient to substitute $P_1, \ldots, P_n$ for $p_1, \ldots, p_n$.

Corollary 2 *If $A(P)$ is the universal closure of a formula definite in P, then the circumscription* $\mathrm{CIRC}[A(P); P]$ *is equivalent to*

$$\forall x[P(x) \equiv \forall p(A(p) \supset p(x))]. \tag{38}$$

Proof According to Proposition 8, $\mathrm{CIRC}[A(P); P]$ is equivalent to

$$A(P) \wedge \forall x[P(x) \equiv \forall p(A(p) \supset p(x))],$$

and consequently to

$$A(\lambda x \forall p(A(p) \supset p(x))) \wedge \forall x[P(x) \equiv \forall p[A(p) \supset p(x)]].$$

According to the lemma, the first conjunctive term is universally valid and can be dropped.

The explicit definition (38) says simply that P is the smallest predicate satisfying $A(P)$.

4 Circumscriptive Theories of Inheritance

As discussed in Chapter 3 of this volume, there are two kinds of approaches to formalizing nonmonotonic inheritance reasoning: "direct" and "translational." The language of circumscriptive theories (Section 2.6) can be used as the target of a translational semantics. In this section, we show how some simple examples of inheritance reasoning can be represented by circumscriptive theories.

4.1 Three Examples

We have already seen one example of inheritance reasoning formalized using circumscription—the blocks world example from the introduction. The block B_2 "inherits" its property *Ontable* from the class of blocks. In the notation of Chapter 3, it can be represented by a mixed inheritance network with 4 links:

$$\{Block \rightarrow Ontable, B_1 \not\Rightarrow Ontable, B_1 \Rightarrow Block, B_2 \Rightarrow Block\}.$$

The corresponding circumscriptive theory is

$$\begin{array}{c} Block(x) \wedge \neg Ab(x) \supset Ontable(x), \\ \neg Ontable(B_1), \\ Block(B_1),\ Block(B_2),\ B_1 \neq B_2, \\ \mathbf{circ}\ Ab\ \mathbf{var}\ Ontable. \end{array} \tag{39}$$

We will discuss three modifications of this example, and show how each of them can be represented by a circumscriptive theory.

First, instead of saying that B_1 is an exception in the strong sense that it is definitely not on the table, we can express that B_1 is a *possible* exception, and thus leave the question about its location open. (This modification cannot be directly expressed in the network notation of Chapter 3.) To this end, we replace the axiom $\neg Ontable(B_1)$ in (39) by $Ab(B_1)$. For the new theory, the computation of the value of Ab leads to the same result as in the original example, that is,

$$Ab(x) \equiv x = B_1$$

(see Section 3.2). The application of the default assumption about the positions of blocks to B_1 is disabled, and neither $Ontable(B_1)$ nor $\neg Ontable(B_1)$ can be proved.

Second, we can describe how blocks inherit several different properties, rather than just one property *Ontable*. For instance, we can assume additionally that blocks can be "heavy," and that, by default, a block is assumed to be not heavy. In terms of inheritance networks, this corresponds to adding the defeasible link $Block \not\to Heavy$.

To formalize this as a circumscriptive theory, we need two abnormality predicates, corresponding to two defaults. The postulates of the new theory are:

$$\begin{gathered}
Block(x) \wedge \neg Ab_1(x) \supset Ontable(x), \\
Block(x) \wedge \neg Ab_2(x) \supset \neg Heavy(x), \\
\neg Ontable(B_1), \\
Block(B_1),\ Block(B_2),\ B_1 \neq B_2, \\
\mathbf{circ}\ Ab_1\ \mathbf{var}\ Ontable, \\
\mathbf{circ}\ Ab_2\ \mathbf{var}\ Heavy.
\end{gathered}$$

We selected the circumscription policy in accordance with the following informal principle: If the default which is formalized using Ab_i characterizes a predicate Z, then Z is varied as Ab_i is circumscribed. The methods of Sections 3.1, 3.2, applied to these two circumscriptions, give:

$$Ab_1(x) \equiv x = B_1, \quad Ab_2(x) \equiv false.$$

If we add the axiom $Heavy(B_2)$, then the right-hand side of the last equivalence will turn into $x = B_2$.

Let us consider now a case of two conflicting defaults. Blocks are typically on the table, except for heavy blocks, that are typically not on the table. B_1 is heavy and B_2 is not. In network notation:

$$\begin{gathered}
\{Block \to Ontable, HeavyBlock \not\to Ontable, HeavyBlock \Rightarrow Block, \\
B_1 \Rightarrow HeavyBlock, B_2 \Rightarrow Block, B_2 \not\Rightarrow HeavyBlock\}.
\end{gathered}$$

Here are some of the postulates we can use:

$$\begin{gathered}
Block(x) \wedge \neg Ab_1(x) \supset Ontable(x), \\
HeavyBlock(x) \wedge \neg Ab_2(x) \supset \neg Ontable(x), \\
HeavyBlock(x) \supset Block(x), \\
HeavyBlock(B_1),\ Block(B_2),\ \neg HeavyBlock(B_2),\ B_1 \neq B_2, \\
\mathbf{circ}\ Ab_1\ \mathbf{var}\ Ontable, \\
\mathbf{circ}\ Ab_2\ \mathbf{var}\ Ontable.
\end{gathered} \tag{40}$$

There is one more idea that needs to be incorporated in this circumscriptive theory. Since the class of heavy blocks is a subclass of blocks, default assumptions about properties of heavy blocks are more specific than assumptions about blocks in general, and consequently should be given precedence in case of conflict. In particular, the conflict between the first and second of the axioms (40) should be resolved in favor of the second one.

This idea can be expressed by disabling the first default for all heavy blocks:

$$HeavyBlock(x) \supset Ab_1(x). \tag{41}$$

Axioms like this are known as *cancellation of inheritance* axioms. With this axiom added to the list (40), the effect of circumscription is described by the formulas:

$$Ab_1(x) \equiv HeavyBlock(x), \quad Ab_2(x) \equiv false.$$

Using these equivalences, we can derive from the axioms that B_2 is on the table, and B_1 is not.

4.2 Priorities

Instead of adding (41) to the axiom set, we could have expressed the specificity principle in this example by changing the circumscription policy. Let us replace the last policy declaration in (40) by

$$\mathbf{circ}\ Ab_2\ \mathbf{var}\ Ontable, Ab_1. \tag{42}$$

The new circumscription policy requires that Ab_2 be minimal in a stronger sense than Ab_1: In a model of this theory, the extent of Ab_2 cannot be made smaller even at the price of changing the extent of Ab_1. In this sense, the new policy assigns to Ab_2 a "higher priority" than to Ab_1.

More generally, whenever, in a circumscriptive theory, the goals of circumscribing two predicates are in conflict with each other, these goals can be "prioritized" by placing one of the predicates in the list of varied constants in the policy declaration for the other.[4]

[4] An alternative approach to expressing priorities is based on the concept of prioritized circumscription (Section 7.2).

Consider, for instance, the theory

$$\begin{array}{c} Q(x) \supset P_1(x) \vee P_2(x), \\ \textbf{circ } P_1, \\ \textbf{circ } P_2. \end{array} \tag{43}$$

(As in classical axiomatic theories, a formula with free variables represents its universal closure.) There is tension between the tasks of minimizing P_1 and P_2, because, according to the axiom, $P_1(x)$ and $P_2(x)$ cannot be both false whenever the value of x satisfies Q. Since neither P_1 nor P_2 is varied as the other is circumscribed, the circumscription policy in (43) requires that the extent of each of the two predicates be made smaller only as far as this can be done without changing the extent of the other. The effects of both circumscriptions can be easily determined by predicate completion; they are equivalent, respectively, to

$$\forall x(P_1(x) \equiv Q(x) \wedge \neg P_2(x))$$

and

$$\forall x(P_2(x) \equiv Q(x) \wedge \neg P_1(x)).$$

Any structure in which P_1 and P_2 form a partition of Q is a model of (43). Let us see, on the other hand, what happens if P_2 is included as a varied predicate in the policy declaration for P_1:

$$\begin{array}{c} Q(x) \supset P_1(x) \vee P_2(x), \\ \textbf{circ } P_1 \textbf{ var } P_2, \\ \textbf{circ } P_2. \end{array} \tag{44}$$

The new policy circumscribes P_1 "with a higher priority" than P_2. The effect of this circumscription is to make P_1 identically false: $\forall x \neg P_1(x)$. In the presence of this formula, the result of the second circumscription can be further simplified: $\forall x(P_2(x) \equiv Q(x))$. Thus assigning a higher priority to the minimization of P_1 has the effect of making P_2 equivalent to Q.

Since the introduction of priorities is accomplished by extending a list of varied predicates, it can only make the theory stronger.

