General Domain Circumscription and its First-Order Reduction

Patrick Doherty and Witold Łukaszewicz and Andrzej Szałas

Abstract

We first define *general domain circumscription* (GDC) and provide it with a semantics. GDC subsumes existing domain circumscription proposals in that it allows varying of arbitrary predicates, functions, or constants, to maximize the minimization of the domain of a theory. We then show that for the class of *semi-universal* theories without function symbols, that the domain circumscription of such theories can be constructively reduced to logically equivalent first-order theories by using an extension of the DLS algorithm, previously proposed by the authors for reducing second-order formulas. We also show that for a certain class of domain circumscribed theories, that any arbitrary second-order circumscription policy applied to these theories is guaranteed to be reducible to a logically equivalent first-order theory. In the case of semi-universal theories with functions and arbitrary theories which are not *separated*, we provide additional results, which although not guaranteed to provide reductions in all cases, do provide reductions in some cases. These results are based on the use of fixpoint reductions.

General Domain Circumscription and its First-Order Reduction

Patrick Doherty and Witold Lukaszewicz and Andrzej Szałas

Abstract

We first define *general domain circumscription* (GDC) and provide it with a semantics. GDC subsumes existing domain circumscription proposals in that it allows varying of arbitrary predicates, functions, or constants, to maximize the minimization of the domain of a theory. We then show that for the class of *semi-universal* theories without function symbols, that the domain circumscription of such theories can be constructively reduced to logically equivalent first-order theories by using an extension of the DLS algorithm, previously proposed by the authors for reducing second-order formulas. We also show that for a certain class of domain circumscribed theories, that any arbitrary second-order circumscription policy applied to these theories is guaranteed to be reducible to a logically equivalent first-order theory. In the case of semi-universal theories with functions and arbitrary theories which are not *separated*, we provide additional results, which although not guaranteed to provide reductions in all cases, do provide reductions in some cases. These results are based on the use of fixpoint reductions.

1 Introduction

In many common-sense reasoning scenarios, we are given a theory T specifying general laws and domain specific facts about the set of phenomena under investigation. In addition, one provides a number of closure axioms circumscribing the domain of individuals and certain properties and relations among individuals. The closure machinery normally involves the use of non-monotonic rules of inference, or in the case of circumscription, a second-order axiom. In order for a circumscribed theory to be useful, it is necessary to find a means of computing inferences from the circumscribed theory in an efficient manner. Unfortunately, the second-order nature of circumscription axioms creates an obstacle towards doing this.

In previous work [3], we proposed the use of an algorithm (DLS) which when given a second-order formula as input would terminate with failure, or output a logically equivalent first-order formula. Since circumscription axioms are simply second-order formulas, we showed that the DLS algorithm could be used as a basis for efficiently computing inferences for a broad class of circumscribed theories by first reducing the circumscription axiom to a logically equivalent first-order formula and then using classical theorem proving techniques to compute inferences from the original theory augmented with the output of the algorithm. In [4], the DLS algorithm was generalized using a reduction theorem from [10]. It was shown that a broad subset of second-order logic can be reduced into fixpoint logic. Moreover, a class of fixpoint formulas was characterized which can be reduced into their first-order equivalents.

In this paper, we extend the previous work in three ways:

1. We define a general form of domain circumscription which subsumes existing domain circumscription proposals in the literature ([9], [2], [5], [6], and [7]). We call the generalization *general domain circumscription* (GDC). GDC distinguishes itself from other proposals in the following manner. When circumscribing the domain of a theory T, it is permitted to vary

arbitrary predicates, functions, or constants, to maximize the minimization of the domain of individuals.

2. We characterize a class of theories which when circumscribed using GDC are guaranteed to be reducible to equivalent first-order theories which are constructively generated as output from extended versions of the original DLS algorithm. Included in this class are theories for which both McCarthy's original domain circumscription [9] and Hintikka's mini-consequence [6] are always reducible to first-order logic.

3. We characterize a class of theories which, when first circumscribed using GDC and then circumscribed using an arbitrary circumscription policy, are guaranteed to be reducible to equivalent first-order theories which are constructively generated as output from the extended versions of the original DLS algorithm mentioned in the previous item.

We approach the characterization and reduction problems in the following manner.

- Given a theory T, we show that if the domain closure axiom is entailed by the domain circumscribed theory $Circ_D(T)$, then $Circ_D(T)$ is always reducible to a logically equivalent first-order theory.

- We then characterize a class of theories where the domain closure axiom is not only entailed by the domain circumscribed theory, but can be automatically generated and used in the extended algorithm to reduce theories from this class to their corresponding first-order equivalents.

- Given a theory in the class characterized above and an arbitrary circumscription policy applied to that theory, we show that the extended version of the DLS algorithm will always generate a first-order theory logically equivalent to the second-order circumscribed theory.

The key to the approach is determining when a domain circumscribed theory $Circ_D(T)$, entails it's domain closure axiom. Semantically, a possible answer is when the cardinalities of all minimal models of the domain circumscribed theory have the same finite upper bound. Syntactically, we can characterize two classes of theories that provide such constraints when minimized:

1. Universal theories without function symbols, where the general domain circumscription policy can include arbitrary constants and predicates that vary.

2. Semi-universal theories without function symbols, where the general domain circumscription policy can include arbitrary constants and predicates that vary.

The class of semi-universal theories is a broad class of theories much more expressive than universal theories which have previously been studied in the context of restricted forms of domain circumscription. In the case of universal and semi-universal theories with function symbols, where the general domain circumscription policy can include arbitrary constants, predicates and functions that vary, reducible classes of theories are difficult to characterize. In this case, we provide additional results which guarantee reduction non-constructively and additional methods which, although not guaranteed to provide first-order reductions in all cases, do provide reductions in some cases.

The paper is organized as follows. Section 2 consists of preliminary definitions and notation. In Section 3, general domain circumscription is introduced together with it model-preferential semantics. In Section 4, the original DLS algorithm is briefly described together with two limitations associated with the basic algorithm. In Section 5, two generalizations of the basic DLS algorithm are described which deal with the limitations previously described. In Section 6, reducibility results concerning different specializations of general domain circumscription are presented together

with a number of concrete examples. In Section 7, we consider the potential for reducing a larger class of arbitrarily circumscribed theories which are first circumscribed using general domain circumscription.

2 Preliminaries

In this paper, the term *theory* always refers to a finite set of sentences of first-order logic. Since each set is equivalent to the conjunction of its members, a theory may be always viewed as a single first-order sentence. In the sequel, we shall never distinguish between a theory T and the sentence being the conjunction of all members of T. Unless stated otherwise, the term *function symbol* refers to a function symbol of arity n, where $n > 0$.

2.1 Notation

An n-ary *predicate expression* is any expression of the form $\lambda \overline{x}.\ A(\overline{x})$, where $\overline{x}$ is a tuple of n individual variables and $A(\overline{x})$ is any formula of first-order classical logic. If U is an n-ary predicate expression of the form $\lambda \overline{x}.\ A(\overline{x})$ and $\overline{\alpha}$ is a tuple of n terms, then $U(\overline{\alpha})$ stands for $A(\overline{\alpha})$. As usual, a predicate constant P is identified with the predicate expression $\lambda \overline{x}.\ P(\overline{x})$. Similarly, a predicate variable Φ is identified with the predicate expression $\lambda \overline{x}.\ \Phi(\overline{x})$.

An n-ary *function expression* is any expression of the form $\lambda \overline{x}.\ \tau(\overline{x})$, where $\overline{x}$ is a tuple of n individual variables and $\tau(\overline{x})$ is any term of first-order classical logic. If u is an n-ary function expression of the form $\lambda \overline{x}.\ \tau(\overline{x})$ and $\overline{t}$ is a tuple of n terms, then $u(\overline{t})$ stands for $\tau(\overline{t})$. An n-ary $(n \geq 0)$ function constant f is identified with the function expression $\lambda \overline{x}.\ f(\overline{x})$. An n-ary $(n \geq 0)$ function variable ϕ is identified with the function expression $\lambda \overline{x}.\ \phi(\overline{x})$. Note that $0 - ary$ function variables are simply individual variables.

Let $\overline{U} = (U_1, \ldots, U_n)$ and $\overline{V} = (V_1, \ldots, V_n)$ (resp. $\overline{u} = (u_1, \ldots, u_n)$ and $\overline{v} = (v_1, \ldots, v_n)$) be tuples of predicate (resp. function) expressions. $\overline{U}$ and $\overline{V}$ (resp. $\overline{u}$ and $\overline{v}$) are said to be *similar* iff, for each i $(1 \leq i \leq n)$, U_i and V_i (resp. u_i and v_i) are predicate (resp. function) expressions of the same arity.

Truth values *true* and *false* are denoted by $\top$ and $\bot$, respectively.

If U and V are predicate expressions of the same arity, then $U \leq V$ stands for $\forall \overline{x}\ U(\overline{x}) \supset V(\overline{x})$. If $\overline{U} = (U_1, \ldots, U_n)$ and $\overline{V} = (V_1, \ldots, V_n)$ are similar tuples of predicate expressions, i.e. U_i and V_i are of the same arity, $1 \leq i \leq n$, then $\overline{U} \leq \overline{V}$ is an abbreviation for $\bigwedge_{i=1}^{n}[U_i \leq V_i]$.

