Reasoning with Qualitative Preferences and Cardinalities using Generalized Circumscription*

Patrick Doherty†
†Department of Computer Science
Linköping University
S-581 83 Linköping, Sweden
email: patdo@ida.liu.se

Andrzej Szałas†,‡
‡Institute of Informatics
University of Warsaw
02-097 Warsaw, Banacha 2, Poland
email: andsz@mimuw.edu.pl

Abstract

The topic of preference modeling has recently attracted the interest of a number of sub-disciplines in artificial intelligence such as the nonmonotonic reasoning and action and change communities. The approach in these communities focuses on qualitative preferences and preference models which provide more natural representations from a commonsense perspective. In this paper, we show how generalized circumscription can be used as a highly expressive framework for qualitative preference modeling. Generalized circumscription proposed by Lifschitz allows for predicates (and thus formulas) to be minimized relative to arbitrary pre-orders (reflexive and transitive). Although it has received little attention, we show how it may be used to model and reason about elaborate qualitative preference relations. One of the perceived weaknesses with any type of circumscription is the 2nd-order nature of the representation. The paper shows how a large variety of preference theories represented using generalized circumscription can in fact be reduced to logically equivalent first-order theories in a constructive way. Finally, we also show how preference relations represented using general circumscription can be extended with cardinality constraints and when these extensions can also be reduced to logically equivalent first-order theories.

Introduction

Reasoning about preferences is ubiquitous in almost all aspects of life. Humans (and artificial agents) are continually faced with alternatives. Contextual preferences for choosing among alternatives often guide decision making and action. Modeling and reasoning about such preferences is an inherent and important part of commonsense reasoning. Classical decision theory focuses on numeric preference models, but it is often difficult in commonsense scenarios to acquire the appropriate numeric preferences and weights for maximizing the utility of a decision based on such weighted alternatives. Recently, preference modeling has attracted the interest of many sub-disciplines in artificial intelligence (Junker *et al.* 2004) such as the action and change, nonmonotonic reasoning and constraint programming communities. The approach in these communities focuses on qualitative preferences and preference models which provide more natural representations from a commonsense perspective.

In logic-based AI, a common approach is to take an existing nonmonotonic logic such as default logic or logic programming with answer set semantics and extend these formalisms with preferences in one form or another. Preferences may be placed on individuals, formulas, rules, sets of objects, etc. In the case of default logic, preferences are placed on rules, either in the meta language (Brewka & Eiter 2000; Delgrande & Schaub 2000) or in the object language by adding names for default rules. In logic programs with ordered disjunction (Brewka, Niemelä, & Syrjänen 2004; Brewka, Niemelä, & Truszczynski 2005), preferences are placed on literals in the heads of rules which induce a preferred ordering on answer sets. The intuition is that commonsense preferences on individuals or sets specified in the object language of a formalism induce a preferred ordering on models, answer sets, extensions, etc. This in turn constrains the logical consequences of the formalism to satisfy the specified preference relations characterized in the theories in question. Much recent work has investigated qualitative preferences using logic programming with answer set semantics and default logics as the base nonmonotonic formalisms. Preferences on sets, in various forms, have also been considered in (Brafman *et al.* 2005; Chomicki & Zhang 2008; Guha *et al.* 2003; desJardins & Wagstaff 2005).

Circumscription is another nonmonotonic formalism well-suited to the study and development of a general representational framework for qualitative preferences, but much less effort has been expended in investigating its use as such a framework (but see, e.g., (Satoh 1996; Wakaki, Satoh, & Nitta 1997), where prioritized circumscription is employed to model preferences between rules). This is surprising since it has a number of features which make it a natural candidate for such a study. To name a few, it is a classical formalism framed in first- and second-order logic; preferences are generally placed on predicates which means implicitly that we can define preference orderings on individuals, atoms, formulas and sets. These orderings in turn induce preferences on models for the theories in question. Most importantly,

*Supported in part by the MNiSW grant N N206 399134 and SSF Excellence Center MOVIII.

there is a rather straightforward integration of object and meta-level concepts in the object language due to the use of 2nd-order logic.

In fact, we will target one of the least investigated forms of circumscription, generalized circumscription (Lifschitz 1984), which has an additional and essential feature in this context, the ability to define arbitrary pre-orders over predicates in the object language. Generalized circumscription modifies the standard definition of predicate circumscription (McCarthy 1980; 1986; Lifschitz 1991; Łukaszewicz 1990). Instead of preferring predicates w.r.t. set inclusion, one focuses on a general notion of preference (pre-orders) which may be used to great advantage in encoding quite complex qualitative preferences.

One of the misconceptions about various forms of circumscription is that in general, they are not amenable to practical use due to the second-order nature of the formalisms. In fact, it is often the case that restricted, but quite general uses of circumscription allow for the constructive reduction of such second-order theories to logically equivalent first-order theories (Doherty, Łukaszewicz, & Szałas 1997; 1996; 1998). We will show this for the combined case of preference modeling and generalized circumscription.

The structure of the paper is as follows. We first present generalized circumscription and propose a methodology for representing preference models which uses primary preference constraints and cardinality constraints. We then present a quantifier elimination algorithm which can in some cases reduce second-order generalized circumscription theories to first-order equivalents. In these cases, we place a number of restrictions on our preference theories. In a subsequent section, we extend the DLS algorithm to relax these restrictions. In fact, the DLS and DLS extension results are stand alone results and may be used for any applications of generalized circumscription. Examples are used throughout to demonstrate the methodology and reduction techniques.

Generalized Circumscription and Preference and Cardinality Constraints

First-order preferences are preferences on objects of an underlying domain while second-order preferences are preferences on sets (relations). For example, consider a universe consisting of writers. One might prefer Dickens to Shaw, which is an example of a first-order preference. On the other hand, one might prefer crime novels to thrillers. In such a case we express a preference on collections, i.e., sets of objects, which is then a second-order preference.

We consider preference relations $\preceq$ with the intuitive meaning that $A \preceq B$ denotes the fact that B is "not worse than" A. It is then natural to assume that $\preceq$ is reflexive and transitive. This is common, e.g., to applications in economics, where traditionally preorders are considered rather than partial orders (see, e.g., (Doyle 2004)), since weak antisymmetry can, in many situations, be violated. For example, consider preferences $\leq$ on decisions, where a decision is preferred if it is less risky. Then $d_1 \leq d_2$ and $d_2 \leq d_1$ means that decisions d_1 and d_2 are equally risky, but not that $d_1 = d_2$.

Definition 1 By a *preference relation* over a set of relations $\mathcal{R}$ we understand a binary reflexive and transitive relation $\preceq$ on $\mathcal{R}$. $\lhd$

Generalized circumscription has been proposed by Lifschitz (Lifschitz 1984). For our purposes, we provide a simplified version (which is also generalized in the sense that we allow relations of third-order, too).

Let $\mathcal{L}$ be a fixed first-order language with equality. We write $T(P_1, \ldots, P_n)$ to indicate that some (but not necessarily all) of the relation symbols occurring in T are among $P_1, \ldots, P_n$. We consider finite theories only.

In what follows we will always assume that any orderings considered, including $\preceq$, are pre-orders. They are reflexive and transitive, i.e., satisfy:

– $\forall X[X \preceq X]$
– $\forall X \forall Y \forall Z\big[(X \preceq Y \wedge Y \preceq Z) \rightarrow X \preceq Z\big]$.

Unlike standard forms of circumscription, we will often maximize rather then minimize preferences, so in such cases the ordering sign $\succeq$ is defined by

$$x \succeq y \stackrel{\mathrm{def}}{=} y \preceq x. \tag{1}$$

Whenever we use a symbol representing preferences (such as $\preceq, \leq$), we also allow typical associated symbols (such as $\prec, \succ, \succeq <, >, \geq$) defined by $x \prec y \stackrel{\mathrm{def}}{=} x \preceq y \wedge \neg(y \preceq x)$, $x \succ y \stackrel{\mathrm{def}}{=} x \succeq y \wedge \neg(y \succeq x)$, and similarly for other ordering symbols.

