TR-513

Formalizing Soft Constraints by Interpretation Ordering

by

K. Satoh

October, 1989

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

Formalizing Soft Constraints by Interpretation Ordering

Ken Satoh
Institute for New Generation Computer Technology
4-28 Mita 1-Chome Minato-ku, Tokyo 108 Japan
email: ksatoh@icot.jp

October 5, 1989
Revised: May 22, 1990

Abstract

In the area of synthesis problem such as scheduling, design and planning, there are two kinds of constraints. One kind is a hard constraint which a solution must satisfy. The other is a soft constraint which represents preference over solutions. In this paper, we provide a logical foundation of soft constraints. Let hard constraints be represented as first-order formulas. Then a solution to those hard constraints becomes an interpretation which satisfies the axiom set, and soft constraints can be regarded as providing an order over those interpretations, and the most preferred solutions are the most preferred interpretations in that order. We use a meta-language which presents a preference order directly. This meta-language can be translated into the second-order formula to provide a syntactical definition of the most preferred solutions. We also give a method to calculate all the most preferred solutions based on the proof theory and the model theory.

1 Introduction

There are a lot of research papers on constraints[5, 12]. But most of those papers only consider *hard constraints* which every solution must satisfy. However, in the area of *synthesis* problem such as job shop scheduling, circuit design and planning, there is another kind of constraints, that is, *soft constraints* which provides preferences over solutions [3, 4, 11].

This paper is an attempt to give a logical foundation of soft constraints by using an interpretation ordering which is a generalization of *circumscription* [7].

The idea of formalizing soft constraints is as follows. Let hard constraints be represented in the first-order formulas. Then an interpretation which satisfies all of those first-order formulas can be regarded as a solution. Then soft constraints can be regarded as an order over those interpretations because soft constraints represent criteria over solutions to choose the most preferable ones.

A model theory in circumscription defines an order over interpretations of first-order formulas based on set-theoretic inclusion of extensions of certain predicates. However, if we use circumscription for soft constraints, we must specify which predicates are minimized to get a desired result and it is not always clear how to choose those predicates. Instead, in this paper we use a meta-language [9] which is a generalized form of circumscription and provides a direct representation of soft constraints.

The structure of the paper is as follows. In section 2, we consider some aspects of soft constraints. In section 3, we review a meta-language and use it to provide syntactical definition of the most preferred models based on a second-order language. In section 4, we give examples of soft constraints expressed by meta-language. In section 5, we consider two kinds of proof methods. One is a proof-theoretic method which uses inference rules of second-order logic and the other is a model-theoretic procedure which computes all the most preferred models in propositional logic and the first-order logic with domain closure axioms. In section 6, we compare with some related works. And finally, we talk about future research. Proofs of the theorems are in the appendix.

2 Soft Constraints

In this section, we consider some aspects of soft constraints.

Suppose the following situation where we make schedule for meeting of the president, vice president and a manager of a company.

1. The president *must* attend the meeting.
2. The vice president should *preferably* attend the meeting.
3. The manager also should *preferably* attend the meeting. However, the schedule of the vice president is *prioritized* to the schedule of the manager.

The first condition is a required constraint and a solution must satisfy the condition. However, the second and the third conditions are soft constraints and are regarded as criteria to choose the most preferable models among the solutions of hard constraints. In this case, if there are solutions both of which satisfy the hard constraint, we choose solutions which also satisfy the soft constraints. However, if there is no solution which satisfies soft constraints, we just ignore those soft constraints. Soft constraints, therefore, may not always be satisfied.

Looking into the third conditions, there is a note that the second condition is prioritized to the third condition. If there are two solutions in one of which the second condition is satisfied and the third is not satisfied, and in the other of which the third condition is satisfied and the second is not, then we choose the former because of the priority.

Consider another situation where a value in a part of a solution should be as large as possible. In this case, we can regard those conditions as a criterion which chooses a solution which has a larger value for that part.

To summarize, there are the three kinds of soft constraints stated below.

1. **Soft constraints without priorities:**
   In this case, the most preferable solutions are the ones that satisfy soft constraints as much as possible. However, if one solution satisfies a set of soft constraints and another solution satisfies a different set of soft constraints and the former set does not include the latter set and vice versa, we cannot say which solution is better.
2. **Soft constraints with priorities:**
   In this case, we may be able to distinguish solutions which we cannot

distinguish in the previous case. The most preferable solutions are the ones that satisfy as many prioritized constraints as possible.

3. **General soft constraints:**
   Any order over solutions can be regarded as a general case of soft constraints. If we can define an order of solutions, we choose the most preferable solutions in the order. The above two cases are an instance of this case because those cases define an order of solutions.

The most preferable solutions can be defined as follows by an order of solutions. Let S be a set of the most preferable solutions and S_0 be a set of the solutions satisfying hard constraints and $<$ be an order over solutions.

$$S \stackrel{\text{def}}{=} \{\sigma | \sigma \in S_0 \text{ and there exists no } \theta \in S_0 \text{ such that } \theta < \sigma\}.$$

where smaller solutions are preferable solutions.

We can paraphrase the above definition into first-order logic as follows. Hard constraints can be regarded as the first-order axiom set which solutions must satisfy. Then a solution to those hard constraints becomes an interpretation which satisfies the axiom set, and soft constraints can be regarded as providing an order over those interpretations, and the most preferred solutions are the most preferred interpretations in that order. Then, we can define a set of the most preferable solutions as follows. Let C be a formula which represents a conjunction of hard constraints, and M, M' be logical interpretations, and $\prec$ be an order over interpretations.

$$\{M | M \models C \text{ and } M \text{ is comparable with } M' \text{ and there exists no } M' \text{ such that } M' \models C \text{ and } M' \prec M\}.$$

where smaller interpretations are preferable.

Then, we can see that the logical interpretation in the above set is minimal model with respect to the order $\prec$.

3 Interpretation Ordering by Meta Language

In this section, we review our previous work [9] on meta-language by which we represent order over interpretations.

3.1 Preliminaries

In second-order logic, we shall use predicate variables in addition to symbols from first-order logic. When we talk about a variable v, v is an individual variable or a predicate variable.

An *interpretation* M consists of a nonempty set D, called the *domain* of the interpretation, and the following assignment mapping.

1. Each individual constant a is mapped onto an element $(a)^M$ of D.
2. Each n-ary function constant f^n is mapped onto a function $(f^n)^M$ from D^n to D.
3. Each n-ary predicate constant P^n is mapped onto a subset $(P^n)^M$ of D^n, and each propositional constant is mapped onto one of the truth values, **T** or **F**.

We consider the following assignment function ϕ with respect to domain D.

1. Each individual variable x is mapped onto an element of D.
2. Each n-ary predicate variable p^n is mapped onto a subset $\phi(p^n)$ of D^n, and each propositional variable is mapped onto one of the truth values.

We denote all assignment functions (with respect to D) as Φ_D. We denote an assignment function ϕ_v, which differs from ϕ in at most the assignment of the variable v. We write as $\phi_{v_1 \dots v_n}$ an assignment function which differs from $\phi_{v_1 \dots v_{n-1}}$ in at most the assignment of v_n.

Let M be an interpretation with domain D, and ϕ be an assignment function with respect to the domain D, and t be a term. We extend an assignment function ϕ to a function ϕ^M that assigns to each term t an element $\phi^M(t)$ in D as follows.

1. If t is an individual constant, then $\phi^M(t) = (t)^M$
2. If t is an individual variable, then $\phi^M(t) = \phi(t)$.
3. If t is of the form $f^n(t_1, ..., t_n)$, then $\phi^M(t) = (f^n)^M(\phi^M(t_1), ..., \phi^M(t_n))$.