If A is a formula, $\bar{\sigma} = (\sigma_1, \ldots, \sigma_n)$ and $\bar{\delta} = (\delta_1, \ldots, \delta_n)$ are tuples of any expressions, then $A(\bar{\sigma} \leftarrow \bar{\delta})$ stands for the formula obtained from A by simultaneously replacing *each* occurrence of σ_i by δ_i $(1 \leq i \leq n)$. For any tuple $\bar{x} = (x_1, \ldots x_n)$ of individual variables and any tuple $\bar{t} = (t_1, \ldots t_n)$ of terms, we write $\bar{x} = \bar{t}$ to denote the formula $x_1 = t_1 \wedge \cdots \wedge x_n = t_n$. We write $\bar{x} \neq \bar{t}$ as an abbreviation for $\neg(\bar{x} = \bar{t})$.

2.2 Definitions

Definition 2.1 A theory T is said to be *existential* (*universal*) iff all of its axioms are of the form $\exists \overline{x}\ T_1$ (resp. $\forall \overline{x}\ T_1$), where T_1 is quantifier free. ∎

Definition 2.2 A theory is called *semi-universal* if its axioms do not contain existential quantifiers in the scope of universal quantifiers. ∎

Definition 2.3 Let T be a theory without function symbols of positive arity and suppose that, for $n > 0$, $c_1, \ldots c_n$ are all the individual constants occurring in T. The *domain closure axiom for* T, written $DCA(T)$, is the sentence

$$\forall x. \, x = c_1 \vee \cdots \vee x = c_n.$$

Let $\overline{c}$ be a tuple of individual constants. By $DCA^{-\overline{c}}(T)$ we shall denote the sentence $\forall x \; x = c_1 \vee \ldots \vee x = c_n$, where $c_1, \ldots, c_n$ are all the individual constants of T excluding constants from $\overline{c}$. For $k \in \omega$, by $DCA^{+k}(T)$ we shall denote the sentence

$$\exists z_1 \cdots \exists z_k \forall x \; x = z_1 \vee \cdots \vee x = z_k \vee x = c_1 \vee \ldots \vee x = c_n,$$

where $c_1, \ldots, c_n$ are all the individual constants of T. We also use notation $DCA^{-\overline{c}+k}(T)$ as a combination of the above. ∎

Definition 2.4 A predicate variable Φ occurs *positively* (resp. *negatively*) in a formula A if the conjunctive normal form of A contains a subformula of the form $\Phi(\overline{t})$ (resp. $\neg\Phi(\overline{t})$). A formula A is said to be *positive* (resp. *negative*) w.r.t. Φ iff all occurrences of Φ in A are positive (resp. negative). ∎

Definition 2.5 Let Φ be either a predicate constant or a predicate variable and $\overline{\Phi}$ be a tuple of predicate constants or a tuple of predicate variables. Then a formula $T(\Phi)$ is said to be *separated* w.r.t. Φ iff it is of the form $T_1(\Phi) \wedge T_2(\Phi)$ where $T_1(\Phi)$ is positive w.r.t. Φ and T_2 is negative w.r.t. Φ. ∎

3　General Domain Circumscription

In this section, we provide a definition of general domain circumscription (GDC) and its model-preferential semantics. GDC subsumes both McCarthy's original domain circumscription, introduced in [9], studied in [2], and substantially improved in [5], and Hintikka's mini-consequence, formulated in [6] and studied in [7].

Definition 3.1 Let $\overline{P} = (P_1, \ldots, P_n)$ be a tuple of different predicate constants, $\overline{f} = (f_1, \ldots, f_k)$ be a tuple of different function constants (including, perhaps, individual constants), $T(\overline{P}, \overline{f})$ be a theory and let Φ be a one-place predicate variable, $\overline{\Psi}$ be a tuple of predicate variables similar to $\overline{P}$, and $\overline{\psi}$ be a tuple of function variables similar to $\overline{f}$. By $Axiom(\Phi, \overline{P}, \overline{f})$, sometimes abbreviated by $Axiom(\Phi)$, we shall mean the conjunction of:

- $\Phi(a)$, for each individual constant a in T not occurring in $\overline{f}$,

- $\Phi(\psi_i)$, for each individual constant a in T such that a is f_i,

- $\forall x_1 \ldots x_n[\Phi(x_1) \wedge \cdots \wedge \Phi(x_n) \supset \Phi(f(x_1, \ldots x_n))]$, for each n-ary ($n > 0$) function constant f in T not occurring in $\overline{f}$, and

- $\forall x_1 \ldots x_n[\Phi(x_1) \wedge \cdots \wedge \Phi(x_n) \supset \Phi(\psi_i(x_1, \ldots x_n))]$, for each n-ary ($n > 0$) function constant f in T such that f is f_i.

T^Φ stands for the result of rewriting $T(\overline{\Psi}, \overline{\psi})$, replacing each occurrence of $\forall x$ and $\exists x$ in $T(\overline{\Psi}, \overline{\psi})$ with "$\forall x \; \Phi(x) \supset$" and "$\exists x \; \Phi(x) \wedge$", respectively. ∎

Definition 3.2 Let $\overline{P} = (P_1, \ldots, P_n)$, $\overline{f} = (f_1, \ldots, f_k)$ and $T(\overline{P}, \overline{f})$ be as in Definition 3.1. The *general domain circumscription for* $T(\overline{P}, \overline{f})$ *with variable* $\overline{P}$ *and* $\overline{f}$, written $CIRC_D(T; \overline{P}; \overline{f})$, is the following sentence of second-order logic:

$$T(\overline{P}, \overline{f}) \wedge \forall \Phi \forall \overline{\Psi} \forall \overline{\psi}[(\exists x \Phi(x) \wedge Axiom(\Phi, \overline{P}, \overline{f}) \wedge T^{\Phi}) \supset \forall x \Phi(x)]. \tag{1}$$

A formula α is said to be a *consequence of* $CIRC_D(T; \overline{P}; \overline{f})$ iff $CIRC_D(T; \overline{P}; \overline{f}) \models \alpha$, where "$\models$" denotes the entailment relation of classical second-order logic.

The second conjunct of the sentence (1) is called the *domain circumscription axiom*. ∎

It is not difficult to see that (1) asserts that the domain of discourse (represented by Φ) is minimal with respect to T, where $\overline{P}$ and $\overline{f}$ are allowed to vary during the minimization.

We shall write $CIRC_D(T)$ as an abbreviation for $CIRC_D(T; (); ())$, i.e. if neither predicate nor function constants are allowed to vary. This simplest form of domain minimization corresponds closely to McCarthy's original domain circumscription [9] with the augmentation described in [5].[1]

We shall write $CIRC_D(T; \overline{P})$ as an abbreviation for $CIRC_D(T; \overline{P}; ())$, i.e. if some predicate constants, but not function constants, are allowed to vary. If $\overline{P}$ includes all predicate constants occurring in a theory T, then $CIRC_D(T; \overline{P})$ is exactly *mini-consequence*, introduced in [6] and improved in [7]. Following [7], this form of minimization will be referred to as *variable domain circumscription*.[2]

Example 3.1 Consider a theory T consisting of $\forall x \ P(x) \wedge Q(x) \wedge P(a) \wedge Q(b)$. We shall minimize the domain of T without varying predicate or function constants. $CIRC_D(T)$ is given by

$$T \wedge \forall \Phi[(\exists x \Phi(x) \wedge \Phi(a) \wedge \Phi(b) \wedge \forall x(\Phi(x) \supset (P(x) \wedge Q(x) \wedge P(a) \wedge Q(b)))) \supset \forall x \Phi(x)]. \tag{2}$$

Substituting $\lambda x.x = a \vee x = b$ for Φ, we get

$$\begin{aligned} T \wedge [\exists x(x = a \vee x = b) \wedge (a = a \vee a = b) \wedge (b = a \vee b = b) \wedge \\ \forall x[(x = a \vee x = b) \supset (P(x) \wedge Q(x) \wedge P(a) \wedge Q(b))] \supset \\ \forall x(x = a \vee x = b)]. \end{aligned} \tag{3}$$

Since (3) is equivalent to $T \wedge (\forall x \ x = a \vee x = b)$, we conclude that the domain closure axiom for T, i.e. the sentence $\forall x \ x = a \vee x = b$ is a consequence of $CIRC_D(T)$. ∎

Example 3.2 Let T consist of $P(a) \wedge P(b)$. We minimize the domain of T with a allowed to vary during the minimization. $CIRC_D(T; (), (a))$ is given by[3]

$$T \wedge \forall \Phi \forall x_a[\exists x \Phi(x) \wedge \Phi(x_a) \wedge \Phi(b) \wedge P(x_a) \wedge P(b) \supset \forall x \Phi(x)]. \tag{4}$$

Substituting $\lambda x.x = b$ for Φ and b for x_a, one easily calculates that (4) implies $T \wedge \forall x \ x = b$. Accordingly, we conclude that the domain of T consists of one object, referred to by both a and b. ∎

---

[1] In fact, $CIRC_D(T)$ is slightly stronger in that it is based on a second-order axiom rather than on a first-order schema.

[2] Note that in variable domain circumscription all predicate constants, but no function constants, are allowed to vary during the minimization process.

[3] Since a is an individual constant, the variable corresponding to a is, in fact, an individual variable. Accordingly, we denote it by x_a, rather than by ψ.