The specificity principle in inheritance reasoning is only one of many applications of this idea. Here is an example of a different kind. There are two sources of information about the world, and any event reported by either of the sources is presumed to have happened:

$$Reported_1(x) \wedge \neg Ab_1(x) \supset Happened(x),$$
$$Reported_2(x) \wedge \neg Ab_2(x) \supset Happened(x).$$

Two different announcements are made by these sources:

$$Reported_1(A), \quad \neg Reported_2(A),$$
$$Reported_2(B), \quad \neg Reported_1(B),$$

and these reports contradict each other:

$$\neg(Happened(A) \wedge Happened(B)).$$

The circumscriptive theory with these axioms and with the declarations

$$\mathbf{circ}\ Ab_1\ \mathbf{var}\ Happened,$$
$$\mathbf{circ}\ Ab_2\ \mathbf{var}\ Happened$$

has models of two kinds: In some models $Happened(A)$ is true, in the others $Happened(B)$. Neither the former nor the latter is a theorem. Giving a higher priority to one of the abnormality predicates would allow us to arrive at a definite conclusion about these sentences. If, for instance, we consider the first source more reliable, then we replace the first declaration by

$$\mathbf{circ}\ Ab_1\ \mathbf{var}\ Happened, Ab_2.$$

In this theory we can prove $Happened(A)$ and $\neg Happened(B)$.

Further examples of using priorities for representing knowledge can be found in the next section.

Let us go back now to the formalization of the blocks world example which uses the policy declaration (42). It is easy to check that it leads to the same values of Ab_1 and Ab_2 as the formalization using cancellation of inheritance. In some other cases, these methods lead to somewhat different results. This happens, for instance, if we assume that B_1 is on the table, contrary to the default assumption about heavy blocks in general. With the formula $Ontable(B_1)$ added to (40), the formulation with cancellation of inheritance will give:

$$Ab_1(x) \equiv HeavyBlock(x), \quad Ab_2(x) \equiv x = B_1.$$

The prioritization method gives a different value for Ab_1:

$$Ab_1(x) \equiv HeavyBlock(x) \wedge x \neq B_1.$$

However, both formalizations lead to the same conclusions about the locations of blocks, namely: B_1 and B_2 are on the table, and all the other blocks are not. Since the abnormality predicates play an auxiliary role in these formalizations, we can say that the difference is not particularly essential.

A hierarchy may include more than two levels of conflicting defaults, and then a many-level prioritization will be required. Consider, for example, the additional "top level" assumption that a thing in general is, typically, not to be found on the table. This assumption is less specific than the default assumptions about the locations of blocks and of heavy blocks introduced earlier. The extended example can be formalized as follows:

$$\begin{gathered}
\neg Ab_0(x) \supset \neg Ontable(x), \\
Block(x) \wedge \neg Ab_1(x) \supset Ontable(x), \\
HeavyBlock(x) \wedge \neg Ab_2(x) \supset \neg Ontable(x), \\
HeavyBlock(x) \supset Block(x), \\
HeavyBlock(B_1),\ Block(B_2),\ \neg HeavyBlock(B_2),\ B_1 \neq B_2, \\
\textbf{circ } Ab_0 \textbf{ var } Ontable, \\
\textbf{circ } Ab_1 \textbf{ var } Ontable, Ab_0, \\
\textbf{circ } Ab_2 \textbf{ var } Ontable, Ab_0, Ab_1.
\end{gathered}$$

This theory establishes 3 priority levels: Ab_0 is "the least important," Ab_1 is "intermediate," and Ab_2 has "the highest priority."

4.3 Unresolved Conflicts in Inheritance Hierarchies

In some inheritance hierarchies, there are no grounds for deciding which of two opposite conclusions, suggested by different paths in the hierarchy, should be given precedence. Then we expect the formalization to be incomplete. The standard example involves Quakers, who are normally pacifists, and Republicans, who are normally nonpacifists; these conflicting defaults do not lead to a definite conclusion in the case of a person who is both a Quaker and a Republican. This example is represented by the following circumscriptive theory:

$$\begin{gathered} \mathit{Quaker}(x) \wedge \neg Ab_1(x) \supset \mathit{Pacifist}(x), \\ \mathit{Republican}(x) \wedge \neg Ab_2(x) \supset \neg \mathit{Pacifist}(x), \\ \textbf{circ } Ab_1 \textbf{ var } \mathit{Pacifist}, \\ \textbf{circ } Ab_2 \textbf{ var } \mathit{Pacifist} \end{gathered}$$

Using the methods of Sections 3.1 and 3.2, we can show that this theory is equivalent to the conjunction of its two axioms with the following formulas, relating Ab_1 and Ab_2:

$$\begin{aligned} Ab_1(x) &\equiv \mathit{Quaker}(x) \wedge \mathit{Republican}(x) \wedge \neg Ab_2(x), \\ Ab_2(x) &\equiv \mathit{Quaker}(x) \wedge \mathit{Republican}(x) \wedge \neg Ab_1(x). \end{aligned}$$

It follows that we can decide about each person who is a Quaker or a Republican *but not both*, whether he is a pacifist or not:

$$\begin{gathered} \mathit{Quaker}(x) \wedge \neg \mathit{Republican}(x) \supset \mathit{Pacifist}(x), \\ \mathit{Republican}(x) \wedge \neg \mathit{Quaker}(x) \supset \neg \mathit{Pacifist}(x). \end{gathered}$$

No conclusion can be made about the value of $\mathit{Pacifist}(x)$ whenever x satisfies both *Quaker* and *Republican*.

In Chapter 3, this example is discussed in the framework of the nontranslational approach to inheritance reasoning.

5 Circumscriptive Theories of Action

The formalism of circumscriptive theories is used in this section for representing facts about action and change. Nonmonotonic reasoning is relevant here in many ways, and a future volume of this *Handbook* will contain a special chapter on nonmonotonic temporal reasoning. Here we examine one particularly promising approach in some detail.

The syntax for describing actions which is known as the "situation calculus" is based on the view that an *action*, executed in a particular *situation*, leads to a new situation, called the *result* of the action. Intuitively, the snapshot of the world at any instant of time can be thought of as a situation, although some situations may be never realized in the actual course of history. A *fluent* is a function defined on situations. For instance, "the number of blocks on the table" is a fluent whose values are nonnegative integers;

"block B_1 is on the table" is a *propositional* fluent, that is, a fluent whose possible values are *false* and *true*.

As a preliminary step, we will develop a formal theory of situations and propositional fluents for the blocks world. Then the theory will be extended by postulates for moving blocks.

5.1 Situations and States

We will use variables of three sorts: for blocks (x,y,z,x_1,y_1,z_1,...), for situations (s,s_1,...), and for propositional fluents (f,f_1,...). The atomic formula $Holds(f,s)$ expresses that the value of f in the situation s is *true*. The specific fluents that are of interest to us in this example will be represented using the function constants *On* and *Ontable*. For instance, the atomic formulas $Holds(On(x,y),s)$ and $Holds(Ontable(z),s)$ say that, in the situation s, x is on y and z is on the table. Notice that, in this language, the expressions $On(x,y)$ and $Ontable(x)$ are terms, and not formulas. These terms represent distinct fluents for different values of x and y:

$$\begin{gathered} On(x_1,y_1) = On(x_2,y_2) \supset x_1 = x_2 \wedge y_1 = y_2, \\ Ontable(x) = Ontable(y) \supset x = y, \\ On(x,y) \neq Ontable(z). \end{gathered} \tag{45}$$

The fluents $On(x,y)$ and $Ontable(x)$ play the role of a "coordinate frame" in the space of situations, in the sense that specific configurations of blocks can be described by combinations of values of these "frame fluents." For instance, the configuration in which all blocks are on the table can be characterized by saying that the value of each of the fluents $Ontable(x)$ is *true*.

Our goal is to describe the effects of moving blocks on the frame fluents. We want to be able to prove, in other words, that placing x on top of y leads to a situation in which $On(x,y)$ is true, $On(x,z)$ for $z \neq y$ is false, $Ontable(x)$ is false, and all fluents $On(u,v)$ and $Ontable(u)$ for $u \neq x$ have the same values as before. To this end, we need to describe first when a combination of values of the frame fluents is "consistent," that is, attained in some situation. Indeed, if there is no situation in which x is on y while u is on the table, then any action that makes $On(x,y)$ true will also make $Ontable(u)$ false. On the other hand, the conclusion that $On(x,z)$ for $z \neq y$ and $Ontable(x)$ become false after x is placed on top of y is based on the fact that there are no situations in which an object is located in two places at once.

Questions related to combinations of values of frame fluents can be conveniently formulated in terms of "states." A *state* is a function assigning values to the frame fluents. Every situation defines a certain state, but one state may correspond to many different situations. For instance, two different situations s_1 and s_2 may share the property that all blocks are located on the table; perhaps the difference between s_1 and s_2 is in the exact locations of blocks, or these situations may correspond to different instants of time. Given a state, the question now is: does it correspond to at least one situation?

We will assume as a default that the answer to this question is yes. This default will be be supplemented by the axioms saying that a block cannot be located in two places at once:

$$\begin{gathered} Holds(On(x,y_1),s) \wedge Holds(On(x,y_2),s) \supset y_1 = y_2, \\ Holds(On(x,y),s) \supset \neg Holds(Ontable(x),s). \end{gathered} \tag{46}$$

These axioms guarantee that some states are exceptional.[5]

The following notation will be used. The unary predicate *FrameFluent* singles out the fluents that are included in the frame. It is characterized by the postulates:

$$\begin{gathered} FrameFluent(On(x,y)), \\ FrameFluent(Ontable(x)). \end{gathered} \tag{47}$$

It will be circumscribed, to ensure that the frame contains no fluents other than these.

