自 然 言 語 処 理 109－6
（1995. 9. 14）

フレーム構造を用いた名詞句「ＡのＢ」の意味解析と階層構造，属性継承の取り扱い

廣重拓司，西原典孝，横山晶一

山形大学工学部 電子情報工学科

日本語文においては「ＡのＢ」型の名詞句が頻繁に現れ、そこでの「の」の働きは多岐に渡る。本稿では、束論を土台とする形式的意味論によって定義されたフレーム構造を与え、これによって名詞句「ＡのＢ」の意味解析を行う。そのために、実際の名詞句の意味をフレーム構造上のオブジェクトに反映させる翻訳規則を定義する。これらを用いることにより、名詞句上で成り立っている is-a, has-a-property-of の関係、およびそれらの継承をオブジェクト上において表現することができる。

Semantic Analysis for Japanese Noun Phrase "A no B" and Treatment of Its Hierarchy and Inheritance Based on Frame-structure

HIROSHIGE Takuji, NISHIHARA Noritaka, YOKOYAMA Shoichi

Department of Electrical and Information Engineering,
Faculty of Engineering, Yamagata University

In Japanese sentences, noun phrase "A no B" is used commonly and relation between A and B has great variations. In this paper, we propose "frame-structure" which is a data structure with a formal semantics on a lattice. We provide rules to translate actual noun phrases into objects on the Frame-structure, and simulate the relations on noun phrases, like "is-a" or "has-a-property-of" and their inheritance, with this framework.

1 Introduction

In Japanese sentences, noun phrase "A no B" is used commonly. Relation between A and B has great variations, and several methods are proposed to analyze them semantically[2][3].

In this paper, we propose "Frame-structure", which can treat semantic structure of noun phrases "A no B". It is a data structure with a formal semantics on a lattice and can properly treat class hierarchy and attribute inheritance between objects. Its basic idea is from that of Deductive Object Oriented Database(DOOD)[1], and the expressive ability of the Frame-structure is a subset of $\mathcal{QUIXOTE}$[4][5]. In other words, it is simpler version which is closed under strict semantics.

We consider how the meaning of a noun phrase "A no B" is constructed from each meaning of A and B, and provide rules to translate noun phrases into objects on the Frame-structure.

Produced objects reflect the semantic structure of the noun phrases "A no B". We simulate the relations on actual noun phrases, as hierarchy and inheritance, with the corresponding objects.

2 Frame-structure

In this section, we define a Frame-structure, which has a formal semantics based on a lattice structure. It can treat class hierarchy and attribute inheritance between objects. In this framework, relations between compound objects can be deduced from those between simpler objects.

2.1 Definition of Objects and Relations

Definition 2.1 (semantics)

1. Basic Symbols:
   - object symbols: $p, q, \cdots$
   - single-valued slot names: $a, b, \cdots$
   - set-valued slot names: $A, B, \cdots$
   - property slot: *prop*
   - relations: $\preceq, \ll$
2. Objects: Let s be an object symbol, $a_1, \cdots, a_k$ be single-valued slot names, $A_1, \cdots, A_l$ be set-valued slot names, $p_1, \cdots, p_k$ be objects, and $P_1, \cdots, P_l, \delta$ be sets of objects (maybe a null set). Following expressions are also objects provided that $\alpha_i \neq \alpha_j, A_i \neq A_j$ for any $i, j (i \neq j)$.
   (a) $s[a_1 \rightarrow p_1, \cdots, a_k \rightarrow p_k, A_1 \rightarrow P_1, \cdots, A_l \rightarrow P_l]$
   (b) $s[a_1 \rightarrow p_1, \cdots, a_k \rightarrow p_k, A_1 \rightarrow P_1, \cdots, A_l \rightarrow P_l, prop \rightarrow \delta]$

   "$s[\,]$" is called *primitive object*, and it may abbreviated to "s". Objects other than primitive objects are called *compound objects*.
3. Object relations: Let α, β be objects. Relations $\alpha \preceq \beta, \alpha \ll \beta$ are *object relations*. If α, β are primitive objects, their relation is called *primitive object relation*.

