

Disambiguation by Prioritized Circumscription

Ken Satoh

Hokkaido University
N13W8 Kita-ku Sapporo 060 Japan
ksatoh@db.huee.hokudai.ac.jp

Abstract

This paper presents a method of resolving ambiguity by using a variant of circumscription, *prioritized circumscription*. In a disambiguation task, human seems to use various preferences which have various strength. In prioritized circumscription, we can express these preferences as defeasible constraints with various strength and we infer the most preferable logical models which satisfy stronger constraints as much as possible. This representation is very natural for disambiguation since we can regard a logical interpretation as a possible reading and the most preferable logical models as the most preferable readings. We argue that prioritized circumscription is another promising method for the task. We also discuss an implementation of prioritized circumscription by a hierarchical logic programming (HCLP) language.

1 Introduction

This paper presents a method of disambiguation task by a variant of circumscription, *prioritized circumscription* (McCarthy, 1986; Lifschitz, 1985) and discuss its implementation by a hierarchical constraint logic programming (HCLP) language such as (Borning et al., 1989).

Disambiguation is a very important task in natural language processing. To resolve ambiguity, humans seem to use not only syntactic constraints but also various levels of heuristics such as grammatical preferences (Hobbs, 1990) and semantic preferences (Wilks, 1975).

For example, suppose that we have the following sentences.

- John just saw a man with a telescope.** (a)
He bought the telescope yesterday. (b)

Although there is an ambiguity on meaning of the phrase, "with a telescope" (the telescope is either

used by John or carried by the man), we might conclude the preferred reading as follows.

From the above sentences, "He" would be equal to John because the subject tends to be continued to the next sentence and John probably had a telescope at the time of seeing a man from the sentence (b) and inertia of possession. Therefore, from this preferred reading, we conclude that the telescope is used as a device to see a man.

However, this reading is not final since at least the following preferences are involved in the above reading and these preferences can be defeated by stronger information.

Syntactic preference: The subject tends to be continued.

Semantic preference: If a person buys something at time i , then he should have it at time j where $i < j$.

In order to demonstrate defeasibility of preference rules, suppose the following sentence is added after the above sentences (a) and (b).

But, he gave the man the telescope this morning. (c)

Then, we might change a preferred reading that the man should have had a telescope and therefore, the telescope was carried by the man at the time of John's seeing the man. In this reading, at least, the following preference rule of another inertia of possession is used.

If a person gives something to the other person at time i , then the other person should have it at time j where $i < j$.

This conflicts with the former semantic preference of inertia of possession by buying, but the above preference is stronger than the former since the time of giving is later than the time of buying. Thus, the former preference becomes no longer applicable by the new sentence.

This kind of revision of reading cannot be represented by inference in classical logic since in classical logic, once we get a inferred result, we can no longer retract the result (*monotonic property*). Therefore, to understand the phenomena, we need other reasoning methods and in fact,

many researches have been using general reasoning frameworks in Artificial Intelligence such as abduction (Hobbs et al., 1993), probabilistic network (Charniak and Goldman, 1989), truth maintenance system (Zernik and Brown, 1988), default logic (Quantz, 1993) and conditional logic (Lascarides, 1993). In this paper, we propose another alternative, that is, *circumscription* (McCarthy, 1986; Lifschitz, 1985). Even though circumscription is one of the most popular formalisms in the community of nonmonotonic reasoning research, it is surprising that very few has examined feasibility of circumscription for disambiguation. Our work of disambiguation by interpretation ordering is originated from (Satoh, 1991) and in a more recent work, Kameyama (Kameyama, 1994) has independently proposed usage of circumscription for interpretation of pronominal anaphora.

In this paper, we explore this direction further. In circumscription, we give a preference order over logical interpretations and consider the most preferable models. This representation naturally corresponds with a disambiguation task since we can regard a logical interpretation as a possible reading, and disambiguation as a task to get the most preferable reading among possible readings. Among variants of circumscription, *prioritized circumscription* is suitable to represent various strength of preference rules. In prioritized circumscription, we can divide preference rules into hierarchy and this hierarchy gives a priority relation over preferences. Therefore, we directly represent rules in the hierarchy in prioritized circumscription.

