A CONTROLLED QUANTIFICATION IN PARSING OF MONTAGUE GRAMMAR

Key-Sun CHOI and Gil Chang KIM

Department of Computer Science, Korea Advanced Institute of Science and Technology,
P.O. Box 150, Cheongryang, Seoul 131, Korea (South)

Communicated by W.L. Van der Poel
Received 23 January 1985

Keywords: Computational linguistics, Montague grammar, quantifications, parsing, logical translation

1. Introduction

Montague grammar [5] provides a good framework for a direct translation from a natural language sentence into its logical formulas. This is enabled by the fact that every syntactic rule has its corresponding translation rule. As soon as a subphrase is recognized by a syntactic rule, its corresponding translation rule translates the recognized subphrase into its logical formula. M-grammar of Landsbergen [3] and semantic equivalence parsing (SEP) of Warren and Friedman [7] are parsing algorithms that are based on the above framework of Montague grammar.

In this paper, only a version of Montague grammar is discussed, that is, the grammar for the English fragments in "The proper treatment of quantification in ordinary English" (PTQ) of Montague [4]. This version is peculiar but most well known.

Syntactic rules of Montague grammar consist of context-free rules and noncontext-free rules. The context-free rules scan the given input sentence and recognize its syntactic structure. The noncontext-free rules are the quantification rules whose roles are to treat semantic problems as follows [2]:

(1) They bind anaphoric pronouns to their coreferential noun phrases (e.g., "a woman walks and she talks").

(2) They handle the de re/de dicto (or specific/nonspecific) ambiguities in opaque contexts (e.g., "John seeks a unicorn").

(3) They handle the scope ambiguities which are caused by more than one quantifier in a sentence (e.g., "every man loves a woman").

However, quantification rules can be applied to any phrase which does not need any quantification. At every parsing step, it cannot be determined whether a quantification rule should be applied or not. This nondeterminism causes wrong applications of quantification rules, or redundant parsing paths.

SEP [7] is a nondeterministic program such that every parsing path is tried. If an application of a quantification rule results in a wrong parse, SEP backtracks and one of the other alternative rules is tried. On the other hand, if distinct parsing paths translate an input sentence into equivalent logical formulas, only the first one is printed. Here, the equivalency of logical formulas is determined by an equivalency test. However, it is an undecidable problem to test the equivalence of intensional logical formulas [7]. (The intensional logic is the underlying logic of Montague grammar.) Hence, SEP uses the following weaker definition of equivalence: "logical formulas are equivalent if they are identical within change of bound variable, after logical reduction and extensionalization." By this definition only, the equivalence of intensional logical formulas cannot be completely tested.

In this paper, a parsing algorithm is proposed,
which directly translates an input English sentence into intensional logical formulas as SEP does. However, unnecessary parsing paths will not be tried by controlling the quantifications. In this sense, we call our parsing algorithm the controlled semantic equivalence parsing (CSEP). The goals of CSEP are as follows:

1. It produces the intensional logical formulas that are nonequivalent to one another.
2. It does not try any unnecessary parse.

The main idea of CSEP is to apply the quantification rules after every context-free rule.

2. Preliminaries

For the illustration in this paper, three context-free rules and a quantification rule of Montague grammar are introduced: a sentence-recongnizing rule $S_4$ concatenates a noun phrase and a verb phrase to give a sentence; a sentence-coordination rule $S_{12}$ conjoins two sentences by a connective such as "and" or "or"; an NP-recognition rule $S_2$ makes a noun phrase by concatenating a determiner and a common noun; and a quantification rule $S_{14}$ substitutes a syntactic variable $he_i$ by a noun phrase.

A parsing path will be represented by an analytic description surrounded by parentheses whose first component represents a rule identifier like $S_2$ and remaining ones are the analytic descriptions which denote the constituents for applying the rule of the first component. For example, let us consider an analytic description such as

\[(S_4 \, (S_2 \, \text{every man}) \, \text{walks}).\]

The first component $S_4$ is the rule which concatenates two subphrases: a noun phrase "every man" parsed by $S_2$, and a verb phrase "walks".

3. Problems of quantification

In this section, three cases of confusing the application of quantification rules will be discussed. Each case is compared with one of three roles of quantification rules, and its problem is investigated to achieve the goals of CSEP that were introduced in Section 1.

First, any sentence including no anaphoric pronoun may be analyzed by quantification rules. For example, let us consider the following sentence:

"John walks".