Intuitively, object relations correspond to the sentences as follows:

- $\alpha \preceq \beta$ — α is-a β, .
- $\alpha \ll \beta$ — α has-a-property-of β.

Definition 2.2 Assume that U is a lattice with semi-order $\preceq'$. Mapping $\psi : U \rightarrow 2^U$ which satisfies the following condition is called an *attribution function* on U:

$$\text{if } s \preceq' t (s, t \in U), \\ \text{then for any } x \in \psi(t),\ x \in \psi(s) \text{ holds.}$$

Intuitively, the value of $\psi(t)$ may be regarded as the set of all properties t has.

Definition 2.3 Let T be a set of object symbols. (U, ϕ, ψ) is called an interpretation, where U is a lattice with $\preceq'$, ϕ is a mapping $T \rightarrow U$, and ψ is an attribution function on U.

Definition 2.4 Let (U, ϕ, ψ) be some interpretation, α, β be objects and P, Q be sets of objects. Object relations:

- $(U, \phi, \psi) \models \alpha \preceq \beta$
- $(U, \phi, \psi) \models \alpha \ll \beta$
- $(U, \phi, \psi) \models P \subseteq_F Q$

are defined as follows.

1. If $\alpha = s[a_1 \rightarrow p_1, \cdots, a_k \rightarrow p_k, A_1 \rightarrow P_1, \cdots, A_l \rightarrow P_l], \beta = t[b_1 \rightarrow q_1, \cdots, b_m \rightarrow q_m, B_1 \rightarrow Q_1, \cdots, B_n \rightarrow Q_n]$ holds,
   $(U, \phi, \psi) \models \alpha \preceq \beta$ iff

- $\phi(s) \preceq' \phi(t)$, and
- for all $b_i (1 \le i \le m)$, there exists some a_j s.t. $a_j = b_i$ and $(U, \phi, \psi) \models p_j \preceq q_i$, and
- for all $B_i (1 \le i \le n)$, there exists some A_j s.t. $A_j = B_i$ and $(U, \phi, \psi) \models P_j \subseteq_F Q_i$.

2. If $\alpha = s[a_1 \to p_1, \cdots, a_k \to p_k, A_1 \to P_1, \cdots, A_l \to P_l, prop \to \delta], \beta = t[b_1 \to q_1, \cdots, b_m \to q_m, B_1 \to Q_1, \cdots, B_n \to Q_n, prop \to \eta]$ holds,

   $(U, \phi, \psi) \models \alpha \preceq \beta$ iff

   - $(U, \phi, \psi) \models s[a_1 \to p_1, \cdots, a_k \to p_k, A_1 \to P_1, \cdots, A_l \to P_l] \preceq t[b_1 \to q_1, \cdots, b_m \to q_m, B_1 \to Q_1, \cdots, B_n \to Q_n]$, and
   - for all $x \in \eta$, either of the conditions below holds:
     - for some $y \in \delta$, $(U, \phi, \psi) \models y \preceq x$, or
     - $(U, \phi, \psi) \models s \ll x$.

3. if α has "$prop \to \delta$" and β doesn't have "$prop \to \eta$", rule 2 is applied, regarding β as having "$prop \to \{\}$", and vice versa.

4. $(U, \phi, \psi) \models \alpha \ll \beta$ iff either of the conditions below holds:

   - α can be represented as $s[\cdots]$ (including the case $\alpha = s[\ ]$), and $(U, \phi, \psi) \models \psi(s) \subseteq_F \{\beta\}$, or
   - α can be represented as $s[\cdots, prop \to \delta]$, and $(U, \phi, \psi) \models \delta \subseteq_F \{\beta\}$.

5. If $P = \{\alpha_1, \cdots, \alpha_m\}, Q = \{\beta_1, \cdots, \beta_n\}$ for $m, n \ge 0$,

   $(U, \phi, \psi) \models P \subseteq_F Q$ iff for all $\beta_i (1 \le i \le m)$, there exists some a_j s.t. $(U, \phi, \psi) \models \alpha_j \preceq \beta_i$.

If $(U, \phi, \psi) \models \alpha \preceq \beta$ (or $\alpha \ll \beta$) holds, it is said that (U, ϕ, ψ) satisfies an object relation $\alpha \preceq \beta$ (or $\alpha \ll \beta$).