We believe that circumscription has the following advantages in the task of resolving ambiguity.

- Since we use a first-order predicate calculus for a basic language, we can represent various kinds of information such as grammatical rules and semantic rules in one framework.
- There is only one extra underlying mechanism besides inference rules for the first-order predicate calculus, that is, introducing an order over logical interpretations. Therefore, reasoning process can be understood easily compared to other mechanism using numerical reasoning or complex inference rules.
- We do not need to assign detailed numerical values to preference rules in order to express priority over preference rules, but just specify a preference level of the rules. This representation can be regarded as an assignment of qualitative strength for preference rules and reduces a burden of representing a priority over preference rules greatly. Moreover, this prioritization is general since we can represent a various kind of priority besides specificity.
- It is important to retain possible readings if we can not resolve ambiguity yet. In circumscription, we can consider *multiple preferable models*,

not necessary the *single preferable model*. So, if there are yet multiple possible readings as a result of disambiguation, we can keep these possible readings as multiple preferable models.

In this paper, we also discuss an implementation by using hierarchical constraint logic programming (HCLP) language such as (Borning et al., 1989). HCLP language is similar to constraint logic programming language except that we can represent a constraint hierarchy. Thus, there is a correspondence between a solution of an HCLP language and the most preferable models of prioritized circumscription. In this paper, we use our HCLP language based on a boolean constraint solver to get the most preferable models from preference rules represented as boolean constraints in the HCLP language. We demonstrate how the above example of the disambiguation is treated in the HCLP language.

2 Prioritized Circumscription

In this section, we briefly review prioritized circumscription. For simplicity sake, we modify the definition of prioritized circumscription by (McCarthy, 1986; Lifschitz, 1985). The difference is that we let all predicates vary and maximize preference rules whereas Lifschitz minimize abnormal predicates for preference rules.

Let $\Phi(\mathbf{x})$ and $\Psi(\mathbf{x})$ be formulas with the same number of free variables \mathbf{x} . We say that Φ and Ψ are similar. $\Phi \succeq \Psi$ stands for $\forall \mathbf{x}(\Psi(\mathbf{x}) \supset \Phi(\mathbf{x}))$. We extend this notation to tuples of formulas Φ , Ψ where $\Phi = \Phi_1, \dots, \Phi_n$ and $\Psi = \Psi_1, \dots, \Psi_n$ and Φ and Ψ are similar (each Φ_j and Ψ_j are similar): $\Phi \succeq \Psi$ stands for $\bigwedge_{j=1}^n \Phi_j \succeq \Psi_j$. We also write $\Phi \succeq \Psi \wedge \Psi \succeq \Phi$ as $\Phi = \Psi$ and $\Phi \succeq \Psi \wedge \neg(\Psi \succeq \Phi)$ as $\Phi \succ \Psi$.

Let a tuple of formulas Φ be broken into disjoint parts $\Phi^1, \Phi^2, \dots, \Phi^k$. Let Ψ^i be similar to Φ^i . We define $\Phi \succeq \Psi \stackrel{def}{=} \bigwedge_{i=1}^k (\bigwedge_{j=1}^{i-1} \Phi^j = \Psi^j \supset \Phi^i \succeq \Psi^i)$. We also write $\Phi \succeq \Psi \wedge \neg(\Psi \succeq \Phi)$ as $\Phi \succ \Psi$.

Definition 1 Let $A(\mathbf{P})$ be a formula and $\Phi(\mathbf{P})$ be a tuple of formulas which is broken into $\Phi^1(\mathbf{P}), \Phi^2(\mathbf{P}), \dots, \Phi^k(\mathbf{P})$ where \mathbf{P} is a tuple of predicates used in these formulas.

The syntactic definition of prioritized circumscription is as follows:

$$A(\mathbf{P}) \wedge \neg \exists \mathbf{p}(A(\mathbf{p}) \wedge \Phi(\mathbf{p}) \succ \Phi(\mathbf{P})), \quad (1)$$

where

1. \mathbf{p} is a tuple of predicate variables each of which has the same arity as the corresponding predicate constant in \mathbf{P} ,
2. $A(\mathbf{p})(\Phi(\mathbf{p}))$ is a formula obtained by replacing every occurrence in $A(\Phi)$, respectively) of a predicate constant in \mathbf{P} by the corresponding predicate variable in \mathbf{p} .