In Montague grammar, there are two analytic descriptions as follows:

\[(S_4 \, \text{John walks}),\]
\[(S_{14} \, \text{John} \, (S_4 \, he_0 \, \text{walks})).\]

But, their logical formulas are equivalent such as

\[\text{walk(John)}.\]

Here, the problem is how to select one of two parsing paths that yield the identical logical formulas.

Secondly, a noun phrase in an opaque context may not cause a de dicto reading. For example, let us consider the following sentence:

"John seeks a unicorn and it runs".

The pronoun "it" is coreferential with "a unicorn" in the opaque context caused by an intensional verb "seek". In this case, "a unicorn" should be quantified over the whole sentence; hence, only the de re reading is possible. A possible analytic description is the following:

\[(S_{14} \, \text{a unicorn} \, (S_{12} \, (S_4 \, \text{John seeks he_0}) \, \text{and} \, (S_4 \, he_0 \, \text{runs})).\]

Here, the problem is how to determine whether a noun phrase in an opaque context causes de re/de dicto ambiguities or not.

Lastly, more than one quantifier in a sentence may not cause any scope ambiguity. Scope ambiguity are caused by the different ordering of quantifications of at least two noun phrases in a sentence $S$, only if a verb in $S$ is translated into a predicate whose number of arguments is at least two. For example, in the sentence

"every many loves a woman",

"loves" is translated into a two-place predicate such as love($x, y$), where $x$ and $y$ are variables bound to "every man" and "a woman" respec-
(a) (S12 (S4 every man walks) and (S4 a woman talks)).
(b) (S12 (S14̄ every man (S4 he₀ walks)) and (S4 a woman talks)).
(c) (S12 (S4 every man walks) and (S14, a woman (S4 he₁ talks))).
(d) (S12 (S14̄ every man (S4 he₀ walks)) and (S14, a woman (S4 he₁ talks))).
(e) (S14̄ every man (S12 (S4 he₀ walks) and (S4 a woman talks))).
(f) (S14, a woman (S12 (S4 every man walks) and (S4 he₁ talks))).
(g) (S14̄ every man (S14, a woman (S12 (S4 he₀ walks) and (S4 he₁ talks))))
(h) (S14, a woman (S14̄ every man (S12 (S4 he₀ walks) and (S4 he₁ talks))).

Fig. 1. Eight analyses for "every man walks and a woman talks".

Fig. 2. Five logical formulas translated from "every man walks and a woman talks". (Each scope of quantifiers is specified under each formula.)
Example 3.2. Let us consider the following logical formula:

\[ \forall x [\text{love(x, y, z)}] \]

This formula is not canonical. \( \forall x \) causes a vacuous quantification, for the predicate \( \text{love} \) does not contain the variable \( x \) for its argument.

Example 3.3. The logical formula (Fig. 2(a)-(d)) is canonical, for the scope of two quantifiers is not overlapping each other. Its degree of overlapping is smaller than those of Fig. 2(e)-(h).

Example 3.4. For “every man loves a woman”, there are two canonical logical formulas which reflect its scope ambiguities as follows:

\[ \exists y [\text{woman(y)} \land \forall x [\text{man(x) = love(x, y) \land woman(y)}]], \]

\[ \forall x [\text{man(x) = love(x, y)}]. \]

4. Controlled semantic equivalence parsing

The goals of CSEP that were introduced in Section 1 can be restated as follows:

1. It produces the canonical logical formulas only.
2. It does not try any unnecessary parse.

These goals can be converted to the following two problems respectively: (i) how to know the accurate scope of each quantifier in order to keep the scope as narrow as possible, and (ii) how to make the ordering of rule applications deterministic.

In Section 3, three cases of confusing the application of quantification rules were investigated. Their common problem is: when a noun phrase is recognized, it cannot be determined whether the noun phrase may be directly combined with its neighboring constituents or it should be quantified. Such a decision can be fixed after all noun phrases are recognized.

