

# General Domain Circumscription and its First-Order Reduction

Patrick Doherty<sup>1\*</sup>, Witold Lukaszewicz<sup>2\*\*</sup>, Andrzej Szalas<sup>2\*\*\*</sup>

<sup>1</sup> Department of Computer and Information Science,  
Linköping University, S-581 83 Linköping, Sweden,  
e-mail: patdo@ida.liu.se

<sup>2</sup> Institute of Informatics, Warsaw University,  
ul. Banacha 2, 02-097 Warsaw, Poland,  
e-mail: witlu,szalas@mimuw.edu.pl.

**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 isolate a class of domain circumscribed theories, such 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.

---

\* Supported in part by the Swedish Council for Engineering Sciences (TFR).

\*\* Supported in part by KBN grant 3 P406 019 06.

\*\*\* Supported in part by KBN grant 3 P406 019 06.

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 [11]. 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 ([10], [2], [6], [7], and [8]). 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 [10] and Hintikka's mini-consequence [7] 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 its 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 its 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.

We refer the reader to [5], for an extended version of this paper which includes all proofs, additional methods for reduction based on fixpoint methods, and additional examples.

## 2 Preliminaries

In this paper, the term *theory* always refers to a finite set of sentences of first-order logic. Since each such 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  $\lambda\bar{x}. A(\bar{x})$ , where  $\bar{x}$  is a tuple of  $n$  individual variables and  $A(\bar{x})$  is any formula of first-order classical logic. If  $U$  is an  $n$ -ary predicate expression of the form  $\lambda\bar{x}. A(\bar{x})$  and  $\bar{\alpha}$  is a tuple of  $n$  terms, then  $U(\bar{\alpha})$  stands for  $A(\bar{\alpha})$ . As usual, a predicate constant  $P$  is identified with the predicate expression  $\lambda\bar{x}. P(\bar{x})$ . Similarly, a predicate variable  $\Phi$  is identified with the predicate expression  $\lambda\bar{x}. \Phi(\bar{x})$ .

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

Let  $\bar{U} = (U_1, \dots, U_n)$  and  $\bar{V} = (V_1, \dots, V_n)$  (resp.  $\bar{u} = (u_1, \dots, u_n)$  and  $\bar{v} = (v_1, \dots, v_n)$ ) be tuples of predicate (resp. function) expressions.  $\bar{U}$  and  $\bar{V}$  (resp.  $\bar{u}$  and  $\bar{v}$ ) 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  $\perp$ , respectively.

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

If  $A$  is a formula,  $\bar{\sigma} = (\sigma_1, \dots, \sigma_n)$  and  $\bar{\delta} = (\delta_1, \dots, \delta_n)$  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  $\sigma_i$  by  $\delta_i$  ( $1 \leq i \leq n$ ). For any tuple  $\bar{x} = (x_1, \dots, x_n)$  of individual variables and any tuple  $\bar{t} = (t_1, \dots, t_n)$  of terms, we write  $\bar{x} = \bar{t}$  to denote the formula  $x_1 = t_1 \wedge \dots \wedge x_n = t_n$ . We write  $\bar{x} \neq \bar{t}$  as an abbreviation for  $\neg(\bar{x} = \bar{t})$ .

## 2.2 Definitions

**Definition 1.** A theory  $T$  is said to be *existential (universal)* iff all of its axioms are of the form  $\exists\bar{x} T_1$  (resp.  $\forall\bar{x} T_1$ ), where  $T_1$  is quantifier free.

**Definition 2.** A theory is called *semi-universal* if its axioms do not contain existential quantifiers in the scope of universal quantifiers.