Definition 2 Let $\bar{P} = \langle P_1, \ldots, P_n \rangle$ be a tuple of distinct relation symbols, $\bar{S} = \langle S_1, \ldots, S_m \rangle$ be a tuple of distinct relation symbols disjoint with $\bar{P}$, $\preceq$ be a pre-order[1] and let $T(\bar{P}, \bar{S})$ be a theory. The *generalized circumscription* of $\bar{P}$ *in* $T(\bar{P}, \bar{S})$ *with* $\bar{S}$ varied w.r.t. $\preceq$, denoted by $\mathrm{CIRC}_{\preceq}(T; \bar{P}; \bar{S})$, is the sentence

$$T(\bar{P}, \bar{S}) \wedge \forall \bar{X} \forall \bar{Y}\big\{[T(\bar{X}, \bar{Y}) \wedge \bar{X} \preceq \bar{P}] \rightarrow \bar{P} \preceq \bar{X}\big\} \tag{2}$$

where $\bar{X} = \langle X_1, \ldots, X_n \rangle$ and $\bar{Y} = \langle Y_1, \ldots, Y_m \rangle$ are tuples of relation variables similar to $\bar{P}$ and $\bar{S}$, respectively.[2] $\lhd$

Remark 3 Observe that in Definition 2 we minimize predicates according to a defined relation $\preceq$. As motivated earlier, we often maximize predicates in order to reflect the assumed preferences. We shall then sometimes use $\succeq$, defined by (1) rather than $\preceq$. In such cases the sentence

$$T(\bar{P}, \bar{S}) \wedge \forall \bar{X} \forall \bar{Y}\big\{[T(\bar{X}, \bar{Y}) \wedge \bar{X} \succeq \bar{P}] \rightarrow \bar{P} \succeq \bar{X}\big\} \tag{3}$$

is denoted by $\mathrm{CIRC}_{\succeq}(T; \bar{P}; \bar{S})$. $\lhd$

[1] Writing $P \preceq R$ we require that P and R are of the same arities. Formally for each arity n we need a separate symbol $\preceq_n$. However, in order to simplify notation we omit the subscript n which will always be known from the context.

[2] $T(\bar{X}, \bar{Y})$ is the sentence obtained from $T(\bar{P}, \bar{S})$ by replacing all occurrences of $P_1, \ldots, P_n$ by $X_1, \ldots, X_n$, respectively, and all occurrences of $S_1, \ldots, S_m$ by $Y_1, \ldots, Y_m$, respectively.

We will call $\preceq$ in Definition 2, a *primary preference constraint*. Observe that, in general, it is a third-order relation, as it takes relations as arguments. However, one can often assume that the relation itself is specified by means of a finite first-order or second-order theory $T_{\preceq}(\bar{P}, \bar{P}')$, i.e.,

$$\bar{P} \preceq \bar{P}' \stackrel{\text{def}}{\equiv} T_{\preceq}(\bar{P}, \bar{P}').$$

In such cases (2) can be rewritten as

$$\begin{array}{l} T(\bar{P}, \bar{S}) \wedge \\ \forall \bar{X} \forall \bar{Y} \big\{ [T(\bar{X}, \bar{Y}) \wedge T_{\preceq}(\bar{X}, \bar{P})] \rightarrow T_{\preceq}(\bar{P}, \bar{X}) \big\}. \end{array} \tag{4}$$

When defining $T_{\preceq}(\bar{P}, \bar{P}')$, we may often use other preference or ordering constraints on individuals (or sets) in the definition. Constraints on these preferences or orderings are specified in the theory $T(\bar{P}, \bar{S})$ in Definition 2. We will call such constraints *secondary preference or ordering constraints*. Note that secondary preference constraints will generally be defined in the theory $T(\bar{P}, \bar{S})$.

Example 4 Below we show some examples of $\preceq$. For other examples and a more systematic study of obtaining such preferences on the basis of preferences on object attributes, see (Chomicki & Zhang 2008).

1. Standard predicate circumscription is obtained when

   $$\bar{P} \preceq \bar{P}' \stackrel{\text{def}}{\equiv} \bigwedge_{i=1}^{n} \big\{ \forall \bar{x}_i [P_i(\bar{x}_i) \rightarrow P_i'(\bar{x}_i)] \big\},$$

   where $\bar{P} = \langle P_1, \ldots, P_n \rangle$ and $\bar{P}' = \langle P_1', \ldots, P_n' \rangle$.

2. Assume that $\leq$ is a secondary ordering constraint on tuples of domain elements. Then it is reasonable to consider, for example, the following definitions of the primary preference constraint $\preceq$, where for simplicity we deal with the case of single rather than tuples of relations,

   (a) $P \preceq P' \stackrel{\text{def}}{\equiv} \forall \bar{x} \forall \bar{y} [(P(\bar{x}) \wedge P'(\bar{y})) \rightarrow \bar{x} \leq \bar{y}]$
   (all tuples satisfying P' are "at least as good as" all tuples satisfying P)

   (b) $P \preceq P' \stackrel{\text{def}}{\equiv} \exists \bar{x} \big[P'(\bar{x}) \wedge \forall \bar{y} [P(\bar{y}) \rightarrow \bar{y} \leq \bar{x}] \big]$
   (there is a tuple satisfying P' which is "at least as good as" all tuples satisfying P).

3. Assume that R is a ternary relation on tuples of domain elements. The intended meaning of $R(\bar{x}, \bar{y}, \bar{z})$ is that $\bar{y}$ is more similar to $\bar{x}$ than $\bar{z}$ is. Then one might consider

   $$P \preceq P' \stackrel{\text{def}}{\equiv} \forall \bar{x} \forall \bar{y} \big[(P(\bar{x}) \wedge P'(\bar{y})) \rightarrow \exists \bar{z} [S(\bar{z}) \wedge R(\bar{z}, \bar{y}, \bar{x})] \big],$$

   where S is a given relation symbol[3] meaning that P' is "better than" P when tuples in P' are "more similar" to tuples in S than tuples in P are. Such a relation $\preceq$ might be useful, e.g., in pattern recognition. $\triangleleft$

We now introduce an additional type of constraint called a cardinality constraint which can be used in formulas in $T(\bar{P}, \bar{S})$ in Definition 2.

[3] Representing an "ideal prototype situation".

Definition 5 By a *cardinality constraint* we understand an expression of the form $|A(\bar{x})| \leq k$ or $|A(\bar{x})| \geq k$, where A is a formula with $\bar{x}$ consisting of all free variables of A and k is a natural number.

If the number of free variables of $A(\bar{x})$ is n then $|A(\bar{x})| \leq k$ (respectively, $|A(\bar{x})| \geq k$) is true in a given model if the number of n-tuples satisfying $A(\bar{x})$ is not greater than k (respectively, not less than k), where we assume that the number of 0-tuples satisfying A is 0 when A is not true in the considered model, otherwise it is equal to the cardinality of the model's domain. $\triangleleft$

Example 6

1. The constraint $|married(x, y)| \geq k$ expresses the fact that the number of pairs satisfying $married(x, y)$ is not less than k.

2. Formula $\forall x [|married(x, y)| \leq 1]$ expresses the fact that for any person x there is at most one y satisfying $married(x, y)$. $\triangleleft$

Let us now look at an encoding of a standard example of Brewka using logic programming with answer set semantics.

Example 7 Brewka (Brewka 2006) considers the following example with cardinality constraints and conditionalized preferences:

$$\begin{array}{l} 4\{starter, main, dessert, drink\}4 \leftarrow dinner \\ 1\{soup, salad\}1 \leftarrow starter \\ 1\{fish, beef, lasagne\}1 \leftarrow main \\ lasagne \leq fish \vee beef \\ wine \leq beer \leftarrow beef \\ beer \leq wine \leftarrow \neg beef. \end{array}$$

The preferred answer sets are configurations (satisfying the cardinality constraints) that are optimal w.r.t. $\leq$.