2.2 Object Hierarchy and Property Inheritance

Definition 2.5 Let S be a set of primitive object relations, and R be a set of object relations. We define $S \models R$ as follows.

> $S \models R$ iff under every interpretation which satisfies all relations in S, all relations in R are also satisfied.

In such cases, R is said to hold under premises S. Given a primitive object relations S, this means that the relation between compound objects R is deduced from S.

Hereafter, $t \preceq s, s \preceq r$ may be abbreviated to $t \preceq s \preceq r$.

Theorem 2.1 Assume that S is any set of primitive object relations and α, β, γ are any objects. The following relations hold:

1. $S \models \alpha \preceq \alpha$
2. $S \models \alpha \preceq \beta$ and $S \models \beta \preceq \gamma \Rightarrow S \models \alpha \preceq \gamma$
3. $S \models \alpha \preceq \beta$ and $S \models \beta \ll \gamma \Rightarrow S \models \alpha \ll \gamma$
4. $S \models \alpha \ll \beta$ and $S \models \beta \preceq \gamma \Rightarrow S \models \alpha \ll \gamma$.

(Proof) Omitted.

Examples 2.1 Followings are examples of handling object hierarchy and property inheritance.

Mt.Aso $\preceq$ volcano $\preceq$ mountain, eruption $\preceq$ action
$\models$ eruption[$subj \to$ Mt.Aso]
$\preceq$ eruption[$subj \to$ volcano]
$\preceq$ eruption[$subj \to$ mountain]
$\preceq$ eruption,
eruption[$subj \to$ Mt.Aso]
$\preceq$ action[$subj \to$ mountain].

prof_H $\preceq$ professor,
prof_H $\ll$ baldness
$\models$ prof_H $\preceq$ professor[$prop \to$ {baldness}].

(with the premises above),
prof_H $\ll$ Spanish_influenza,
Spanish_influenza $\preceq$ influenza $\preceq$ sickness
$\models$ prof_H $\ll$ sickness,
prof_H $\preceq$ professor[$prop \to$ {sickness, baldness}].

3 Semantic Analysis for Noun Phrases

In this section, we consider how the meaning of a noun phrase "A no B" is constituted of each meaning of A and B, and provide rules to translate noun phrases into objects on the Frame-structure. Produced objects reflect the semantic structure of the noun phrases "A no B". Furthermore, the relations on the objects(i.e., $\preceq, \ll$) can properly simulate the corresponding relations on actual noun phrases(i.e., is-a, has-a-property-of, respectively). Notice that our purpose is an analysis on the semantic structure of "A no B", and we do not refer to the way how the given noun phrase "A no B" is classified. For reference, see [3] for the practical classification.

3.1 Types of objects

We assume that every primitive noun is classified into common noun(i.e., a name of some thing or notion), noun derived from vt(transitive verb), noun derived from vi(intransitive verb) or functional noun(i.e., a noun whose meaning is determined with some arguments). Nouns "neko(cat), benkyô(study), sibô(death), haha(mother)" are examples for each type respectively. Noun A is translated into an object whose name is A as following:

- Common noun is translated as $A[\,]$.
- Noun derived from vt is translated as $A[subj \rightarrow \Box]$.
- Noun derived from vi is translated as $A[subj \rightarrow \Box,\ obj \rightarrow \Box]$.
- Functional noun is translated as $A[arg \rightarrow \Box]$.

Translating operation here is represented with the symbol "′". e.g., A′ = $A[\,]$. Object $A[\,]$ can be abbreviated to A . Slot value "□" means "not filled yet", and may be substituted later with other objects.

3.2 Addition of a slot to objects

In noun phrase "A no B", part "A no" usually works as a restriction or a modification for B, and such function can also be represented by the Frame-structure by adding a slot pair corresponding to the semantic role of noun phrase A to the object B′ .

3.2.1 Addition of a single-valued slot

In "Taro no hon"(Taro's book), "Taro no" is a restriction on "hon" whose owner is "Taro". This can be represented by adding a slot pair "$owner \rightarrow$ Taro" to a object for "hon".