Since in our example all frame fluents are propositional, a state can be identified with the set of fluents to which it assigns the value *true*, and predicate variables can be used for expressing assertions about states. The symbols

[5]This approach classifies many states that are impossible to realize physically, and even difficult to imagine, as consistent. It is impossible to build "circular" configurations, when, for instance, x is located on top of y, y on top of z, and z on top of x (or, worse than that, x is located on its own top); since we do not include these limitations in the axiom set, it will follow that such states are consistent. This may seem counterintuitive. Recall, however, that we do not think of every situation as necessarily realized at some point in the actual course of events. There is nothing wrong then with admitting "ideal" situations, corresponding to some of the physically impossible states. The reason why the constraints (46) are included is that they help us give a more compact description of the effects of moving a block.

σ,σ_1,... will serve as unary predicate variables whose argument is a fluent variable. We write $\mathit{IsState}(\sigma)$ for

$$\forall f[\sigma(f) \supset \mathit{FrameFluent}(f)],$$

and $\mathit{State}[s]$ for

$$\lambda f[\mathit{FrameFluent}(f) \wedge \mathit{Holds}(f,s)].$$

Obviously, the formula

$$\forall s\ \mathit{IsState}(\mathit{State}[s])$$

is universally valid.

The formula

$$\mathit{IsState}(\sigma) \supset \exists s(\sigma = \mathit{State}[s])$$

asserts that every state corresponds to at least one situation. Alternatively, this can be expressed in the Skolemized form:

$$\mathit{IsState}(\sigma) \supset \sigma = \mathit{State}[\mathit{Sit}(\sigma)].$$

The function *Sit* maps a state into one of the corresponding situations. Since we want to assume this only as a default, the assumption that σ is not abnormal needs to be added to the antecedent of this formula. The symbol *Inconsistent* will be used as an abnormality predicate:

$$\mathit{IsState}(\sigma) \wedge \neg\mathit{Inconsistent}(\sigma) \supset \sigma = \mathit{State}[\mathit{Sit}(\sigma)]. \tag{48}$$

We expect that circumscribing *Inconsistent* will lead to the conclusion that the constraints (46) represent all existing dependencies between the values of frame fluents. One more assumption is needed in order to make this work:

$$\mathit{IsState}(\sigma_1) \wedge \mathit{IsState}(\sigma_2) \wedge \mathit{Sit}(\sigma_1) = \mathit{Sit}(\sigma_2) \supset \sigma_1 = \sigma_2. \tag{49}$$

This axiom guarantees that the cardinality of the universe of situations is sufficiently large.

The predicates *FrameFluent* and *Inconsistent* will be circumscribed in accordance with the following policy:

$$\begin{array}{l}\mathbf{circ}\ \mathit{FrameFluent}\ \mathbf{var}\ \mathit{Inconsistent},\\ \mathbf{circ}\ \mathit{Inconsistent}\ \mathbf{var}\ \mathit{Holds}.\end{array} \tag{50}$$

The first of these declarations expresses that *FrameFluent* is given a higher priority than *Inconsistent*. This is because we want to think of the extent of the frame as already fixed when we decide which states are consistent and which are not.

By BW_1 we denote the circumscriptive theory whose postulates are (45)–(50). The effect of the circumscriptions included in BW_1 can be determined using the methods of Section 3. It turns out that BW_1 is equivalent to the conjunction of its axioms with explicit definitions of the circumscribed predicates. The definition of *FrameFluent* is obtained by applying predicate completion to (47):

$$FrameFluent(f) \equiv \exists xy[f = On(x,y)] \lor \exists x[f = Ontable(x)], \tag{51}$$

The definition of *Inconsistent* is a longer formula:

$$\begin{aligned} Inconsistent(\sigma) \equiv IsState(\sigma) & \\ \land\neg[\forall x y_1 y_2[\sigma(On(x,y_1)) \land \sigma(On(x,y_2)) \supset y_1 = y_2] & \\ \land \forall xy[\sigma(On(x,y)) \supset \neg\sigma(Ontable(x))]]. & \end{aligned} \tag{52}$$

Of course, we could have used (51) as an axiom, instead of postulating (47) and circumscribing *FrameFluent*. The advantage of our method is that, should we decide to extend the formalization and add more fluents to the frame, we will be able to accomplish that simply by including additional axioms.

5.2 Actions

Now we are ready to introduce actions. We begin with the language of BW_1, and add to it variables for actions $a,a_1,\ldots$. We also add the binary function constant *Result*, whose arguments are an action and a situation and whose value is a situation, and the binary function constant *Move*, whose arguments are blocks and whose value is an action. Intuitively, the term $Move(x,y)$ represents the action of placing x on top of y. (Moving blocks onto the table can be easily added.) These terms represent distinct actions for different values of x and y:

$$Move(x_1,y_1) = Move(x_2,y_2) \supset x_1 = x_2 \land y_1 = y_2. \tag{53}$$

The effect of the action $Move(x,y)$ on the fluent $On(x,y)$ can be expressed by the formula

$$Holds(On(x,y), Result(Move(x,y),s)). \tag{54}$$

This formula is actually somewhat stronger than what we would like to postulate, because, in some situations, it may be impossible to place x on top of y. First, if some blocks are already on top of y, then there may be not enough room for x there. Second, the action may be impossible if there are blocks on top of x (think of a robot's arm trying to grasp x from above in order to move it). Third, in any situation, it is impossible to place x on top of x.

We will use the atomic formula $Possible(a,s)$ to express that it is possible to carry out the action a in the situation s. Instead of (54), we postulate:

$$Possible(Move(x,y),s) \supset Holds(On(x,y), Result(Move(x,y),s)). \tag{55}$$

A sufficient condition for the possibility of $Move(x,y)$ is given by the formula

$$\begin{aligned}\neg\exists z\, Holds(On(z,x),s) \wedge \neg\exists z\, Holds(On(z,y),s) \wedge x \neq y \\ \supset Possible(Move(x,y),s).\end{aligned} \tag{56}$$

The axiom (55) tells us how the action $Move(x,y)$ affects the location of x, but it does not say whether it affects the locations of any other blocks. This is an instance of the *frame problem*—the problem of describing which frame fluents do *not* change their values when an action is performed.

For any particular domain, the frame problem can be solved by including explicit "frame axioms." In our theory of moving blocks, the following two frame axioms would suffice:

$$\begin{aligned}&Possible(Move(x_1,y_1),s) \wedge x_1 \neq x_2\\ &\quad \supset [Holds(On(x_2,y_2), Result(Move(x_1,y_1),s)) \equiv Holds(On(x_2,y_2),s)],\\ &Possible(Move(x_1,y_1),s) \wedge x_1 \neq x_2\\ &\quad \supset [Holds(Ontable(x_2), Result(Move(x_1,y_1),s)) \equiv Holds(Ontable(x_2),s)].\end{aligned} \tag{57}$$

For a theory including fluents and actions of many different kinds, the number of necessary frame axioms may be quite large.

Circumscription allows us to replace all frame axioms by one axiom of a very general nature. This axiom formalizes the default principle known as the "commonsense law of inertia": *Normally, things remain as they are.*

The assertion that a leaves the value of all frame fluents unchanged can be expressed by the formula

$$FrameFluent(f) \wedge Possible(a,s) \supset [Holds(f, Result(a,s)) \equiv Holds(f,s)].$$

Since we want to assume this only as a default, the assumption that the triple (f,a,s) is not abnormal needs to be added to the antecedent. The symbol *Noninertial* will be used as a new abnormality predicate, and the commonsense law of inertia will be expressed by the formula

$$\begin{array}{r} FrameFluent(f) \wedge Possible(a,s) \wedge \neg Noninertial(f,a,s) \\ \supset [Holds(f, Result(a,s)) \equiv Holds(f,s)], \end{array} \tag{58}$$

By BW_2 we denote the circumscriptive theory whose axiom set is obtained by adding (53), (55), (56) and (58) to the axioms of BW_1, and whose circumscription policy is

$$\begin{array}{l} \textbf{circ}\ \mathit{FrameFluent}\ \textbf{var}\ \mathit{Inconsistent}, \mathit{Noninertial}, \\ \textbf{circ}\ \mathit{Inconsistent}\ \textbf{var}\ \mathit{Holds}, \mathit{Result}, \mathit{Possible}, \mathit{Noninertial}, \\ \textbf{circ}\ \mathit{Noninertial}\ \textbf{var}\ \mathit{Result}. \end{array} \tag{59}$$

This policy circumscribes *Noninertial* with a lower priority than the other two predicates, because we think of the structure of the space of situations as already determined when the effects of actions are described. Since the intention of the law of inertia is to characterize the result of executing a, the function *Result* is varied as *Noninertial* is circumscribed.

Using the methods of Section 3, we can show that the theory BW_2 is equivalent to the conjunction of its axioms with the formulas (51) and (52) and with an explicit definition of *Noninertial*. As expected, the frame axioms (57) are theorems of BW_2.

5.3 Preconditions

One other possible application of circumscription to the theory of action is related to the treatment of preconditions. The antecedent of the axiom (56) is the conjunction of 3 terms corresponding to the preconditions for the successful execution of $Move(x,y)$. For an action with many preconditions, the corresponding axiom would be very long. We may wish to express precondition information in a modular way, with each precondition described by a separate axiom. This is an aspect of what is known as the *qualification problem*.