This specification can be represented using our approach by the following theory, Th_{dinner}, where $di(x)$ stands for "x is a part of the dinner", $st(x)$ stands for "x is a starter" and so on.

$$\begin{array}{l} \forall x \big[di(x) \rightarrow \big(st(x) \vee ma(x) \vee de(x) \vee dr(x) \big) \big] \wedge \\ |di(x)| \leq 4 \wedge |di(x)| \geq 4 \wedge \\ \forall x \big[st(x) \rightarrow \big(su(x) \vee sa(x) \big) \big] \wedge \\ |st(x)| \leq 1 \wedge |st(x)| \geq 1 \wedge \\ \forall x \big[ma(x) \rightarrow \big(fi(x) \vee bf(x) \vee la(x) \big) \big] \wedge \\ |ma(x)| \leq 1 \wedge |ma(x)| \geq 1 \wedge \\ \forall x \big[la(x) \preceq \big(fi(x) \vee bf(x) \big) \big] \wedge \\ \forall x \Big[\big(di(x) \wedge bf(x) \big) \rightarrow \\ \qquad \forall y \big[\big(di(y) \wedge dr(y) \big) \rightarrow \big(wi(y) \preceq br(y) \big) \big] \Big] \wedge \\ \forall x \Big[\big(di(x) \wedge \neg bf(x) \big) \rightarrow \\ \qquad \forall y \big[\big(di(y) \wedge dr(y) \big) \rightarrow \big(br(y) \preceq wi(y) \big) \big] \Big]. \end{array}$$

Note that this theory includes cardinality constraints and a secondary preference constraint, $\preceq$. In the example, we are generally interested in reasoning about preferred dinners that satisfy the secondary preference and cardinality constraints. In order to reason about preferred dinners, we have great flexibility when using generalized circumscription. Our primary preference constraint ($\preceq_1$ or $\preceq_2$) will be placed on the predicate $di(x)$ representing dinner and we can define it in terms of the secondary preference constraint $\preceq$. Note the flexibility and control provided by the formalism we deal with. For example, the following primary preference constraints could be used for reasoning about preferred dinners:

$$di \preceq_1 di' \stackrel{\text{def}}{\equiv} \forall x \forall y \big[(di(x) \wedge di'(y)) \to \neg(y \preceq x)\big]$$

or another relation

$$di \preceq_2 di' \stackrel{\text{def}}{\equiv} \forall x \big[di(x) \to \exists y[di'(y) \wedge x \preceq y]\big].$$

The optimal configurations are represented by $\textsc{Circ}_{\succeq_1}(Th_{dinner}; di; \ldots)$ or by $\textsc{Circ}_{\succeq_2}(Th_{dinner}; di; \ldots)$, respectively. The choice of varied relations gives another degree of freedom. Observe that in both cases, we want to compute the maximal definition of $di(x)$. In the case of $\textsc{Circ}_{\succeq_1}$, this results in only preferring the maximal choices for each part of a dinner relative to the secondary preference ordering $\preceq$ defined in Th_{dinner}. $\lhd$

Our Methodology

Before providing examples and going into details as to how we will reason with preferences using generalized circumscription, let us state the representational methodology used because due to its generality, we have a variety of choices that can be made in representing a problem.

- We begin by defining the theory $T(\bar{P}, \bar{S})$ in Definition 2. This theory includes the definition and use of secondary preference or ordering constraints and cardinality constraints. The secondary preference or ordering constraints are often constraints on individuals.
- We then define the primary preference constraint $T_{\preceq}(\bar{P}, \bar{P}')$ on a selected set of predicates which may include in its definition, the use of the secondary preference or ordering constraints and other predicates in $T(\bar{P}, \bar{S})$.
- We then conjoin the generalized circumscription axiom which uses $\preceq$ as defined by $T_{\preceq}(\bar{P}, \bar{P}')$.
- We can then apply the DLS algorithm or its extensions to the resulting second-order theory and if it is in an appropriate syntactic form, we can successfully reduce the theory to a first-order logically equivalent theory. In addition, we can often extract maximal and/or minimal definitions of our predicates $\bar{P}$ relative to $\preceq$, constructively. Reasoning about the theory may then proceed in a standard manner using any automated theorem proving techniques or efficient specializations of them when applicable.

The DLS Algorithm

In the previous sections, we have shown how generalized circumscription can be used as a very flexible framework for qualitative preference modeling with cardinality constraints. We now consider how we can reason with second-order generalized circumscription theories by reducing them to logically equivalent first-order theories under certain assumptions which will be made clear. The basis for doing this will be to use the DLS algorithm (Doherty, Łukaszewicz, & Szałas 1997) which is an algorithm for eliminating second-order quantifiers and is based on a lemma of Ackermann (Ackermann 1935).

This method and its extension to fixpoints, called the DLS* algorithm (Doherty, Łukaszewicz, & Szałas 1996) which is based on results proved in (Nonnengart & Szałas 1998), has been shown to be quite powerful in the context of standard forms of circumscription.

For a discussion of related methods see, e.g., (Gabbay, Schmidt, & Szałas 2008). In the following sections, we will show how the DLS algorithm and new extensions to it can be used for reasoning with preference and cardinality constraints in the context of generalized circumscription. The following definitions and lemmas provide the formal basis for eliminating quantifiers using the DLS algorithm.

Notation 8 Let $B(X)$ be a second-order formula, where X is a k-argument relational symbol or variable and let $A(\bar{x}, \bar{t})$ be a first-order formula with free variables $\bar{x}$ and maybe some other arguments $\bar{t}$, being terms, i.e., free or bound variables, constants or functional expressions. Then by $B[X(\bar{x})/A(\bar{x}, \bar{t})]$ we mean the formula obtained from $B(X)$ by substituting each occurrence of X of the form $X(\bar{s})$ in $B(X)$ by $A(\bar{s}, \bar{t})$, renaming the bound variables in $A(\bar{x}, \bar{t})$ with fresh variables. If variables $\bar{x}$ in $B[X(\bar{x})/A(\bar{x}, \bar{t})]$ are clear from context, we simplify the notation to $B[X/A(\bar{x}, \bar{t})]$. $\lhd$

The above notation is needed in the formulation of Ackermann's lemma (Lemma 11). The intuition behind it is that in Lemma 11 we substitute all occurrences of X by its definition given by (see also Remark 12):

$$\forall \bar{x}[X(\bar{x}) \stackrel{\text{def}}{\equiv} A(\bar{x}, \bar{t})]. \tag{5}$$

When we apply (5) in the substitution, $X(\bar{t})$ is to be substituted by $A(\bar{t}, \bar{t})$. The assumption as to renaming bound variables is necessary to avoid conflicts between different variables.

Example 9 Let $B(X) \equiv \forall z[X(y, z) \vee X(f(y), g(x, z))]$, where X is a relational variable and let $C(x, y)$ be the formula $\exists z[R(x, y, z)]$. Then $B[X/C(x, y)]$ is

$$\forall z[\underbrace{\exists z' R(y, z, z')}_{C'(y,z)} \vee \underbrace{\exists z' R(f(y), g(x, z), z')}_{C'(f(y), g(x,z))}],$$

where $C'(x, y)$ is obtained from $C(x, y)$ by renaming the bound variable z with z'. $\lhd$

Definition 10 Let $B(X)$ be a formula, where X is a k-argument relational symbol or variable. Assume that $B(X)$ contains a single occurrence of X. We say that formula $B(X)$ is up-monotone w.r.t. X if for any model $\mathcal{M}$ and for any formulas $A(\bar{x}), C(\bar{x})$:

$$\mathcal{M} \models \forall \bar{x}[A(\bar{x}) \to C(\bar{x})] \tag{6}$$

implies

$$\mathcal{M} \models \forall \bar{x}\big[B[X/A(\bar{x})] \rightarrow B[X/C(\bar{x})]\big] \tag{7}$$

and down-monotone w.r.t. X, if (6) implies

$$\mathcal{M} \models \forall \bar{x}\big[B[X/C(\bar{x})] \rightarrow B[X/A(\bar{x})]\big]. \tag{8}$$

We say that formula $D(X)$ is *up-monotone* (respectively *down-monotone*) w.r.t. X if for any formula D' obtained from D by replacing all except one occurrences of X by a new relation symbol, $D'(X)$ is up-monotone (respectively down-monotone) w.r.t. X. If D contains no occurrences of X then it is assumed to be both up- and down-monotone w.r.t. X. $\lhd$

The following lemma is based on the lemma proved by Ackermann (Ackermann 1935).

Lemma 11 [Ackermann's Lemma] Let X be a predicate variable and $A(\bar{x}, \bar{t})$, $B(X)$ be formulas such that A contains no occurrences of X. Then

$$\exists X \big\{\forall \bar{x}[A(\bar{x}, \bar{t}) \rightarrow X(\bar{x})] \wedge B(X)\big\} \equiv B[X/A(\bar{x}, \bar{t})] \tag{9}$$

provided that $B(X)$ is down-monotone w.r.t. X, and

$$\exists X \big\{\forall \bar{x}[X(\bar{x}) \rightarrow A(\bar{x}, \bar{t})] \wedge B(X)\big\} \equiv B[X/A(\bar{x}, \bar{t})] \tag{10}$$

provided that $B(X)$ is up-monotone w.r.t. X. $\lhd$

Let us briefly explain how Ackermann's lemma works. We consider (10) (the case of (9) is analogous).