Here it is assumed that the value of the slot "*owner*" is unique. Consequently, if the slot has been already filled with some object as its value, the addition should not be made. Formal definition for this operation is given as follows.

Definition 3.1 (operator ∘) To a given object $\beta = t[b_1 \rightarrow q_1, \cdots, b_m \rightarrow q_m, B_1 \rightarrow Q_1, \cdots, B_n \rightarrow Q_n]$, operator ∘ adding a single-valued slot "$c \rightarrow r$" to a object is defined as:

If $b_i \neq c$ for all $i (1 \leq i \leq m)$,

$$\beta \circ [c \rightarrow r] \;=\; t[b_1 \rightarrow q_1, \cdots, b_m \rightarrow q_m, c \rightarrow r, B_1 \rightarrow Q_1, \cdots, B_n \rightarrow Q_n],$$

and if $b_i = c$ and $q_i \neq \Box$ for some i, $\beta \circ [c \rightarrow r]$ is not defined.

If $b_i = c$ and $q_i = \Box$ for some i,

$$\beta \circ [c \rightarrow r] \;=\; t[b_1 \rightarrow q_1, \cdots, b_i \rightarrow r, \cdots, b_m \rightarrow q_m, B_1 \rightarrow Q_1, \cdots, B_n \rightarrow Q_n].$$

3.2.2 Addition of a set-valued slot

Noun phrase "(heya no (doa no nobu))"(knob of a door of a room) means that the knob is a part of a door, and is also a part of a room. Thus, slot "*part_of*" may have multiple values and should be treated as a set-valued slot. Operation for adding a set-valued slot is given as follows.

Definition 3.2 (operator •) Operator • adding a set-valued slot "$C \rightarrow R$" to a object is defined as:

If $B_i \neq C$ for all $i (1 \leq i \leq n)$,

$$\beta \bullet [C \rightarrow R] = t[b_1 \rightarrow q_1, \cdots, b_m \rightarrow q_m, B_1 \rightarrow Q_1, \cdots, B_n \rightarrow Q_n, C \rightarrow \{R\}],$$

and if $B_i = C$ for some i,

$$\beta \bullet [C \rightarrow R] = t[b_1 \rightarrow q_1, \cdots, b_m \rightarrow q_m, B_1 \rightarrow Q_1, \cdots, B_i \rightarrow Q_i \cup R, \cdots, B_n \rightarrow Q_n].$$

3.3 Translation of actual noun phrases

If a sentence "(A no B) is-a B" is semantically valid, translation of this sentence into the Frame-structure should be done so that the object relation (A no B)′ $\preceq$ B′ holds.

3.3.1 Case that B is a common noun

(1) Case that A is joined with ∘

A owns B: Translation $(A \text{ no } B)' = B' \circ [owner \rightarrow A']$ leads to:

$$((\text{Taro no inu}) \text{ no omotya})' = \text{omotya} \circ [\mathit{owner} \rightarrow (\text{Taro no inu})'] = \text{omotya}[\mathit{owner} \rightarrow \text{inu}[\mathit{owner} \rightarrow \text{Taro}]].$$

Here we have regarded two "no"s as representing a *owning* relation. "omotya[*owner* → inu[*owner* → Taro]]" stands for "toy[*owner* → dog[*owner* → Taro]]".

$$(\text{Taro no } (\text{inu no omotya}))' = (\text{inu no omotya})' \circ [\mathit{owner} \rightarrow \text{Taro}] = \text{omotya}[\mathit{owner} \rightarrow \text{inu}] \circ [\mathit{owner} \rightarrow \text{Taro}] \rightarrow (\text{not defined}).$$

Noun phrase "Taro no inu no omotya" may be syntactically ambiguous as shown above. But semantically the latter has no meaning and translation rule shown here excludes such a case. Of course, the latter parenthesization may be valid regarding "no"s as representing other relations.

A makes or produces B: Translation $(A \text{ no } B)' = B' \circ [\mathit{made\_by} \rightarrow A']$ leads to:

$$\text{Eri no si} = \text{si}[\mathit{made\_by} \rightarrow \text{Eri}].$$

"si[*made_by* → Eri]" stands for "poem[*made_by* → Eri]".