According to the result of (Lifschitz, 1985), we give a model theoretic definition of the above formula (1) as follows.

Definition 2 We define an order \geq over logical interpretations as follows:

$$M' \geq M$$

where

1. M' and M have the same domain.
2. every constant and function symbol has the same interpretation in M' and M .
3. $\Phi(\mathbf{p}) \succeq \Phi(\mathbf{q})$ is true in M' (or, equivalently, in M) for $M'[\mathbf{P}]$ as \mathbf{p} and $M[\mathbf{P}]$ as \mathbf{q} where $M'[\mathbf{P}](M[\mathbf{P}])$ is a tuple of the extensions for $M'(M)$, respectively of predicates in \mathbf{P}

In the above order, a greater interpretation is more preferable. The above order intuitively means that logical interpretations which maximally satisfy a subset of Φ^1 are preferable, and if there are interpretations which satisfy the same formulas in Φ^1 , then interpretations which maximally satisfy a subset of Φ^2 are preferable, and... and if there are interpretations which satisfy the same formulas in Φ^{k-1} , then interpretations which maximally satisfy a subset of Φ^k are preferable.

Let A be a formula. We say that a logical interpretation M is the *most preferable model in the class of models of A w.r.t. \geq* if there is no model M' of A in the class such that $M' \geq M$ and not $M \geq M'$.

According to the result of (Lifschitz, 1985), we have the following correspondence between syntactic definition and semantic definition.

Theorem 1 A logical interpretation M is a model of (1) iff M is the most preferable model w.r.t. \geq in the class of models of A .

3 Disambiguation by Prioritized Circumscription

In order to use prioritized circumscription for a disambiguation task, we make the following correspondence between formulas in the definition of prioritized circumscription and information in natural language. In the syntactic definition of prioritized circumscription in Section 2, we correspond A with information about given sentences and background knowledge which is always true in any situation. And, we regard Φ as a tuple of preference rules. Note that preference rules are put into hierarchy according to strength of preference rules. Then, the most preferable models correspond with the most preferable readings since each model satisfies stronger preference rules as much as possible and therefore, the syntactic definition becomes a specification of the preferable readings by Theorem 1.

In the subsequent subsections, we firstly fix an experimental logical representation of sentences, background knowledge and preferences. Then, we

treat the example in Section 1 by the logical representation.

3.1 Logical Representation of Sentences and Background Knowledge

We use an adaptation of Kowalski's event calculus (Kowalski and Sergot, 1986). However, the idea of disambiguation in this paper does not depend on a particular representation. We assume that each sentence expresses an event. For example, a sentence "John gave the telescope to the man" is represented as the following formula.

$$\begin{aligned} &act(E, Give) \wedge actor(E, John) \wedge \\ &object(E, Telescope) \wedge recipient(E, Man) \end{aligned}$$

A complex sentence is supposed to be decomposed into a set of simple sentences which is translated into the above representation. Ambiguities are expressed by disjunctions. For example, the sentence "John saw a man with a telescope" is expressed as follows.

$$\begin{aligned} &time(E, T) \wedge act(E, See) \\ &\wedge actor(E, John) \wedge object(E, Man) \\ &\wedge (device(E, Telescope) \\ &\vee (time(E', T) \wedge act(E', Have) \\ &\wedge actor(E', Man) \wedge object(E', Telescope))) \end{aligned}$$

The last conjunct expresses ambiguity in the phrase "with a telescope" (used as a device or carried by the man).