Assume that $\exists X \big\{\forall \bar{x}[X(\bar{x}) \rightarrow A(\bar{x}, \bar{t})] \wedge B(X)\big\}$. The greatest (w.r.t. inclusion) relation X satisfying the implication $\forall \bar{x}[X(\bar{x}) \rightarrow A(\bar{x}, \bar{t})]$ is $X(\bar{x}) \stackrel{\text{def}}{\equiv} A(\bar{x}, \bar{t})$. By assumption, there is an X' satisfying this implication as well as $B(X')$. So such an X' implies $A(\bar{x}, \bar{t})$ which implies X, too. Now observe that $B(X)$ is up-monotone w.r.t. X so from the facts that $B(X')$ holds and that X' implies X we have $B(X)$.

If, on the other hand, $B[X/A(\bar{x}, \bar{t})]$ holds, then there is an X (defined by $X(\bar{x}) \stackrel{\text{def}}{\equiv} A(\bar{x}, \bar{t})$) satisfying the lefthand side of equivalence (10).

Based on the above discussion we also have an important remark.

Remark 12 Note that one of the important outcomes of the method based on Ackermann's lemma is that formula (5) provides a definition:[4]

- of the smallest (w.r.t. inclusion) relation satisfying the first-order part of formula (9) in the case of applying (9)
- of the greatest (w.r.t. inclusion) relation satisfying the first-order part of formula (10) in the case of applying (10).

Therefore we have "canonical" definitions of the required maximal and minimal relations which can be used to extract definitions of circumscribed relations constructively, under the assumption that we can eliminate quantifiers. The same results also apply to the fixpoint approach of (Nonnengart & Szałas 1998) used in the DLS* algorithm. $\lhd$

The original lemma of Ackermann assumes that some formulas are positive (respectively, negative) w.r.t. certain relation symbols. This follows from the fact that, in general, checking up- or down- monotonicity is not decidable, while checking positivity and negativity can be done in time linear in the length of the considered formula. Since positivity implies up-monotonicity and negativity implies down-monotonicity, it is algorithmically convenient to use positivity and negativity rather than monotonicity. To make Lemma 11 algorithmically friendly, we will use the following definition and extend it later (see Definition 18) to deal with other monotone operators (such as cardinality constraints and preference relations).

Definition 13 Let $A(P)$ be a formula. Assume that $A(P)$ contains a single occurrence of P and formula C contains no occurrences of P. Then $\|A(P)\|_P$, counting the number of negations under which P occurs, is defined inductively to be the natural number satisfying the following clauses:[5]

$$\|A(P)\|_P \stackrel{\text{def}}{=} \begin{cases} 0 & \text{for } A(P) = P \\ \|B(P)\|_P & \text{for } A(P) = B(P) \circ C \\ & \text{or } A(P) = C \circ B(P), \\ & \quad \text{where } \circ \in \{\vee, \wedge\} \\ \|B(P)\|_P & \text{for } A(P) = Q[B(P)], \\ & \quad \text{where } Q \in \{\exists, \forall\} \\ \|B(P)\|_P & \text{for } A(P) = C \rightarrow B(P) \\ \|B(P)\|_P + 1 & \text{for } A(P) = B(P) \rightarrow C \\ \|B(P)\|_P + 1 & \text{for } A(P) = \neg B(P). \end{cases}$$

$\lhd$

We have the following important proposition.

Proposition 14 Let $A(P)$ be a formula containing at least one occurrence of P.

If, for any formula $A'(P)$ obtained from $A(P)$ by replacing all except one occurrences of P by a new relation symbol, $\|A'(P)\|_P$ is even then $A(P)$ is up-monotone w.r.t. P.

If for any such formula $A'(P)$, $\|A'(P)\|_P$ is odd then $A(P)$ is down-monotone w.r.t. P.

Any formula without occurrences of P is both up- and down-monotone w.r.t. P. $\lhd$

Second-order quantifier elimination based on Lemma 11 depends on transforming a formula into one of the forms (9) or (10) required in the lemma. The DLS algorithm defined in (Doherty, Łukaszewicz, & Szałas 1997) uses such transformations. It has been implemented and is available online – see (Gustafsson 1996; Magnusson 2005).

[4] Assuming that such an X exists, i.e., that $B[X/A(\bar{x}, \bar{t})]$ in the respective equivalences (9) and (10) of Lemma 11 holds.

[5] For simplicity we disallow the connective of equivalence here. One can deal with $p \equiv q$, replacing it by $(p \rightarrow q) \wedge (q \rightarrow p)$ and using Proposition 14.

Observe that in the context of database technology, the reduction of second-order theories to fixpoint calculus is also important, since the complexity of computing second-order queries is PSPACE-complete. Circumscription over first-order theories leads (at least) to CO-NPTIME complexity. On the other hand, computing fixpoint queries is in PTIME. For results on the complexity of queries see, e.g., (Abiteboul, Hull, & Vianu 1996; Ebbinghaus & Flum 1995).

Using the DLS Algorithm with Generalized Circumscription

In this section, we show how the DLS algorithm can be applied to generalized circumscription. Initially, we will place two restrictions on our theories that will be relaxed in the next section. The first restriction is that cardinality constraints are excluded and must not be used. The second restriction is that we will assume that the pre-order $\preceq$ representing the primary preference constraint is defined explicitly by a first- or second-order theory $T_{\preceq}$, i.e., that

$$\models \bar{P} \preceq \bar{Q} \stackrel{\text{def}}{\equiv} T_{\preceq}(\bar{P}, \bar{Q}).$$

The DLS algorithm considers the negated second-order part of the formula (4). It is equivalent to

$$\neg\exists\bar{X}\exists\bar{Y}\{T(\bar{X},\bar{Y}) \wedge T_{\preceq}(\bar{X},\bar{P}) \wedge \neg T_{\preceq}(\bar{P},\bar{X})\}. \quad (11)$$

In addition, $\preceq$ must not appear as an operator in $T(\bar{P},\bar{S})$.

The method based on the DLS algorithm consists of two steps:

1. finding a first-order equivalent A for the generalized circumscription formula $\text{CIRC}_{\preceq}(T;\bar{P};\bar{S})$, if possible, and
2. extracting $\bar{P}$ satisfying A.

Observe that extracting a definition of the greatest or smallest (w.r.t. inclusion) optimal $\bar{P}$ from A can be done either directly by analyzing the resulting formula or by considering the formula $\exists\bar{P}[A]$ and applying the method discussed in Remark 12. However, there can be cases when there is no greatest nor smallest such relation (and then extracting $\bar{P}$ from $\exists\bar{P}[A]$ cannot be done using Ackermann's lemma, as A cannot be transformed into a required form).

Consider Example 15 in formalizing a form of commonsense reasoning where we try to maximize preferences for certain actions at each time point which meet some application dependent criteria.

Example 15 Consider actions $b, c, d, e, \ldots$ and a preference relation $\geq$ defined on actions. Let $C(t,a)$ mean that action a is chosen at time point t. Assume that C satisfies a theory $T(C,\bar{Q})$, reflecting constraints on C and maybe some other relations $\bar{Q}$.

One principle for maximizing preferences on action choices meeting such constraints is reflected by

$$C' \succeq C'' \stackrel{\text{def}}{\equiv} \forall t\forall x\forall y\Big[\big(C'(t,x) \wedge C''(t,y)\big) \rightarrow x \geq y\Big]. \quad (12)$$

Formula (12) states that we prefer to choose actions at time points that are preferable relative to a secondary preference relation on actions, $\geq$, which can be defined in a domain dependent manner.