This goal can be achieved in the following way. Instead of treating *Possible* as a primitive predicate, we can define it in terms of the primitive predicate *Precondition*:

$$Possible(a, s) \equiv \forall f(Precondition(f, a) \supset Holds(f, s)).$$

Then individual preconditions can be described by axioms like these:

$$\begin{gathered} Precondition(Clear(x), Move(x, y)), \\ Precondition(Clear(y), Move(x, y)), \\ Precondition(False, Move(x, x)). \end{gathered}$$

The expressions $Clear(x)$ and *False* represent fluents and are characterized by the axioms

$$\begin{gathered} Holds(Clear(x), s) \equiv \neg\exists y Holds(On(y, x), s), \\ \neg Holds(False, s). \end{gathered}$$

The predicate *Precondition* needs to be circumscribed, to ensure that actions have no preconditions other than those given by the axioms.

6 Well-Foundedness

In the definition of circumscription, the idea that the extent of P is "small" is expressed by imposing a minimality condition on models. Any model of A that is not minimal is rejected, because it can be "improved" by making P smaller. If the new model is not minimal either, then it can be further "improved."

It may happen, however, that this process of gradually "improving" a given model M of A does not lead to a model of the circumscription, because any structure obtained from M in this way can be "improved" further. If this is the case, then it is not clear that the identification of "small" with "minimal" is intuitively justified. We would like to avoid circumscriptions of this kind. The worst thing happens if *every* model of A can be "improved."

Then the circumscription has no models—it is not satisfiable.[6]

The well-foundedness condition, discussed in this section, makes this distinction between acceptable and unacceptable circumscriptions precise. Not surprisingly, this condition guarantees that the application of circumscription will not lead to an inconsistency. It is also used in the statements of some other theorems that establish the limits of what can be achieved by applying circumscription.

6.1 Definition and Examples

Consider a classical theory Γ whose language contains a predicate constant P and constants Z. We say that Γ is *well-founded* with respect to $(P; Z)$ if for every model M of Γ there exists a model M' of Γ minimal relative to $\leq^{P;Z}$ and such that $M' \leq^{P;Z} M$. If Γ is finite, then this condition can be stated as follows: There exists a model M' of CIRC $[\bigwedge_{A \in \Gamma} A; P; Z]$ such that $M' \leq^{P;Z} M$. When Z is empty, we say "well-founded with respect to P."

For example, the sentence $P(a)$ (more precisely, the theory $\{P(a)\}$) is well-founded (with respect to P), because any model of $P(a)$ can be transformed into a model of $\forall x(P(x) \equiv x = a)$ by (possibly) making the extent of P smaller. The sentence $\exists x P(x)$ is well-founded, because every nonempty set contains a singleton. Any sentence of the form

$$\forall x(F(x) \supset P(x)), \tag{60}$$

where $F(x)$ does not contain P, is well-founded, because any model of (60) can be transformed into a model of $\forall x(F(x) \equiv P(x))$ by making P smaller.

It is clear that Γ is well-founded with respect to $(P; Z)$ if, in each model of Γ, the extent of P is finite (in particular, if all models of Γ are finite).

On the other hand, consider the theory with 3 axioms:

$$\begin{gathered} \exists x[P(x) \wedge \forall y(P(y) \supset x \neq s(y))], \\ \forall x[P(x) \supset P(s(x))], \\ \forall xy[s(x) = s(y) \supset x = y]. \end{gathered} \tag{61}$$

[6] A sentence or theory is *satisfiable* if it has a model. For first order theories, this is the same as consistency; in view of the inevitable incompleteness of any formalization of second order logic, these two notions have to be distinguished in the case of a second order theory.

It is clear that this theory has models (for instance, take s to be the successor function on the set of natural numbers, and make P identically true). But it is not well-founded with respect to P. Any model M of this theory can serve as a counterexample. Indeed, let x^* be an element of M satisfying the first axiom as x, and let s^* be the function representing s. Consider the sequence

$$x^*, s^*(x^*), s^*(s^*(x^*)), \ldots.$$

It is clear that its members belong to $M[\![P]\!]$, and are distinct. We can construct a model M' which is $<^P M$ by defining

$$M'[\![P]\!] = \{s^*(x^*), s^*(s^*(x^*)), \ldots\}.$$

6.2 Syntactic Sufficient Conditions

We will consider now some syntactic conditions that imply well-foundedness. The most important fact of this kind is that every universal theory is well-founded, provided that the list of varied constants Z contains only predicate symbols. Recall that a formula is *universal* if it has the form $\forall x A$, where x is a tuple of object variables, and A is quantifier-free. If Z is empty, then this condition can be slightly generalized, and it is convenient to discuss this result first.

We say that a formula is *almost universal* with respect to a predicate constant P if it has the form $\forall x A$, where x is a tuple of object variables, and A does not contain positive occurrences of P in the scope of quantifiers.

Clearly, every universal formula is almost universal with respect to any P. If a formula is negative with respect to P, then it is almost universal with respect to P. Any formula of the form (60), where $F(x)$ does not contain P, is almost universal.

Any formula almost universal with respect to P is equivalent to a conjunction of formulas of the form

$$\forall x[B(x) \vee P(t_1(x)) \vee \cdots \vee P(t_k(x))], \tag{62}$$

where x is a tuple of object variables, $B(x)$ a formula negative relative to P, and $t_1(x), \ldots, t_k(x)$ are tuples of terms. Indeed, take an almost universal formula $\forall x A$ and write A as the conjunction of disjunctions of its propositional components and their negations; it is clear that each disjunction will have the form $B(x) \vee P(t_1(x)) \vee \cdots \vee P(t_k(x))$.

Proposition 9 *If every axiom of* Γ *is almost universal with respect to* P*, then* Γ *is well-founded with respect to* P*.*

Proof Without loss of generality, we can assume that every axiom has the form (62). Take a model M_0 of Γ, and let $\mathcal{M}$ be the set of models M of Γ such that $M \leq^P M_0$. Our goal is to show that $\mathcal{M}$ has a minimal element. By Zorn's lemma, it suffices to show that for every totally ordered subset U of $\mathcal{M}$ there exists a model M^* of Γ such that $M^* \leq^P M$ for every $M \in U$. We define M^* by the condition:

$$M^*[\![P]\!] = \bigcap_{M \in U} M[\![P]\!].$$

It remains to check that M^* is a model of Γ. Take any axiom (62) and any tuple x^* of elements of the universe, of the same length as x. Assume that each $P(t_i(x))$ is false in M^* for x^* as x. Then, for each $i = 1, \ldots, k$, there exists a model $M_i \in U$ such that $P(t_i(x))$ is false in M_i for x^* as x. Since U is totally ordered, there is a smallest among the models $M_1,\ldots,M_k$. Assume for definiteness that this is M_1. Then all formulas $P(t_i(x))$ are false in M_1 for x^* as x. But (62) is true in M_1; this is possible only if $B(x)$ is true in M_1 for x^* as x. Since all occurrences of P in $B(x)$ are negative, and $M^* \o^P M_1$, it follows that $B(x)$ is true in M^* for x^* as x. Hence (62) is true in M^*.

Proposition 10 *If* Γ *is universal and* Z *contains predicate symbols only, then* Γ *is well-founded with respect to* $(P; Z)$*.*

Proof Take any model M of Γ. Let Γ' be the set of all universal sentences that are entailed by Γ and do not contain symbols from Z. Γ' is a universal subtheory of Γ whose language does not contain the predicate symbols Z; by Proposition 9, it is well-founded with respect to P. Consequently, there is a model M' of Γ' minimal relative to $\leq^P$, such that $M' \leq^P M$. The proof will be completed if we show that M' can be extended by defining interpretations of the symbols Z so that the axioms of Γ will be satisfied. Extend the language of Γ' by names for the elements of M', and let $\Delta_{M'}$ be the diagram of M' (the set of the atomic sentences and negated atomic sentences in the extended language which are true in M'). First we want to show that $\Delta_{M'}$ is consistent with Γ. Assume that it is not. Then, by the compactness of first order logic, some finite set of sentences $\{D_1, D_2, \ldots\} \subset \Delta_M$ is inconsistent with Γ, so that

$$\Gamma \vdash \neg(D_1 \wedge D_2 \wedge \ldots).$$

Let $D'_1, D'_2, \ldots$ be obtained from $D_1, D_2, \ldots$ by substituting distinct variables for the names of the elements of M'. Since the names do not occur in Γ,

$$\Gamma \vdash \neg(D'_1 \wedge D'_2 \wedge \ldots).$$

Hence the universal closure of $\neg(D'_1 \wedge D'_2 \wedge \ldots)$ belongs to Γ'. Consequently, it is true in M'. But this is impossible, because each D_i belongs to $\Delta_{M'}$. This contradiction shows that $\Delta_{M'}$ is indeed consistent with Γ. Consider then a model M'' of $\Delta_{M'} \cup \Gamma$. Since M'' is a model of the diagram of M', it has a substructure M''' which is isomorphic to M', and in which, in addition, the predicate constants Z are interpreted. Since Γ is universal, M''' is a model of Γ.

The statement of Proposition 10 would be incorrect if Z were allowed to contain object or function symbols. Let Γ be the Skolemized form of the theory (61), that is, the result of replacing its first axiom by

$$P(c) \wedge \forall y (P(y) \supset c \neq s(y)),$$

where c is an object constant. It is easy to see that Γ is not well-founded with respect to $(P; c)$.