In this stage, the fundamental principle of CSEP can be derived. First, every noun phrase should be substituted by a quantification rule, after the given input sentence is completely scanned. Secondly, before quantification rules are applied, the accurate scope of quantifiers should be calculated. Thus, CSEP consists of two phases as follows:

- **English sentence**
  - Phase I:
    - parsing with context-free rules
  - **skeletal logical formula**
  - **environment**
  - Phase II:
    - controlling of quantification rules
  - **canonical logical formulas**

Only the context-free rules are applied through Phase I, and then the quantification rules are applied through Phase II. As SEP does, whenever Phase I applies a syntactic rule, its corresponding translation rule is also applied. Hence, Phase I proceeds with making a logical formula and its environment. This logical formula of Phase I will be called the **skeletal logical formula**, in which each noun phrase is substituted by a syntactic variable. Its environment keeps the bindings between such variables and noun phrases. In the environment, the surface form of noun phrase is not stored. Instead, its logical formula and other related information are stored in order to control and apply the quantification rules in Phase II, whose result is the **canonical logical formulas**.

The concept of environment of CSEP is similar to the environment of SEP [7] and the NP-storage of Cooper [1], in view of the fact that the bindings between each noun phrase and its substituted variable are stored and retrieved later in order to apply a quantification rule. But, the environment of CSEP contains more information to control the quantifications, which will be explained in Section 5 below.

Example 4.1. Let us consider a sentence “every man walks and a woman talks”. Suppose that “every man” and “a woman” are substituted by variables \( x \) and \( y \) respectively. Its skeletal logical formula is:

\[ \text{walk}(x) \land \text{talk}(y). \]
The order of rule applications follows that of Fig. 1(h); but its canonical logical formula is not Fig. 2(h) but Fig. 2(a)-(d).

Such a gap is removed by calculating the accurate scope of two quantifiers, and then applying the quantification rules over the accurate range of components in a skeletal logical formula.

4.1. Scope of quantifiers

In order to solve the first problem on “how to know the accurate scope of quantifiers”, what information is required?

Firstly, without resolving the anaphoric pronouns, the scope problem is meaningless. In Montague grammar, when determining the antecedent of an anaphoric pronoun, it is resolved by the gender agreement. The gender of every noun phrase should be stored in the environment when its lexical entry is retrieved through Phase I. Moreover, since the anaphora resolution should be tried for pronouns only, the more basic information is whether a variable stored in the environment is bound to a pronoun or not. Therefore, every variable has three attributes: the logical formula, the gender, and the subcategory of its binding noun phrase.

Secondly, what representation is used for finding the accurate scope of quantifiers? If the skeletal logical formula is used, arguments of every predicate should be tested. An argument is either a variable or a formula containing it. If every argument is just a variable, the problem is simple. But, if it is a complex formula, the problem is not simple. For example, let us consider “John seeks a unicorn and it runs”. If “John” and “a unicorn” are substituted by variables x and y respectively, its skeletal logical formula is

\[ \text{seek}(x, \lambda P[P(y)]) \& \text{run}(y). \]

Here, the necessary information is that y appears in arguments of both “seek” and “run”. It motivates a new representation which reveals only the necessary information: the distribution of variables in the given sentence structure. This new representation will be called the entity-structure, for a syntactic variable is an entity type in Montague grammar. It will be described concretely in Section 5 below.

4.2. Ordering of rule applications

In CSEP, quantification rules are applied after other rules. Hence, the second problem for embodying the second goal of CSEP can be reduced to: “how to make the application ordering of quantification rules deterministic”. Here, the determinism is acquired by calculating what quantifiers cause semantic ambiguities.

Firstly, in order to determine whether quantifiers cause scope ambiguities or not, the following relations are defined.

Definition 4.2. If more than one quantifier in a sentence causes scope ambiguities, the relation among their bound variables is called dependent; otherwise, it is called independent.

Example 4.3. In “every man loves a woman”, the relation between “every man” and “a woman” is dependent; but, in “every man walks and a woman talks”, their relation is independent.

If variables are in the dependent relation, each different ordering of quantifications produces a different reading. If independent, any different ordering does not contribute to a different reading. In Section 6, this will be discussed formally.

Secondly, before applying quantification rules, it should be determined whether a noun phrase in an opaque context causes a de dicto reading or not. If the noun phrase is coreferential with a pronoun out of the opaque context like “John seeks a unicorn and it runs”, the quantification rule for its de dicto reading should not be applied. In CSEP, a set OpaqueSet is prepared. It only contains the variables bound to quantifiers that cause de re/de dicto ambiguities.

Lastly, what representation is desirable for solving the above-mentioned problems: the independent/dependent relations, and the de re/de dicto ambiguities of noun phrases in opaque contexts?

The independent/dependent relations are found by analyzing the distribution of noun
phrases in a conjoined sentence. By the same reason as the scope problem in Section 4.1, it is desirable to use the entity-structure which reflects the conjoinedness of the given sentence.