Let M be an interpretation with domain D. An assignment function ϕ (with respect to the domain D) satisfies a wff A in M (written as $M \models_\phi A$) if and only if the following conditions are satisfied.

1. If A is of the form $P^n(t_1, ..., t_n)$ where P^n is a n-ary predicate constant, then $< \phi^M(t_1), ..., \phi^M(t_n) > \in (P^n)^M$. If A is a propositional constant P, then $(P)^M = \mathbf{T}$.
2. If A is of the form $p^n(t_1, ..., t_n)$ where p^n is a n-ary predicate variable, then $< \phi^M(t_1), ..., \phi^M(t_n) > \in \phi(p^n)$. If A is a propositional variable p, then $(p)^M = \mathbf{T}$.
3. If A is of the form $\neg B$, not $M \models_\phi B$ (written as $M \not\models_\phi B$).
4. If A is of the form $B \supset C$, either $M \not\models_\phi B$ or $M \models_\phi C$.
5. If A is of the form $\forall v B$ where v is a variable, for every ϕ_v in Φ_D, $M \models_{\phi_v} B$.

If for every $\phi \in \Phi_D$, $M \models_\phi A$ then we write $M \models A$ and we call M a *model* of A.

3.2 Wdr(Well-Defined Relation)

Now, we define a *well-defined relation(wdr)* with respect to *comparable* interpretations M and M' as follows.

Let M and M' be interpretations. M and M' are *comparable* with respect to a tuple of predicate constants $\mathbf{P}$ if and only if the following conditions are satisfied.

1. M and M' have the same domain D.
2. For every individual constant, function constant, and predicate constant not in $\mathbf{P}$, M and M' have the same interpretation.

Definition 1 *Well-Defined Relation (wdr)*

Let M and M' be comparable interpretations with respect to a tuple of predicate constants $\mathbf{P}$. We say the following expressions are well-defined relations (wdrs) with a top-level assignment function (taf) ϕ with respect to M and M'.

1. If A is a wff and ϕ is an assignment function, then $M \models_\phi A$, $M' \models_\phi A$ is a wdr with a taf ϕ called an *atomic wdr*.
2. If $\mathcal{A}$ is a wdr with a taf ϕ, then $\neg\mathcal{A}$ is also a wdr with a taf ϕ.
3. If $\mathcal{A}$ and $\mathcal{B}$ are wdrs with a taf ϕ, then $\mathcal{A} \supset \mathcal{B}$ is also a wdr with a taf ϕ.

4. If $\mathcal{A}$ is a wdr with a taf ϕ_v, then $(\forall\phi_v\in\Phi_D)\mathcal{A}$ is a wdr with a taf ϕ called a *quantified wdr*.

Do not confuse the meta-logical connectives with ordinary logical connectives. The meta-logical connectives are the following abbreviations of English sentences used in the definition of satisfaction.

1. $M \models_\phi A$ means "ϕ satisfies A in M".
2. $\neg\mathcal{A}$ means "$\mathcal{A}$ is not true".
3. $\mathcal{A} \supset \mathcal{B}$ means "either $\mathcal{A}$ is not true or $\mathcal{B}$ is true".
4. $(\forall\phi_v\in\Phi_D)\mathcal{A}$ means "for every ϕ_v in Φ_D which differs from ϕ in at most the assignment of v, $\mathcal{A}$ is true".

If $\mathcal{A}$ and $\mathcal{B}$ are wdrs with a taf ϕ, then $\mathcal{A} \wedge \mathcal{B}$, $\mathcal{A} \vee \mathcal{B}$, $\mathcal{A} \equiv \mathcal{B}$ are abbreviations for $\neg(\mathcal{A} \supset \neg\mathcal{B})$, $(\neg\mathcal{A}) \supset \mathcal{B}$, $(\mathcal{A} \supset \mathcal{B}) \wedge (\mathcal{B} \supset \mathcal{A})$ respectively. And if $\mathcal{A}$ is a wdr with a taf ϕ_v, $(\exists\phi_v\in\Phi_D)\mathcal{A}$ is an abbreviation for $\neg((\forall\phi_v\in \Phi_D)\neg\mathcal{A})$.

3.3 Translation from a Wdr to an Atomic Meta-relation

In this subsection, we introduce a translation from a wdr to an atomic meta-relation of M.

A predicate variable and a predicate constant are *similar* if and only if they have the same arity. A tuple of predicate variables $\mathbf{p}(= \langle p_1, ..., p_n\rangle)$ and a tuple of predicate constants $\mathbf{P}(= \langle P_1, ..., P_n\rangle)$ are *similar* (or we say $\mathbf{p}$ is similar to $\mathbf{P}$) if and only if each variable p_i of $\mathbf{p}$ and each corresponding constant P_i of $\mathbf{P}$ are similar. We write a wff A with some of the predicate constants $P_1, ..., P_n$ in a tuple of predicate constants $\mathbf{P}$ as $A(\mathbf{P})$. Then we write as $A(\mathbf{p})$ the result of substituting in A the predicate variables $p_1, ..., p_n$ for all occurrences of $P_1, ..., P_n$, respectively. By the definition of wdr, we can convert any wdr to an atomic wdr of M by the following translation.

Translation: From a wdr to an atomic wdr of M
Let M' and M be comparable with respect to $\mathbf{P}$, and let $\mathcal{R}(M', M)_\phi$ be a wdr with a taf ϕ. Let $\mathbf{p}$ be similar to $\mathbf{P}$ such that every predicate variable in $\mathbf{p}$ is not contained in $\mathcal{R}(M', M)_\phi$.

1. $\mathcal{R}(M', M)_\phi$ is of the form $M \models_\phi A$. It is translated into itself.

2. $\mathcal{R}(M', M)_\phi$ is of the form $M' \models_\phi A(\mathbf{P})$. It is translated into $M \models_\phi A(\mathbf{p})$.
3. $\mathcal{R}(M', M)_\phi$ is of the form $\neg\mathcal{A}(M', M)_\phi$. It is translated into $M \models_\phi \neg A(\mathbf{p})$, where $\mathcal{A}(M', M)_\phi$ is translated into $M \models_\phi A(\mathbf{p})$.
4. $\mathcal{R}(M', M)_\phi$ is of the form $\mathcal{A}(M', M)_\phi \supset \mathcal{B}(M', M)_\phi$. It is translated into $M \models_\phi A(\mathbf{p}) \supset B(\mathbf{p})$, where where $\mathcal{A}(M', M)_\phi$ is translated into $M \models_\phi A(\mathbf{p})$, and $\mathcal{B}(M', M)_\phi$ is translated into $M \models_\phi B(\mathbf{p})$.
5. $\mathcal{R}(M', M)_\phi$ is of the form $(\forall\phi_v \in \Phi_D)\mathcal{A}(M', M)_{\phi_v}$. It is translated into $M \models_\phi \forall v A(\mathbf{p})$, where $\mathcal{A}(M', M)_{\phi_v}$ is translated into $M \models_{\phi_v} A(\mathbf{p})$.