6.3 Satisfiability of Circumscription

Proposition 11 *If a sentence A is satisfiable and well-founded with respect to $(P; Z)$, then* $\mathrm{CIRC}[A; P; Z]$ *is satisfiable.*

Proof Let M be a model of A. Take a model M' of A minimal relative to $\leq^{P;Z}$ and such that $M' \leq^{P;Z} M$. By Proposition 1, M' is a model of $\mathrm{CIRC}[A; P; Z]$.

Using Propositions 9 and 10, we conclude:

Corollary 3 *If a sentence A is satisfiable and almost universal with respect to P, then* $\mathrm{CIRC}[A; P]$ *is satisfiable.*

Corollary 4 *If a universal sentence A is satisfiable, and Z contains predicate symbols only, then* $\mathrm{CIRC}[A;P;Z]$ *is satisfiable.*

Without the requirement that Z consist of predicate symbols, the statement of Corollary 4 would be incorrect. For instance, let Γ be the theory described at the end of Section 6.2. The circumscription $\mathrm{CIRC}\,[\,\bigwedge_{A\in\Gamma} A;P;c]$ is not satisfiable.

These results can be used to prove the existence of models for some circumscriptive theories with one policy declaration. But they are not directly applicable to theories with several declarations. Such a theory represents a conjunction of several circumscriptions, and the satisfiability of all these circumscriptions does not imply, of course, that the conjunction is satisfiable. In fact, a very simple circumscriptive theory may be unsatisfiable if its policy declarations assign "conflicting priorities" to two circumscribed predicates (see Section 4.2). For example, the propositional circumscriptive theory

$$\begin{gathered} P \vee Q, \\ \mathbf{circ}\ P\ \mathbf{var}\ Q, \\ \mathbf{circ}\ Q\ \mathbf{var}\ P \end{gathered}$$

is clearly unsatisfiable.

If, however, a theory with several declarations does not have such cycles, then Corollaries 3 and 4, in combination with Propositions 9 and 10, may allow us to prove the existence of a model. Consider, for instance, a circumscriptive theory of the form

$$\begin{gathered} A, \\ \mathbf{circ}\ P\ \mathbf{var}\ Q, R, \\ \mathbf{circ}\ Q\ \mathbf{var}\ S, \end{gathered} \tag{63}$$

where P,Q,R,S are predicate constants, and A is a satisfiable universal sentence. By Corollary 4, the circumscription

$$\mathrm{CIRC}[A;P;Q,R,S]$$

is satisfiable; take any model M of this circumscription. By Proposition 10, A is well-founded with respect to $(Q;S)$; consequently, there is a model M' of $\mathrm{CIRC}[A;Q;S]$ such that $M' \leq^{Q;S} M$. It is clear that M' satisfies the circumscription $\mathrm{CIRC}[A;P;Q,R,S]$ also; consequently, it is a model of (63).

6.4 Limitations of Circumscription

In all our applications of circumscription to formalizing defaults, we varied the predicates and functions that the defaults were supposed to characterize. For instance, in the examples of inheritance reasoning in the introduction and in Section 4.1, the predicates *Ontable* and *Heavy* were varied. In the Quaker-Republican example (Section 4.3), we varied *Pacifist*. In the circumscriptive theory solving the frame problem in the blocks world (Section 5.2), we varied *Noninertial*.

There was a good reason for doing so. Under the assumption of well-foundedness, a circumscription can provide no new information about the functions and predicates that are fixed. As far as the circumscribed predicate P is concerned, only "negative" information about it can be obtained—which is not surprising, because the idea of circumscription is to make the extent of P small. The following proposition makes this claim precise.

Proposition 12 *Let A be a sentence well-founded with respect to $(P; Z)$, and $F(P)$ a sentence which is positive with respect to P and contains no symbols from Z. $F(P)$ is entailed by* $\mathrm{CIRC}[A; P; Z]$ *if and only if it is entailed by A.*

For instance, circumscribing P can help us prove a negative literal $\neg P(a)$, but not an atom $P(a)$ or a positive disjunction $P(a) \vee P(b)$.

By taking *false* as $F(P)$, we can get Proposition 11 as a special case of Proposition 12.

As another example, notice that if F is a sentence built from atoms of the form $x = y$, where x and y are variables, then the application of circumscription will not help us conclude F, no matter how P and Z are selected (under the assumption of well-foundedness). For instance, the assertion that the cardinality of the universe is n $(n = 1, 2, \ldots)$ can be expressed by a sentence of this form. Circumscription cannot lead to any new conclusions about the size of the universe.

Proof of Proposition 12 The if part is obvious. Assume that $F(P)$ is entailed by $\mathrm{CIRC}[A; P; Z]$, and let M be a model of A. Take a model M' of A minimal relative to $\leq^{P;Z}$ and such that $M' \leq^{P;Z} M$. By Proposition 1, M' is a model of $\mathrm{CIRC}[A; P; Z]$, so that $F(P)$ is true in M'. Since $M' \leq^{P;Z} M$, the constant P is the only symbol occurring in F whose interpretation in M' can differ from its interpretation in M. Moreover, $M'[\![P]\!] \subset M[\![P]\!]$. Since $F(P)$ is positive with respect to P, it follows that F is true in M also.

6.5 A Characterization of Minimal Models

The following theorem characterizes the minimal models of a well-founded sentence:

Proposition 13 *Let A be a sentence well-founded with respect to $(P;Z)$. A model M of A is minimal relative to $\leq^{P;Z}$ iff it satisfies every sentence F such that*

(i) F does not contain symbols from Z, and

(ii) for every sentence G which is positive in P and does not contain symbols from Z, if G is entailed by $A \wedge F$, then G is entailed by A.

Proof To prove the implication right to left, observe that the second conjunctive term of CIRC$[A;P;Z]$ satisfies conditions (i) and (ii) as F. For (ii), this fact follows from Proposition 12. Let us prove the implication left to right. Consider any sentence $F(P)$ such that, for every sentence $G(P)$ positive in P, if $G(P)$ is entailed by $A(P,Z) \wedge F(P)$, then $G(P)$ is entailed by $A(P,Z)$. We need to prove that CIRC$[A;P;Z]$ entails $F(P)$. Take $G(P)$ to be

$$\exists pz(A(p,z) \wedge F(p) \wedge p \leq P). \tag{64}$$

Clearly, (64) is entailed by $A(P,Z) \wedge F(P)$; hence, by (ii), (64) is entailed by $A(P,Z)$. Assume CIRC$[A;P;Z]$. Then $A(P,Z)$, and consequently (64). Take p, z such that

$$A(p,z) \wedge F(p) \wedge p \leq P.$$

Using the second conjunctive term of CIRC$[A;P;Z]$, we derive from this formula $p = P$ and then $F(P)$.

7 Circumscribing Several Predicates

In the definition of CIRC$[A;P;Z]$ (Section 2.3), P is a predicate constant. We will extend this definition to the case when P is a tuple of predicate constants—first, assuming that the predicates are circumscribed "in parallel," and then assuming that they can be assigned different "priorities."

7.1 Parallel Circumscription

First, we need to generalize the notation $P = Q$, $P \leq Q$, $P < Q$, introduced in Section 2.1. Let P and Q be tuples $P_1, \ldots, P_n$ and $Q_1, \ldots, Q_n$ of predicate symbols such that, for each $i = 1, \ldots, n$, Q_i has the same arity as P_i. Then $P = Q$ stands for

$$P_1 = Q_1 \wedge \ldots \wedge P_n = Q_n,$$

$P \leq Q$ stands for

$$P_1 \leq Q_1 \wedge \ldots \wedge P_n \leq Q_n,$$

and $P < Q$ is again

$$(P \leq Q) \wedge \neg(P = Q).$$

Now $\text{CIRC}[A; P; Z]$ is defined as shorthand for the same expression

$$A(P, Z) \wedge \neg\exists pz[A(p, z) \wedge p < P]$$

that was used in the case when P was a single predicate; p stands now for a tuple of distinct predicate variables of appropriate arities.

The parallel circumscription of several predicates has a simple model-theoretic characterization, similar to the one presented in Section 2.5. When P is a tuple $P_1, \ldots, P_n$, the relation $M_1 \leq^{P;Z} M_2$ between structures M_1, M_2 is defined as before, except that the condition $M_1[\![P]\!] \subset M_2[\![P]\!]$ is replaced by $M_1[\![P_i]\!] \subset M_2[\![P_i]\!]$ for all $i = 1, \ldots, n$. It is clear then that Proposition 1 and its proof are applicable to parallel circumscription.

Circumscribing a tuple of predicates, like circumscribing one predicate, means selecting among the models of A those that cannot be "improved"; to "improve" a model means now to make the extent of at least one predicate from the tuple P smaller, while the others remain at least as small as they were.

Consider, for example, the parallel circumscriptions

$$\text{CIRC}[P_1(a) \wedge P_2(b); P_1, P_2] \tag{65}$$

and

$$\text{CIRC}[\forall x(P_1(x) \equiv P_2(x)); P_1, P_2]. \tag{66}$$

(In each formula, the comma between P_1 and P_2 shows that the predicates are circumscribed in parallel, so that the list of varied constants Z is empty;

a semicolon in this position would indicate that P_1 is circumscribed with P_2 varied.) The first circumscription is equivalent to

$$\forall x(P_1(x) \equiv x = a) \wedge \forall x(P_2(x) \equiv x = b);$$

the second is equivalent to

$$\forall x \neg P_1(x) \wedge \forall x \neg P_2(x).$$

Many of the definitions and results of Sections 3 and 6 can be easily extended to parallel circumscription.