Next, in order to determine the de re/de dicto ambiguities, first of all, it should be known what verb invokes an opaque context. In Phase II, this information is acquired by retrieving the lexical entry for each functor in the given skeletal logical formula. It is the redundant retrieval, for the same lexical entry was retrieved in Phase I. Furthermore, it should be known whether a noun phrase in an opaque context is coreferential with other noun phrases outside the opaque context or not. It requires the analysis of the entity-structure. Thus, the most desirable solution is to make one unified representation. That is the entity-structure which reflects the opaque contexts also.

5. Environment

In CSEP, an environment keeps the information to control the quantification rules, which was discussed in Section 4. An environment consists of four components as follows.

Definition 5.1. An environment ENV is a four-tuple, ENV = (B, E, F, Rs). An NP-storage B is a set of bindings between variables and noun phrases. An entity-structure E represents the distribution of noun phrases in a sentence, except noun phrases in relative clauses. A flattened-entity-structure F is an extended entity-structure including noun phrases in relative clauses. Rs is a list of relative-environments R. A relative-environment R is an environment for a relative clause.

In this paper, a list denotes an ordered bag, whose elements are surrounded by a pair of brackets “[ ]”.

Definition 5.2. An NP-storage is a set of bindings. A binding is represented by

\[ x@[(\langle Np \rangle), \text{Subtype}, \text{Gender}] \]

where the binding operator “@” binds a variable x to the list of three attributes of a noun phrase whose input string is Np, such that \( \langle \langle Np \rangle \rangle \) denotes the logical formula of Np; Subtype is one of quantified, name, or pronoun; Gender is one of masculine, feminine, or neuter.

Example 5.3. Let us consider a sentence “Mary likes a book”. Its NP-storage consists of two bindings as follows:

\[ x@[\langle \langle \text{Mary} \rangle \rangle, \text{name}, \text{feminine}] \]
\[ y@[\langle \langle \text{a book} \rangle \rangle, \text{quantified}, \text{neuter}] \]

As was discussed in Section 4, an entity-structure only concerns the distribution of noun phrases, and only reflects the necessary information for the analyses in Phase II as follows.

Definition 5.4. For each sentence \( S_i \) (\( i = 0, 1, 2, \ldots \)), there is its corresponding entity-structure \( E_i \) that is a list of variables bound to noun phrases in \( S_i \), except noun phrases in relative clauses, such that the position of each variable in \( E_i \) is the same as that of its binding noun phrase in \( S_i \), as follows:

(i) if a sentence \( S_0 \) is conjoined by two sentences \( S_1 \) and \( S_2 \) with a logical connective LC, its entity-structure \( E_0 \) is represented by

\[ [\text{LC}, E_1, E_2] \]

(ii) if a sentence contains an opaque context, its entity-structure is represented by a nested list like

\[ [\ldots, x_1, \ldots, x_n, \text{Cat}, \ldots] \]

where the inner list denotes an opaque context, each \( x_i \) (\( i = 1, \ldots, n \)) is a variable, and Cat denotes a category of opaque context such as S (sentence), NP (noun phrase), or VP (verb phrase).

(iii) if a sentence is tensed and/or negated, the last element of entity-structure is the symbol “tensed” like

\[ [\ldots, \text{tensed}] \]

In Definition 5.4(ii), the value of Cat in an opaque context depends on what intensional verb invokes the opaque context. For example, three intensional verbs “believe”, “try to”, and “seek” invoke different categories of opaque contexts: S,
VP, and NP respectively. This feature is required, because for a de dicto reading, different categories of opaque contexts need different quantification rules.

**Example 5.5.** For a conjoined sentence with an opaque context “John seeks a unicorn and it runs”, its entity-structure is represented by

\[
[&, [x, [y, NP]], [y]],
\]

where \( x \) and \( y \) are bound to “John” and “a unicorn” respectively, and “it” is coreferential with “a unicorn”.

**Example 5.6.** For a negated sentence “John does not walk”, if “John” is bound to \( x \), its entity-structure is

\[
[x, \text{tensed}].
\]

The third component of an environment, flattened-entity-structure, contains variables bound to noun phrases in relative clauses, which were excluded in an entity-structure.

**Definition 5.7.** A flattened-entity-structure has the same structure as its corresponding entity-structure, except that variables bound to noun phrases in relative clauses are positioned after their antecedents.