In addition to the semantic representation, we also use syntactical information from a parser so that grammatical preference rules can be expressed. For example, we show some of the grammatical information of the sentence "John gave the telescope to the man" as follows. (We assume that sentence number is 1).

$$\begin{aligned} &subj(1, John) \wedge verb(1, Give) \\ &\wedge direct_obj(1, Telescope) \\ &\wedge indirect_obj(1, Man) \\ &\wedge in_the_sentence(1, John) \end{aligned}$$

By using these basic predicates, we can represent background knowledge which are always valid. For example, background knowledge "If a_1 has o at time i , and a_1 is not equal to a_2 , then a_2 does not have o at time i " can be expressed in the following formula¹.

$$\begin{aligned} &\forall e \forall i \forall a_1 \forall a_2 \forall o \forall e_1 (\\ &(time(e, i) \wedge act(e, Have) \wedge actor(e, a_1) \\ &\wedge object(e, o) \wedge \neg eq(a_1, a_2)) \supset \\ &(((time(e_1, i) \wedge act(e_1, Have) \\ &\wedge actor(e_1, a_2)) \supset \neg object(e_1, o))) \end{aligned} \quad (2)$$

3.2 Logical Representation of Preferences

We represent a preference rule as a formula in Φ in the syntactic definition of prioritized circumscription and handle a priority among preferences by

¹We ignore joint ownership for simplicity. If we would like to consider the possibility, we can represent the formula as the strongest preference.

putting stronger preferences into a stronger hierarchy of preferences.

For example, consider the following two grammatical preferences.

1. If “*He*” appears in a sentence as the subject and the subject in the previous sentence is male, then it is preferable that “*He*” refers to the previous subject.
2. If “*He*” appears in a sentence as the subject and someone in the previous sentence is male, then it is preferable that “*He*” refers to the one in the previous sentence.

Suppose that the former is stronger than the latter. This priority of the preferences means that the formula:

$$(isa(a, Male) \wedge subj(i, a) \wedge in_the_sentence(i+1, He)) \supset eq(a, He) \quad (3)$$

should be satisfied as much as possible for every a and i , and if it is maximally satisfied then the following formula:

$$(isa(a, Male) \wedge in_the_sentence(i, a) \wedge in_the_sentence(i+1, He)) \supset eq(a, He) \quad (4)$$

should be satisfied as much as possible for every a and i .

We can represent semantic preferences as well. For example, a preference “If a_1 sees a_2 , then a_2 and a_1 are not equal” means that the following expression should be satisfied as much as possible for every e, a_1 and a_2 :

$$(act(e, See) \wedge actor(e, a_1) \wedge object(e, a_2)) \supset \neg eq(a_2, a_1) \quad (5)$$

Note that the above is a preference rule because there is a possibility of reflexive use of “see”.

3.3 Example

Now, we are ready to treat disambiguation of the sentences used in Section 1 by prioritized circumscription.

We consider the following background knowledge which is always true. We denote the conjunctions of the following axioms as $A_0(\mathbf{P})$ where

$$\mathbf{P} \stackrel{def}{=} \langle eq, is, time, act, actor, object, recipient, device, subj, in_the_sentence \rangle.$$

1. If a_1 is equal to a_2 then a_2 is equal to a_1 .
 $\forall a_1 \forall a_2 (eq(a_1, a_2) \supset eq(a_2, a_1))$
2. If a_1 and a_2 are equal and a_2 and a_3 are equal, then a_1 and a_3 are equal.
 $\forall a_1 \forall a_2 \forall a_3 ((eq(a_1, a_2) \wedge eq(a_2, a_3)) \supset eq(a_1, a_3))$
3. If a_1 is equal to a_2 , then a_2 is an actor of a_1 's action, too.
 $\forall e \forall a_1 \forall a_2 ((eq(a_1, a_2) \wedge actor(e, a_1)) \supset actor(e, a_2))$
4. If a use o as a device at time i then a has o at time i .

$$\begin{aligned} & \forall e \forall i \forall a \forall o ((time(e, i) \wedge actor(e, a) \\ & \wedge device(e, o)) \supset \\ & \exists e_1 (time(e_1, i) \wedge act(e_1, Have) \\ & \wedge actor(e_1, a) \wedge object(e_1, o))) \end{aligned}$$

5. If a_1 has o at time i , and a_1 is not equal to a_2 , then a_2 does not have o at time i .
This is same as (2).

We consider the following preferences.