Let us consider the generalized circumscription $\text{CIRC}_{\succeq}(T(C,\bar{Q});C;)$ of C in $T(C,\bar{Q})$ w.r.t. the primary preference constraint $\succeq$ defined by (12).[6] In this case formula (3) specializes to the conjunction of $T(C,\bar{Q})$ and

$$\begin{aligned}\forall X\Big\{&\Big[T(X,\bar{Q})\wedge \\ &\forall t\forall x\forall y\Big[\big(X(t,x) \wedge C(t,y)\big) \rightarrow x \geq y\Big]\Big] \rightarrow \\ &\forall t\forall x\forall y\Big[\big(C(t,x) \wedge X(t,y)\big) \rightarrow x \geq y\Big]\Big\},\end{aligned} \quad (13)$$

which is also equivalent to

$$\begin{aligned}\neg\exists X\Big\{&T(X,\bar{Q})\wedge \\ &\forall t\forall x\forall y\big[\big(X(t,x) \wedge C(t,y)\big) \rightarrow x \geq y\big]\wedge \\ &\exists t\exists x\exists y\big[C(t,x) \wedge X(t,y) \wedge \neg x \geq y\big]\Big\}.\end{aligned} \quad (14)$$

The DLS algorithm transforms (14) into the equivalent:

$$\begin{aligned}\neg\exists t\exists x\exists y\exists X\Big\{&\forall u\forall v\big[(t=u \wedge y=v) \rightarrow X(u,v)\big]\wedge \\ &\forall t'\forall x'\forall y'\Big[\big(X(t',x') \wedge C(t',y')\big) \rightarrow x' \geq y'\Big]\wedge \\ &T(X,\bar{Q}) \wedge C(t,x) \wedge \neg x \geq y\Big\}.\end{aligned}$$

Assuming that $T(X,\bar{Q})$ is down-monotone w.r.t. X, an application of Lemma 11 results in

$$\begin{aligned}\neg\exists t\exists x\exists y\Big\{&T(X,\bar{Q})[X(u,v)/(t=u \wedge y=v)]\wedge \\ &\forall t'\forall x'\forall y'\Big[\big((t=t' \wedge y=x') \wedge C(t',y')\big) \rightarrow x' \geq y'\Big]\wedge \\ &C(t,x) \wedge \neg x \geq y\Big\}.\end{aligned}$$

Moving the negation inside and simplifying the result, we obtain

$$\begin{aligned}\forall t\forall x\forall y\Big\{&\Big[T(X,\bar{Q})[X(u,v)/(t=u \wedge y=v)]\wedge \\ &\forall y'\big[C(t,y') \rightarrow y \geq y'\big] \wedge C(t,x)\Big] \\ &\rightarrow x \geq y\Big\}.\end{aligned} \quad (15)$$

We assumed that $T(X,\bar{Q})$ is down-monotone w.r.t. X. Since $T(C,\bar{Q})$ is assumed to hold, we conclude that $T(X,\bar{Q})[X(u,v)/(t=u \wedge y=v)]$ is TRUE. Therefore (15)) reduces to

$$\forall t\forall x\forall y\big\{\big[\forall y'[C(t,y') \rightarrow y \geq y'] \wedge C(t,x)\big] \rightarrow x \geq y\big\}, \quad (16)$$

meaning that whenever x is chosen at time point t then it is preferred to any y which is maximal w.r.t. $\geq$. Since $\geq$ is reflexive, we can in fact conclude that such x is equally

[6] Recall that the definition of $\text{CIRC}_{\succeq}(;;)$ is provided in Remark 3 (equivalence (3)).

preferred to any such y. This means that any maximal (w.r.t. $\succeq$ defined by (12)) choice C consists of maximally preferred (w.r.t. $\geq$) actions, assuming that $T(C, \bar{Q})$ is satisfied for such C.

Consider, as an example the theory $T(C, \bar{Q})$ given by

$$\forall t\big[\big(C(t,d) \vee C(t,c)\big) \rightarrow Q(t)\big],$$

where Q stands for "meeting quality criteria". Observe that $T(C, \bar{Q})$ is down-monotone w.r.t. C. If one wants to make optimal w.r.t. $\succeq$ choices that guarantee $Q(t)$ then d or c is to be chosen. If at least one of these actions is maximal w.r.t. $\geq$ then optimal choices guaranteeing quality criteria can be made. If these actions are not maximal w.r.t. $\geq$ then preferred actions do not guarantee quality. In fact, this often happens in the real world, where one often prefers cost effective choices. However, such choices often do not guarantee quality. $\lhd$

Extending DLS for Cardinality Constraints and Implicit Primary Preference Constraints

In the previous section, we placed two restrictions on our preference models which we will now relax. In doing so, we will extend the DLS algorithm so it can handle the use of cardinality constraints and also implicit definitions of the primary preference constraint, $\preceq$. In order to do this, we will use the properties of up- and down-monotonicity.

Extending the Language

In order to express additional constraints, such as cardinality or preference constraints, we have to extend the language by adding new operators. Usually such operators take formulas as arguments, and so they are (at least) of the third order.

Definition 16 Let $X_1(\bar{x}_1), \ldots, X_n(\bar{x}_n)$ be relational variables. By a third-order operator on $X_1, \ldots, X_n$ we understand any expression of the form $\Gamma(X_1(\bar{x}_1), \ldots, X_n(\bar{x}_n))$, where Γ is a third-order relation symbol. $\lhd$

Example 17

1. The usual propositional connectives and quantifiers are examples of third-order operators.
2. For any natural number k, the cardinality constraint $|\ldots|\leq k$ is a unary third-order operator.
3. The preference relation $\preceq$ is a binary third-order operator. $\lhd$

We extend the logical language by assuming that some third-order operators are atomic formulas, too.

Whenever the language is extended that way and we want to make use of Ackermann's lemma and the DLS algorithm, we have to investigate the monotonicity of the introduced operators. Once monotonicity is established, we can further use it in the lemma of Ackermann and thus in the DLS algorithm.[7] This approach is also a contribution of the current paper and has applications much wider than handling preferences and cardinality constraints, as it concerns arbitrary monotone operators.

Definition 18 Let $\Gamma(\bar{X})$ be a third-order operator and $A(P)$ be a formula containing a single occurrence of P. Then $\|A(P)\|_P$ is defined to satisfy Definition 13 and additionally:

$$\|A(P)\|_P \stackrel{\text{def}}{=} \begin{cases} \|B(P)\|_P & \text{for } A(P) = \Gamma(\ldots, B(P), \ldots) \\ & \text{and } \Gamma \text{ up-monotone in the} \\ & \text{coordinate where } B(P) \text{ occurs} \\ \|B(P)\|_P + 1 & \text{for } A(P) = \Gamma(\ldots, B(P), \ldots) \\ & \text{and } \Gamma \text{ down-monotone in the} \\ & \text{coordinate where } B(P) \text{ occurs,} \end{cases}$$

or is assumed to be undefined if neither clauses of Definition 13 nor the above clauses are applicable. $\lhd$

With the above extension of the definition of $\|A(P)\|_P$, Proposition 14 holds and can be applied as an efficient sufficient condition for checking up- and down-monotonicity.

In what follows, by the *extended* DLS algorithm we shall mean a modified version of the DLS algorithm which works with formulas where cardinality and preference constraints are allowed and also makes use of Definitions 13 and 18, as well as Proposition 14.

Monotonicity of Cardinality Constraints

Let us first analyze the monotonicity of cardinality constraints. Observe that constraints of the form $|X(\bar{x})|\leq k$ can be considered as infinitely many third-order operators $\Gamma_k(X(\bar{x}))$, for k ranging over the natural numbers, and similarly in the case of constraints of the form $|X(\bar{x})|\geq k$.

Lemma 19

1. The constraint $|X(\bar{x})|\leq k$ is down-monotone in X.
2. The constraint $|X(\bar{x})|\geq k$ is up-monotone in X.

PROOF. Assume that the number of free variables in $X(\bar{x})$ is n. Let $\mathcal{M}$ be a model and $A(\bar{x}), B(\bar{x})$ be formulas with n free variables $\bar{x}$, satisfying $\mathcal{M} \models \forall\bar{x}[A(\bar{x}) \rightarrow B(\bar{x})]$, meaning that the set of tuples satisfying A is included in the set of tuples satisfying B. As an immediate consequence we have that $|B(\bar{x})|\leq k$ implies $|A(\bar{x})|\leq k$ and $|A(\bar{x})|\geq k$ implies $|B(\bar{x})|\geq k$. $\lhd$

Example 20 Formula $|P(x)|\leq k$ is down-monotone w.r.t. the relation symbol P and formula $|P(x)|\geq k$ is up-monotone w.r.t. P. Therefore, e.g.,

$$\exists P[\forall x\big[P(x) \rightarrow Q(x)\big] \wedge |P(y)|\geq k\big]$$

is, by application of Lemma 11, equivalent to $|Q(y)|\geq k$.