**Definition 3.** Let  $T$  be a theory without function symbols and suppose that, for  $n > 0$ ,  $c_1, \dots, c_n$  are all the individual constants occurring in  $T$ . The *domain closure axiom for T*, written  $DCA(T)$ , is the sentence

$$\forall x. x = c_1 \vee \dots \vee x = c_n.$$

Let  $\bar{c}$  be a tuple of individual constants. By  $DCA^{-\bar{c}}(T)$  we shall denote the sentence  $\forall x x = c_1 \vee \dots \vee x = c_n$ , where  $c_1, \dots, c_n$  are all the individual constants

of  $T$  excluding constants from  $\bar{c}$ . For  $k \in \omega$ , by  $DCA^{+k}(T)$  we shall denote the sentence

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

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

**Definition 4.** A predicate variable  $\Phi$  occurs *positively* (resp. *negatively*) in a formula  $A$  if the conjunctive normal form of  $A$  contains a subformula of the form  $\Phi(\bar{t})$  (resp.  $\neg\Phi(\bar{t})$ ). A formula  $A$  is said to be *positive* (resp. *negative*) w.r.t.  $\Phi$  iff all occurrences of  $\Phi$  in  $A$  are positive (resp. negative).

**Definition 5.** Let  $\Phi$  be either a predicate constant or a predicate variable and  $\bar{\Phi}$  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.  $\Phi$  iff it is of the form  $T_1(\Phi) \wedge T_2(\Phi)$  where  $T_1(\Phi)$  is positive w.r.t.  $\Phi$  and  $T_2$  is negative w.r.t.  $\Phi$ .

### 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 [10], studied in [2], and substantially improved in [6], and Hintikka's mini-consequence, formulated in [7] and studied in [8].

**Definition 6.** Let  $\bar{P} = (P_1, \dots, P_n)$  be a tuple of different predicate constants,  $\bar{f} = (f_1, \dots, f_k)$  be a tuple of different function constants (including, perhaps, individual constants),  $T(\bar{P}, \bar{f})$  be a theory and let  $\Phi$  be a one-place predicate variable,  $\bar{\Psi}$  be a tuple of predicate variables similar to  $\bar{P}$ , and  $\bar{\psi}$  be a tuple of function variables similar to  $\bar{f}$ . By  $Axiom(\Phi, \bar{P}, \bar{f})$ , sometimes abbreviated by  $Axiom(\Phi)$ , we shall mean the conjunction of:

- $\Phi(a)$ , for each individual constant  $a$  in  $T$  not occurring in  $\bar{f}$ ,
- $\Phi(\psi_i)$ , for each individual constant  $a$  in  $T$  such that  $a$  is  $f_i$ ,
- $\forall x_1 \dots x_n [\Phi(x_1) \wedge \cdots \wedge \Phi(x_n) \supset \Phi(f(x_1, \dots, x_n))]$ , for each  $n$ -ary ( $n > 0$ ) function constant  $f$  in  $T$  not occurring in  $\bar{f}$ , and
- $\forall x_1 \dots x_n [\Phi(x_1) \wedge \cdots \wedge \Phi(x_n) \supset \Phi(\psi_i(x_1, \dots, x_n))]$ , for each  $n$ -ary ( $n > 0$ ) function constant  $f$  in  $T$  such that  $f$  is  $f_i$ .

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

**Definition 7.** Let  $\bar{P} = (P_1, \dots, P_n)$ ,  $\bar{f} = (f_1, \dots, f_k)$  and  $T(\bar{P}, \bar{f})$  be as in Definition 6. The *general domain circumscription for  $T(\bar{P}, \bar{f})$  with variable  $\bar{P}$  and  $\bar{f}$* , written  $CIRC_D(T; \bar{P}; \bar{f})$ , is the following sentence of second-order logic:

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

A formula  $\alpha$  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  $\Phi$ ) 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 [10] with the augmentation described in [6].<sup>3</sup>

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 [7] and improved in [8]. Following [8], this form of minimization will be referred to as *variable domain circumscription*.<sup>4</sup>

*Example 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

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

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

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

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 2.* Let  $T$  consist of  $P(a) \wedge P(b)$ . We minimize the domain of  $T$  with the constant  $a$  allowed to vary during the minimization.  $CIRC_D(T; (), (a))$  is given by<sup>5</sup>

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

Substituting  $\lambda x. x = b$  for  $\Phi$  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  $b$ .