Example 1 *(Translation from a wdr into an atomic wdr)*

Let M and M' be comparable with respect to $\langle P \rangle$, let $\langle p \rangle$ be similar to $\langle P \rangle$, and let ϕ be an assignment function. Let $\mathcal{R}(M', M)_\phi$ be the following wdr. We show a process of translation.

$$(\forall\phi_x \in \Phi_D)((M' \models_{\phi_x} P(x)) \supset (M \models_{\phi_x} P(x))) \wedge$$
$$\neg(\forall\phi_x \in \Phi_D)((M \models_{\phi_x} P(x)) \supset (M' \models_{\phi_x} P(x)))$$
$$\Longrightarrow (\forall\phi_x \in \Phi_D)((M \models_{\phi_x} p(x)) \supset (M \models_{\phi_x} P(x))) \wedge$$
$$\neg(\forall\phi_x \in \Phi_D)((M \models_{\phi_x} P(x)) \supset (M \models_{\phi_x} p(x)))$$
$$\Longrightarrow (\forall\phi_x \in \Phi_D)(M \models_{\phi_x} (p(x) \supset P(x))) \wedge \neg(\forall\phi_x \in \Phi_D)(M \models_{\phi_x} (P(x) \supset p(x)))$$
$$\Longrightarrow (M \models_\phi \forall x(p(x) \supset P(x))) \wedge \neg(M \models_\phi \forall x(P(x) \supset p(x)))$$
$$\Longrightarrow (M \models_\phi \forall x(p(x) \supset P(x))) \wedge (M \models_\phi \neg\forall x(P(x) \supset p(x)))$$
$$\Longrightarrow M \models_\phi (\forall x(p(x) \supset P(x)) \wedge \neg\forall x(P(x) \supset p(x))) \quad \square$$

3.4 Syntactic Definition of the Most Preferable Solutions in the Second-order Formula

In this section, we show a syntactic definition of the most preferable solutions by combining hard constraints represented as the first-order axioms and soft constraints represented in meta language.

If M and M' are comparable and ϕ satisfies the following condition in Lemma 1, we can show that a wdr is true if and only if an atomic wdr of its translation is true.

Lemma 1 *Let M' and M be interpretations with domain D which are comparable with respect to* $\mathbf{P}$, *and let* $\mathcal{R}(M', M)_\phi$ *be a wdr. Let* $\mathbf{p}$ *be similar to* $\mathbf{P}$ *such that every predicate variable in* $\mathbf{p}$ *is not contained in* $\mathcal{R}(M', M)_\phi$. *Let* $M \models_\phi R(\mathbf{p})$ *be an atomic wdr of its translation. If for every P_i in* $\mathbf{P}$ *and p_i in* $\mathbf{p}$, $\phi(p_i) = (P_i)^{M'}$, *then* $\mathcal{R}(M', M)_\phi$ *is true if and only if* $M \models_\phi R(\mathbf{p})$ *is true.*

Now we show the following theorem closely related to a link between minimal models in preference order and second-order wff.

Theorem 1 *Let M' and M be interpretations with domain D which are comparable with respect to* $\mathbf{P}$, *and let* $\mathcal{R}(M', M)_\phi$ *be a wdr. Let* $\mathbf{p}$ *be similar to* $\mathbf{P}$ *such that every predicate variable in* $\mathbf{p}$ *is not contained in* $\mathcal{R}(M', M)_\phi$. *Let its translation using* $\mathbf{p}$ *be* $M \models_\phi R(\mathbf{p})$. *There exists M' such that* $\mathcal{R}(M', M)_\phi$ *is true if and only if* $M \models_\phi \exists p_1 ... \exists p_n R(\mathbf{p})$ *is true.*

We say that M is a *minimal model* with respect to a first-order formula $A(\mathbf{P})$ and a wdr $\mathcal{R}(M', M)_\phi$ if and only if the following condition is satisfied.

> *M is a model of $A(\mathbf{P})$ and for every interpretation M' which is comparable with M, if M' is a model of $A(\mathbf{P})$ then for every assignment function ϕ, $\neg\mathcal{R}(M', M)_\phi$ is true.*

Then next corollary shows that if an interpretation satisfies a second order formula obtained from $A(\mathbf{P})$ and an atomic wdr translated from a wdr, it is a minimal model with respect to $A(\mathbf{P})$ and that wdr.

Corollary 1 *M is a minimal model with respect to a first-order formula $A(\mathbf{P})$ and a wdr $\mathcal{R}(M', M)_\phi$ if and only if M is a model of:*

$$A(\mathbf{P}) \wedge \neg\exists p_1 ... \exists p_n (A(\mathbf{p}) \wedge R(\mathbf{p})),$$

where a translation of the wdr using $\mathbf{p}$ *is* $M \models_\phi R(\mathbf{p})$.

Example 2 *(Syntactic Definition of a Minimal Model)*

Consider the wdr $\mathcal{R}(M', M)_\phi$ of Example 1. The wdr $\mathcal{R}(M', M)_\phi$ means that M' is more preferable than M if and only if M' have a smaller extension of P than M. Since the wdr is translated into:

$$M \models_\phi (\forall x(p(x) \supset P(x)) \wedge \neg \forall x(P(x) \supset p(x))),$$

M is a minimal model w.r.t $A(P)$ and the wdr if and only if M is a model of the following formula:

$$A(P) \wedge \neg \exists p(A(p) \wedge \forall x(p(x) \supset P(x)) \wedge \neg \forall x(P(x) \supset p(x))),$$

which is a definition of circumscription to minimize P. □

If we regard $A(\mathbf{P})$ as hard constraints and $\mathcal{R}(M', M)_\phi$ as soft constraints, and assume that M' is preferable, then minimal models are the most preferable models of hard constraints in the order defined by soft constraints. By Corollary 1, minimal models are models of:

$$A(\mathbf{P}) \wedge \neg \exists p_1 ... \exists p_n (A(\mathbf{p}) \wedge R(\mathbf{p})).$$

This formula, therefore, gives a syntactical definition of the most preferable solutions.

4 Representation of Soft Constraints

In this section, we provide a way of representing soft constraints. Before doing that, we introduce notations for brevity. Let $\mathbf{P}$ be a tuple of predicate constants or a tuple of predicate variables, and $x_1, ..., x_n$ be individual variables. $E(\mathbf{P}, x_1, ..., x_n)$ denotes a formula which includes some of those predicate constants and those individual variables as free variables.
$M' \leq_\phi^{E(\mathbf{P}, x_1, ..., x_n)} M$ is an abbreviation of the following wdr:

$$\begin{aligned} &\forall \phi_{x_1} \in \Phi_D ... \forall \phi_{x_1 x_2 ... x_n} \in \Phi_D (\\ &\quad (M \models_{\phi_{x_1 ... x_n}} E(\mathbf{P}, x_1, ..., x_n)) \supset (M' \models_{\phi_{x_1 ... x_n}} E(\mathbf{P}, x_1, ..., x_n))). \end{aligned}$$

And $M' =_\phi^{E(\mathbf{P}, x_1, ..., x_n)} M$ is an abbreviation of the following wdr:

$$(M' \leq_\phi^{E(\mathbf{P}, x_1, ..., x_n)} M) \wedge (M \leq_\phi^{E(\mathbf{P}, x_1, ..., x_n)} M')$$

$E(\mathbf{P}, x_1, ..., x_n) \leq E(\mathbf{Q}, x_1, ..., x_n)$ is an abbreviation of the following formula:

$$\forall x_1 \forall x_2 ... \forall x_n (E(\mathbf{Q}, x_1, ..., x_n) \supset E(\mathbf{P}, x_1, ..., x_n))$$

And $E(\mathbf{P}, x_1, ..., x_n) = E(\mathbf{Q}, x_1, ..., x_n)$ is an abbreviation of the following formula:

$$(E(\mathbf{P}, x_1, ..., x_n) \leq E(\mathbf{Q}, x_1, ..., x_n)) \wedge (E(\mathbf{Q}, x_1, ..., x_n) \leq E(\mathbf{P}, x_1, ..., x_n)).$$