1. If a_1 sees a_2 , then a_1 and a_2 are not equal.
 $\Phi_1^1(\mathbf{P}, e, a_1, a_2) = (5)$
2. If a is male and a is the subject of i -th sentence and “*He*” is in the next sentence, then a is equal to “*He*”.
 $\Phi_1^2(\mathbf{P}, e, a, i) = (3)$
3. If a is male and a is in i -th sentence and “*He*” is in the next sentence, then a is equal to “*He*”.
 $\Phi_1^3(\mathbf{P}, a, i) = (4)$
4. If someone gives o to a at time i , then a has o at time $i+1$. This expresses inertia of ownership.
 $\Phi_1^4(\mathbf{P}, e, a, o, i) =$
 $(act(e, Give) \wedge object(e, o)$
 $\wedge recipient(e, a) \wedge time(e, i)) \supset$
 $\exists e_1 (act(e_1, Have) \wedge actor(e_1, a)$
 $\wedge object(e_1, o) \wedge time(e_1, i+1))$
5. If a buys o at time i , then a has o at time $i+2$. This preference of another inertia of ownership is weaker than the former preference because time interval is longer than the former preference.
 $\Phi_1^5(\mathbf{P}, e, a, o, i) =$
 $(act(e, Buy) \wedge actor(e, a)$
 $\wedge object(e, o) \wedge time(e, i)) \supset$
 $\exists e_1 (act(e_1, Have) \wedge actor(e_1, a)$
 $\wedge object(e_1, o) \wedge time(e_1, i+2))$

We assume that Φ_1^1 is a formula which should be satisfied in the first place, Φ_1^2 in the second place, Φ_1^3 in the third place, Φ_1^4 in the fourth place and Φ_1^5 in the fifth place.

Example 1 We consider the following sentences.
John just saw a man with a telescope.
He bought the telescope yesterday.

A logical representation of the above sentences is as follows and we denote it as $A_1(\mathbf{P})$.

$$\begin{aligned} & time(E_1, 2) \wedge act(E_1, See) \wedge actor(E_1, John) \\ & \wedge object(E_1, Man) \wedge isa(John, Male) \\ & \wedge isa(Man, Male) \wedge subj(1, John) \\ & \wedge in_the_sentence(1, John) \\ & \wedge in_the_sentence(1, Man) \\ & \wedge (device(E_1, Telescope) \\ & \vee (actor(E'_1, Man) \wedge time(E'_1, 2) \\ & \wedge act(E'_1, Have) \wedge object(E'_1, Telescope))) \end{aligned}$$

$$\begin{aligned} & \wedge \text{time}(E_2, 0) \wedge \text{act}(E_2, \text{Buy}) \\ & \wedge \text{actor}(E_2, \text{He}) \wedge \text{object}(E_2, \text{Telescope}) \\ & \wedge \text{in_the_sentence}(2, \text{He}) \end{aligned}$$

Note that we represent “just” as time 2 and “yesterday” as time 0.

In the syntactic definition of the most preferable reading (1), we let $A(\mathbf{P})$ be $A_0(\mathbf{P}) \wedge A_1(\mathbf{P})$ and k be 5.

We show an intuitive explanation of inference of getting the most preferable reading as follows. From the preference 2, “He” preferably refers to John. Note that although the preference 3 seems to be applicable, it is not actually used since the stronger preference 2 overrides the preference 3.

Then, from the preference 5, John had the telescope at time 2. From the preference 1, John is not equal to the man. Then, the man cannot have the telescope at time 2 from the background knowledge 5 and therefore, the telescope was used as a device from the disjunction in $A_1(\mathbf{P})$. We can actually prove that $\text{device}(E_1, \text{telescope})$ is true in the most preferable readings.

Example 2 Suppose we add the following sentence to the previous sentences.

But, he gave the telescope to the man this morning.

A logical representation related to this sentence is as follows. We denote the formula as $A_2(\mathbf{P})$.

$$\begin{aligned} & \text{time}(E_3, 1) \wedge \text{act}(E_3, \text{Give}) \wedge \text{actor}(E_3, \text{He}) \\ & \wedge \text{object}(E_3, \text{Telescope}) \wedge \text{recipient}(E_3, \text{Man}) \end{aligned}$$

Note that we represent “this morning” as time 1.