We now proceed to give a semantics for general domain circumscription. We start with some notational conventions. Given a model M, we shall write $\mid M \mid$ to denote the domain of M. By $V_v^M(A)$ we denote the truth-value of formula A in M with respect to a valuation v. Note that if A is a formula of second-order logic, then "v" provides an interpretation not only for individual variables, but also for predicate (and possibly function) variables. If v is a valuation for a model M, d is an element of $\mid M \mid$ and x is an individual variable, then $v[x \leftarrow d]$ denotes an assignment that is identical to v except for the variable x which is assigned the value d.

Definition 3.3 Let $\overline{P}$, $\overline{f}$ and $T(\overline{P}, \overline{f})$ be as in Definition 3.1. Let M_1 and M_2 be models of T. We say that M_1 is a $(\overline{P}; \overline{f})$-*submodel* of M_2, written $M_1 \leq^{(\overline{U}, \overline{u})} M_2$, iff $\mid M_1 \mid \subseteq \mid M_2 \mid$, and for each predicate, function or individual constant C, occurring neither in $\overline{P}$ nor in $\overline{f}$, the interpretation of C in M_1 is the restriction of the corresponding interpretation in M_2 to $\mid M_1 \mid$. A model is said to be $(\overline{P}; \overline{f})$-*minimal* iff it has no proper $(\overline{P}; \overline{f})$-submodels. ∎

Theorem 3.1 Let $\overline{P}$, $\overline{f}$ and $T(\overline{P}, \overline{f})$ be as in Definition 3.1. A formula A is a consequence of $CIRC_D(T; \overline{P}; \overline{f})$ iff A is true in all $(\overline{P}; \overline{f})$-minimal models of T.

<u>Proof</u> It suffices to show the following: a model M of T is a model of $CIRC_D(T, \overline{P}, \overline{f})$ iff M is a $(\overline{P}; \overline{f})$-minimal model of T.

($\Rightarrow$) Assume to the contrary that M is a model of $CIRC_D(T, \overline{P}, \overline{f})$ and M is not $(\overline{P}; \overline{f})$-minimal. Thus, there is a model N of T which is a proper $(\overline{P}; \overline{f})$-submodel of M. Clearly, $\mid N \mid \subset \mid M \mid$. Since M is a model of $CIRC_D(T, \overline{P}, \overline{f})$,

$$V_v^M(\exists x \Phi(x) \wedge Axiom(\Phi) \wedge T^\Phi \supset \forall x \Phi(x)) = 1 \tag{5}$$

for each assignment v. Suppose that $v(\Phi)$ is $\mid N \mid$ and $v(\Psi_i)$ (resp. $v(\psi_i)$) is the relation (resp. function) corresponding to the interpretation of P_i (resp. f_i) in N. Since N is a $(\overline{P}; \overline{f})$-submodel of T, this assignment is well-defined. It is easily verified that $V_v^M(\exists x \Phi(x) \wedge Axiom(\Phi) \wedge T^\Phi) = 1$. Thus, in view of (5), we infer that $V_v^M(\forall x \Phi(x)) = 1$. This means that $\mid M \mid = \mid N \mid$. A contradiction.

($\Leftarrow$) Assume to the contrary that M is a $(\overline{P}; \overline{f})$-minimal model of T and M is not a model of $CIRC_D(T, \overline{P}, \overline{f})$. Thus, there is an assignment v such that

$$V_v^M(\exists x \Phi(x) \wedge Axiom(\Phi) \wedge T^\Phi) = 1 \tag{6}$$

$$V_v^M(\forall x \Phi(x)) = 0. \tag{7}$$

Consider the set S given by: $d \in S$ iff $V_{v(x \leftarrow d)}^M(\Phi(x)) = 1$. In view of (7), S is a proper subset of $\mid M \mid$. Moreover, in view of (6), S is non-empty. We define a model N such that $\mid N \mid$ is S and the interpretation of each predicate and function constant, not in $\overline{P}, \overline{f}$ in N is the restriction of the corresponding interpretation in M to $\mid N \mid$. Using (6), one easily checks that N is a model of T. Thus, since $\mid N \mid \subset \mid M \mid$, N is a proper $(\overline{P}; \overline{f})$-submodel of M. A contradiction. ∎

3.1   An Optimization Technique

In this section, we propose a technique that allows one to reduce the size of a domain circumscription axiom. The technique allows one to sometimes remove counterparts of universal formulas from the axiom. More precisely, let theory T consist of axioms, including universal axioms of the form $\forall x_1 \cdots \forall x_n\ A(x_1, \cdots, x_n)$, and suppose that all predicate and/or function constants occurring in

$A(x_1, \cdots, x_n)$ are not allowed to vary during the minimization. Then each such axiom reappears in (1) as a part of T^ϕ, equivalent to the formula

$$\forall x_1 \cdots \forall x_n \; \neg \Phi(x_1) \vee \cdots \vee \neg \Phi(x_n) \vee A(x_1, \cdots, x_n). \tag{8}$$

Since (8), together with the corresponding axiom of T, reduces to $\top$, it can be removed from T^ϕ. We thus have the following principle:

> Remove a counterpart of every universal axiom A of T from T^ϕ in the domain circumscription axiom, provided that none of the predicate and/or function constants occurring in A are allowed to vary. If B is the resulting formula, then $T \wedge B$ is equivalent to $CIRC_D(T; \overline{P}; \overline{f})$.

Observe also that one can remove formula $\exists x \Phi(x)$ from (1) whenever T contains a constant symbol. This follows from the fact that for each constant symbol, say a, one has $\Phi(a)$ as a conjunct of $Axiom(\Phi)$. Thus $\exists x \Phi(x)$ follows from $Axiom(\Phi)$ and can be removed.

It should be emphasized that the DLS algorithm works successfully without the above mentioned optimizations. However, as shall be seen in the examples in Section 6, they usually considerably decrease the complexity of the reduced formula.

4 DLS Algorithm

4.1 The Basic DLS Algorithm

In this section, we briefly describe the DLS algorithm mentioned in the introduction. Its complete formulation can be found in [3]. The algorithm was originally formulated in a weaker form in [12], in the context of modal logics. It is based on Ackermann's techniques developed in connection with the elimination problem (see [1]). The DLS algorithm is based on the following lemma, proved by Ackermann in 1934 (see [1]). The proof can also be found in [12].

Lemma 4.1 (Ackermann Lemma) Let Φ be a predicate variable and $A(\bar{x}, \bar{z})$, $B(\Phi)$ be formulas without second-order quantification. Let $B(\Phi)$ be positive w.r.t. Φ and let A contain no occurrences of Φ at all. Then the following equivalences hold:

$$\exists \Phi \forall \bar{x} [\Phi(\bar{x}) \vee A(\bar{x}, \bar{z})] \wedge B(\Phi \leftarrow \neg \Phi) \;\equiv\; B(\Phi \leftarrow A(\bar{x}, \bar{z})) \tag{9}$$

$$\exists \Phi \forall \bar{x} [\neg \Phi(\bar{x}) \vee A(\bar{x}, \bar{z})] \wedge B(\Phi) \;\equiv\; B(\Phi \leftarrow A(\bar{x}, \bar{z})) \tag{10}$$

where in the righthand formulas the arguments $\bar{x}$ of A are each time substituted by the respective actual arguments of Φ (renaming the bound variables whenever necessary). ∎

The DLS algorithm is based on eliminating second-order quantifiers of the input formula using a combination of applications of Lemma 4.1 together with various syntactic transformations which preserve equivalence.

4.2   Problems with the Basic DLS Algorithm

There are two weaknesses associated with the basic DLS algorithm which cause it to terminate with failure:

1. Non-separated input problem.

2. Unskolemization problem.

In order for the DLS algorithm to reduce an input formula, it must be possible for the formula to be transformed into separated form. If the input formula consists of clauses which contain both positive and negative occurrences of the predicate variable being eliminated, then the basic DLS algorithm will return with failure.

Another limitation of the basic DLS algorithm involves unskolemization. Skolemization is sometimes required either due to the original form of the input formula, or to one of the phases in the algorithm which may introduce new existential quantifiers. When applying Ackermann's Lemma, all existentially quantified individual variables have to be removed from the prefix of the formula being reduced. For this purpose, Skolemization is performed using the equivalence,

$$\forall \overline{x} \exists y \ A(\overline{x}, y) \equiv \exists f \forall \overline{x} \ A(\overline{x}, y \leftarrow f(\overline{x})), \tag{11}$$

where f is a new function variable. After application of Ackermann's Lemma, one tries to remove the newly introduced function variables using equivalence (11) in the other direction. Unfortunately, unskolemization is not always successful.

5   Extending the DLS Algorithm

There are two generalizations of the basic DLS algorithm that extend the class of input formulas that can be successfully reduced to include non-separated input formulas and formulas which would normally fail to be reduced due to unskolemization problems.

The first method appeals to the observation that for a particular class of theories whose domain closure axiom is entailed by the corresponding general domain circumscribed theory, both the non-separated input and unskolemization problems can be avoided by combining the basic DLS algorithm with the additional constraints contributed by the domain closure axiom associated with the input theory. Although this method can be used for a particular class of input formulas, it can not be used for all non-separated input formulas.