**Example 5.8.** Let us consider the following two sentences and their NP-storages:

1. (la) **sentence:**
   “a man walks”,
   **NP-storage:**
   \( x@[(a \text{ man})], \text{quantified, masculine}] \)

2. (lb) **sentence:**
   “a man such that he seeks a unicorn walks”,
   **NP-storage:**
   \( x@[(a \text{ man such that he seeks a unicorn})], \text{quantified, masculine}],
   y@[⟨⟨a unicorn⟩⟩], \text{quantified, neuter}] \)

(In Montague’s PTQ [4], relative clauses are not formed with the WH-pronouns like “who”, but formed with “such that”.) Sentences (1a) and (1b) have the same entity-structure “[x]” and also the same skeletal logical formula “walk(x)”. But, their flattened-entity-structures are different as follows:

\[
(2a) [x], \quad (2b) [x, x, y].
\]

In (2b), the second \( x \) means that “he” is coreferential with its antecedent “a man”, and \( y \) is bound to “a unicorn”.

The fourth component of an environment, a list of relative-environments, keeps the information about relative clauses.

**Definition 5.9.** A relative-environment \( R \) is a five-type, \( R = (X_r, E_r, F_r, L_r, R_r) \). For a relative clause, \( X_r \) is a variable bound to the antecedent. \( E_r, F_r, \) and \( L_r \) are the entity-structure, the flattened-entity-structure, and the logical formula of the relative clause respectively. \( R_r \) is the list of relative-environments of relative clauses embedded in the current relative clause.

**Example 5.10.** The relative-environment for (lb) is

\[
(x, [x, [y, NP]], [x, [y, NP]],
\langle \langle \text{he loves a unicorn} \rangle \rangle, [\ ]),
\]

where “[ ]” means an empty list, for there is no embedded relative clause.

### 6. Quantification

Phase II of CSEP produces canonical logical formulas. Fig. 3 shows that phase II consists of seven steps.

The first step resolves the anaphoric pronouns. This step follows the principle of Right-to-Left Resolution: “A pronoun is bound to a noun phrase of the same gender on its left.”

**Example 6.1(i).** Let us consider a sentence, its entity-structure, and its NP-storage as follows:

3. (3a) **sentence:**
   “John believes that he runs”,

213
Step 1. Bind each anaphoric pronoun to its coreferential noun phrase.

Step 2. Remove all bindings for pronouns from NP-storage.

Step 3. Quantify names over the skeletal logical formula.

Step 4. Remove all bindings for names from NP-storage.

Step 5. Extract OpaqueSet.

Step 6. Find the independent and dependent relations.

Step 7. Quantify every noun phrase in NP-storage, whose ordering of quantification in every case of their permutation with observing OpaqueSet and independent/dependent relations.

Example 6.1(iii). Let us consider sentence (3a) again. The logical formula for “John” in (3c’) is quantified over the skeletal logical formula (4a) and its canonical logical formula (4b) is produced in the third step:

\[(4a) \text{believe}(x, ^\text{run}(x)) \]
\[(4b) \text{believe}(\text{John}, ^\text{run}(\text{John}))\]

Then, every name is removed from the NP-storage (3c’); it becomes an empty list. In this case, its parsing terminates.

The fifth step excludes the variables that cannot give a de dicto reading in spite of being in an opaque context, as explained in Section 4. In this step, a set OpaqueSet is produced, which contains variables that cause de re/de dicto ambiguities. OpaqueSet is generated by analyzing the flattened-entity-structure as follows.

Let O be the set of variables inside opaque contexts in the flattened-entity-structure. Next, let C be the set of variables in more than one list which is not nested in one another. In other words, for a conjoined flattened-entity-structure [LC, E1, E2], C is the set of variables in both lists E1 and E2. OpaqueSet is the set of variables, which is the result of subtracting C from O.