4.1 Soft Constraints without Priorities

Let conditions which a solution should satisfy as much as possible be represented as the following formulas:

$$E_1(\mathbf{P}, x_1, ..., x_n), ..., E_m(\mathbf{P}, x_1, ..., x_n).$$

Then, we can represent a soft constraint by the above formulas as follows:

$$\mathcal{R}(M', M)_\phi \stackrel{\text{def}}{=} (M' \leq_\phi M) \wedge \neg(M \leq_\phi M'),$$

where $M' \leq_\phi M$ is an abbreviation of the following wdr:

$$\bigwedge_{i=1}^{m}(M' \leq_\phi^{E_i(\mathbf{P}, x_1, ..., x_n)} M).$$

This relation intuitively means that an interpretation which satisfies $E_1, ..., E_m$ as much as possible is preferable. Then, we can show a syntactic definition of the most preferable solutions in the above order which satisfy hard constraints denoted as $A(\mathbf{P})$:

$$\begin{aligned}&A(\mathbf{P}) \wedge \neg\exists\mathbf{p}(A(\mathbf{p})\wedge\\ &\quad\bigwedge_{i=1}^{m}(E_i(\mathbf{p}, x_1, ..., x_n) \leq E_i(\mathbf{P}, x_1, ..., x_n)) \wedge\\ &\quad\neg\bigwedge_{i=1}^{m}(E_i(\mathbf{P}, x_1, ..., x_n) \leq E_i(\mathbf{p}, x_1, ..., x_n))).\end{aligned}$$

For example, if $P(x) \supset Q(x)$ should be satisfied as much as possible, then the syntactic definition of the most preferable models are as follows:

$$\begin{aligned}&A((P, Q)) \wedge \neg\exists p\exists q(A((p, q))\wedge\\ &\quad\forall x((P(x) \supset Q(x)) \supset (p(x) \supset q(x)))\wedge\\ &\quad\neg\forall x((p(x) \supset q(x)) \supset (P(x) \supset Q(x))))\end{aligned}$$

4.2 Soft Constraints with Priorities

Let formulas which should be satisfied in the first place be

$$E_1^1(\mathbf{P}, x_1, ..., x_n), ..., E_{m_1}^1(\mathbf{P}, x_1, ..., x_n),$$

and formulas which should be satisfied in the second place be

$$E_1^2(\mathbf{P}, x_1, ..., x_n), ..., E_{m_2}^2(\mathbf{P}, x_1, ..., x_n), ...$$

and formulas which should be satisfied in the k-th place be

$$E_1^k(\mathbf{P}, x_1, ..., x_n), ..., E_{m_k}^k(\mathbf{P}, x_1, ..., x_n).$$

Then a soft constraint by the above formulas is defined as follows:

$$\mathcal{R}(M', M)_\phi \stackrel{\text{def}}{=} (M' \leq_\phi M) \wedge \neg(M \leq_\phi M').$$

where $M' \leq_\phi M$ is an abbreviation of $(M' \leq_\phi^1 M) \wedge ... \wedge (M' \leq_\phi^k M)$ and $M' \leq_\phi^i M$ is an abbreviation of the following wdr:

$$(\bigwedge_{j=1}^{i-1} \bigwedge_{l=1}^{m_j}(M' =_\phi^{E_l^j(\mathbf{P}, x_1, ..., x_n)} M)) \supset (\bigwedge_{l=1}^{m_i}(M' \leq_\phi^{E_l^i(\mathbf{P}, x_1, ..., x_n)} M)),$$

where $M' \leq_\phi^1 M$ is a wdr without conditional part.

This relation means that interpretations which satisfy $E_1^1, ..., E_{m_1}^1$ as much as possible is preferable and if there are interpretations which satisfy the same formulas in the first place, then interpretations which satisfy $E_1^2, ..., E_{m_2}^2$ as much as possible are preferable and, ... if there are interpretations which satisfy the same formulas in the $(k-1)$-th place, then interpretations which satisfy $E_1^k, ..., E_{m_k}^k$ as much as possible are preferable.

Then, we can show a syntactic definition of the most preferable solutions in the above order which satisfy hard constraints denoted as $A(\mathbf{P})$:

$$A(\mathbf{P}) \wedge \neg\exists \mathbf{p}(A(\mathbf{p}) \wedge (\mathbf{p} \leq \mathbf{P}) \wedge \neg(\mathbf{P} \leq \mathbf{p})),$$

where $\mathbf{p} \leq \mathbf{P}$ is an abbreviation of $(\mathbf{p} \leq^1 \mathbf{P}) \wedge ... \wedge (\mathbf{p} \leq^k \mathbf{P})$ and $\mathbf{p} \leq^i \mathbf{P}$ is an abbreviation of the following formula:

$$(\bigwedge_{j=1}^{i-1} \bigwedge_{l=1}^{m_j}(E_l^j(\mathbf{p}, x_1, ..., x_n) = E_l^j(\mathbf{P}, x_1, ..., x_n))) \supset$$
$$(\bigwedge_{l=1}^{m_i}(E_l^i(\mathbf{p}, x_1, ..., x_n) \leq E_l^i(\mathbf{P}, x_1, ..., x_n)))$$

The meeting scheduling problem in Section 2 belongs to this case and we explain it in the next subsection.

4.3 Meeting Scheduling

We show a wdr which corresponds to soft constraints in the meeting scheduling. $C(x)$ represents that the meeting will be held on day x and $P(x), V(x), M(x)$ represent that the president, the vice president and the manager attend the meeting on day x. Let $\mathbf{P}$ be $\{C, P, V, M\}$. Let us represent hard constraints for inconvenient dates. For example, if day x is inconvenient for the president then $\neg P(x)$ is included in hard constraints. As another example, a hard constraint in which the president must attend the meeting is represented as follows:

$$\forall x(C(x) \supset P(x)).$$

The soft constraint in which the vice president should preferably attend the meeting means that the following formula should be satisfied as much as possible:

$$E_1^1(\mathbf{P}, x) = C(x) \supset V(x).$$

And the soft constraint in which the manager should preferably attend the meeting means that the following formula should be satisfied as much as possible:

$$E_1^2(\mathbf{P}, x) = C(x) \supset M(x).$$

And the priority of the schedule of the vice president to the schedule of the manager means that $E_1^1(\mathbf{P}, x)$ should be satisfied more preferably than $E_1^2(\mathbf{P}, x)$. Then the soft constraint by the above formulas is defined as follows:

$$\begin{aligned}
&\mathcal{R}(M', M)_\phi \stackrel{\text{def}}{=} \\
&\quad (M' \leq_\phi^{E_1^1(\mathbf{P},x)} M) \wedge ((M' =_\phi^{E_1^1(\mathbf{P},x)} M) \supset (M' \leq_\phi^{E_1^2(\mathbf{P},x)} M)) \wedge \\
&\quad \neg((M \leq_\phi^{E_1^1(\mathbf{P},x)} M') \wedge ((M =_\phi^{E_1^1(\mathbf{P},x)} M') \supset (M \leq_\phi^{E_1^2(\mathbf{P},x)} M'))).
\end{aligned}$$