The second method generalizes Ackermann's Lemma (4.1) by transforming an input formula into a (possibly) non-separated form which can be shown to be logically equivalent to a fixpoint formula in a fixpoint calculus. In the case where the fixpoint formula is bounded, the non-separated input formula can be reduced to a logically equivalent first-order formula.

In Section's 5.1 and 5.2, the formal justification on which the two methods are based will be described.

5.1   DLS Algorithm with the Domain Closure Axiom

As mentioned before, the DLS algorithm may fail due to non-separatedness and unskolemizaton problems. On the other hand, whenever it is known that the domain closure axiom (DCA) follows from the theory considered, separatedness and unskolemization are always possible. This

is particularly important in cases when one combines domain circumscription with other second-order formalisms, like e.g. second-order circumscription.

Assume that T is a theory. Then, for each formula A, $DCA(T)$ implies:

$$\exists x\ A(x)\ \equiv\ (A(c1) \vee \cdots \vee A(c_n)) \tag{12}$$

and

$$\forall x\ A(x)\ \equiv\ (A(c1) \wedge \cdots \wedge A(c_n)). \tag{13}$$

The following example illustrates the use of equivalences (12) and (13).

Example 5.1 Assume that $\forall x\ x = a \vee x = b$ holds. An application of equivalence (12) to formula $\forall y \exists z\ P(y, z)$ results in $\forall y (P(y, a) \vee P(y, b))$. An application of equivalence (13) to this formula results in $(P(a, a) \vee P(a, b)) \wedge (P(b, a) \vee P(b, b))$. ∎

Using equivalence (12) one can remove existential quantifiers that would require Skolemization. This solves the unskolemization problem associated with the DLS algorithm. Observe that in order to make the DLS algorithm work one could also use equivalence (13) in order to remove universal quantifiers preceding the existential quantifiers, whenever necessary.

The second reason the DLS algorithm fails is when formulas cannot be separated w.r.t. predicate Φ. In the canonical case, this occurs when a universally quantified clause contains both positive and negative occurrences of Φ. Using equivalence (13), one can remove the universal quantifiers from the clause prefix. This, together with certain distributions across subformulas, is guaranteed to transform the initially non-separated formula into a separated formula.

Of course, the above technique can easily be modified if it is known that $DCA^{-\overline{c}}(T)$, $DCA^{+k}(T)$ or $DCA^{-\overline{c}+k}(T)$ is entailed from T. Before we introduce this modification, consider the following simple example.

Example 5.2 Assume that $\exists z \forall x\ x = z \vee x = a$ holds. An application of equivalence (13) to formula $\forall y\ P(y)$ results in $\exists z [\forall x (x = z \vee x = a) \wedge P(z) \wedge P(a)]$.

Observe that, unlike Example (5.1), the DCA reappears in the result. This is due to the existential quantifier $\exists z$ that has to bind both the DCA and the resulting formula. ∎

The following theorem justifies the technique.

Theorem 5.1 Assume that for a given second-order theory T,

$$T\ \models\ DCA^{-\overline{c}+k}(T).$$

Then T is equivalent to a first-order formula.

<u>Proof</u>   Let $DCA^{-\overline{c}+k}(T)$ be the formula

$$\exists z_1 \cdots \exists z_k \forall x\ x = z_1 \vee \cdots \vee x = z_k \vee x = c_1 \vee \ldots \vee x = c_n, \tag{14}$$

where $c_1, \ldots, c_n$ are all the individual constants of T excluding those in $\overline{c}$. In order to prove that all second-order quantifiers can be eliminated from T, we show that both the Skolemization and

9

non-separatedness problems associated with the DLS algorithm can be avoided. Since these are the only reasons why the DLS algorithm may fail, this will prove the theorem.

Suppose a second-order quantifier from a subformula of T is being eliminated. Since only existential second-order quantifiers are considered,[4] a subformula of T of the form $\exists\Phi(A(\Phi))$, where A is a first-order formula is the only case requiring consideration.

In order to make things simple, A is first transformed into prenex conjunctive normal form. Formula (14) is then Skolemized and its equivalent[5]

$$\forall x \; x = b_1 \vee \cdots \vee x = b_k \vee x = c_1 \vee \ldots \vee x = c_n, \tag{15}$$

is obtained, where $b_1, \cdots, b_k$ are Skolem constants replacing $z_1, \cdots, z_k$.

Now in order to prove our theorem, we *run* the DLS algorithm.

Each time the DLS algorithm would perform Skolemization, the respective existential quantifier, say $\exists u \alpha(u)$, is eliminated by using equivalence (12). In this case $\exists u \alpha(u)$ is replaced by the disjunction $\alpha(b_1) \vee \cdots \vee \alpha(b_k) \vee \alpha(c_1) \vee \cdots \vee \alpha(c_n)$.

Observe that the only reason a formula can not be separated is when a universal quantifier binds a clause containing both positive and negative occurrences of the predicate variable being eliminated , e.g. we have a clause in A of the form $\forall x \forall y (P(x) \vee P(y) \vee \alpha(x,y))$. Each time DLS would fail due to the non-separatedness problem, the problematic universal quantifier, say $\forall u \alpha(u)$, is eliminated by using equivalence (13). In this case, $\forall u \alpha(u)$ is replaced by the conjunction $\alpha(b_1) \wedge \cdots \wedge \alpha(b_k) \wedge \alpha(c_1) \wedge \cdots \wedge \alpha(c_n)$.

This solves the problems associated with the original DLS algorithm and ensures that it runs successfully, eliminating all second-order quantifiers of T. ∎

Since $CIRC_D(T, \overline{P}, \overline{f})$ is a second-order sentence, we have the following corollary.

Corollary 5.1 Assume that $CIRC_D(T, \overline{P}, \overline{f}) \models DCA^{-\overline{e}+k}(T)$. Then $CIRC_D(T, \overline{P}, \overline{f})$ is equivalent to a first-order formula.∎

Theorem 5.1 allows us to modify the DLS algorithm in such a way that whenever there is a Skolemization or separatedness problem, one applies formulas (12) and (13), respectively. We will denote this modification of the DLS algorithm by DLS^*.

The following example illustrates the use of the DLS^* algorithm.

Example 5.3 Assume that the DCA is of the form $\exists z \forall x \; x = z \vee x = a$, and let T be the second-order formula

$$\exists\Phi\forall x\forall y[(\Phi(x) \vee \neg\Phi(y)) \wedge \exists z\neg\Phi(z) \wedge \exists u\Phi(u)] \wedge (\exists z\forall x \; x = z \vee x = a). \tag{16}$$

We first Skolemize DCA and obtain $\forall x(x = b \vee x = a)$. We then try to eliminate the quantifier $\exists\Phi$ from formula (16) using the DLS algorithm. In this case, one first Skolemizes one of the first-order existential quantifiers. Whichever is chosen, we are then faced with a non-separated formula. Due

---

[4] Universal second-order quantifiers are processed by first negating the formula, applying the DLS algorithm to the negated formula stripped of its negation sign and then negating the output of the algorithm.

[5] We use second-order Skolemization given by formula (11). This means, that we always have a prefix of existential quantifiers binding Skolem constants in front of the whole theory. In this case, unskolemization is always possible – see also Examples 5.2 and 5.3

to this, the DLS algorithm fails. If instead one uses the DLS^* algorithm, we first eliminate one of the existential quantifiers, say $\exists u$, by applying equivalence (12) and obtain

$$\exists b[DCA \wedge \exists \Phi \forall x \forall y((\Phi(x) \vee \neg \Phi(y)) \wedge \exists z \neg \Phi(z) \wedge (\Phi(b) \vee \Phi(a)))]. \tag{17}$$

Formula (17) is not separated. We thus apply equivalence (13) to quantifier $\forall y$ and obtain:

$$\exists b[DCA \wedge \exists \Phi \forall x((\Phi(x) \vee \neg \Phi(b)) \wedge (\Phi(x) \vee \neg \Phi(a))) \wedge \exists z \neg \Phi(z) \wedge (\Phi(b) \vee \Phi(a)))]. \tag{18}$$

(18) is equivalent to

$$\exists b[DCA \wedge [\exists \Phi(\exists z \neg \Phi(z) \wedge \forall x \Phi(x) \wedge (\Phi(b) \vee \Phi(a)))] \vee \tag{19}$$
$$[\exists \Phi(\exists z \neg \Phi(z) \wedge \neg \Phi(b) \wedge \neg \Phi(a) \wedge (\Phi(b) \vee \Phi(a)))]$$

It is easily observed that each disjunct is in separated form, and no additional skolemization is necessary, so application of the basic DLS algorithm results in a first-order formula equivalent to (19). ∎

5.2 Fixpoint Calculus

Let $\mathcal{L}_\mathcal{I}$ be the classical first-order logic. In order to define the *fixpoint calculus* $\mathcal{L}_\mathcal{F}$, we extend $\mathcal{L}_\mathcal{I}$ by allowing the least fixpoint operator $\mu\Phi.A(\Phi)$, where A is positive w.r.t. Φ. We abbreviate a formula of the form $\neg\mu\neg\Phi.\neg A(\neg\Phi)$ by $\nu\Phi.A(\Phi)$. It is sometimes convenient to indicate the individual variables that are bound by the fixpoint operators μ and ν. We write $\mu.\Phi(\bar{x})$ and $\nu.\Phi(\bar{x})$ to indicate that the tuple $\bar{x}$ of variables is bound by a fixpoint operator.