In this case, we let $A(\mathbf{P})$ be $A_0(\mathbf{P}) \wedge A_1(\mathbf{P}) \wedge A_2(\mathbf{P})$ in the syntactic definition. Then, reading of “with a telescope” is changed. From the preference 4, the man should have had the telescope at time 2. If the telescope were used as a device at time 2, John would also have the telescope at the same time according to background knowledge 4 and it contradicts background knowledge 5. Then, the weaker preference 5 is retracted to avoid contradiction and the stronger preference 4 is survived. Therefore, in the most preferable reading, the man had the telescope at time 2.

4 HCLP language

Now, we discuss an implementation of prioritized circumscription by HCLP. Firstly, we briefly review a hierarchical constraint logic programming(HCLP) language. We follow the definition of (Borning et al., 1989).

An HCLP program consists of rules of the form:

$$h: \neg b_1, \dots, b_n$$

where h is a predicate and each of b_1, \dots, b_n is a predicate or a constraint or a labeled constraint. A labeled constraint is of the form:

$$\text{label } C$$

where C is a constraint in specific domain and **label** is a label which expresses strength of the constraint C .

The operational semantics for HCLP is similar to CLP except manipulating a constraint hierarchy. In HCLP, we accumulate labeled constraints to form a constraint hierarchy by each label while executing CLP until CLP solves all goals and gives a reduced required constraints. Then, we solve constraint hierarchy with required constraints.

To solve constraint hierarchy, we firstly find a maximal subset of constraints for the strongest level which is consistent with the required constraints. Then, we try to find a maximal subset of constraints in the second strongest level with respect to the union of the required constraints and the maximal consistent subset for the strongest level,..., and so on until a maximal consistent subset of constraints in the k -th strongest level is added. Then, an assignment which satisfies the final set of constraints is called a *solution*.

Let θ and σ be assignments and C_θ^1 (and C_σ^1) be a set of constraints in the strongest level of the hierarchy satisfied by θ (and σ), and C_θ^2 (and C_σ^2) be a set of constraints in the second strongest level of the hierarchy satisfied by θ (and σ),..., and C_θ^k (and C_σ^k) be a set of constraints in the k -th strongest level of the hierarchy satisfied by θ (and σ).

θ is *locally-predicate-better* (Borning et al., 1989) than σ w.r.t. the constraint hierarchy if there exists $i(1 \leq i \leq k)$ such that for every $j(1 \leq j \leq i - 1)$, $C_\sigma^j = C_\theta^j$ and $C_\sigma^i \subset C_\theta^i$.

We can prove that if θ is a *solution*, then there is no assignment σ which satisfies the required constraints and is *locally-predicate-better* than θ .

Note that the definition of locally-predicate-better comparator is similar to the definition of the order over logical interpretation in the prioritized circumscription. The difference is that locally-predicate-better comparator considers assignments for variables in constraints in HCLP whereas the order over logical interpretation considers assignments of truth-value for formulas in prioritized circumscription.

5 Implementation by HCLP language

In order to use HCLP language for implementation of prioritized circumscription, we need to change formulas in prioritized circumscription into constraints in HCLP. It is done as follows. We introduce a domain closure axiom so that we only consider relevant constants used in the given sentences. Then, we instantiate universal-quantified variables in background knowledge and free variables in preferences with the relevant constants and introduce Skolem functions for existential-quantified variables.

For example, we have the following formula by instantiating preference 4 in Section 3.3 with E_3 for c and the man for a and the telescope for o and 1 for i and introducing a Skolem function f :

$$\begin{aligned} & (act(E_3, Give) \wedge object(E_3, Telescope) \\ & \wedge recipient(E_3, Man) \wedge time(E_3, 1)) \supset \\ & (act(f(E_3, Man, Telescope, 1), Have) \\ & \wedge actor(f(E_3, Man, Telescope, 1), Man) \\ & \wedge object(f(E_3, Man, Telescope, 1), Telescope) \\ & \wedge time(f(E_3, Man, Telescope, 1), 2)) \end{aligned}$$

By this translation, every formula becomes ground and we regard a different ground atom as a different propositional symbol. Then, every formula in prioritized circumscription can be regarded as a boolean constraint in HCLP. We translate all formulas in the syntactic definition of the background knowledge and the sentences in Examples 1 and 2 into boolean constraints in our HCLP language (Satoh, 1990). Then, from the two sentences in Example 1, our HCLP language gives the following result as a part of a solution:

$$\begin{aligned} time(E_1, 2) &= \text{true} \\ actor(E_1, John) &= \text{true} \\ object(E_1, Man) &= \text{true} \\ act(E_1, See) &= \text{true} \\ device(E_1, Telescope) &= \text{true} \end{aligned}$$

which means that the telescope is used as a device.