[7] The same applies to fixpoint reductions and DLS*, but this subject is outside of the scope of the current paper.

Formula $\neg\big((|P(y)|\leq k) \wedge R\big)$ is up-monotone w.r.t. P, since $\|\neg\big((|P(y)|\leq k)\wedge R\big)\|_P = 2$ and formula $\neg\big((|P(y)|\geq k)\wedge R\big)$ is down-monotone w.r.t. P, since $\|\neg\big((|P(y)|\geq k)\wedge R\big)\|_P = 1$. Therefore, e.g.,

$$\exists P[\forall x\big[\big(Q(x) \wedge S(x)\big) \to P(x)\big] \wedge \neg\big((|P(y)|\geq k)\wedge R\big)\big]$$

is, by application of Lemma 11, equivalent to

$$\neg\big((|Q(y) \wedge S(y)|\geq k)\wedge R\big). \qquad \lhd$$

Observe that one can easily define the cardinality constraint $|A(\bar{x})|{=}k$ by the conjunction $|A(\bar{x})|{\leq}k \wedge |A(\bar{x})|{\geq}k$, but we do not allow $|A(\bar{x})|{=}k$ as an atomic formula, since it is neither up- nor down-monotone, while the constraints of Definition 5 behave monotonically, as Lemma 19 shows.

Monotonicity of Primary Preference Constraints

In our second relaxation, we do not assume that the primary preference constraint has to have an explicit definition $T_{\preceq}(\bar{P}, \bar{P}')$, but that it may be defined implicitly in the theory $T(\bar{P}, \bar{S})$, where the operator $\preceq$ can occur in formulas. Then, in order to make quantifier elimination possible, the use of $\preceq$ in such formulas has to be shown to be monotone.

In the method we propose here we shall assume that whenever in a given model $\mathcal{M}$, $\mathcal{M} \models \forall\bar{x}[P(\bar{x}) \to Q(\bar{x})]$ then $\mathcal{M} \models P \preceq Q$. Such an assumption does not have to hold in general, but seems very natural in applications we consider. For example, whenever to each tuple there is associated a payoff being a nonnegative number then $P \subseteq Q$ implies that the total payoff of P is not greater than the total payoff of Q, so it is reasonable to assume that Q is at least as preferred as P. Also, when $\preceq$ is considered as a preference relation on formulas then $P \preceq Q$ means that Q is preferred to P. Suppose that P implies Q. Then, whenever one chooses to satisfy P, Q is also satisfied, but not conversely. So Q has some preference over P.

We have the following important lemma, indicating that $\preceq$ behaves similarly to implication when monotonicity is considered.

Lemma 21 Assume that for any formulas $A(\bar{x}), B(\bar{x})$ with all free variables among $\bar{x}$, and any model $\mathcal{M}$,

$$\text{if } \mathcal{M} \models \forall\bar{x}[A(\bar{x}) \to B(\bar{x})] \text{ then } \mathcal{M} \models A \preceq B \qquad (17)$$

and that $\preceq$ is transitive.

Under these assumptions $\preceq$ is down-monotone w.r.t. its first coordinate A and up-monotone w.r.t. its second coordinate B.

PROOF. Let $\mathcal{M}$ be a model and A, B be formulas such that $\mathcal{M} \models \forall\bar{x}[A(\bar{x}) \to B(\bar{x})]$.

Let us first prove down-monotonicity defined by (8). Let C be a formula and $\mathcal{M} \models B \preceq C$. By assumption (17) we have that $\mathcal{M} \models A \preceq B$. Therefore, by the transitivity of $\preceq$ we have $\mathcal{M} \models A \preceq C$.

To prove up-monotonicity (defined by (7)) assume that C is a formula and $\mathcal{M} \models C \preceq A$. As before we use assumption (17) and the transitivity of $\preceq$ and obtain $\mathcal{M} \models C \preceq B$. $\lhd$

The following example shows up- and down-monotonicity of formulas when $\preceq$ is allowed in them.

Example 22 The occurrence of relation symbol P in formula $P \preceq Q$ is down-monotone and it is up-monotone in $Q \preceq P$. Dually, the occurrence of P in $Q \succeq P$ is down-monotone and it is up-monotone in $P \succeq Q$. Therefore, e.g.,

$$\exists P\big[\forall x\big[Q(x) \to P(x)\big] \wedge P(a) \preceq R(a)\big]$$

is, by application of Lemma 11, equivalent to $Q(a) \preceq R(a)$.

The occurrence of P in formula $\neg\big((\neg P) \preceq R\big)$ is down-monotone, since $\big\|\neg\big((\neg P) \preceq R\big)\big\|_P = 3$. On the other hand, the occurrence of P in $\neg\big(R \preceq (\neg P)\big)$ is up-monotone, since $\big\|\neg\big(R \preceq (\neg P)\big)\big\|_P = 2$. Therefore, e.g.,

$$\exists P\big[\forall x\big[P(x) \to \big(Q(x) \vee S(x)\big)\big] \wedge \neg\big(R(b) \preceq (\neg P(c))\big)\big]$$

is, by application of Lemma 11, equivalent to

$$\neg\big(R(b) \preceq \big(\neg(Q(c) \vee S(c))\big)\big). \qquad \lhd$$

Example 23 Assume that we have the following theory $T(W, R, T, E)$:

$$\begin{array}{l} W(x) \preceq R(x) \wedge W(x) \preceq T(x) \wedge \\ \quad \forall x\big[\big(W(x) \vee R(x) \vee T(x)\big) \to E(x)\big], \end{array}$$

meaning that, taking into account the necessary effort, a person x prefers to go to a restaurant ($R(x)$) or to go to a theater ($T(x)$) rather than to walk ($W(x)$) and that all choices imply some effort $E(x)$.

Assume that we minimize E w.r.t. $\preceq$ varying W, i.e., we consider $\text{CIRC}_{\preceq}(T(W, R, T, E); E; W)$:

$$\begin{array}{l} T(W, R, T, E) \wedge \\ \forall X \forall Y \Big\{ \Big[Y(x) \preceq R(x) \wedge Y(x) \preceq T(x) \wedge \\ \qquad \forall x\big[\big(Y(x) \vee R(x) \vee T(x)\big) \to X(x)\big] \wedge \\ \qquad\qquad X(x) \preceq E(x) \Big] \to E(x) \preceq X(x) \Big\}. \end{array} \qquad (18)$$

The second order part of (18) is equivalent to

$$\begin{array}{l} \neg \exists X \exists Y \Big\{ Y(x) \preceq R(x) \wedge Y(x) \preceq T(x) \wedge \\ \qquad \forall x\big[\big(Y(x) \vee R(x) \vee T(x)\big) \to X(x)\big] \wedge \\ \qquad\qquad X(x) \preceq E(x) \wedge \neg\big(E(x) \preceq X(x)\big) \Big\}. \end{array} \qquad (19)$$

All occurrences of Y in (19) are down-monotone, so (19) is equivalent to

$$\begin{array}{l} \neg \exists X \Big\{ \text{FALSE} \preceq R(x) \wedge \text{FALSE} \preceq T(x) \wedge \\ \qquad \forall x\big[\big(\text{FALSE} \vee R(x) \vee T(x)\big) \to X(x)\big] \wedge \\ \qquad\qquad X(x) \preceq E(x) \wedge \neg\big(E(x) \preceq X(x)\big) \Big\}. \end{array} \qquad (20)$$

Since FALSE $\to A$, for any formula A, by assumption (17) we obtain FALSE $\preceq A$, for any A. Thus formula (20) simplifies to

$$\begin{array}{l} \neg \exists X \Big\{ \forall x\big[\big(R(x) \vee T(x)\big) \to X(x)\big] \wedge \\ \qquad\qquad X(x) \preceq E(x) \wedge \neg\big(E(x) \preceq X(x)\big) \Big\}. \end{array} \qquad (21)$$

Observe that all occurrences of X in the second line of (21) are down-monotone. Therefore an application of Ackermann's lemma results in

$$\neg\big\{\big(R(x)\vee T(x)\big)\preceq E(x)\wedge\neg\big(E(x)\preceq\big(R(x)\vee T(x)\big)\big)\big\}.$$

This is equivalent to

$$\big[\big(R(x)\vee T(x)\big)\preceq E(x)\big]\to\big[E(x)\preceq\big(R(x)\vee T(x)\big)\big]. \quad (22)$$

Observe that

$$(R(x)\vee T(x))\to(W(x)\vee R(x)\vee T(x))$$

and, by theory T,

$$(W(x)\vee R(x)\vee T(x))\to E(x).$$

Therefore, by assumption (17) and the transitivity of $\preceq$, we have that $\big(R(x)\vee T(x)\big)\preceq E(x)$. Thus formula (22) reduces to $E(x)\preceq\big(R(x)\vee T(x)\big)$, so we have

$$\big(\big(R(x)\vee T(x)\big)\preceq E(x)\preceq\big(R(x)\vee T(x)\big).$$

Intuitively, the result indicates that the best or most preferred choice of activity from the point of view of using minimal effort (w.r.t. $\preceq$) is either to go to a restaurant or go to the theater. $\triangleleft$

Combining Cardinality and Preference Constraints

In this section, we show an example which uses both cardinality and preference constraints and show how it is reduced.