This definition intuitively means that $C(x) \supset V(x)$ should be satisfied as much as possible and then $C(x) \supset M(x)$ should be satisfied as much as possible in a situation where $C(x) \supset V(x)$ are maximally satisfied. By this definition, we first consider the vice president's schedule and then consider the manager's schedule. Let hard constraints be represented as $A((C, P, V, M))$. Then a syntactic definition of the most preferable solutions is as follows:

$$\begin{aligned}
&A((C, P, V, M)) \wedge \neg\exists c\exists p\exists v\exists m(A((c, p, v, m)) \wedge \\
&\quad ((c, p, v, m) \leq (C, P, V, M)) \wedge \neg((C, P, V, M) \leq (c, p, v, m))),
\end{aligned}$$

where $(c, p, v, m) \leq (C, P, V, M)$ is an abbreviation of the following formula:

$$\begin{aligned}
&\forall x((C(x) \supset V(x)) \supset (c(x) \supset v(x))) \wedge \\
&(\forall x((C(x) \supset V(x)) \equiv (c(x) \supset v(x))) \supset \\
&\quad \forall x((C(x) \supset M(x)) \supset (c(x) \supset m(x)))).
\end{aligned}$$

4.4 General Soft Constraints

By a wdr, we can represent an order which prefers a solution which has a larger value in a part of a solution. For example, if we would like to have a solution where x of $Q(x)$ is maximum, a soft constraint for that order is defined as follows [1]:

$$\mathcal{R}(M', M)_\phi \stackrel{\text{def}}{=} \\ \forall \phi_x \in \Phi_D \forall \phi_{xy} \in \Phi_D(((M' \models Q(x)) \wedge (M \models Q(y))) \supset (M' \models x > y)).$$

This order means that an interpretation in which the value of x of $Q(x)$ is larger is preferred. A syntactic definition of the most preferable solutions is as follows:

$$A(Q) \wedge \neg \exists q(A(q) \wedge \forall x \forall y((q(x) \wedge Q(y)) \supset x > y)).$$

5 Method of Calculating the Most Preferable Models

There are two kinds of methods of calculating the most preferable models. One is a proof theoretic method and the other is model theoretic method. In this paper, we describe both methods.

5.1 Proof Theoretic Method

In the proof theoretic method, we calculate a formula which is true in all the most preferable models by inference rules of second-order logic [2]. We show an example of the method by using the meeting scheduling problem. In the meeting scheduling problem, the syntactic definition of the most prefererable solutions is as follows:

$$A((C, P, V, M)) \wedge \neg \exists c \exists p \exists v \exists m(A((c, p, v, m)) \wedge \\ ((c, p, v, m) \leq (C, P, V, M)) \wedge \neg((C, P, V, M) \leq (c, p, v, m))), \qquad (1)$$

[1] However, we must have a hard constraint in which x of $Q(x)$ is unique.

[2] However, in the second-order logic, no inference rule is complete. That is, there may be a formula which is true in all the most preferable models and cannot be derived by any inference rules.

where $(c, p, v, m) \leq (C, P, V, M)$ is an abbreviation of the following formula:

$$\begin{aligned}&\forall x((C(x) \supset V(x)) \supset (c(x) \supset v(x)))\wedge\\&(\forall x((C(x) \supset V(x)) \equiv (c(x) \supset v(x))) \supset\\&\quad \forall x((C(x) \supset M(x)) \supset (c(x) \supset m(x)))).\end{aligned}$$

Now, using the above definition of soft constraints, we calculate the most preferable meeting date. Suppose we have the following hard constraints.

1. The meeting must be held:

$$\exists x C(x). \tag{2}$$

2. The president must attend the meeting:

$$\forall x(C(x) \supset P(x)). \tag{3}$$

3. We consider a meeting schedule for day 1, 2 and 3. Then we have the following domain closure axiom:

$$\forall x((x = 1 \vee x = 2 \vee x = 3)). \tag{4}$$

And we have the following axiom because 1,2,3 are all different:

$$1 \neq 2 \wedge 2 \neq 3 \wedge 3 \neq 1. \tag{5}$$

4. The president cannot attend the meeting on day 1:

$$\neg P(1). \tag{6}$$

5. The manager cannot attend the meeting on day 2:

$$\neg M(2). \tag{7}$$

Then $A((C, P, V, M))$ becomes conjunction of those formulas. Now, we calculate most preferable meeting date under the above hard constraints.
(1) is equivalent to the following formula:

$$\begin{aligned}&A((C, P, V, M)) \wedge \forall c \forall p \forall v \forall m((A((c, p, v, m)) \wedge\\&\quad ((c, p, v, m) \leq (C, P, V, M))) \supset ((c, p, v, m) = (C, P, V, M))),\end{aligned} \tag{8}$$

where $(c, p, v, m) = (C, P, V, M)$ is an abbreviation of the following formula:

$$\forall x((C(x) \supset V(x)) \equiv (c(x) \supset v(x))) \wedge \forall x((C(x) \supset M(x)) \equiv (c(x) \supset m(x))).$$

Any formula obtained by replacing c, p, v, m in (8) by any predicates must be true. Suppose we replace each of c, p, v, m by $\lambda x(x = 3)$. Then, $A((c, p, v, m)) \wedge ((c, p, v, m) \leq (C, P, V, M))$ in (8) becomes true. And $(c, p, v, m) = (C, P, V, M)$ becomes the following:

$$\forall x(C(x) \supset V(x)) \wedge \forall x(C(x) \supset M(x)).$$

Then from (7), we get $\neg C(2)$. And, from (3) and (6), we get $\neg C(1)$. Therefore, from (2) and (4) and (5), we get $C(3)$. This means that day 3 is the most preferable meeting date in this situation.

The conclusion may be withdrawn by adding another constraint. For example, suppose a new constraint that the vice president cannot attend the meeting on day 3 is added. That is, the following constraint is added.

$$\neg V(3). \tag{9}$$

Then, $A((C, P, V, M))$ becomes conjunction of the previous $A((C, P, V, M))$ and (9). Suppose we replace each of c, p, v in (8) by $\lambda x(x = 2)$ and m by $\lambda x False$. Then, we can get the following from (8).

$$\forall x(C(x) \supset V(x)).$$

Then from (9), we get $\neg C(3)$. And, from (3) and (6), we get $\neg C(1)$. Therefore, from (2) and (4) and (5), we get $C(2)$. This means that day 2 is the most preferable meeting date in this new situation because the schedule of the vice president has the priority to the schedule of the manager. This expresses nonmonotonic character of soft constraints.

5.2 Model Theoretic Method

In the model theoretic method, we pick up all the most preferable models by interpretation ordering.

This method consists of two steps. In the first step, we calculate all models which satisfy hard constraints. And, in the second step, we check whether a model is a minimal model in the order over interpretations.

We describe this procedure in propositional logic. Let hard constraints be represented as $A(\mathbf{P})$ and an order over interpretation with respect to a tuple of propositional constants $\mathbf{P}$ be represented as $\mathcal{R}(M', M)_\phi$.

1. We construct an interpretation by assigning every propositional constant to a truth value and check if the interpretation satisfies $A(\mathbf{P})$. If it satisfies $A(\mathbf{P})$, register it as a model. Let a set of models of $A(\mathbf{P})$ be $\mathcal{M}$.
2. For all $M \in \mathcal{M}$, we check if there exists $M' \in \mathcal{M}$ such that $\mathcal{R}(M', M)_\phi$ is true. If there exists no such model, then register M as a minimal model.

Both the first and the second step will terminate in propositional logic, because the number of models is finite. However, the procedure may not terminate in the first-order logic, because the domain of models may be infinite. If we have a finite domain closure axiom in which we can count all of objects in the domain, we can translate all quantified formulas (and also meta-quantified formulas) into the formula which consists of ground sentences only. $\forall x P(x)$ becomes $\bigwedge_{i=1}^{n} P(a_i)$ where $a_1, ..., a_n$ are all terms in domain closure axiom. $\forall \phi_v \in \Phi_D \mathcal{A}$ becomes $\bigwedge_{i=1}^{n} \mathcal{A}_{a_i|v}$ where $\mathcal{A}_{a_i|v}$ is a wdr obtained by replacing v by a_i. Then, we can calculate all minimal models in a similar way to propositional logic by regarding an atomic formula in the translated formula as a proposition.