Other idiomatic cases: This case includes idiomatic usages of "A no B". Slot names depend on the meaning of B, e.g., translations

$$\begin{aligned}(A \text{ no } B)' &= B' \circ [\mathit{produces} \rightarrow A'] \\ (A \text{ no } B)' &= B' \circ [\mathit{teaches} \rightarrow A'] \\ (A \text{ no } B)' &= B' \circ [\mathit{sells} \rightarrow A'] \\ (A \text{ no } B)' &= B' \circ [\mathit{cause} \rightarrow A'] \\ (A \text{ no } B)' &= B' \circ [\mathit{purpose} \rightarrow A'] \\ (A \text{ no } B)' &= B' \circ [\mathit{topic} \rightarrow A']\end{aligned}$$

lead to:

(baiku no kôzyô)′ = kôzyô[*produces* → baiku]
(eikaiwa no gakkô)′ = gakkô[*teaches* → eikaiwa]
(kutu no mise)′ = mise[*sells* → kutu]
(tabako no kaji) ′ = kaji[*cause* → tabako]
(densya no kippu) ′ = kippu[*purpose* → densya]
(kome no mondai)′ = mondai[*topic* → kome].

Each objects stand for "factory[*produces* → bike], school[*teaches* → English_conversation], shop[*sells* → shoes], fire[*cause* → cigarette], ticket[*purpose* → train], problem[*topic* → rice]".

(2) Case that A is joined with •

A is a property of B: Generally, a sentence "(A no B) is-a B" is semantically valid. But there are some cases where "(A no B) has-a-property-of A" is also valid. In such cases, both $(\text{A no B})' \preceq B'$, $(\text{A no B})' \ll A'$ should hold in object expression. In such cases, translation $(A \text{ no } B)' = B' \bullet [\mathit{prop} \rightarrow A']$ leads to:

$$(\text{midori no } (\text{tyûko no kuruma}))' = (\text{tyûko no kuruma})' \bullet [\mathit{prop} \rightarrow \text{midori}] = \text{kuruma}[\mathit{prop} \rightarrow \{\text{tyûko}\}] \bullet [\mathit{prop} \rightarrow \text{midori}] = \text{kuruma}[\mathit{prop} \rightarrow \{\text{tyûko, midori}\}].$$

"kuruma[*prop* → {tyûko, midori}]" stands for "car[*prop* → {used, green}]".

(3) Case that A is joined with • and rule TR

A has B as a part: Translation $(A \text{ no } B)' = B' \bullet [\mathit{part\_of} \rightarrow A']$ leads to:

$$(\text{kodomo no } (\text{te no tume})' = (\text{kodomo no tume})' \bullet [\mathit{part\_of} \rightarrow \text{te}] = \text{tume}[\mathit{part\_of} \rightarrow \{\text{kodomo}\}] \bullet [\mathit{part\_of} \rightarrow \text{te}] = \text{tume}[\mathit{part\_of} \rightarrow \{\text{kodomo, te}\}].$$

Another parenthesization may be valid:

$$((\text{kodomo no te}) \text{ no tume})' = \text{tume} \bullet [\mathit{part\_of} \rightarrow (\text{kodomo no te})'] = \text{tume}[\mathit{part\_of} \rightarrow \{ \text{ te}[\mathit{part\_of} \rightarrow \{\text{kodomo}\}] \}].$$

In this case, objects are nested with the same slot name "part_of". To simulate the inheritance, we assume that a slot "*part_of*" requires a special rule as follows:

(TR) if $\gamma = u[\cdots, \mathit{tr\_slot} \rightarrow \beta, \cdots]$, where $\beta = t[\cdots, \mathit{tr\_slot} \rightarrow Q_i, \cdots]$, then it is transformed as $\gamma = u[\cdots, \mathit{tr\_slot} \rightarrow \text{grind}(\beta, \mathit{tr\_slot}), \cdots]$.