Example 6.2. Let us consider the following sentence (5a), its skeletal logical formula (5b) and its environment (5c)-(5f):

\[(5a) \text{“a man such that he seeks a unicorn walks and it talks”}\]

\[(5b) \text{skeletal logical formula:}\]
\[\text{walk}(x) \& \text{talk}(y)\]

\[(5c) \text{NP-storage:}\]
\[x@[\langle\text{a man such that he seeks a unicorn}\rangle], \text{quantified}, \text{masculine}]\]
\[y@[\langle\text{a unicorn}\rangle], \text{quantified}, \text{neuter}]\]
(5d) entity-structure:
\[ \&, \{x\}, \{y\} \],

(5e) flattened-entity-structure:
\[ \&, \{x, x, y\}, \{y\} \],

(5f) relative-environment:
\[ (x, \{x, \{y, \text{NP}\}\}, \{x, \{y, \text{NPI}\}\}, \langle \langle \text{he seeks a unicorn} \rangle, [ ] \rangle). \]

In the entity-structure of (5f), y is in the opaque context O. But, y also belongs to C which is the intersection of \{x, x, y\} and \{y\} in (5e). So, OpaqueSet = [ ]. This is the reason why the flattened-entity-structure should be analyzed.

The sixth step finds the independent and dependent relations, that were given by Definition 4.2. Here, these relations are formally redefined as follows.

**Definition 6.3.** (i) The independent relation I is a binary relation over a power set of variables bound to quantifiers in an entity-structure. Let Q₁ and Q₂ be sets of such variables, which are disjoint. Then, Q₁ is independent of Q₂ if and only if each member of Q₁ is independent of each member of Q₂, that is, two such members do not invoke any semantic ambiguity. This is denoted by \( (Q₁, Q₂) \subseteq I \).

(ii) The dependent relation D is an n-ary relation over a set of variables bound to quantifiers in an entity-structure for some n = 2, 3, .... Such n variables (say, \( x₁, ..., xₙ \)) are dependent of one another if and only if each different ordering of quantifications produces a different reading. This is denoted by \( (x₁, ..., xₙ) \subseteq D \).

From Definition 6.3(i), the next lemma is immediately derived.

**Lemma 6.4.** Any different ordering of quantifications of noun phrases in the independent relation produces no different reading.

From this lemma, if two sets are in the independent relation, every member in one set should be quantified prior to every member in the other set not to try any redundant parse. In this sense, the next theorem follows.

**Theorem 6.5.** The independent relation is a quasi (or strictly partial) order [6].

**Example 6.6.** Let us consider the following sentence and variables beneath their binding noun phrases:

(6a) \( \frac{\text{every man loves a woman and}}{u \ v} \)
(6b) \( \frac{\text{every boy finds a girl and}}{x \ y} \)
(6c) \( \frac{\text{every unicorn likes a centaur.}}{w \ z} \)

Its entity-structure, independent relation, and dependent relation are shown in (6b)–(6d) respectively as follows:

(6b) \( \{\&, \{u, v\}, \&, \{x, y\}, \{w, z\}\} \),
(6c) \( I = \{\langle\{u, v\}, \{x, y, w, z\}\rangle, \langle\{x, y\}, \{w, z\}\rangle\} \),
(6d) \( D = \{\langle u, v \rangle, \langle x, y \rangle, \langle w, z \rangle \} \).

Formula (6c) reads "\( \{u, v\} \) should be quantified-in prior to \( \{x, y, w, z\} \), and also \( \{x, y\} \) be prior to \( \{w, z\} \)". Hence, quantification rules should be applied in the order of \( \{u, v\}, \{x, y\}, \{w, z\} \). Here, since variables in each set are in the dependent relation, they make different readings. Thus, eight canonical logical formulas are produced in the last step. If their quantifications are not controlled, (6a) is translated into \( 2^6 = 64 \) formulas, where the number of unnecessary parses is \( 64 - 8 = 56 \).

The last step quantifies all noun phrases in the NP-storage, in every case of permutation with keeping the quasi order in the independent relation. If a variable does not belong to OpaqueSet, it does not make its de dicto reading. Here, the accurate scope of quantifiers can be detected by analyzing the entity-structures.

**7. Conclusion**

A parsing algorithm CSEP has been described to deal with the quantification rules determinis-
cally for the language of Montague grammar [4] by separating quantification rules from the syntax. Here, the determinism means that the unnecessary parsing is completely prevented. The skeletal logical formula and the environment for quantifications and obtained as soon as the syntactic structure is recognized. Every noun phrase is not quantified in the sentence until the anaphoric pronouns are resolved and the scope of quantifiers is analyzed. Since the semantic ambiguities can be foreseen by the above analyses, any unnecessary parsing is avoided. The output of CSEP are canonical logical formulas. These analyses are accomplished by the entity-structures. An entity-structure is a concise representation to reveal the necessary information only; it is easy to handle, and it has the same structure as its given input sentence.

References