Every formula $A(\Phi)$ which is positive w.r.t. Φ is monotone. Consequently, by the Knaster & Tarski fixpoint theorem, we are assured that the fixpoints we consider are well defined. Moreover, the fixpoints have the following nice characterization:

$$\mu\Phi.A(\Phi) \;\equiv\; \bigvee_{\beta \in \alpha} A^\beta(\bot) \tag{20}$$

for an ordinal α.

Definition 5.1 The least ordinal α for which equivalence (20) holds is called the *closure ordinal for $A(\Phi)$*.

A fixpoint formula is called *bounded* iff it contains only fixpoint operators with finite closure ordinals. ∎

Note that $\mu\Phi(\bar{x}).A(\Phi)$ is the least (w.r.t. $\leq$) formula $B(\bar{x})$ such that

$$B(\bar{x}) \;\equiv\; A(\Phi \leftarrow B(\bar{x})).$$

Proposition 5.1 Let T be a first-order theory. If there is $i \in \omega$ such that $T \models A^i(\bot) \equiv A^{i+1}(\bot)$ then $T \models \mu\Phi.A(\Phi) \;\equiv\; \bigvee_{j \leq i} A^j(\bot)$. ∎

The following theorem is proved in [10].

Theorem 5.2 Assume that all occurrences of the predicate variable Φ in the formula B bind only variables.

- if A and B are negative w.r.t. Φ then the closure ordinal for $A(\neg\Phi)$ is less than or equal to ω, and

$$\exists\Phi\forall\bar{y}\,[\Phi(\bar{y}) \vee A(\neg\Phi)] \wedge [B(\neg\Phi)] \;\equiv\; B[\neg\Phi \leftarrow \nu\neg\Phi(\bar{y}).A(\neg\Phi)] \tag{21}$$

- if A and B are positive w.r.t. Φ then the closure ordinal for $A(\Phi)$ is less than or equal to ω, and

$$\exists\Phi\forall\bar{y}[\neg\Phi(\bar{y}) \vee A(\Phi)] \wedge [B(\Phi)] \;\equiv\; B[\Phi \leftarrow \nu\Phi(\bar{y}).A(\Phi)] \tag{22}$$

where the above substitutions exchange the variables bound by fixpoint operators by the corresponding actual variables of the substituted predicate. ∎

Note the similarity between Theorem 5.2 and Ackermann's Lemma 4.1. In this case though, the second-order formula on the lhs is not necessarily separated and the logically equivalent formula on the rhs is not a first-order formula, but a fixpoint formula which is potentially reducible to a first-order formula if it can be shown to be bounded (see [4]).

6 Reducing General Domain Circumscription

In this section we provide some reducibility results concerning various variants of general domain circumscription. In what follows, we assume that theories under consideration contain at least one individual constant symbol.

6.1 Fixed GDC

6.1.1 Universal Theories

In Example 3.1, we saw that domain circumscription may allow the derivation of the domain closure axiom. It turns out that for universal theories without function constants this is always the case. Moreover, as the next theorem shows, if T is a theory of that type, then the domain circumscription of T is equivalent to $T \wedge DCA(T)$.

Theorem 6.1 Let T be a universal theory without function symbols. Then $CIRC_D(T)$ is always reducible into first-order logic using the DLS algorithm. Moreover, if A is the resulting formula, then A is equivalent to $T \wedge DCA(T)$.

<u>Proof</u>   Let, for some $k > 0$, $c_1, \cdots, c_k$ be all the individual constants occurring in T. The optimization technique from Section 3.1, allows us to write $CIRC_D(T)$ in the form

$$T \wedge \forall\Phi[(\Phi(c_1) \wedge \cdots \wedge \Phi(c_k)) \supset \forall z\ \Phi(z)]. \tag{23}$$

After negating the second-order conjunct of (23), we obtain

$$\exists\Phi[\Phi(c_1) \wedge \cdots \wedge \Phi(c_k) \wedge \exists z\neg\Phi(z)]. \tag{24}$$

The DLS algorithm transforms (24) into

$$\exists z \exists \Phi[\Phi(c_1) \wedge \cdots \wedge \Phi(c_k) \wedge \forall u(\neg \Phi(u) \vee z \neq u)]. \tag{25}$$

Applying Ackermann's lemma to (25), results in

$$\exists z[z \neq c_1 \wedge \cdots \wedge z \neq c_k]. \tag{26}$$

Negating (26), we obtain

$$\forall z[z = c_1 \vee \cdots \vee z = c_k].$$

Thus

$$CIRC_D(T) \equiv T \wedge \forall z[z = c_1 \vee \cdots \vee z = c_k].$$

∎

Observe that according to our assumption, we consider only theories that contain at least one individual constant. This is only a technical assumption. If T has no individual constant symbols then

$$CIRC_D(T) \equiv T \wedge \forall \Phi[\exists x\ \Phi(x) \supset \forall z\ \Phi(z)].$$

After negating the second conjunct of this formula we obtain

$$\exists \Phi[\exists x\ \Phi(x) \wedge \exists z \neg \Phi(z)],$$

which, after applying the DLS algorithm, results in the equivalent $\exists x \exists z[z \neq x]$. Thus $CIRC_D(T) \equiv T \wedge \forall x \forall z(x = z)$.

6.1.2   Semi-Universal Theories

As regards semi-universal theories without function symbols we have the following theorem:

Theorem 6.2 Let T be a semi-universal theory without function symbols. Then $CIRC_D(T)$ is always reducible into first-order logic using the DLS algorithm. Moreover, if A is the resulting formula, then A implies $DCA^{+k}(T)$, where k is the number of existential quantifiers of T.

<u>Proof</u>   Follows from Theorem 6.6 and the fact that the DCA axiom is unnecessary here. (See the proof of Theorem 6.6 and note that the DCA axiom is only needed there to eliminate predicate variables corresponding to varying predicates.) ∎

The following example illustrates a reduction of a semi-universal theory:

Example 6.1 Let T consist of

A1. $\exists x F(x)$
A2. $B(Tweety)$

where B and F stand for *Bird* and *Flies*, respectively. We shall reduce $CIRC_D(T)$, i.e. no predicate or individual constants are allowed to vary during the minimization. The corresponding domain circumscription axiom is

$$\forall \Phi[(\exists x \Phi(x) \wedge \Phi(Tweety) \wedge \exists x(\Phi(x) \wedge F(x))) \supset \forall x \Phi(x)]. \tag{27}$$

After removing $\exists x \Phi(x)$ from (27) and negating the result, we obtain

$$\exists \Phi[\Phi(Tweety) \wedge \exists x(\Phi(x) \wedge F(x)) \wedge \exists x \neg \Phi(x)]. \tag{28}$$

To eliminate quantification over Φ, we transform (28) into the form

$$\exists x' \exists \Phi[\Phi(Tweety) \wedge \exists x(\Phi(x) \wedge F(x)) \wedge \forall z(\neg \Phi(z) \vee z \neq x')]. \tag{29}$$

Applying Ackermann's Lemma, we obtain

$$\exists x'[x' \neq Tweety \wedge \exists x(x' \neq x \wedge F(x))]. \tag{30}$$

After negating (30), we obtain

$$\forall x'[x' = Tweety \vee \forall x(x' = x \vee \neg F(x))] \tag{31}$$

which is equivalent to

$$\forall x \forall x'[x' = Tweety \vee x' = x \vee \neg F(x)]. \tag{32}$$

Thus

$$CIRC_D(T) \equiv \exists x F(x) \wedge B(Tweety) \wedge \forall x \forall x'[x' = Tweety \vee x' = x \vee \neg F(x)]. \tag{33}$$

It is easily seen that (33) implies $\exists z \forall x'[x' = Tweety \vee x' = z]$, i.e. $DCA^{+1}(T)$. ∎

6.2 Variable GDC

For universal and semi-universal theories, we have the following counterparts of Theorems 6.1 and 6.2:

6.2.1 Universal Theories

Theorem 6.3 Let T be a universal theory without function symbols and suppose that $\overline{P}$ is a tuple of predicate symbols occurring in T. Then $CIRC_D(T; (\overline{P}))$ is always reducible into first-order logic using the DLS^* algorithm. Moreover, if A is the resulting formula, then A implies $DCA(T)$.

<u>Proof</u>   Observe that $CIRC_D(T; \overline{P})$ implies $CIRC_D(T)$ thus (by Theorem 6.1) it implies $DCA(T)$. The reduction of $CIRC_D(T; \overline{P})$ now easily follows from the discussion in Section 5.1. ∎

14

Example 6.2 Let theory T consist of $\forall x \; R(x,a) \supset R(a,x)$ and consider $CIRC_D(T;R)$. In this case we cannot apply the optimization technique in Section 3.1 concerning universal axioms since R is varied. Thus the second-order part of $CIRC_D(T;R)$ is equivalent to

$$\forall \Psi \forall \Phi [\Phi(a) \wedge \forall x (\neg \Phi(x) \vee (\Psi(x,a) \supset \Psi(a,x)))] \supset \forall z \Phi(z). \tag{34}$$

In order to eliminate the quantifier Φ, the DLS algorithm negates this formula and transforms it into a form required for application of Ackermann's Lemma,

$$\exists \Psi \exists \Phi [\forall x (\Phi(x) \vee x \neq a) \wedge \forall x (\neg \Phi(x) \vee \neg \Psi(x,a) \vee \Psi(a,x)) \wedge \exists z \neg \Phi(z)]. \tag{35}$$

The output of the DLS algorithm after the elimination of Φ results in

$$\exists \Psi [\forall x (x \neq a \vee \neg \Psi(x,a) \vee \Psi(a,x)) \wedge \exists z (z \neq a)]. \tag{36}$$

The elimination of Ψ is now straightforward. The DLS algorithm determines that the clause $\forall x (x \neq a \vee \neg \Psi(x,a) \vee \Psi(a,x))$ can be removed, resulting in $\exists z (z \neq a)$. Negating the formula results in $\forall z (z = a)$ which is a first-order equivalent of formula (34). ■

For theories with varied individual constants the following theorem holds.