And, our HCLP language gives the following result for the sentences in Example 2:

$$\begin{aligned} time(E_1, 2) &= \text{true} \\ actor(E_1, John) &= \text{true} \\ object(E_1, Man) &= \text{true} \\ act(E_1, See) &= \text{true} \\ device(E_1, Telescope) &= \text{false} \\ actor(E'_1, Man) &= \text{true} \\ time(E'_1, 2) &= \text{true} \\ act(E'_1, Have) &= \text{true} \\ object(E'_1, Telescope) &= \text{true} \end{aligned}$$

which means that the man has the telescope (and it is not used as a device).

6 Conclusion

We believe that the following are contributions of this paper.

1. We examine a feasibility of prioritized circumscription for specifying the most preferable reading by considering a disambiguation task in the concrete examples and show that we can represent the task quite naturally.
2. We discuss an implementation of disambiguation within an HCLP language by showing a correspondence between a priority over preference rules in prioritized circumscription and a constraint hierarchy in HCLP.

As a future research, we need the following.

1. We would like to examine a computational complexity of disambiguation by HCLP.

2. It is better to learn preferences automatically in stead of specifying preferences by user. One approach for learning is to build an interactive system such that the system shows to a user a set of possible readings for given sentences and the user gives an order over possible readings. Then, the system would be able to learn preferences by generalizing the order.

References

- Borning, A., Maher, M., Martindale, A., and Wilson, M. 1989. Constraint Hierarchies and Logic Programming. In *Proceedings of the International Conference on Logic Programming*, pages 149–164.
- Charniak, E., and Goldman, R. 1989. A Semantics for Probabilistic Quantifier-Free First-Order Languages, with Particular Application to Story Understanding. In *Proceedings of IJCAI-89*, pages 1074–1079.
- Hobbs, J. R. 1990. Two Principles of Parse Preference. In *Proceedings of COLING-90*, Vol. 3, pages 162–167.
- Hobbs, J. R., Stickel, M., Martin, P. and Edwards, D. 1993. Interpretation as Abduction. *Artificial Intelligence*, Vol. 63, pages 69–142.
- Kameyama, M., 1994. Infeasible Semantics and Defeasible Pragmatics. *SRI Technical Note 544*, SRI, 30 pages.
- Kowalski, R. and Sergot, M. 1986. A Logic-based Calculus of Events. *New Generation Computing*, Vol. 4, pages 67–95.
- Lascarides, A. and Asher, N. 1993. Temporal Interpretation, Discourse Relations, and Commonsense Entailment. *Linguistics and Philosophy*, Vol. 16, pages 437–493.
- Lifschitz, V. 1985. Computing Circumscription. In *Proceedings of IJCAI-85*, pages 121–127.
- McCarthy, J. 1986. Applications of Circumscription to Formalizing Commonsense Knowledge. *Artificial Intelligence*, Vol. 28, pages 89–116.
- Quantz, J. J., 1993. Interpretation as Exception Minimization. In *Proceedings of IJCAI-93*, pages 1310–1315.
- Satoh, K., 1991. A Logical Foundation of Preference-based Disambiguation. *ICOT-TR-632*, ICOT, 15 pages.
- Satoh, K and Aiba, A. 1990. The Hierarchical Constraint Logic Language: CHAL. *ICOT-TR-592*, ICOT, 18 pages.
- Wilks, Y. 1975. A Preferential, Pattern-Seeking, Semantics for Natural Language Inference. *Artificial Intelligence*, Vol 6, pages 53–74.
- Zernik, U. and Brown, A. 1988. Default Reasoning in Natural Language Processing. In *Proceedings of COLING-88*, pages. 801–805.