There is an important special case when the parallel circumscription of a tuple of predicates is equivalent to the conjunction of the circumscriptions of its members:

Proposition 14 *If $A(P, Z)$ is positive relative to each member P_i of P, then the parallel circumscription* $\mathrm{CIRC}[A(P, Z); P; Z]$ *is equivalent to*

$$\bigwedge_i \mathrm{CIRC}[A(P, Z); P_i; Z]. \tag{67}$$

This fact can be used, for instance, to justify the result of the circumscription (65) given above. The example (66) shows, on the other hand, that the positivity assumption in Proposition 14 is essential: Each of the two circumscriptions $\mathrm{CIRC}[\forall x(P_1(x) \equiv P_2(x)); P_i]$ gives only $\forall x(P_1(x) \equiv P_2(x))$.

Proof of Proposition 14 The circumscription $\mathrm{CIRC}[A(P, Z); P; Z]$ can be equivalently rewritten as

$$A(P, Z) \wedge \neg \exists pz \left(A(p, z) \wedge (p \leq P) \wedge \bigvee_i \neg (p_i = P_i) \right),$$

or

$$A(P, Z) \wedge \bigwedge_i \neg \exists pz (A(p, z) \wedge (p \leq P) \wedge \neg (p_i = P_i)).$$

Thus it is sufficient to show that the second conjunctive term of the sentence $\mathrm{CIRC}[A(P, Z); P_i; Z]$ is equivalent to

$$\neg \exists pz (A(p, z) \wedge (p \leq P) \wedge \neg (p_i = P_i)).$$

To simplify notation, take $i = 1$; we need to prove that

$$\neg\exists p_1 z(A(p_1, P_2, \ldots, P_n, z) \wedge (p_1 < P_1)) \tag{68}$$

is equivalent to

$$\neg\exists p z(A(p, z) \wedge (p \leq P) \wedge \neg(p_1 = P_1)).$$

The last formula can be rewritten as

$$\neg\exists p_1 z(\exists p_2 \ldots p_n(A(p_1, p_2, \ldots, p_n, z) \wedge (p_2 \leq P_2) \wedge \ldots \wedge (p_n \leq P_n)) \wedge (p_1 < P_1)). \tag{69}$$

To show that (68) is equivalent to (69), observe that they only differ by the subformulas $A(p_1, P_2, \ldots, P_n, z)$ and

$$\exists p_2 \ldots p_n(A(p_1, p_2, \ldots, p_n, z) \wedge (p_2 \leq P_2) \wedge \ldots \wedge (p_n \leq P_n)),$$

and these subformulas are equivalent, because $A(P, Z)$ is positive relative to $P_2, \ldots, P_n$.

Parallel circumscription provides an alternative approach to formalizing the instances of default reasoning that involve more than one default. The last example from Section 4.2 was formalized there as a circumscriptive theory with two declarations:

$$\begin{array}{l} \textbf{circ } Ab_1 \textbf{ var } \textit{Ontable}, \\ \textbf{circ } Ab_2 \textbf{ var } \textit{Ontable}. \end{array}$$

This amounts to the use of the conjunction of two circumscriptions

$$\mathrm{CIRC}[A; Ab_1; \textit{Ontable}] \wedge \mathrm{CIRC}[A; Ab_2; \textit{Ontable}],$$

where A is the conjunction of the axioms. Instead, we could have used the parallel circumscription

$$\mathrm{CIRC}[A; Ab_1, Ab_2; \textit{Ontable}].$$

The inspection of the axioms (40) and (41) shows that all occurrences of Ab_1 and Ab_2 in them are positive. It follows from Proposition 14 that the two formalizations are equivalent.

The second circumscriptive theory from Section 4.1 has the declarations

$$\textbf{circ } Ab_1 \textbf{ var } \textit{Ontable},$$
$$\textbf{circ } Ab_2 \textbf{ var } \textit{Heavy}.$$

Here the lists of varied constants in the two circumscriptions are different. Generally, there seems to be no way to replace the conjunction of such circumscriptions by a single parallel circumscription. But in this particlular case exactly the same result would have been achieved with the circumscription policy

$$\textbf{circ } Ab_1 \textbf{ var } \textit{Ontable}, \textit{Heavy}$$
$$\textbf{circ } Ab_2 \textbf{ var } \textit{Ontable}, \textit{Heavy}.$$

It follows that we could have used a parallel circumscription of the form

$$\mathrm{CIRC}[A; Ab_1, Ab_2; \textit{Ontable}, \textit{Heavy}].$$

7.2 Prioritized Circumscription

The concept of prioritized circumscription provides a further generalization of the definition of circumscription from Section 2.1.

Let P be again a tuple of predicate constants. Assume that P is broken into disjoint segments $P^1, \ldots, P^k$. We want to express the idea that the predicates in P^1 should be minimized at a higher priority than the predicates in P^2, P^2 at a higher priority than P^3, etc. If Q is another tuple of predicate symbols, of the same length as P and of corresponding arities, we will write $P \preceq Q$ for

$$\bigwedge_{i=1}^{k} \left(\bigwedge_{j=1}^{i-1} (P^j = Q^j) \supset (P^i ø Q^i) \right),$$

where $Q^1, \ldots, Q^k$ are the parts of the tuple Q corresponding to the parts $P^1, \ldots, P^k$ of P. This formula represents a lexicographic combination of the k order relations $P^i ø Q^i$. If $k = 1$, it turns into $P \leq Q$; if $k = 2$, it becomes

$$(P^1 \leq Q^1) \wedge ((P^1 = Q^1) \supset (P^2 \leq Q^2)).$$

We write $P \prec Q$ for $(P \preceq Q) \wedge \neg(P = Q)$.

The *prioritized circumscription*

$$\mathrm{CIRC}[A; P^1 > \ldots > P^k; Z]$$

is the sentence

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

Parallel circumscription corresponds to $k = 1$. Here is an example of prioritized circumscription with $k = 2$:

$$\mathrm{CIRC}[\forall x(Q(x) \supset P_1(x) \vee P_2(x)); P_1 > P_2] \tag{70}$$

is equivalent to

$$\forall x \neg P_1(x) \wedge \forall x(P_2(x) \equiv Q(x)).$$

The comparison of the last example with the circumscriptive theory (44) from Section 4.2 shows that (70) is equivalent to the conjunction of two circumscriptions:

$$\mathrm{CIRC}[\forall x(Q(x) \supset P_1(x) \vee P_2(x)); P_1; P_2]$$

and

$$\mathrm{CIRC}[\forall x(Q(x) \supset P_1(x) \vee P_2(x)); P_2].$$

In fact, any prioritized circumscription can be reduced to a conjunction of parallel circumscriptions in this way:

Proposition 15 *The circumscription* $\mathrm{CIRC}[A; P^1 > \ldots > P^k; Z]$ *is equivalent to*

$$\bigwedge_{i=1}^{k} \mathrm{CIRC}[A; P^i; P^{i+1}, \ldots, P^k, Z].$$

Lemma 2 *The formula* $p \prec q$ *is propositionally equivalent to*

$$\bigvee_{i=1}^{k} \left(\bigwedge_{j=1}^{i-1} (p^j = q^j) \ \wedge \ (p^i < q^i) \right).$$

Proof It is clear that $\neg(p = q)$ is propositionally equivalent to

$$\bigvee_{i=1}^{k} \neg(p^i = q^i),$$

and consequently to

$$\bigvee_{i=1}^{k}\left(\neg(p^i = q^i) \ \wedge \ \bigwedge_{j=1}^{i-1}(p^j = q^j)\right).$$

It follows that $p \prec q$ can be written as

$$\bigvee_{i=1}^{k}\left(\neg(p^i = q^i) \ \wedge \ \bigwedge_{j=1}^{i-1}(p^j = q^j) \ \wedge \ (p \preceq q)\right). \tag{71}$$

The subformula $p \preceq q$ is the conjunction of the terms

$$\bigwedge_{j=1}^{l-1}(p^j = q^j) \ \supset \ (p^l \leq q^l)$$

for all $l = 1, \ldots, k$. In the presence of $\neg(p^i = q^i)$ and

$$\bigwedge_{j=1}^{i-1}(p^j = q^j), \tag{72}$$

the antecedent of this conditional can be replaced by *true* if $l \leq i$, and by *false* if $l > i$; consequently, $p \preceq q$ in (71) can be replaced by

$$\bigwedge_{l=1}^{i}(p^l \leq q^l).$$

Furthermore, all terms of this conjunction except $p^i \leq q^i$ are propositional consequences of (72), so that (71) can be replaced by

$$\bigvee_{i=1}^{k}\left(\neg(p^i = q^i) \ \wedge \ \bigwedge_{j=1}^{i-1}(p^j = q^j) \ \wedge \ (p^i \leq q^i)\right).$$

It remains to replace $\neg(p^i = q^i) \wedge (p^i \leq q^i)$ by $p^i < q^i$.