Theorem 6.4 Let T be a universal theory without function symbols. Let $\overline{P}$ be a tuple of predicate symbols and $\overline{c}$ be a tuple of individual constants occurring in T. Then $CIRC_D(T;\overline{P};\overline{c})$ is always reducible into first-order logic using the DLS^* algorithm. Moreover, if A is the resulting formula, then A implies $DCA^{-\overline{c}}(T)$.

<u>Proof</u>   Follows from Theorem 6.6 (with $k = 0$). ■

The following example varies an individual constant.

Example 6.3 Consider the theory T given by

A1. $S(c) \wedge S(d)$
A2. $\forall x \; R(x,c) \equiv R(x,d)$
A3. $\forall x \neg R(x,c) \wedge \forall y \neg R(y,d) \wedge \forall z \neg R(z,z)$.

This example is taken from [11]. Here $S(x)$, $R(x,y)$, c and d are to be read "the evidence says that x saw the victim alive", "the evidence says that x saw the victim alive after y saw her alive for the last time", "murderer" and "suspect", respectively. Suppose further that the police tries to find all individuals who satisfy exactly those formulas that the (unknown) murderer c does, by comparing what is provable about the murderer with what is provable about a particular individual. To formalize this type of procedure, we should minimize the domain under consideration with all constant symbols, except that referring to the murderer which is allowed to vary. In our case, we minimize the domain of T with variable c. The intended conclusion is $d = c$.

The second-order part of $Circ_D(T,(),(c))$, after simplifications, is equivalent to

$$\forall x_c \forall \Phi [\Phi(x_c) \wedge \Phi(d) \wedge S(x_c) \wedge \tag{37}$$
$$\forall x (\Phi(x) \supset ((\neg R(x,x_c) \vee R(x,d)) \wedge (R(x,x_c) \vee \neg R(x,d)))$$
$$\wedge \forall x (\Phi(x) \supset \neg R(x,x_c)) \supset \forall s \Phi(s)].$$

Negating (37), we obtain

$$\exists x_c \exists \Phi [\Phi(x_c) \wedge \Phi(d) \wedge S(x_c) \wedge \tag{38}$$
$$\forall x (\neg \Phi(x) \vee \neg R(x, x_c) \vee R(x, d)) \wedge (\neg \Phi(x) \vee R(x, x_c) \vee \neg R(x, d)))$$
$$\wedge \forall x (\neg \Phi(x) \vee \neg R(x, x_c)) \wedge \exists s \neg \Phi(s)]$$

which is transformed by the DLS algorithm to

$$\exists s \exists x_c \exists \Phi \forall x [(\Phi(x) \vee (x \neq x_c \wedge x \neq d)) \wedge S(x_c) \wedge \tag{39}$$
$$\forall x (\neg \Phi(x) \vee \neg R(x, x_c)) \wedge \neg \Phi(s)]$$

and then, after the application of Ackermann's Lemma, to

$$\exists s \exists x_c [S(x_c) \wedge \forall x ((x \neq x_c \wedge x \neq d) \vee \neg R(x, x_c) \wedge \tag{40}$$
$$(s \neq x_c \wedge s \neq d)].$$

Negating (40), we obtain

$$\forall s \forall x_c [S(x_c) \wedge \forall x ((x \neq x_c \wedge x \neq d) \vee \neg R(x, x_c) \supset \tag{41}$$
$$(s \neq x_c \wedge s \neq d)]$$

and so,

$$Circ_D(T, (), (c)) \equiv T \wedge \forall s \forall x_c [S(x_c) \wedge \forall x ((x \neq x_c \wedge x \neq d) \vee \neg R(x, x_c) \supset \tag{42}$$
$$(s \neq x_c \wedge s \neq d)]$$

It is easily seen, substituting d for x_c, that $Circ_D(T, (), (c)) \models d = c$. $\blacksquare$

6.2.2 Semi-Universal Theories

Theorem 6.5 Let T be a semi-universal theory without function symbols and suppose that $\overline{P}$ is a tuple of predicate symbols occurring in T. Then $CIRC_D(T; \overline{P})$ is always reducible into first-order logic using the DLS^* algorithm. Moreover, if A is the resulting formula, then A implies $DCA^k(T)$, where k is the number of existential quantifiers of T.

<u>Proof</u>   Follows from Theorem 6.2 and Section 5.1. $\blacksquare$

The following theorem generalizes Theorem 6.4 to semi-universal theories.

Theorem 6.6 Let T be a semi-universal theory without function symbols. Let $\overline{P}$ be a tuple of predicate symbols and $\overline{c}$ be a tuple of individual constants occurring in T. Then $CIRC_D(T; \overline{P}; \overline{c})$ is always reducible into first-order logic using the DLS^* algorithm. Moreover, if A is the resulting formula, then A implies $DCA^{-\overline{c}+k}(T)$, where k is the number of existential quantifiers of T.

<u>Proof</u>   Since T is semi-universal, it can be be written in the form

$$\bigwedge_i \exists x_{i1} \cdots \exists x_{is_i} \forall y_{i1} \cdots \forall y_{it_i} T_i(\overline{\Psi}, \overline{\psi}) \tag{43}$$

16

where each T is an open formula. Let $c'_1, \cdots, c'_r$ be all individual constants occurring in T excluding those in $\overline{c}$. The second-order part of $CIRC_D(T; \overline{P}; \overline{c})$ is then equivalent to

$$\forall \overline{\psi} \forall \overline{\Psi} \forall \Phi [(\Phi(c'_1) \wedge \cdots \wedge \Phi(c'_r) \wedge \tag{44}$$
$$\bigwedge_i \exists x_{i1} \cdots \exists x_{is_i} \forall y_{i1} \cdots \forall y_{it_i} (\Phi(x_{i1}) \wedge \cdots \wedge \Phi(x_{is_i}) \wedge (\neg \Phi(y_{i1}) \vee \cdots \vee \neg \Phi(y_{it_i}) \vee T_i(\overline{\Psi}, \overline{\psi})))$$
$$\supset \forall z \ \Phi(z)].$$

To show the reduction, we first negate (44), obtaining

$$\exists \overline{\psi} \exists \overline{\Psi} \exists \Phi [(\Phi(c'_1) \wedge \cdots \wedge \Phi(c'_r) \wedge \tag{45}$$
$$\bigwedge_i \exists x_{i1} \cdots \exists x_{is_i} \forall y_{i1} \cdots \forall y_{it_i} (\Phi(x_{i1}) \wedge \cdots \wedge \Phi(x_{is_i}) \wedge (\neg \Phi(y_{i1}) \vee \cdots \vee \neg \Phi(y_{it_i}) \vee T_i(\overline{\Psi}, \overline{\psi})))$$
$$\wedge \exists z \neg \Phi(z)].$$

Formula (45) is transformed by the DLS algorithm into

$$\exists x_{11} \cdots \exists x_{1s_1} \cdots \exists \overline{\psi} \exists \overline{\Psi} \exists \Phi [\tag{46}$$
$$\forall u (\Phi(u) \vee (u \neq c'_1 \wedge \cdots \wedge u \neq c'_r \wedge \bigwedge_i (u \neq x_{i1} \wedge \cdots \wedge u \neq x_{is_i}))) \wedge$$
$$\bigwedge_i \forall y_{i1} \cdots \forall y_{it_i} (\neg \Phi(y_{i1}) \vee \cdots \vee \neg \Phi(y_{it_i}) \vee T_i(\overline{\Psi}, \overline{\psi})))$$
$$\wedge \exists z \neg \Phi(z)].$$

Let $B(u)$ denote formula $u \neq c'_1 \wedge \cdots \wedge u \neq c'_r \wedge \bigwedge_i (u \neq x_{i1} \wedge \cdots \wedge u \neq x_{is_i})$. Then the application of Ackermann's Lemma to formula (46) results in

$$\exists x_{11} \cdots \exists x_{1s_1} \cdots \exists \overline{\psi} \exists \overline{\Psi} [\bigwedge_i \forall y_{i1} \cdots \forall y_{it_i} (B(y_{i1}) \vee \cdots \vee B(y_{it_i}) \vee T_i(\overline{\Psi}, \overline{\psi}))) \wedge \exists z \ B(z)]. \tag{47}$$

Negating (47), we obtain

$$\forall x_{11} \cdots \forall x_{1s_1} \cdots \forall \overline{\psi} \forall \overline{\Psi} [\bigwedge_i \forall y_{i1} \cdots \forall y_{it_i} (B(y_{i1}) \vee \cdots \vee B(y_{it_i}) \vee T_i(\overline{\Psi}, \overline{\psi}))) \supset \forall z \neg B(z)]. \tag{48}$$

By (43), $T \wedge$ (48) implies

$$\exists x_{11} \cdots \exists x_{1s_1} \cdots \forall z \neg B(z), \tag{49}$$

where $\exists x_{11} \cdots \exists x_{1s_1} \cdots$ are all existential quantifiers of T. Substituting $B(z)$ in (49) by its definition, we obtain

$$\exists x_{11} \cdots \exists x_{1s_1} \cdots \forall z (z = c'_1 \vee \cdots \vee z = c'_r \vee \bigvee_i (z = x_{i1} \vee \cdots \vee z = x_{is_i})). \tag{50}$$

It can now easily be observed that (50) is equivalent to $DCA^{-\overline{c}+k}(T)$, where k is the number of existential quantifiers of T. Given this, the elimination of predicate variables corresponding to varying predicate and individual constants follows from our discussion in Section 5.1. ∎

A reduction of variable domain circumscription for a semi-universal theory is illustrated below.