Example 24 A student has to make a choice of lectures to attend during a new term. He must choose at least three lectures. His tutor asks him to restrict his choices of lectures to those related to algorithms or to logic with a preference given to those related to logic. Let $C(x)$ mean that lecture x is one of those that is chosen, $A(x)$ mean that x is a lecture related to algorithms and $L(x)$ mean that x is a lecture related to logic. Then the student's task can be represented as the theory $Th_{st}(C,L,A)$:

$$\forall x\big[C(x)\to(A(x)\vee L(x))\big]\wedge|C(x)|\geq 3\wedge L(x)\succeq A(x),$$

where $\succeq$ is a preference relation defined on choices.

Consider $\textsc{Circ}_{\succeq}(Th_{st};C;)$ which is the conjunction of $Th_{st}(C,L,A)$ and

$$\begin{aligned}&\forall X\big\{\big[\forall x[X(x)\to(A(x)\vee L(x))]\wedge|X(x)|\geq 3\wedge\\ &L(x)\succeq A(x)\wedge X(x)\succeq C(x)\big]\to C(x)\succeq X(x)\big\}.\end{aligned} \quad (23)$$

Formula (23), in the presence of $Th_{st}(C,L,A)$, is equivalent to

$$\begin{aligned}&\neg\exists X\big\{\forall x[X(x)\to(A(x)\vee L(x))]\wedge\\ &|X(x)|\geq 3\ \wedge X(x)\succeq C(x)\wedge\neg(C(x)\succeq X(x))\big\}.\end{aligned} \quad (24)$$

Elimination of $\exists X$ results in the following formula equivalent to (24):

$$\begin{aligned}\neg\Big[&|A(x)\vee L(x)|\geq 3\wedge\big(A(x)\vee L(x)\big)\succeq C(x)\wedge\\ &\neg\big(C(x)\succeq\big(A(x)\vee L(x)\big)\big)\Big].\end{aligned} \quad (25)$$

By $Th_{st}(C,L,A)$, we have that $\forall x[C(x)\to(A(x)\vee L(x))]$, so by (17), $\big(A(x)\vee L(x)\big)\succeq C(x)$ is true. Taking this into account and moving negation inside, we obtain the following equivalent of (25):

$$|A(x)\vee L(x)|\geq 3\to C(x)\succeq\big(A(x)\vee L(x)\big). \quad (26)$$

The cardinality constraint in the theory $|C(x)|\geq 3$ assures us that in any model, any set of lectures $C(x)$ to choose from is always going to be greater than or equal to 3. The theory also states that logical lectures are preferred over algorithmic lectures, $L(x)\succeq A(x)$, but not that $C(x)$ will reflect this, only that $C(x)$ will include logical or algorithmic lectures.The net result of the constraints in the theory together with the maximization of C when circumscribed will in fact reflect our preference of logic lectures over algorithmic lectures in all preferred extensions of C. If, for example, there were available two lectures related to logic and three lectures related to algorithms, then the optimal (w.r.t. $\succeq$ satisfying (26) and Th_{st}) choices would consist of two lectures related to logic and at least one lecture related to algorithms. $\triangleleft$

On the Strength of the Extended DLS

The general situation we consider might be complex and outside of the scope of the extended DLS and DLS* algorithms, as the following example illustrates.[8]

Example 25 [Knapsack problem] The knapsack problem, known to be NPTIME-complete, can be formulated as the problem of maximizing the value of chosen items subject to the restriction of the total volume of items selected. Let X be a unary relation and let $volume(X,z)$ mean that the total volume of items in the set $\{x\mid x\in X\}$ does not exceed z. Even if $volume$ is a third-order relation, it can easily be computed in a tractable manner. Let further $value(X,z)$ mean the total value of items in the set $\{x\mid x\in X\}$ is equal to z. Define the ordering $\succeq$ on unary relations to satisfy:

$$X\succeq Y\stackrel{\text{def}}{\equiv}\exists x\exists y[value(X,x)\wedge value(Y,y)\wedge x\geq y].$$

Then $\textsc{Circ}_{\succeq}(volume(X,v);X;)$, where v is a given constant, describes maximal (w.r.t. $\succeq$) relations X of total volume less than v, i.e., all relations X representing all choices of items of greatest possible value, whose total volume does not exceed the value v. $\triangleleft$

However, there are many theories reducible by the DLS algorithm. Conradie (Conradie 2006) provides a characterization of formulas where DLS succeeds.

Below we shall provide some sufficient conditions for the success of the extended DLS algorithm in the context of generalized circumscription, when we allow precedence and cardinality operators to appear explicitly in the language.

[8]Observe that, in such cases, one cannot expect a reduction to first-order or fixpoint calculus unless PTIME =NPTIME. This follows from the fact that the knapsack problem is NPTIME-complete, while first-order and fixpoint formulas are PTIME-computable over finite domains (see, e.g., (Abiteboul, Hull, & Vianu 1996)).

Such cases are out of scope of results provided in (Conradie 2006).

We have considered two forms of generalized circumscription: $\text{CIRC}_{\preceq}(T; \bar{P}; \bar{S})$, expressed by (2) and $\text{CIRC}_{\succeq}(T; \bar{P}; \bar{S})$, expressed by (3). We discuss them in the next sections.

The Case of Minimization

The second-order part of (2) is equivalent to the negation of

$$\exists \bar{X} \exists \bar{Y} \{T(\bar{X}, \bar{Y}) \wedge \bar{X} \preceq \bar{P} \wedge \neg(\bar{P} \preceq \bar{X})\}. \quad (27)$$

To keep the presentation simple, we consider the case when $\bar{X}$ consists of a single relational variable X and without varied predicates, so (27) reduces in this case to

$$\exists X \{T(X) \wedge X \preceq P \wedge \neg(P \preceq X)\}. \quad (28)$$

Observe that the sub-formula $X \preceq P \wedge \neg(P \preceq X)$ is down-monotone w.r.t. X. Therefore we have the following proposition.

Proposition 26 If $T(X)$ in formula (28) is the conjunction $T_1(X) \wedge T_2(X)$, where

- $T_1(X)$ is the conjunction of formulas of the form $\forall \bar{x}[A(\bar{t}) \rightarrow X(\bar{s})]$, where $\bar{s}, \bar{t}$ are arbitrary terms (in particular containing variables from $\bar{x}$) and A is a formula containing no occurrences of X
- $T_2(X)$ is down-monotone w.r.t. X,

then the extended DLS algorithm succeeds in the reduction of (28) to first-order logic with cardinality and preference constraints. ◁

The following example shows that even the class of formulas considered in Proposition 26 is not trivial and far out of scope of other known approaches.