We explain the above translation and a process of calculating all minimal models by using the meeting scheduling problem. For example, let the meeting be scheduled in one day out of 1, 2, 3. Then we have a domain closure axiom:

$$\forall x((x = 1 \vee x = 2 \vee x = 3)).$$

We have the following axiom because 1,2,3 are all different.

$$1 \neq 2 \wedge 2 \neq 3 \wedge 3 \neq 1.$$

Given the above axioms, we only need to consider the following atomic formulas.

$$C(1), C(2), C(3), P(1), P(2), P(3),$$
$$V(1), V(2), V(3), M(1), M(2), M(3).$$

Henceforth, we consider the above atomic formulas as the following propositions:

$$C_1, C_2, C_3, P_1, P_2, P_3, V_1, V_2, V_3, M_1, M_2, M_3$$

Then, an interpretation can be constructed by giving a truth value to every proposition.

We represent an interpretation as a set of propositional constants or negation of propositional constants which are true in the interpretation. For example, an interpretation in which C_2, P_2, V_2 are true and the others are false can be expressed as:

$$\{\neg C_1, C_2, \neg C_3, \neg P_1, P_2, \neg P_3, \neg V_1, V_2, \neg V_3, \neg M_1, \neg M_2, \neg M_3\}$$

We have the following hard constraints.

1. The meeting must be held:

$$\exists x C(x)$$

From the domain closure axiom, the above is equivalent to:

$$C_1 \vee C_2 \vee C_3.$$

2. The president must attend the meeting:

$$\forall x (C(x) \supset P(x)).$$

This is equivalent to:

$$(C_1 \supset P_1) \wedge (C_2 \supset P_2) \wedge (C_3 \supset P_3).$$

3. Since $P(x)$ expresses that the president attends the meeting on day x, if it is true, $C(x)$ (the meeting is held on day x) is also true:

$$\forall x (P(x) \supset C(x)).$$

This is equivalent to:

$$(P_1 \supset C_1) \wedge (P_2 \supset C_2) \wedge (P_3 \supset C_3).$$

4. The same thing holds if the vice president or the manager attends the meeting. We can expand these constraints as follows.

$$(V_1 \supset C_1) \wedge (V_2 \supset C_2) \wedge (V_3 \supset C_3),$$
$$(M_1 \supset C_1) \wedge (M_2 \supset C_2) \wedge (M_3 \supset C_3).$$

5. Suppose the president cannot attend the meeting on day 1 and the manager cannot attend the meeting on day 2.

$$\neg P_1 \wedge \neg M_2.$$

In the first step, we find out all models which satisfy the above hard constraints by producing all combinations of truth value for all propositional constants and check each of them if it satisfies the hard constraints. Then in the second step, for each model M, we check whether it is a minimal model or not. It is done by checking if there exists M' such that $\mathcal{R}(M', M)_\phi$ holds or not. If there exists no such model, M is a minimal model.

From the domain closure axiom, we can expand the previous wdr of the meeting scheduling as follows.

$$\mathcal{R}(M', M)_\phi = (M' \preceq_\phi M) \wedge \neg(M \preceq_\phi M')$$

where $M' \preceq_\phi M$ is the following abbreviation.

$$
\begin{array}{l}
((M \models_\phi C_1 \supset V_1) \supset (M' \models_\phi C_1 \supset V_1)) \wedge \\
((M \models_\phi C_2 \supset V_2) \supset (M' \models_\phi C_2 \supset V_2)) \wedge \\
((M \models_\phi C_3 \supset V_3) \supset (M' \models_\phi C_3 \supset V_3)) \wedge \\
((((M \models_\phi C_1 \supset V_1) \equiv (M' \models_\phi C_1 \supset V_1)) \wedge \\
\quad ((M \models_\phi C_2 \supset V_2) \equiv (M' \models_\phi C_2 \supset V_2)) \wedge \\
\quad ((M \models_\phi C_3 \supset V_3) \equiv (M' \models_\phi C_3 \supset V_3))) \supset \\
\quad (((M \models_\phi C_1 \supset M_1) \supset (M' \models_\phi C_1 \supset M_1)) \wedge \\
\quad ((M \models_\phi C_2 \supset M_2) \supset (M' \models_\phi C_2 \supset M_2)) \wedge \\
\quad ((M \models_\phi C_3 \supset M_3) \supset (M' \models_\phi C_3 \supset M_3))))
\end{array}
$$

We compare models by this wdr. For example, the following two interpretations are models which satisfy the above hard constraints.

$$
\begin{array}{l}
I_1 = \{\neg C_1, \neg C_2, C_3, \neg P_1, \neg P_2, P_3, \neg V_1, \neg V_2, V_3, \neg M_1, \neg M_2, M_3\}. \\
I_2 = \{\neg C_1, C_2, \neg C_3, \neg P_1, P_2, \neg P_3, \neg V_1, V_2, \neg V_3, \neg M_1, \neg M_2, \neg M_3\}.
\end{array}
$$

In I_1, the meeting will be held on day 3 and all members will attend it, whereas in I_2, the meeting will be held on day 2 and the manager does not attend it. We can see that I_1 is preferred to I_2 because $\mathcal{R}(I_1, I_2)_\phi$ holds. This result coincides with the intended order of solutions. And from this result, I_2 is not a minimal model. In this way, we select all minimal models. In this case, only I_1 is the minimal model.

And, suppose a new constraint that the vice president cannot attend the meeting on day 3 is added. That is, the following constraint is added.

$$\neg V_3.$$

Since I_1 is no longer a model, we have to find another minimal model. Then, the minimal model becomes I_2. The following interpretation is a model but not a minimal model.

$$I_3 = \{\neg C_1, \neg C_2, C_3, \neg P_1, \neg P_2, P_3, \neg V_1, \neg V_2, \neg V_3, \neg M_1, \neg M_2, M_3\}.$$

In I_2, the president and the vice president will attend the meeting, whereas in I_3, the president and the manager will attend the meeting. From the priorities of the soft constraint, we choose I_2 by giving high priority for the schedule of the vice president.

Finally, if a wdr is a (strict) partial order, we can calculate minimal models a little more efficiently. We say $\mathcal{R}(M', M)_\phi$ is a *(strict)partial order* with respect to $\mathbf{P}$ if for every Φ_D and every ϕ, the following conditions are satisfied.

1. For every interpretation M, $\mathcal{R}(M, M)_\phi$ does not hold.
2. For every M and for every M' and M'' which are comparable with M with respect to $\mathbf{P}$, if $\mathcal{R}(M'', M')_\phi$ and $\mathcal{R}(M', M)_\phi$ hold, $\mathcal{R}(M'', M)_\phi$ holds.

When the above conditions are true, we can calculate minimal models by the following procedure.

1. We calculate the set of models of hard constraints, $A(\mathbf{P})$. Let $\mathcal{M}$ be this set.
2. Let $\mathcal{N}$ be empty in the first place.
3. Take a model $M \in \mathcal{M}$ which is not checked yet.
   (a) If there is no $M' \in \mathcal{N}$ such that $\mathcal{R}(M', M)_\phi$ holds, add M to $\mathcal{N}$.
   (b) Remove all $M' \in \mathcal{N}$ from $\mathcal{N}$ such that $\mathcal{R}(M, M')_\phi$ holds.
4. If we check all models in $\mathcal{M}$, output $\mathcal{N}$ as a set of all minimal models.