Example 6.4 Let T consist of $\exists x \ Q(x) \wedge [Q(a) \supset \exists y \ y \neq a]$. This example is taken from [7]. The intended conclusion of domain circumscription with Q allowed to vary is $\exists y \forall x \ x = y \vee x = a$.

We reduce $CIRC_D(T;(Q);())$. The second-order part of $CIRC_D(T;(Q);())$ (after removing $\exists\Phi(x)$) is

$$\forall\Psi\forall\Phi[[\Phi(a) \wedge \exists x(\Phi(x) \wedge \Psi(x)) \wedge (\Psi(a) \supset \exists y(\Phi(y) \wedge y \neq a))] \supset \forall z\Phi(z)]. \tag{51}$$

Negating (51), we obtain

$$\exists\Psi\exists\Phi[\Phi(a) \wedge \exists x(\Phi(x) \wedge \Psi(x)) \wedge (\Psi(a) \supset \exists y(\Phi(y) \wedge y \neq a))] \wedge \exists z\neg\Phi(z),$$

which is equivalent to

$$\exists x\exists y\exists z\exists\Psi\exists\Phi[\Phi(a) \wedge \Phi(x) \wedge \Psi(x) \wedge (\neg\Psi(a) \vee (\Phi(y) \wedge y \neq a)) \wedge \neg\Phi(z)],$$

which is equivalent to

$$\exists x\exists y\exists z\exists\Psi\exists\Phi[\Phi(a) \wedge \Phi(x) \wedge \Psi(x) \wedge (\neg\Psi(a) \vee (\Phi(y) \wedge y \neq a)) \wedge \forall t(\neg\Phi(t) \vee t \neq z)]. \tag{52}$$

Applying Ackermann's Lemma to (52), we eliminate Φ and obtain

$$\exists x\exists y\exists z\exists\Psi[a \neq z \wedge x \neq z \wedge \Psi(x) \wedge (\neg\Psi(a) \vee (y \neq z \wedge y \neq a))],$$

which is equivalent to

$$\exists x\exists y\exists z\exists\Psi[a \neq z \wedge x \neq z \wedge \forall t(\Psi(t) \vee t \neq x) \wedge (\neg\Psi(a) \vee (y \neq z \wedge y \neq a))]. \tag{53}$$

Applying Ackermann's Lemma to (53), we eliminate Ψ and obtain

$$\exists x\exists y\exists z[a \neq z \wedge (a \neq x \vee (y \neq z \wedge y \neq a))]. \tag{54}$$

The negation of (54) is

$$\forall x\forall y\forall z[a = z \vee (a = x \wedge (y = z \vee y = a))].$$

Thus,

$$CIRC_D(T;(Q);()) \equiv T \wedge \forall x\forall y\forall z[a = z \vee (a = x \wedge (y = z \vee y = a))].$$

It is easily verified that $CIRC_D(T;(Q);())$ implies $\exists y\forall x\ x = y \vee x = a$.

6.2.3 Arbitrary Theories

For non-semi-universal theories without functions, neither the DLS algorithm nor its fixpoint generalization (G-DLS, for short) guarantee a reduction to classical first-order logic. However, the reduction can sometimes be obtained as observed in the following example:

Example 6.5 Let theory T consist of

A1. $\forall x\exists y\ S(x,y)$,
A2. $\forall x\forall y\ S(x,y) \supset x = y$.

18

Thus the second-order part of $CIRC_D(T)$ is equivalent to

$$\forall \Phi[(\exists x \Phi(x) \wedge \forall x(\Phi(x) \supset \exists y(\Phi(y) \wedge S(x,y)))) \supset \forall z\ \Phi(z)]. \tag{55}$$

In order to reduce (55) we first negate it,

$$\forall \Phi[(\exists x \Phi(x) \wedge \forall x(\Phi(x) \supset \exists y(\Phi(y) \wedge S(x,y)))) \wedge \exists z \neg \Phi(z)].$$

This formula is transformed to

$$\exists z \exists x \exists \Phi[\Phi(x) \wedge \forall x(\neg \Phi(x) \vee (z \neq x \wedge \exists y(\Phi(y) \wedge S(x,y))))].$$

Now we apply Theorem 5.2,

$$\exists z \exists x[\nu \Phi(x)(z \neq x \wedge \exists y(\Phi(y) \wedge S(x,y))))].$$

After negating this formula and making minor transformations, we get

$$\forall z \forall x[\mu \Phi(x)(z = x \vee \forall y(S(x,y) \supset \Phi(y))))].$$

As usual, we unfold the fixpoint formula,

$$\mu \Phi(x)(z = x \vee \forall y(S(x,y) \supset \Phi(y)))) \tag{56}$$

Let $\Gamma(\Phi(x)) = (z = x \vee \forall y(S(x,y) \supset \Phi(y))))$. Now,

$$\Gamma^0(\bot) \equiv \bot$$
$$\Gamma^1(\bot) \equiv z = x \vee \forall y(S(x,y) \supset \bot) \equiv (\text{by } A1)\ z = x$$
$$\Gamma^2(\bot) \equiv z = x \vee \forall y(S(x,y) \supset z = y).$$

We show that $\Gamma^1(\bot) \equiv \Gamma^2(\bot)$. It is easily observed that $\Gamma^1(\bot) \supset \Gamma^2(\bot)$. We thus have to show that $\Gamma^2(\bot) \supset \Gamma^1(\bot)$, i.e.

$$(z = x \vee \forall y(S(x,y) \supset z = y)) \supset z = x.$$

Suppose that $z \neq x$ and $\forall y(S(x,y) \supset z = y)$. Thus, substituting y by x we obtain that this formula implies $\neg S(x,x)$, which (using $A1$ and $A2$) implies $\bot$. Thus formula (56) reduces to $z = x$.

Formula (55) reduces now to $\forall z \forall x\ z = x$. ∎

6.3   Variable GDC with Functions

For theories containing function symbols, the DLS algorithm always fails. However, it is sometimes possible to reduce domain circumscription for such theories using G-DLS (see Examples 6.6 and 6.7). We also have the following theorems:

Theorem 6.7 Let T be a semi-universal theory. Then $CIRC_D(T)$ is always reducible into fixpoint logic using the G-DLS algorithm.

19

<u>Proof</u>   It can easily be observed that negation of the second-order part of $CIRC_D(T)$ consists of clauses containing at most one positive occurrence of Φ. Thus it can be transformed into the form required in Theorem 5.2, i.e. $CIRC_D(T)$ can be reduced to a fixpoint formula. ■

Theorem 6.8   Let T be a theory and let $\overline{P}$ be the tuple of all predicate symbols occurring in T. Suppose further that T has a $(\overline{P}, \overline{f})$-minimal model and cardinalities of all such models have the same finite common upper bound. Then $CIRC_D(T; \overline{P}; \overline{f})$ can be reduced into an equivalent first-order sentence.

<u>Proof</u>   According to Theorem 3.1, a formula A is a consequence of $CIRC_D(T; \overline{P}; \overline{f})$ iff A is true in all $(\overline{P}; \overline{f})$-minimal models of T. Since cardinalities of all such models have the same finite common upper bound, say n, we conclude that there is a sentence $S \equiv \exists x_1 \cdots \exists x_n \forall y(y = x_1 \vee \cdots \vee y = x_n)$ which is a consequence of $CIRC_D(T; \overline{P}; \overline{f})$. Since S is a formula of the form discussed in the end of Section 5.1, the domain circumscription axiom can be reduced by applying the DLS* algorithm to $T \cup \{S\}$. ■

Example 6.6 Let T consist of $\forall x \, f(x) = a \vee f(x) = b$. We reduce $CIRC_D(T)$. The corresponding domain circumscription axiom (after simplifications described in section 3.1) is

$$\forall \Phi[\Phi(a) \wedge \Phi(b) \wedge \forall x(\Phi(x) \supset \Phi(f(x)))] \supset \forall x \Phi(x). \tag{57}$$

After negating formula (57), we obtain

$$\exists \Phi[\Phi(a) \wedge \Phi(b) \wedge \forall x(\Phi(x) \supset \Phi(f(x)))] \wedge \exists x' \neg \Phi(x'), \tag{58}$$

which can be further transformed to

$$\exists x' \exists \Phi[\Phi(a) \wedge \Phi(b) \wedge \forall x(\neg \Phi(x) \vee (x \neq x' \wedge \Phi(f(x))))]. \tag{59}$$