<sup>3</sup> 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.

<sup>4</sup> Note that in variable domain circumscription all predicate constants, but no function constants, are allowed to vary during the minimization process.

<sup>5</sup> 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  $\psi$ .

We now proceed to give a semantics for general domain circumscription. Given a model  $M$ , we shall write  $|M|$  to denote the domain of  $M$ .

**Definition 8.** Let  $\overline{P}$ ,  $\overline{f}$  and  $T(\overline{P}, \overline{f})$  be as in Definition 6. 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  $M_1 \leq M_2$ , iff  $|M_1| \subseteq |M_2|$ , 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  $|M_1|$ . A model is said to be  $(\overline{P}; \overline{f})$ -minimal iff it has no proper  $(\overline{P}; \overline{f})$ -submodels.

**Theorem 9.** Let  $\overline{P}$ ,  $\overline{f}$  and  $T(\overline{P}, \overline{f})$  be as in Definition 6. 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  $T$ .

### 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  $\forall x_1 \cdots \forall x_n A(x_1, \dots, x_n)$ , and suppose that all predicate and/or function constants occurring in  $A(x_1, \dots, x_n)$  are not allowed to vary during the minimization. Then each such axiom reappears in (1) as a part of  $T^\phi$ , equivalent to the formula

$$\forall x_1 \cdots \forall x_n \neg \Phi(x_1) \vee \cdots \vee \neg \Phi(x_n) \vee A(x_1, \dots, x_n). \quad (5)$$

Since (5), together with the corresponding axiom of  $T$ , reduces to  $\top$ , it can be removed from  $T^\phi$ . We thus have the following principle:

Remove a counterpart of every universal axiom  $A$  of  $T$  from  $T^\phi$  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 [13], 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 [13].

**Lemma 10 Ackermann Lemma.** *Let  $\Phi$  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.  $\Phi$  and let  $A$  contain no occurrences of  $\Phi$  at all. Then the following equivalences hold:*

$$\exists\Phi\forall\bar{x}[\Phi(\bar{x}) \vee A(\bar{x}, \bar{z})] \wedge B(\Phi \leftarrow \neg\Phi) \equiv B(\Phi \leftarrow A(\bar{x}, \bar{z})) \quad (6)$$

$$\exists\Phi\forall\bar{x}[\neg\Phi(\bar{x}) \vee A(\bar{x}, \bar{z})] \wedge B(\Phi) \equiv B(\Phi \leftarrow A(\bar{x}, \bar{z})) \quad (7)$$

where in the righthand formulas the arguments  $\bar{x}$  of  $A$  are each time substituted by the respective actual arguments of  $\Phi$  (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 10 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,

$$\forall\bar{x}\exists y A(\bar{x}, y) \equiv \exists f\forall\bar{x} A(\bar{x}, y \leftarrow f(\bar{x})), \quad (8)$$

where  $f$  is a new function variable. After application of Ackermann's Lemma, one tries to remove the newly introduced function variables using equivalence (8) 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 (10) 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.

Due to page limitations, we will concentrate on the first method whose formal justification is described in Section 5.1. We refer the reader to [5], for a detailed description of both methods.

### 5.1 DLS Algorithm with the Domain Closure Axiom

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

$$\exists x A(x) \equiv (A(c_1) \vee \cdots \vee A(c_n)) \quad (9)$$

and

$$\forall x A(x) \equiv (A(c_1) \wedge \cdots \wedge A(c_n)). \quad (10)$$

The following example illustrates the use of equivalences (9) and (10).

*Example 3.* Assume that  $\forall x x = a \vee x = b$  holds. An application of equivalence (9) to formula  $\forall y \exists z P(y, z)$  results in  $\forall y (P(y, a) \vee P(y, b))$ . An application of equivalence (10) to this formula results in  $(P(a, a) \vee P(a, b)) \wedge (P(b, a) \vee P(b, b))$ .