We do not have to compare a model with every model in $\mathcal{M}$ by this method.

6 Related Work

In this section, we compare our framework with prioritized circumscription [8], Shoham's Preferential Logic [10], and HCLP(Hierarchical Constraint Logic Programming) [1].

6.1 Prioritized Circumscription

We can express soft constraints with priority by prioritized circumscription [8] by minimizing the negation of desired conditions. However, it is not so clear which predicate should be minimized in circumscription to express general soft constraints such as maximizing a value in a part of solutions, whereas we can express those kind of soft constraints in the meta-language directly. However, since there is strong connection between prioritized circumscription and stratified logic program [6], it is possible to translate a class of soft constraints to logic programs for computation.

Besides this technical difference, we have a different motivation from circumscription. The motivation of circumscription is to supplement lack of incomplete information, whereas the motivation of using the meta-language is to provide preferences over solutions with complete information in this paper.

6.2 Preferential Logic

Shoham [10] gives a general semantic framework on various formalisms of nonmonotonic reasoning. His framework is to define a new logic by augmenting a standard logic by introducing a strict partial order over its interpretations. Although his framework is very general, he does not provide any proof theory. In this paper, we represent an order over interpretations in the first-order logic by defining meta-language and provide a syntactic definition in the second-order logic.

Like circumscription, he regards nonmonotonic reasoning as supplementary role in a situation with incomplete information.

6.3 HCLP

HCLP [1] is the first attempt to introduce constraint hierarchy into constraint logic programming. We can express prioritized constraints in a body of a Horn clause. However, we can only express constraints as an atomic formula and we cannot compare solutions obtained from different derivation in HCLP. On the other hand, our framework can express constrains in any form of formula and is declarative because it is based on a model theory.

7 Conclusion

The main contributions of this paper are as follows.

1. **Proposal for more expressive representation of constraints:**

   We propose a logical representation of a soft constraint which expresses a preference over solutions. This representation is regarded as a new knowledge representation for synthesis problems.

2. **A new application of nonmonotonic reasoning:**

   The original aim of nonmonotonic reasoning is to formalize human inference in a situation with incomplete information. In this paper, we use the same method but apply it to formalizing preference over solutions with complete information. Therefore, this paper can be regarded as giving a new application to nonmonotonic reasoning.

We think we must pursue the following research.

1. **Efficient implementation:**

   The described method to calculate all minimal models has a bottleneck in the first step where we calculate all models. If we need only a formula which is true in all minimal models, we can check a part of models related to the formula. We think that an enhanced method is closely related to ATMS [2].

2. **Extension to the first order logic without domain closure axiom:**

   In the described method, a domain of models must be finite. Since a domain may be infinite in the first-order logic, our method cannot be applied to the first-order logic in general. However, soft constraints are formalized in the second-order logic, so the best we can do is to find some useful subclasses in the first-order logic to be computable. We think that the relationship between prioritized circumscription and stratified logic program [6] is very important because we can express soft constraints with priority by using prioritized circumscription.

3. **Preference of derivation:**

   There is another meta-constraint which is related to efficient derivation. For example, in scheduling problems, an expert have a constraint that a critical schedule must be allocated in the early stages in reasoning process. This preference is different from preference over solutions, but it is very important to get a solution very efficiently. To do this, we have to have another knowledge representation which deals with a process of derivation.

4. **Preference over solutions with incomplete information:**

Planning with incomplete information is often needed in robot planning. In this case, we have to combine original usage of nonmonotonic reasoning to supplement unknown information and soft constraints to choose possible plans.

Acknowledgment: I would like to thank Yasuo Nagai from Toshiba and Mikito Ishikawa, Akira Aiba, Ryuzo Hasegawa, David Hawley, Katsumi Inoue, Jun Arima, Kuniaki Mukai, Kôiti Hasida, Koichi Furukawa and Nicolas Helft from ICOT and Wlodek Zadrozny, Benjamin Grosof from IBM for helpful discussion.

References

[1] Borning, A., Maher, M., Martindale, A. and Wilson, M.: *Constraint Hierarchies and Logic Programming, Proc. of ICLP89*, pp.149 – 164 (1989)

[2] de Kleer, J.: *An assumption-based TMS, Artif. Intell.*, Vol. 28, pp.127 – 162 (1986).

[3] Descotte, Y. and Latombe, J.: *Making Compromises among Antagonist Constraints in a Planner, Artif. Intell.*, Vol. 27, pp.183 – 217 (1985).

[4] Fox, M. S., Allen, B. P., Smith, S. F. and Strohm, G. A.: *ISIS: A Constraint-Directed Reasoning Approach to Job Shop Scheduling*, CMU-RI TR 83-8, Carnegie-Mellon University (1983).

[5] J. Jaffar and J-L. Lassez: *Constraint Logic Programming*, Proc. of the 14th ACM Principles of Programming Languages Conference, pp111 – 119 (1987).

[6] Lifschitz, V.: *On the Declarative Semantics of Logic Programs with Negation, Foundations of Deductive Databases and Logic Programming*, Morgan Kaufmann, pp.177 – 192 (1987).

[7] McCarthy, J.: *Circumscription - a Form of Non-monotonic Reasoning, Artif. Intell.*, Vol. 13, p.27 – 39 (1980).

[8] McCarthy, J.: *Applications of Circumscription to Formalizing Commonsense Knowledge, Artif. Intell.*, Vol. 28, pp.89 – 116 (1986).

[9] Satoh, K.: *Formalizing Nonmonotonic Reasoning by Preference Order*, ICOT–TR–440, ICOT, Japan (1988).

[10] Shoham, Y.: *Nonmonotonic Logics: Meaning and Utility, Proc. of IJ-CAI87*, pp.388 – 393 (1987).

[11] Smith, S. F., Fox, M. S. and Ow, P. S.: *Constructing and Maintaining Detailed Production Plans: Investigations into the Development of Knowledge-Based Factory Scheduling Systems*, *AI Magazine*, Vol. 7, pp.45 – 61 (Fall 1986).

[12] Sussman, G. J. and Steel, G. L.: *CONSTRAINTS – A Language for Expressing Almost-Hierarchical Descriptions*, *Artif. Intell.*, Vol. 14, pp.1 – 39 (1980).

Appendix: Proofs of Theorems

Lemma 1 *Let M' and M be interpretations with domain D which are comparable with respect to $\mathbf{P}$, and let $\mathcal{R}(M', M)_\phi$ be a wdr. Let $\mathbf{p}$ be similar to $\mathbf{P}$ such that every predicate variable in $\mathbf{p}$ is not contained in $\mathcal{R}(M', M)_\phi$. Let $M \models_\phi R(\mathbf{p})$ be an atomic wdr of its translation. If for every P_i in $\mathbf{P}$ and p_i in $\mathbf{p}$, $\phi(p_i) = (P_i)^{M'}$, then $\mathcal{R}(M', M)_\phi$ is true if and only if $M \models_\phi R(\mathbf{p})$ is true.*

Proof. Induction on the number r of meta-logical connectives and quantifiers in $\mathcal{R}(M', M)_\phi$. Assume the result holds for all integers $< r$.

1. $\mathcal{R}(M', M)_\phi$ is of the form $M \models_\phi A$. This case is trivial.