Proof of Proposition 15 The lemma shows that the second conjunctive term of CIRC$[A; P^1 > \ldots > P^k; Z]$ can be written as

$$\neg\exists pz\left[A(p,z) \wedge \bigvee_{i=1}^{k}\left(\bigwedge_{j=1}^{i-1}(p^j = P^j) \ \wedge \ (p^i < P^i)\right)\right],$$

or

$$\bigwedge_{i=1}^{k} \neg \exists p z \left[A(p,z) \;\wedge\; \bigwedge_{j=1}^{i-1} (p^j = P^j) \;\wedge\; (p^i < P^i) \right].$$

For every i, the i-th term of this conjunction is equivalent to

$$\neg \exists p^i p^{i+1} \ldots p^k z [A(P^1, \ldots, P^{i-1}, p^i, p^{i+1}, \ldots, p^k, z) \wedge (p^i < P^i)],$$

which is the second conjunctive term of the circumscription

$$\mathrm{CIRC}[A; P^i; P^{i+1}, \ldots, P^k, Z].$$

8 Bibliographical and Historical Remarks

The word "circumscription" was introduced by McCarthy in [1977]. He wrote:

> There is an intuition that not all human reasoning can be translated into deduction in some formal system of mathematical logic, and therefore mathematical logic should be rejected as a formalism for expressing what a robot should know about the world. The intuition in itself doesn't carry a convincing idea of what is lacking and how it might be supplied.
>
> We can confirm part of the intuition by describing a previously unformalized mode of reasoning called *circumscription*, which we can show does not correspond to deduction in a mathematical system. The conclusions it yields are just conjectures and sometimes even introduce inconsistency. We will argue that humans often use circumscription, and robots must too. The second part of the intuition—the rejection of mathematical logic—is not confirmed; the new mode of reasoning is best understood and used within a mathematical logical framework and co-ordinates well with mathematical logical deduction. We think *circumscription* accounts for some of the successes and some of the errors of human reasoning.

> The intuitive idea of *circumscription* is as follows: We know some objects in a given class and we have some ways of generating more. We jump to the conclusion that this gives all the objects in the class. Thus we *circumscribe* the class to the objects we know how to generate.

The technical implementation of this idea proposed in that paper is called now "domain" circumscription—to distinguish it from the concept of "predicate" circumscription, which is subject of this chapter. The relation between the two forms of circumscription is discussed in Section 5 of [McCarthy, 1980] and in Section 7.4 of [Etherington, 1988].

The abnormality predicate was introduced in [McCarthy, 1986]. It is applied there to "aspects" of objects, rather than to objects themselves, because an object can be normal in some aspects (relative to some defaults), and abnormal in others. This is similar to the use of several abnormality predicates in Section 4.2.

The basic case of circumscription is defined by McCarthy [1980] using a first order schema, instead of second order quantifiers. Properties of this "first order circumscription" are studied in [Perlis and Minker, 1986]. Another way to avoid using predicate variables in the theory of circumscription is to replace them by variables for sets [Perlis, 1987].

The idea of varying some predicates and functions is mentioned in [McCarthy, 1980] in passing, as well as the use of second order formulas. These two ideas were explored in more detail in [McCarthy, 1986]. The definition of "formula" circumscription introduced there shows, moreover, how one can circumscribe a predicate expressed by a formula, and not merely a predicate constant. Circumscribing equality was investigated by Etherington, Mercer and Reiter [1985].

The exposition in Section 2.3 follows [Lifschitz, 1985b]. The need to circumscribe higher order predicates was demonstrated by Baker [1989], in connection with the applications discussed here in Section 5.1.

The model-theoretic view of circumscription presented in Section 2.5 was developed in [McCarthy, 1980] for the basic case and extended to the general case in [Lifschitz, 1985b]. A further development of this idea can be found in Shoham's concept of "preferential entailment" [Shoham, 1987]. Regarding the reference to infinite axiom sets in Section 2.5 we should say that such a set cannot be stored, of course, in the memory of a computer; but it would be a mistake to conclude that infinite axiom sets are out of place in applications of logic to AI. Besides being useful as theoretical tools, they can be also implemented as knowledge bases, if all axioms except for a finite number are represented by means of procedural attachment (see [Genesereth and Nilsson, 1987], Section 4.4).

The idea of representing circumscription policies by metamathematical declarations belongs to McCarthy [1986]. The view that a circumscriptive theory corresponds to a conjunction of several circumscriptions, applied to the same formula, was developed in [Lifschitz, 1989b].

Predicate completion was first introduced in the context of logic programming by Clark [1978], and its relation to circumscription was investigated by Reiter [1982]. The methods presented in Sections 3.1 and 3.2 were developed in [Lifschitz, 1985b]. Propositions 4 and 5 are from [Lifschitz, 1987b]. Proposition 6, due to Rabinov [1989], is a generalization of the "separable" case of circumscription [Lifschitz, 1985b]. Two cases when circumscription is expressible in the first order language that are not discussed here are existential formulas [Kolaitis and Papadimitriou, 1988] and monadic formulas ([McCarthy, 1986], Section 13). Proposition 7 is from [Lifschitz, 1991].

Kolaitis and Papadimitriou [1988] study an important subclass of definite first order formulas from the point of view of the expressibility of their circumscriptions in the first order language, which includes, for instance, Examples 1, 3, 5, 7, 8, 11 and 12. They show that the circumscription CIRC[A; P] for a sentence A from this subclass is equivalent to a first order formula if and only if A is "bounded," in the sense that there is a uniform bound on the number of times the implications from A need to be used for establishing that a given tuple of arguments satisfies P, for all interpretations of other predicate symbols. For instance, the sentence defining transitive closure (Example 12) is not bounded, because the number of steps that may be needed for establishing that a given pair satisfies the transitive closure of Q depends on the choice of Q. The other examples mentioned above are bounded. Since the boundedness property is undecidable [Gaifman *et al.*, 1987], it follows that it is an undecidable problem to determine whether or not a circumscription of this form is expressible without second order quantifiers.

One of the reasons why we may wish to represent a circumscription by a first order formula is that this makes it possible, in principle, to use a resolution theorem prover for verifying that a given first order formula is a consequence of the circumscription. But it should be observed that this approach to the automation of circumscription is not the only one possible. In fact, it may be unattractive, in view of the fact that it requires preprocessing the circumscription in a manner that is not related in any way to the goal formula. “Backward chaining” approaches to implementing special cases of circumscription, based on the ideas of logic programming and on Proposition 13 from Section 6.5, are discussed in [Gelfond and Lifschitz, 1989], [Ginsberg, 1989], [Baker and Ginsberg, 1989], [Przymusinski, 1989], [Inoue and Helft, 1990].

The discussion of inheritance reasoning in Section 4 generally follows [McCarthy, 1986]; Brewka [1987] describes a similar approach to representing inheritance by circumscription in a general form, as a translation from a frame-based language. The method of expressing priorities used in Section 4.2 was introduced in [Lifschitz, 1989b]. The example with conflicting reports is taken from the same paper. The Quaker-Republican example is due to Reiter and Criscuolo [1981].

The standard reference for the situation calculus is [McCarthy and Hayes, 1969]. The first attempt to apply circumscription to the frame problem [McCarthy, 1986] was unsuccessful [Hanks and McDermott, 1987]. The formulation presented in Section 5 follows [Baker, 1991] and [Lifschitz, 1991]. The treatment of preconditions outlined at the end of Section 5.2 is based on [Lifschitz, 1987a].

The concept of well-foundedness, for the case when Z is empty, was invented by Etherington, Mercer and Reiter [1985], as well as the example (61). They observe also that well-foundedness implies satisfiability; furthermore, they point out that universal theories are well-founded (in the absence of varied predicates), and that this fact can be proved by the method due to Bossu and Siegel [1985]. Proposition 9 and Corollary 3 are from [Lifschitz, 1986]). The generalization of well-foundedness to the case of varied constants was proposed independently by Etherington [1988] and Lifschitz [1986]. Proposition 10 belongs to Etherington [1988]; the special case when Γ is a single formula was found independently by Lifschitz [1986]. Corollary 4 was also proved independently by Etherington [1988] and Lifschitz [1986]. Proposition 12 is a generalization of results from [Etherington *et al.*, 1985], [Etherington, 1988] and [Gelfond *et al.*, 1989]. Proposition 13 generalizes a theorem from [Gelfond *et al.*, 1989].

The idea of parallel circumscription goes back to [McCarthy, 1980]. Priorities were first introduced in [McCarthy, 1986], where they are applied to "aspects," rather than predicates. Our exposition follows [Lifschitz, 1985b]. Proposition 14 is from [Lifschitz, 1987b], and Proposition 15 is from [Lifschitz, 1985b]. One other equivalent transformation of parallel circumscription is proposed by de Kleer and Konolige [1989].

Two modifications of the definition of circumscription not discussed in this chapter are "pointwise circumscription" [Lifschitz, 1987b] and "autocircumscription" [Perlis, 1988]. The latter is quite different from predicate (or domain) circumscription, because it expresses the idea of introspection, rather than minimization. "Introspective circumscription" [Lifschitz, 1989a] is a modification of autocircumscription which includes predicate circumscription as a special case.