Using equivalence (9) 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 (10) 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  $\Phi$ . In the canonical case, this occurs when a universally quantified clause contains both positive and negative occurrences of  $\Phi$ . Using equivalence (10), 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  $DCA^{-\bar{c}}(T)$ ,  $DCA^{+k}(T)$  or  $DCA^{-\bar{c}+k}(T)$  is entailed from  $T$ . Before we introduce this modification, consider the following simple example.

*Example 4.* Assume that  $\exists z \forall x x = z \vee x = a$  holds. An application of equivalence (10) 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 (3), 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 11.** *Assume that for a given second-order theory  $T$ ,*

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

*Then  $T$  is equivalent to a first-order formula.*

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

**Corollary 12.** *Assume that  $CIRC_D(T, \bar{P}, \bar{f}) \models DCA^{-\bar{c}+k}(T)$ . Then  $CIRC_D(T, \bar{P}, \bar{f})$  is equivalent to a first-order formula.*

Theorem 11 allows us to modify the DLS algorithm in such a way that whenever there is a Skolemization or separatedness problem, one applies formulas (9) and (10), or their generalizations, respectively. We will denote this modification of the DLS algorithm by *DLS\**.

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

*Example 5.* 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

$$\exists \Phi \forall x \forall y [(\Phi(x) \vee \neg \Phi(y) \vee R(a)) \wedge \exists z \neg \Phi(z) \wedge \exists u \Phi(u)] \wedge (\exists z \forall x x = z \vee x = a). \quad (11)$$

We first Skolemize *DCA* and obtain  $\forall x (x = b \vee x = a)$ . We then try to eliminate the quantifier  $\exists \Phi$  from formula (11) 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 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 (9) and obtain

$$\exists b [DCA \wedge \exists \Phi \forall x \forall y ((\Phi(x) \vee \neg \Phi(y) \vee R(a)) \wedge \exists z \neg \Phi(z) \wedge (\Phi(b) \vee \Phi(a)))]. \quad (12)$$

Formula (12) is not separated. We thus apply equivalence (10) to quantifier  $\forall y$  and obtain:

$$\begin{aligned} \exists b[DCA \wedge \exists \Phi \forall x((\Phi(x) \vee \neg \Phi(b) \vee R(a)) \wedge (\Phi(x) \vee \neg \Phi(a) \vee R(a))) \\ \wedge \exists z \neg \Phi(z) \wedge (\Phi(b) \vee \Phi(a))]. \end{aligned} \quad (13)$$

(13) is equivalent to

$$\begin{aligned} \exists b[DCA \wedge [\exists \Phi(\exists z \neg \Phi(z) \wedge \forall x \Phi(x) \wedge (\Phi(b) \vee \Phi(a)))] \vee \\ [\exists \Phi(\exists z \neg \Phi(z) \wedge (\neg \Phi(b) \vee R(a)) \wedge (\neg \Phi(a) \vee R(a)) \wedge (\Phi(b) \vee \Phi(a)))] \end{aligned} \quad (14)$$

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 (14).

## 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

**Universal Theories** In Example 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 13.** *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  $T \wedge DCA(T)$ .*

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  $CIRC_D(T) \equiv T \wedge \forall x \forall z (x = z)$ .

**Semi-Universal Theories** As regards semi-universal theories without function symbols we have the following theorem:

**Theorem 14.** *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  $T$ .*

## 6.2 Variable GDC

For universal and semi-universal theories, we have the following counterparts of Theorems 13 and 14:

### Universal Theories

**Theorem 15.** *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  $DCA(T)$ .*

For theories with varied individual constants the following theorem holds.

**Theorem 16.** *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  $DCA^{-\overline{c}}(T)$ .*

The following example varies an individual constant.