Here we have introduced a function "grind", which affect to a given object $\beta = t[b_1 \rightarrow q_1, \cdots, b_m \rightarrow q_m, B_1 \rightarrow Q_1, \cdots, B_n \rightarrow Q_n]$ as:
If $B_i \neq \mathit{tr\_slot}$ for all $i (1 \leq i \leq n)$,

$$\text{grind}(\beta, \mathit{tr\_slot}) = \beta,$$

and if $B_i = \mathit{tr\_slot}$ for some i,

$$\begin{aligned}
&\text{grind}(\beta, tr_slot)\\
&= t[b_1 \to q_1, \cdots, b_m \to q_m,\\
&\quad B_1 \to Q_1, \cdots, B_{i-1} \to Q_{i-1},\\
&\quad B_{i+1} \to Q_{i+1}, \cdots, B_n \to Q_n]\\
&\quad \cup \text{grind}(Q_i, tr_slot).
\end{aligned}$$

With this rule, regarding "*tr_slot*" as "*part_of*", the object above is translated as:

$$\stackrel{(TR)}{\to} \text{tume}[part\_of \to \{\text{te, kodomo}\}].$$

"tume[*part_of* → {te, kodomo}]" stands for "nail [*part_of* → {hand, child}].

Rule *TR* reflects the semantic relation between actual sentences, e.g. "(kodomo no te no tume) is-a (kodomo no tume)" and "(kodomo no te no tume) is-a (te no tume)".

A specifies a location of B: In such cases, we apply the rule *TR* regarding "*tr_slot*" as "*loc*". Translation $(A \text{ no } B)' = B' \bullet [loc \to A']$ leads to:

$$\begin{aligned}
&((\text{mati no kôen}) \text{ no buranko})'\\
&= \text{buranko} \bullet [loc \to (\text{mati no kôen})']\\
&= \text{buranko}[loc \to \{\text{kôen}[loc \to \{\text{mati}\}]\}]\\
&\stackrel{(TR)}{\to} \text{buranko}[loc \to \{\text{kôen, mati}\}].
\end{aligned}$$

"buranko[*loc* → {kôen, mati}]" stands for "swing[*loc* → {park, town}]".

A specifies the date or time of B: In such cases, we apply the rule *TR* regarding "*tr_slot*" as "*date*". Translation $(A \text{ no } B)' = B' \bullet [date \to A']$ leads to:

$$\begin{aligned}
&((\text{kinou no yoru}) \text{ no kôen})'\\
&= \text{kôen} \bullet [date \to (\text{kinou no yoru})']\\
&= \text{kôen}[date \to \{\text{yoru}[date \to \{\text{kinou}\}]\}]\\
&\stackrel{(TR)}{\to} \text{kôen}[date \to \{\text{kinou, yoru}\}].
\end{aligned}$$

"kôen[*date* → {kinou, yoru}]" stands for "park[*date* → {yesterday, night}]".

3.3.2 Case that B is a noun derived from vt

A:subject, B:vt: Translation $(A \text{ no } B)' = B' \circ [subj \to A']$ leads to:

$$(\text{Ken no unten})' = \text{unten}[subj \to \text{Ken}].$$

"unten[*subj* → Ken]" stands for "driving[*subj* → Ken]".

A:object, B:vt: Translation $(A \text{ no } B)' = B' \circ [obj \to A']$ leads to:

$$(\text{eigo no benkyô})' = \text{benkyô}[obj \to \text{eigo}].$$

"benkyô[*obj* → eigo]" stands for "study[*obj* → English]".

3.3.3 Case that B is a noun derived from vi

A:subject, B:vi: Translation $(A \text{ no } B)' = B' \circ [subj \to A']$ leads to:

$$(\text{kazan no hunka})' = \text{hunka}[subj \to \text{kazan}].$$

"hunka[*subj* → kazan]" stands for "eruption[*subj* → volcano]".

3.3.4 Case that B is a functional noun

B takes A as an argument: Translation $(A \text{ no } B)' = B' \circ [arg \to A']$ leads to:

$$\begin{aligned}
(\text{kôri no ondo})' &= \text{ondo}[arg \to \text{kôri}],\\
(\text{Taro no haha})' &= \text{haha}[arg \to \text{Taro}].
\end{aligned}$$

Each object stands for "temperature[*arg* → ice], mother[*arg* → Taro]".