Much work has been done on the relationship between circumscription and other nonmonotonic formalisms; some of it is reviewed in other chapters of this volume. Under some conditions, circumscription can be used to express the closed world assumption [Lifschitz, 1985a] and its generalizations [Gelfond and Przymusinska, 1986], as well as the unique names assumption ([McCarthy, 1986], Section 6; see also [Etherington *et al.*, 1985]). On the relation of circumscription to logic programming, see [Lifschitz, 1988], [Przymusinski, 1988], [Gelfond *et al.*, 1989], [Gelfond and Lifschitz, 1989]. Circumscription is related to autoepistemic logic in [Konolige, 1989], and to default logic in [Etherington, 1987]. The modification of default logic introduced in [Lifschitz, 1990] is closer in some ways to circumscription than Reiter's original definition.

Acknowledgements

G. N. Kartha, Norman McCain, John McCarthy, Donald Perlis and Yoav Shoham read the manuscript of this chapter at different stages of its development. John Horty has sent me his comments on Section 4. I am grateful to all of them for criticism and advice. This work was supported in part by NSF grant IRI-9101078.

References

[Baker and Ginsberg, 1989] Andrew Baker and Matthew Ginsberg. A theorem prover for prioritized circumscription. In *Proc. of IJCAI-89*, pages 463–467, 1989.

[Baker, 1989] Andrew Baker. A simple solution to the Yale Shooting Problem. In Ronald Brachman, Hector Levesque, and Raymond Reiter, editors, *Proc. of the First Int'l Conf. on Principles of Knowledge Representation and Reasoning*, pages 11–20, 1989.

[Baker, 1991] Andrew Baker. Nonmonotonic reasoning in the framework of situation calculus. *Artificial Intelligence*, 49:5–23, 1991.

[Bossu and Siegel, 1985] Genevieve Bossu and Pierre Siegel. Saturation, nonmonotonic reasoning, and the closed-world assumption. *Artificial Intelligence*, 25:13–63, 1985.

[Brewka, 1987] Gerhard Brewka. The logic of inheritance in frame systems. In *Proc. IJCAI-87*, pages 483–488, 1987.

[Clark, 1978] Keith Clark. Negation as failure. In Herve Gallaire and Jack Minker, editors, *Logic and Data Bases*, pages 293–322. Plenum Press, New York, 1978.

[de Kleer and Konolige, 1989] Johan de Kleer and Kurt Konolige. Eliminating the fixed predicates from a circumscription. *Artificial Intelligence*, 39(3):391–398, 1989.

[Etherington *et al.*, 1985] David Etherington, Robert Mercer, and Raymond Reiter. On the adequacy of predicate circumscription for closed-world reasoning. *Computational Intelligence*, 1(1), 1985.

[Etherington, 1987] David Etherington. Relating default logic and circumscription. In *Proc. IJCAI-87*, pages 489–494, 1987.

[Etherington, 1988] David Etherington. *Reasoning with Incomplete Information.* Pitman Publishing, London, 1988.

[Gaifman *et al.*, 1987] Haim Gaifman, H. Mairson, Yehoshua Sagiv, and Moshe Vardi. Undecidable optimization problems for database logic programs. In *Proc. 2nd IEEE Symposium on Logic in Computer Science*, pages 106–115, 1987.

[Gelfond and Lifschitz, 1989] Michael Gelfond and Vladimir Lifschitz. Compiling circumscriptive theories into logic programs. In Michael Reinfrank, Johan de Kleer, Matthew Ginsberg, and Erik Sandewall, editors, *Non-Monotonic Reasoning: 2nd International Workshop (Lecture Notes in Artificial Intelligence 346)*, pages 74–99. Springer-Verlag, 1989.

[Gelfond and Przymusinska, 1986] Michael Gelfond and Halina Przymusinska. Negation as failure: Careful closure procedure. *Artificial Intelligence*, 30(3):273–287, 1986.

[Gelfond *et al.*, 1989] Michael Gelfond, Halina Przymusinska, and Teodor Przymusinski. On the relationship between circumscription and negation as failure. *Artificial Intelligence*, 38(1):75–94, 1989.

[Genesereth and Nilsson, 1987] Michael R. Genesereth and Nils J. Nilsson. *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann, San Mateo, CA, 1987.

[Ginsberg, 1989] Matthew Ginsberg. A circumscriptive theorem prover. *Artificial Intelligence*, 39:209–230, 1989.

[Hanks and McDermott, 1987] Steve Hanks and Drew McDermott. Nonmonotonic logic and temporal projection. *Artificial Intelligence*, 33(3):379–412, 1987.

[Inoue and Helft, 1990] Katsumi Inoue and Nicolas Helft. On theorem provers for circumscription. In *Proc. of CSCSI-90*, pages 212–219, 1990.

[Kolaitis and Papadimitriou, 1988] Phokion Kolaitis and Christos Papadimitriou. Some computational aspects of circumscription. In *Proc. of AAAI-88*, volume 2, pages 465–469, 1988.

[Konolige, 1989] Kurt Konolige. On the relation between autoepistemic logic and circumscription. In *Proc. of IJCAI-89*, pages 1213–1218, 1989.

[Lifschitz, 1985a] Vladimir Lifschitz. Closed-world data bases and circumscription. *Artificial Intelligence*, 27, 1985.

[Lifschitz, 1985b] Vladimir Lifschitz. Computing circumscription. In *Proc. of IJCAI-85*, pages 121–127, 1985.

[Lifschitz, 1986] Vladimir Lifschitz. On the satisfiability of circumscription. *Artificial Intelligence*, 28, 1986.

[Lifschitz, 1987a] Vladimir Lifschitz. Formal theories of action (preliminary report). In *Proc. of IJCAI-87*, pages 966–972, 1987.

[Lifschitz, 1987b] Vladimir Lifschitz. Pointwise circumscription. In Matthew Ginsberg, editor, *Readings in nonmonotonic reasoning*, pages 179–193. Morgan Kaufmann, San Mateo, CA, 1987.

[Lifschitz, 1988] Vladimir Lifschitz. On the declarative semantics of logic programs with negation. In Jack Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pages 177–192. Morgan Kaufmann, San Mateo, CA, 1988.

[Lifschitz, 1989a] Vladimir Lifschitz. Between circumscription and autoepistemic logic. In Ronald Brachman, Hector Levesque, and Raymond Reiter, editors, *Proc. of the First Int'l Conf. on Principles of Knowledge Representation and Reasoning*, pages 235–244, 1989.

[Lifschitz, 1989b] Vladimir Lifschitz. Circumscriptive theories: a logic-based framework for knowledge representation. In Richmond Thomason, editor, *Philosophical Logic and Artificial Intelligence*, pages 109–160. Kluwer, 1989.

[Lifschitz, 1990] Vladimir Lifschitz. On open defaults. In John Lloyd, editor, *Computational Logic: Symposium Proceedings*, pages 80–95. Springer, 1990.

[Lifschitz, 1991] Vladimir Lifschitz. Towards a metatheory of action. In James Allen, Richard Fikes, and Erik Sandewall, editors, *Proc. of the Second Int'l Conf. on Principles of Knowledge Representation and Reasoning*, 1991.

[McCarthy and Hayes, 1969] John McCarthy and Patrick Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 4, pages 463–502. Edinburgh University Press, Edinburgh, 1969. Reproduced in [McCarthy, 1990].

[McCarthy, 1977] John McCarthy. Epistemological problems of artificial intelligence. In *Proc. of IJCAI-77*, pages 1038–1044, 1977. Reproduced in [McCarthy, 1990].

[McCarthy, 1980] John McCarthy. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1, 2):27–39,171–172, 1980. Reproduced in [McCarthy, 1990].

[McCarthy, 1986] John McCarthy. Applications of circumscription to formalizing common sense knowledge. *Artificial Intelligence*, 26(3):89–116, 1986. Reproduced in [McCarthy, 1990].

[McCarthy, 1990] John McCarthy. *Formalizing common sense: papers by John McCarthy*. Ablex, Norwood, NJ, 1990.

[Perlis and Minker, 1986] Donald Perlis and Jack Minker. Completeness results for circumscription. *Artificial Intelligence*, 28:29–42, 1986.

[Perlis, 1987] Donald Perlis. Circumscribing with sets. *Artificial Intelligence*, 31:201–211, 1987.

[Perlis, 1988] Donald Perlis. Autocircumscription. *Artificial Intelligence*, 36:223–236, 1988.

[Przymusinski, 1988] Teodor Przymusinski. On the relationship between logic programming and non-monotonic reasoning. In *Proc. AAAI-88*, pages 444–448, 1988.

[Przymusinski, 1989] Teodor Przymusinski. An algorithm to compute circumscription. *Artificial Intelligence*, 38(1):49–73, 1989.

[Rabinov, 1989] Arkady Rabinov. A generalization of collapsible cases of circumscription. *Artificial Intelligence*, 38(1):111–117, 1989.

[Reiter and Criscuolo, 1981] Raymond Reiter and Giovanni Criscuolo. On interacting defaults. In *Proc. of IJCAI-81*, pages 270–276, 1981.

[Reiter, 1982] Raymond Reiter. Circumscription implies predicate completion (sometimes). In *Proc. of IJCAI-82*, pages 418–420, 1982.

[Shoham, 1987] Yoav Shoham. Nonmonotonic logic: meaning and utility. In *Proc. of AAAI-87*, pages 388–393, 1987.