Now one can apply Theorem 5.2. Observe that in this case the fixpoint operator takes the form

$$\nu \Phi(x).x \neq x' \wedge \Phi(f(x)).$$

After negating, the form

$$\mu \Phi(x).x = x' \vee \Phi(f(x))$$

is equivalent to

$$x = x' \vee (x' = x \vee x' = f(x)) \vee (x' = x \vee x' = f(x) \vee x' = f(f(x))) \vee$$
$$(x' = x \vee x' = f(x) \vee x' = f(f(x)) \vee x' = f(f(f(x))) \vee \cdots \tag{60}$$

Observe that T implies that $f(f(f(x)))$ does not introduce new values (since for any x, $f(x) = a \vee f(x) = b$). Thus the third disjunct of (60) is equivalent to the second one, given T. Now by applying Proposition 5.1 we have

$$T \models \mu \Phi(x).x = x' \vee \Phi(f(x)) \equiv (x' = x \vee x' = f(x) \vee x' = f(f(x))).$$

Thus, after performing routine calculations, we conclude that $CIRC_D(T)$ reduces to

$$T \wedge \forall x'[x' = a \vee x' = f(a) \vee x' = f(f(a)) \vee x' = b \vee x' = f(b) \vee x' = f(f(b))]. \tag{61}$$

■

Example 6.7 Let T consist of $\forall x \; x = f(f(x)) \wedge c \neq d$. Here $f(x)$ can be read "a spouse of x". This example is from [11]. We show a reduction of $CIRC_D(T)$. The corresponding circumscription axiom (after simplifications) is

$$\forall \Phi[\Phi(c) \wedge \Phi(d) \wedge \forall x(\Phi(x) \supset \Phi(f(x))) \supset \forall x \; \Phi(x)]. \tag{62}$$

After negating, we get

$$\exists \Phi[\Phi(c) \wedge \Phi(d) \wedge \forall x(\Phi(x) \supset \Phi(f(x))) \wedge \exists s \; \neg\Phi(s),] \tag{63}$$

which is equivalent to

$$\exists \Phi[\Phi(c) \wedge \Phi(d) \wedge \forall x(\neg\Phi(x) \vee (\Phi(f(x)) \wedge x \neq s)))]. \tag{64}$$

Now, we apply Theorem 5.2. Note that in this case the fixpoint operator takes the form

$$\nu \Phi(x).[x \neq s \wedge \Phi(f(x))].$$

After negating this, the form

$$\mu \Phi(x).[x = s \vee \Phi(f(x))]$$

is equivalent to

$$x = s \vee (x = s \vee f(x) = s) \vee (x = s \vee (f(x) = s \vee f(f(x)) = s)) \vee .... \tag{65}$$

Observe that given T, the third disjunct of (65) is equivalent to the second one. Now, applying Proposition 5.1 and performing straightforward calculations, we obtain

$$CIRC_D(T) \equiv T \wedge \forall s[c = s \vee f(c) = s \vee d = s \vee f(d) = s].$$

The above result is rather intuitive. The domain has to contain c and d, and all the results of applications of f. Since $f(f(x)) = x$, the second application of f does not give anything new. The minimal domain is then described by the resulting formula. ∎

7 Combining Domain Circumscription with Arbitrary Circumscriptions

Theorem 5.1 states that given a second-order theory T, if one can show that the domain closure axiom is entailed by T, then T is reducible to a first-order formula using DLS^*. There is a direct connection between this result and the reduction of arbitrary circumscriptive policies applied to a certain class of domain circumscribed theories. The connection works as follows:

1. We know that given a semi-universal theory T, the domain circumscription of T, $Circ_{DC}(T; \bar{P}; \bar{f})^6$, can be reduced to its first-order equivalent using DLS^*. In addition, the DCA used in the DLS^* algorithm can be constructively generated.

---

[6] $\bar{f}$ is restricted to individual constants.

2. Suppose the result of $Circ_{DC}(T; \bar{P}, \bar{f})$ is T'. Observe that for any arbitrary circumscription $Circ_{SO}(T'; \bar{P}; \bar{f})$, applied to T' that

$$Circ_{SO}(T'; \bar{P}; \bar{f}) \equiv Circ_{SO}(T' \wedge DCA; \bar{P}; \bar{Q}).$$

3. Since the DLS algorithm can only fail when unskolemization or non-separatedness occur, and we have shown how to avoid these problems for theories which entail the DCA, it follows that $Circ_{SO}(T'; \bar{P}; \bar{f})$ is always reducible to a first-order formula using DLS^* with the DCA.

In summary, we have the following result.

> For any semi-universal first-order theory without functions and any arbitrary circumscription policy applied to the theory, the DLS^* algorithm will always reduce the circumscribed theory to a logically equivalent first-order formula, provided that the theory is first circumscribed using domain circumscription.

The reduction process is achieved as follows.

1. Given a semi-universal theory T, constructively generate the DCA for T using the procedure described in the proof of Theorem 6.6.

2. Apply DLS^* to T resulting with the output T'.

3. Apply DLS^* to the arbitrary circumscriptive policy applied to T' using the previously generated DCA. This results in T'', a first-order formula logically equivalent to the latter arbitrary circumscription.

The following example illustrates the technique. To save space, we will first domain circumscribe the following theory and then apply an arbitrary circumscription to the original theory in conjunction with the generated DCA.

Example 7.1 *Let T consist of*

> *A1. $Ab(a)$*
> *A2. $\forall x \forall y Ab(x) \wedge S(y, x) \supset Ab(y)$*
> *A3. $\exists x \forall y S(y, x)$*

where Ab and S stand for Abnormal and Son-of, respectively.

The DCA entailed by the domain circumscription $CIRC_D(T)$ is[7]

$$\exists z \forall x (x = z \vee x = a). \tag{66}$$

(66) can be constructively generated using the procedure described in the proof of Theorem 6.6.

In the next phase, we would like to minimize the predicate Ab relative to $T \wedge DCA$. Let T' denote the first-order formula output by application of the DLS^* algorithm to $CIRC_D(T)$. Since $T' \models DCA$ and $CIRC_{SO}(T; Ab) \equiv CIRC_{SO}(T \wedge DCA; Ab)$, the DLS^* algorithm can be applied to $CIRC_{SO}(T; Ab)$ using the DCA with a guarantee that the output of DLS^* will be a formula in classical first-order logic, logically equivalent to $CIRC_{SO}(T; Ab) \wedge DCA$. In fact, the output of DLS^* is

$$\exists z [\quad \forall x (x = z \vee x = a)$$
$$\wedge \forall x (\neg Ab(x) \vee -S(z, x) \vee Ab(z)) \wedge \exists x \forall y S(y, x) \wedge Ab(a)$$
$$\wedge (S(z, a) \vee S(a, a) \vee \forall d \exists c \neg S(c, d) \vee \forall e(a = e \vee \neg Ab(e)))].$$

---

[7] In this example, by DCA, we mean DCA^{+1}.

References

[1] Ackermann, W. (1935) *Untersuchungen über das Eliminationsproblem der mathematischen Logik*, Mathematische Annalen, 110, 390-413.

[2] Davis, M. (1980) *The Mathematics of Non-Monotonic Reasoning*, Artificial Intelligence J., **13**, 73-80.

[3] Doherty, P., Lukaszewicz, W., Szałas, A. (1994) *Computing Circumscription Revisited. A Reduction Algorithm*, Technical Report LiTH-IDA-R-94-42, Linköping University, 1994. Also in, Proc. 14th IJCAI, 1995, Montreal, Canada.[8] Full version to appear in Journal of Automated Reasoning, 1996.

[4] Doherty, P., Lukaszewicz, W., Szałas, A. (1995) *A Reduction Result for Circumscribed Semi-Horn Formulas*, To appear in Fundamenta Informaticae, 1996.

[5] Etherington, D. W., Mercer, R. (1987) *Domain Circumscription: A Revaluation*, Computtational Intelligence, **3**, 94-99.

[6] Hintikka, J. (1988) *Model Minimization – An Alternative to Circumscription*, Journal of Automated Reasoning, **4**,1-13.

[7] Lorenz, S. (1994) *A Tableau Prover for Domain Minimization*, Journal of Automated Reasoning, **13**, 375-390.

[8] Lukaszewicz, W. (1990) *Non-Monotonic Reasoning - Formalization of Commonsense Reasoning*, Ellis Horwood Series in Artificial Intelligence. Ellis Horwood, 1990.

[9] McCarthy, J. (1977) *Epistemological Problems of Artificial Intelligence*, in: Proc. 5th IJCAI, Cambridge, MA, 1977, 1038-1044.

[10] Nonnengart, A., Szałas, A. (1995) *A Fixpoint Approach to Second-Order Quantifier Elimination with Applications to Correspondence Theory*, Report of Max-Planck-Institut für Informatik, MPI-I-95-2-007, Saarbrücken, Germany.

[11] Suchanek, M. A. (1993) *First-Order Syntactic Characterizations of Minimal Entailment, Domain-Minimal Entailment, and Herbrand Entailment*, Journal of Automated Reasoning, **10**, 237-263.

[12] Szałas, A. (1993) *On the Correspondence Between Modal and Classical Logic: an Automated Approach*, Journal of Logic and Computation, **3**, 605-620.

---

[8]URL: http://www.ida.liu.se/labs/rkllab/people/patdo/.