3.4 Hierarchy and Property Inheritance on "A no B"

Noun phrase "kasi no ki no tukue no asi(a leg of a desk, which is made of oak wood)" is parenthesized naturally as "((kasi no ki) no (tukue no asi))" or "(kasi no (ki no (tukue no asi)))", and if proper slot names are chosen somehow, both of these can be translated to the same object "asi[*part_of* → {tukue}, *prop* → {kasi, ki}]". And we can also derive the following relations:

$$\begin{aligned}
&((\text{kasi no ki}) \text{ no } (\text{tukue no asi}))'\\
&\quad \preceq (\text{ki no asi})',\\
&((\text{kasi no ki}) \text{ no } (\text{tukue no asi}))'\\
&\quad \not\preceq (\text{kasi no ki})',\\
&((\text{kasi no ki}) \text{ no } (\text{tukue no asi}))'\\
&\quad \ll (\text{kasi no ki})'.
\end{aligned}$$

Each stands for:

$$\text{leg}[\textit{part\_of} \rightarrow \{\text{desk}\}, \textit{prop} \rightarrow \{\text{wood}[\textit{prop} \rightarrow \{\text{oak}\}]\}] \preceq \text{leg}[\textit{prop} \rightarrow \{\text{wood}\}],$$

$$\text{leg}[\textit{part\_of} \rightarrow \{\text{desk}\}, \textit{prop} \rightarrow \{\text{wood}[\textit{prop} \rightarrow \{\text{oak}\}]\}] \not\preceq \text{wood}[\textit{prop} \rightarrow \{\text{oak}\}],$$

$$\text{leg}[\textit{part\_of} \rightarrow \{\text{desk}\}, \textit{prop} \rightarrow \{\text{wood}[\textit{prop} \rightarrow \{\text{oak}\}]\}] \ll \text{wood}[\textit{prop} \rightarrow \{\text{oak}\}].$$

And they properly reflects the relations on actual linguistic phenomenon as:

((kasi no ki) no (tukue no asi))
is-a (ki no asi),
((kasi no ki) no (tukue no asi))
is-not-a (kasi no ki),
((kasi no ki) no (tukue no asi))
has-a-property-of (kasi no ki).

If linguistic relations

- prof_H is-a professor
- prof_H has-a-property-of baldness
- prof_H has-a-property-of Spanish_influenza
- Spanish_influenza is-a influenza
- influenza is-a sickness

are translated as seen in Example 2.1, we can deduce other relations from them, e.g., last term in the example represents "prof_H is-a professor, who has-a-property-of sickness and baldness".

4 Conclusions

In order to analyze semantic structure of Japanese noun phrase "A no B", we have defined Frame-structure, and proposed a translation rules for noun phrases into objects in this structure. Produced objects suitably reflect the semantic structure of original noun phrases. And by those objects, we can simulate hierarchy and attribute inheritance on noun phrases.

This study will be extended in the future to treat more general noun phrases, e.g., ones modified with relative clauses. For this purpose, we'll extend the Frame-structure, so that it can treat object operations like intersection or union, and the other object relations.

References

[1] Kifer, M. and Lausen, G.: F-logic: A Higher-Order Language for Reasoning About Objects, Inheritance, and Scheme, *Proc. ACM Int. Conf. on Management on Data*, pp. 134–146(1989).

[2] Tomiura, Y. and Hitaka, T.: Semantic Structure of Japanese Noun Phrases "*NP* 'no' *NP* "(in Japanese), *IPSJ SIG reports*, NL100-4, pp. 25–32(1994).

[3] Yokoyama, S., Kato, T. and Hiroshige, T.: Kaku-zyosi 'no' no bunrui to kaiseki (Classification and Analysis of Japanese Case Marker "no", in Japanese), *Proc. Assoc. for NLP*, pp. 137–140(1995).

[4] Morita, Y. and Haniuda, H.: Object Identity in $\mathcal{QUIXOTE}$(in Japanese), *IPSJ SIG Reports*, DBS80-12, pp. 109–118(1990).

[5] Yasukawa, H. and Yokota, K.: Labeled Graphs as Semantics of Objects(in Japanese), *IPSJ SIG Reports*, DBS80-13, pp. 119–127(1990).