2. $\mathcal{R}(M', M)_\phi$ is of the form $M' \models_\phi A(\mathbf{P})$. It is translated into $M \models_\phi A(\mathbf{p})$. $A(\mathbf{P})$ does not contain any predicate variable in $\mathbf{p}$ and for every p_i in $\mathbf{p}$ and corresponding P_i in $\mathbf{P}$, $\phi(p_i) = (P_i)^{M'}$. Since p_i in M is interpreted as same as P_i in M', $M' \models_\phi A(\mathbf{P})$ is true if and only if $M \models_\phi A(\mathbf{p})$ is true.

3. $\mathcal{R}(M', M)_\phi$ is of the form $\neg\mathcal{A}(M', M)_\phi$. It is translated into $\neg\mathcal{A}_\phi$, where $\mathcal{A}(M', M)_\phi$ is translated into $\mathcal{A}_\phi$. By the inductive hypothesis, $\mathcal{A}(M', M)_\phi$ is true if and only if $\mathcal{A}_\phi$ is true. Therefore $\neg\mathcal{A}(M', M)_\phi$ is true if and only if $\neg\mathcal{A}_\phi$ is true.

4. $\mathcal{R}(M', M)_\phi$ is of the form $\mathcal{A}(M', M)_\phi \supset \mathcal{B}(M', M)_\phi$. It is translated into $\mathcal{A}_\phi \supset \mathcal{B}_\phi$, where $\mathcal{A}(M', M)_\phi$ is translated into $\mathcal{A}_\phi$, and $\mathcal{B}(M', M)_\phi$ is translated into $\mathcal{B}_\phi$. By the inductive hypothesis, $\mathcal{A}(M', M)_\phi$ is true

if and only if $\mathcal{A}_\phi$ is true, and $\mathcal{B}(M', M)_\phi$ is true if and only if $\mathcal{B}_\phi$ is true. Therefore, $\mathcal{A}(M', M)_\phi \supset \mathcal{B}(M', M)_\phi$ is true if and only if $\mathcal{A}_\phi \supset \mathcal{B}_\phi$ is true.

5. $\mathcal{R}(M', M)_\phi$ is of the form $(\forall \phi_v \in \Phi_D)\mathcal{A}(M', M)_{\phi_v}$. It is translated into $(\forall \phi_v \in \Phi_D)\mathcal{A}_{\phi_v}$, where $\mathcal{A}(M', M)_{\phi_v}$ is translated into $\mathcal{A}_{\phi_v}$. By the inductive hypothesis, $\mathcal{A}(M', M)_{\phi_v}$ is true if and only if $\mathcal{A}_{\phi_v}$ is true. Therefore, $(\forall \phi_v \in \Phi_D)\mathcal{A}(M', M)_{\phi_v}$ is true if and only if $(\forall \phi_v \in \Phi_D)\mathcal{A}_{\phi_v}$ is true. □

Theorem 1 *Let M' and M be interpretations with domain D which are comparable with respect to* $\mathbf{P}$, *and let $\mathcal{R}(M', M)_\phi$ be a wdr. Let* $\mathbf{p}$ *be similar to* $\mathbf{P}$ *such that every predicate variable in* $\mathbf{p}$ *is not contained in $\mathcal{R}(M', M)_\phi$. Let its translation using* $\mathbf{p}$ *be $M \models_\phi R(\mathbf{p})$. There exists M' such that $\mathcal{R}(M', M)_\phi$ is true if and only if $M \models_\phi \exists p_1 ... \exists p_n R(\mathbf{p})$ is true.*

Proof. Assume there exists M' such that $\mathcal{R}(M', M)_\phi$ is true. Since $\mathcal{R}(M', M)_\phi$ does not contain any variable in $\mathbf{p}$, for every $\phi_{p_1 ... p_n}$ in Φ_D, $\mathcal{R}(M', M)_{\phi_{p_1 ... p_n}}$ is true. Let for every p_i in $\mathbf{p}$ and P_i in $\mathbf{P}$, $\phi_{p_1 ... p_n}(p_i) = (P_i)^{M'}$. Then, by Lemma 1, $\mathcal{R}(M', M)_{\phi_{p_1 ... p_n}}$ is true if and only if $M \models_{\phi_{p_1 ... p_n}} R(\mathbf{p})$ is true. Then, by the definition of satisfaction, $M \models_\phi \exists p_1 ... \exists p_n R(\mathbf{p})$ is true.

Assume $M \models_\phi \exists p_1 ... \exists p_n R(\mathbf{p})$ is true. From the definition of satisfaction, $M \models_{\phi_{p_1 ... p_n}} R(\mathbf{p})$ is true. Take M' such that M' and M are comparable with respect to $\mathbf{P}$, and for every P_i in $\mathbf{P}$ and p_i in $\mathbf{p}$, $(P_i)^{M'} = \phi_{p_1 ... p_n}(p_i)$. Then, by Lemma 1, $M \models_{\phi_{p_1 ... p_n}} R(\mathbf{p})$ is true if and only if $\mathcal{R}(M', M)_{\phi_{p_1 ... p_n}}$ is true. Since $\mathcal{R}(M', M)_{\phi_{p_1 ... p_n}}$ does not contain any variable in $\mathbf{p}$, for every ϕ in Φ_D, $\mathcal{R}(M', M)_\phi$ is true. □

Corollary 1 *M is a minimal model with respect to a first-order formula $A(\mathbf{P})$ and a wdr $\mathcal{R}(M', M)_\phi$ if and only if M is a model of:*

$$A(\mathbf{P}) \wedge \neg \exists p_1 ... \exists p_n (A(\mathbf{p}) \wedge R(\mathbf{p})),$$

where a translation of the wdr using $\mathbf{p}$ *is $M \models_\phi R(\mathbf{p})$.*

Proof. Assume M is a minimal model with respect to a first-order formula $A(\mathbf{P})$ and a partial order relation $\mathcal{R}(M', M)_\phi$. Since for every comparable interpretation M' with M, if M' is a model of $A(\mathbf{P})$ then for every assignment function ϕ, $\neg\mathcal{R}(M', M)_\phi$ is true, there is no comparable interpretation M' with M such that for every assignment function ϕ, $(M' \models_\phi A(\mathbf{P})) \wedge \mathcal{R}(M', M)_\phi$ is true.

Since $(M' \models_\phi A(\mathbf{P})) \wedge \mathcal{R}(M', M)_\phi$ is a wdr, we can translate it into the following atomic wdr:

$M \models_\phi A(\mathbf{p}) \wedge R(\mathbf{p})$.

By Theorem 1, $M \models \neg\exists p_1 ... \exists p_n(A(\mathbf{p}) \wedge R(\mathbf{p}))$ is true. And since M is a model of $A(\mathbf{P})$, $M \models A(\mathbf{P}) \wedge \neg\exists p_1 ... \exists p_n(A(\mathbf{p}) \wedge R(\mathbf{p}))$.

Assume that $M \models A(\mathbf{P}) \wedge \neg\exists p_1 ... \exists p_n(A(\mathbf{p}) \wedge R(\mathbf{p}))$. Then M is a model of $A(\mathbf{P})$. And by Theorem 1, $M \models \neg\exists p_1 ... \exists p_n(A(\mathbf{p}) \wedge R(\mathbf{p}))$ is true if and only if there exists no comparable model M' with M such that for every assignment function ϕ, $(M' \models_\phi A(\mathbf{P})) \wedge \mathcal{R}(M', M)_\phi$ is true. Therefore, for every comparable interpretation M' with M, if M' is a model of $A(\mathbf{P})$ then for every assignment function ϕ, $\neg\mathcal{R}(M', M)_\phi$ is true. Thus, M is a minimal model. □