*Example 6.* 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 [12]. 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 try 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 fixed, 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

$$\begin{aligned} \forall x_c \forall \Phi [\Phi(x_c) \wedge \Phi(d) \wedge S(x_c) \wedge \forall x (\Phi(x) \supset ((\neg R(x, x_c) \vee R(x, d)) \\ \wedge (R(x, x_c) \vee \neg R(x, d))) \wedge \forall x (\Phi(x) \supset \neg R(x, x_c)) \supset \forall s \Phi(s)]. \end{aligned} \quad (15)$$

Negating (15), we obtain

$$\begin{aligned} \exists x_c \exists \Phi [\Phi(x_c) \wedge \Phi(d) \wedge S(x_c) \wedge \forall x (\neg \Phi(x) \vee \neg R(x, x_c) \vee R(x, d)) \wedge \\ (\neg \Phi(x) \vee R(x, x_c) \vee \neg R(x, d)) \wedge \forall x (\neg \Phi(x) \vee \neg R(x, x_c)) \wedge \exists s \neg \Phi(s)] \end{aligned} \quad (16)$$

which is transformed by the DLS algorithm to

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

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

$$\exists s \exists x_c [S(x_c) \wedge \forall x ((x \neq x_c \wedge x \neq d) \vee \neg R(x, x_c)) \wedge (s \neq x_c \wedge s \neq d)]. \quad (17)$$

Negating (17), we obtain

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

and so,  $Circ_D(T, (), (c)) \equiv$

$$T \wedge \forall s \forall x_c [S(x_c) \wedge \forall x ((x \neq x_c \wedge x \neq d) \vee \neg R(x, x_c)) \supset (s \neq x_c \wedge s \neq d)]$$

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

### Semi-Universal Theories

**Theorem 17.** *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  $T$ .*

The following theorem generalizes Theorem 16 to semi-universal theories.

**Theorem 18.** *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  $T$ .*

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

*Example 7.* Let  $T$  consist of  $\exists x Q(x) \wedge [Q(a) \supset \exists y y \neq a]$ . This example is taken from [8]. 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  $\exists x \Phi(x)$ ) is

$$\forall \Psi \forall \Phi [(\Phi(a) \wedge \exists x (\Phi(x) \wedge \Psi(x)) \wedge (\Psi(a) \supset \exists y (\Phi(y) \wedge y \neq a))) \supset \forall z \Phi(z)]. \quad (18)$$

Negating (18), 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

$$\exists x \exists y \exists z \exists \Psi \exists \Phi [\Phi(a) \wedge \Phi(x) \wedge \Psi(x) \wedge (\neg \Psi(a) \vee (\Phi(y) \wedge y \neq a)) \wedge \forall t (\neg \Phi(t) \vee t \neq z)]. \quad (19)$$

Applying Ackermann's Lemma to (19), we eliminate  $\Phi$  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

$$\exists x \exists y \exists z \exists \Psi [a \neq z \wedge x \neq z \wedge \forall t (\Psi(t) \vee t \neq x) \wedge (\neg \Psi(a) \vee (y \neq z \wedge y \neq a))]. \quad (20)$$

Applying Ackermann's Lemma to (20), we eliminate  $\Psi$  and obtain

$$\exists x \exists y \exists z [a \neq z \wedge (a \neq x \vee (y \neq z \wedge y \neq a))]. \quad (21)$$

The negation of (21) 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  $\exists y \forall x x = y \vee x = a$ .

### 6.3 Arbitrary Theories and Functions

A fixpoint generalization of the basic DLS algorithm ( $DLS^{\text{fix}}$ ) is described in [11], applied in [4] to the class of semi-Horn formulas, and applied in [5] to general domain circumscription. Due to page limitations, we can only briefly describe the latter results:

- For arbitrary domain circumscribed theories without functions, neither the DLS algorithm nor its fixpoint generalization  $DLS^{\text{fix}}$ , guarantee a reduction to classical first-order logic. However reductions can sometimes be obtained using  $DLS^{\text{fix}}$ .
- For arbitrary domain circumscribed theories with functions, the DLS algorithm always fails. However, the following results apply.

**Theorem 19.** *Let  $T$  be a semi-universal theory. Then  $CIRC_D(T)$  is always reducible into fixpoint logic using the  $DLS^{\text{fix}}$  algorithm.*

If one can show that the fixpoint formula output by the  $DLS^{\text{fix}}$  algorithm is bounded, then the input formula is reducible to classical first-order logic.