Example 27 Let $T(X)$ be the formula

$$\forall x \Big[\underbrace{\forall y \exists z \big[\big(S(x) \vee |R(x,z)| \leq 5 \big) \wedge T(y) \preceq U(z) \big]}_{A(x)} \rightarrow X(x) \Big] \wedge$$
$$\underbrace{\forall v \Big[\big(X(v) \preceq B(v) \big) \wedge |X(v)| \leq u \wedge \exists w [\neg X(w)] \Big]}_{T_2(X)}.$$

By Proposition 26, in this case formula (28) is reducible to first-order logic with cardinality and preference constraints. The first-order formula equivalent to (28), provided by the extended DLS algorithm is[9]

$$T_2 \Big[X(x) / \forall y \exists z \big[\big(S(x) \vee |R(x,z)| \leq 5 \big) \wedge T(y) \preceq U(z) \big] \Big]$$

which is:

$$\forall v \Big\{ \Big(\forall y \exists z \big[\big(S(v) \vee |R(v,z)| \leq 5 \big) \wedge T(y) \preceq U(z) \big] \Big) \preceq B(v) \wedge$$
$$|\forall y \exists z \big[\big(S(v) \vee |R(v,z)| \leq 5 \big) \wedge T(y) \preceq U(z) \big]| \leq u \wedge$$
$$\exists w \big[\neg \forall y \exists z \big[\big(S(w) \vee |R(w,z)| \leq 5 \big) \wedge T(y) \preceq U(z) \big] \big] \Big\}. \ \triangleleft$$

[9] In fact, we can directly apply Ackermann's lemma here.

The Case of Maximization

The second-order part of (3) is equivalent to the negation of

$$\exists \bar{X} \exists \bar{Y} \{T(\bar{X}, \bar{Y}) \wedge \bar{X} \succeq \bar{P} \wedge \neg(\bar{P} \succeq \bar{X})\}. \quad (29)$$

As before, we consider the case when $\bar{X}$ consists of a single relational variable X and without varied predicates, so (29) reduces in this case to

$$\exists X \{T(X) \wedge X \succeq \bar{P} \wedge \neg(\bar{P} \succeq X)\}. \quad (30)$$

Observe that the sub-formula $X \succeq \bar{P} \wedge \neg(\bar{P} \succeq X)$ is up-monotone w.r.t. X.

Proposition 28 If $T(X)$ in formula (30) is the conjunction $T_1(X) \wedge T_2(X)$, where

- $T_1(X)$ is the conjunction of formulas of the form $\forall \bar{x}[X(\bar{s}) \rightarrow A(\bar{t})]$, where $\bar{s}, \bar{t}$ are arbitrary terms (in particular containing variables from $\bar{x}$) and A is a formula containing no occurrences of X
- $T_2(X)$ is up-monotone w.r.t. X,

then the extended DLS algorithm succeeds in the reduction of (30) to first-order logic with cardinality and preference constraints. ◁

As before, the class of formulas considered in Proposition 28 is not trivial and far out of scope of other known approaches.

Of course, the results provided by Propositions 26 and 28 can be extended to cover varied predicates using, e.g., results in (Conradie 2006).

Conclusions

We have proposed a novel framework for representing and reasoning about qualitative preferences with cardinality constraints using generalized circumscription. We have also shown how, in a number of cases, such second-order circumscription preference theories can be reduced constructively to logically equivalent first-order theories under certain explicitly stated conditions. In fact, these reduction results are independent of applications to preference modeling and open up new opportunities for using generalized circumscription in other contexts.

References

Abiteboul, S.; Hull, R.; and Vianu, V. 1996. *Foundations of Databases*. Addison-Wesley Pub. Co.

Ackermann, W. 1935. Untersuchungen über das eliminationsproblem der mathematischen logik. *Mathematische Annalen* 110:390–413.

Brafman, R. I.; Domshlak, C.; Shimony, S. E.; and Silver, Y. 2005. TCP-nets for preferences over sets. In Veloso, M. M., and Kambhampati, S., eds., *Proceedings of the 20th National Conf. on Artificial Intelligence and the 17th Conf. Innovative Applications*.

Brewka, G., and Eiter, T. 2000. Prioritizing default logic. In *Intellectics and Computational Logic – Papers in honor of Wolfgang Bibel*. Kluwer Academic Publishers.

Brewka, G.; Niemelä, I.; and Syrjänen, T. 2004. Logic programs with ordered disjunction. *Computational Intelligence* 20(2):335–357.

Brewka, G.; Niemelä, I.; and Truszczynski, M. 2005. Prioritized component systems. In Veloso, M. M., and Kambhampati, S., eds., *Proceedings of the 20th National Conf. on Artificial Intelligence and the 17th Conf. Innovative Applications*, 596–601.

Brewka, G. 2006. Preferences in answer set programming. In Marín, R.; Onaindia, E.; Bugarín, A.; and Santos, J., eds., *CAEPIA'2005*, volume 4177 of *Lecture Notes in Computer Science*, 1–10.

Chomicki, J., and Zhang, X. 2008. Profiling sets for preference querying. In *Proc. International Workshop on Logic in Databases (LID)*. To appear.

Conradie, W. 2006. On the strength and scope of DLS. *Journal of Applied Non-Classical Logics* 16(3–4):279–296.

Delgrande, J., and Schaub, T. 2000. Expressing preferences in default logic. *Artificial Intelligence* 123(1-2):41–87.

desJardins, M., and Wagstaff, K. 2005. DD-PREF: A language for expressing preferences over sets. In *Proceedings of the 20th National Conference on Artificial Intelligence*, 620–626.

Doherty, P.; Łukaszewicz, W.; and Szałas, A. 1996. A reduction result for circumscribed semi-Horn formulas. *Fundamenta Informaticae* 28(3-4):261–271.

Doherty, P.; Łukaszewicz, W.; and Szałas, A. 1997. Computing circumscription revisited. *Journal of Automated Reasoning* 18(3):297–336.

Doherty, P.; Łukaszewicz, W.; and Szałas, A. 1998. General domain circumscription and its effective reductions. *Fundamenta Informaticae* 36(1):23–55.

Doyle, J. 2004. Prospects for preferences. *Computational Intelligence* 20(2):111–136.

Ebbinghaus, H.-D., and Flum, J. 1995. *Finite Model Theory*. Heidelberg: Springer-Verlag.

Gabbay, D.; Schmidt, R.; and Szałas, A. 2008. *Second-Order Quantifier Elimination. Foundations, Computational Aspects and Applications*, volume 12 of *Studies in Logic*. College Publications.

Guha, S.; Gunopulos, D.; Koudas, N.; Srivastava, D.; and Vlachos, M. 2003. Efficient approximation of optimization queries under parametric aggregation constraints. In *Proceedings of the 29th VLDB Conference*, 778–789.

Gustafsson, J. 1996. An implementation and optimization of an algorithm for reducing formulas in second-order logic. Technical Report LiTH-MAT-R-96-04, http://www.ida.liu.se/labs/kplab/projects/dls/.

Junker, U.; Delgrande, J.; Doyle, J.; Rossi, F.; and Schaub, T. 2004. Special issue on preferences. *Computational Inteligence* 20(2).

Lifschitz, V. 1984. Some results on circumscription. In *Proceedings of the First AAAI NonMonotonic Reasoning Workshop*, 151–164.

Lifschitz, V. 1991. Circumscription. In Gabbay, D. M.; Hogger, C. J.; and Robinson, J. A., eds., *Handbook of Artificial Intelligence and Logic Programming*, volume 3, 297–352. Oxford University Press.

Łukaszewicz, W. 1990. *Non-Monotonic Reasoning - Formalization of Commonsense Reasoning*. Ellis Horwood Series in Artificial Intelligence. Ellis Horwood.

Magnusson, M. 2005. Implementation of DLS*. http://www.ida.liu.se/labs/kplab/projects/dlsstar/.

McCarthy, J. 1980. Circumscription: A form of non-monotonic reasoning. *Artificial Intelligence J.* 13:27–39.

McCarthy, J. 1986. Applications of circumscription to formalizing commonsense knowledge. *Artificial Intelligence J.* 28:89–116.

Nonnengart, A., and Szałas, A. 1998. A fixpoint approach to second-order quantifier elimination with applications to correspondence theory. In Orłowska, E., ed., *Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa*, 307–328. Springer Physica-Verlag.

Satoh, K. 1996. Disambiguation by prioritized circumscription. In *Proceedings of the 16th Conference on Computational Linguistics*, 901–906. Morristown, NJ, USA: Association for Computational Linguistics.

Wakaki, T.; Satoh, K.; and Nitta, K. 1997. Reasoning about dynamic preferences in circumscriptive theory by logic programming. *Journal of Advanced Computational Intelligence* 1(2):121–129.