**Theorem 20.** *Let  $T$  be a theory and let  $\bar{P}$  be the tuple of all predicate symbols occurring in  $T$ . Suppose further that  $T$  has a  $(\bar{P}, \bar{f})$ -minimal model and cardinalities of all such models have the same finite common upper bound. Then  $CIRC_D(T; \bar{P}; \bar{f})$  can be reduced into an equivalent first-order sentence using the  $DLS^*$  algorithm.*

## 7 Combining Domain Circumscription with Arbitrary Circumscriptions

Theorem 11 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  $T$ ,  $Circ_{DC}(T; \bar{P}; \bar{f})$ <sup>6</sup>, can be reduced to its first-order equivalent using  $DLS^*$ . In addition, the DCA used in the  $DLS^*$  algorithm can be constructively generated.
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

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

---

<sup>6</sup>  $\bar{f}$  is restricted to individual constants.

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  $Circ_{SO}(T'; \bar{P}; f)$  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 18.
2. Apply  $DLS^*$  to  $T$  resulting in 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 a particular circumscription to the original theory in conjunction with the generated DCA.

*Example 8.* 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<sup>7</sup>

$$\exists z \forall x (x = z \vee x = a) \tag{22}$$

(22) can be constructively generated using the procedure described in the proof of Theorem 18.

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

$$\begin{aligned} & \exists z [ \forall x (x = z \vee x = a) \\ & \quad \wedge \forall x (\neg Ab(x) \vee \neg S(z, x) \vee Ab(z)) \wedge \exists x \forall y S(y, x) \wedge Ab(a) \\ & \quad \wedge (S(z, a) \vee S(a, a) \vee \forall d \exists c \neg S(c, d) \vee \forall e (a = e \vee \neg Ab(e))) ]. \end{aligned}$$

<sup>7</sup> 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., Łukaszewicz, W., Szalas, 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.<sup>8</sup> Full version to appear in Journal of Automated Reasoning.<sup>9</sup>
4. Doherty, P., Łukaszewicz, W., Szalas, A. (1995) *A Reduction Result for Circumscribed Semi-Horn Formulas*, To appear in Fundamenta Informaticae, 1996.
5. Doherty, P., Łukaszewicz, W., Szalas, A. (1995) *General Domain Circumscription and its First-Order Reduction*, Technical Report LiTH-IDA-R-96-01, Linköping University.
6. Etherington, D. W., Mercer, R. (1987) *Domain Circumscription: A Reevaluation*, Computational Intelligence, **3**, 94-99.
7. Hintikka, J. (1988) *Model Minimization – An Alternative to Circumscription*, Journal of Automated Reasoning, **4**,1-13.
8. Lorenz, S. (1994) *A Tableau Prover for Domain Minimization*, Journal of Automated Reasoning, **13**, 375-390.
9. Łukaszewicz, W. (1990) *Non-Monotonic Reasoning - Formalization of Commonsense Reasoning*, Ellis Horwood Series in Artificial Intelligence. Ellis Horwood, 1990.
10. McCarthy, J. (1977) *Epistemological Problems of Artificial Intelligence*, in: Proc. 5th IJCAI, Cambridge, MA, 1977, 1038-1044.
11. Nonnengart, A., Szalas, 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.
12. Suchanek, M. A. (1993) *First-Order Syntactic Characterizations of Minimal Entailment, Domain-Minimal Entailment, and Herbrand Entailment*, Journal of Automated Reasoning, **10**, 237-263.
13. Szalas, A. (1993) *On the Correspondence Between Modal and Classical Logic: an Automated Approach*, Journal of Logic and Computation, **3**, 605-620.

This article was processed using the L<sup>A</sup>T<sub>E</sub>X macro package with LLNCS style

---

<sup>8</sup> To try out the algorithm: <http://www.ida.liu.se/labs/kplab/projects/dls/> .

<sup>9</sup> <ftp://ftp.ida.liu.se/pub/labs/kplab/people/patdo/jal-final.ps.gz>.