A Type-theoretical Analysis of Complex Verb Generation

Satoshi Tojo
Mitsubishi Research Institute, Inc.
2-22 Harumi 3, Chuo-ku
Tokyo 104, Japan
e-mail: tojo@mrisun.mri.co.jp, m-tojo@icot.jp

Abstract

Tense and aspect, together with mood and modality, usually form the entangled structure of a complex verb. They are often hard to translate by machines, because of both syntactic and semantic differences between languages. This problem seriously affects upon the generation process because those verb components in interlingua are hardly rearranged correctly in the target language. We propose here a method in which each verb element is defined as a mathematical function according to its type of type theory. This formalism gives each element its legal position in the complex verb in the target language and certifies so-called partial translation. In addition, the generation algorithm is totally free from the stepwise calculation, and is available on parallel architecture.

1 The problem of complex verb translation

1.1 The difference between languages

In this section, we will briefly mention the difficulties of complex verb translation. The term complex verb refers to those verb structures that are formed with modal verbs and other auxiliary verbs in order to express modality or aspects, and which are attached to the original base verb.

For example, in a complex verb:

seems to have been swimming (1)

'swim' is in both the progressive aspect and the past tense, upon which 'seem' is attached with the inflection of agreement for the third person singular. Such a structure is so often hard to translate to other languages both because surface structures become different between languages, and because some aspectual/modal concepts cannot be found in the target language in the generation process. [1]

We admit that the most recalcitrant problem in verb phrase translation is that the differences of tense, aspect, mood, and modality systems. Hence, we can hardly find a verb element in the target language, which corresponds with the original word exactly in meaning. However we also need to admit that the problem of knowledge representation of tense and aspect (namely time), or that of mood and modality is not only the problem of verb translation, but that of the whole natural language understanding. In this paper, in order to realize a type calculus in syntax, we depart from a rather mundane subset of tense, aspects, and modalities, which will be argued in section 2.

1.2 The difference in the structure

In the Japanese translation of (1):

oyoi-de-i-ta-rashii

the syntactic main verb does not change from *oyoi-* (swim), which is the original meaning carrier; and the past tense is replaced by *-ta-* which is indistinguishable between past and perfect in Japanese. The feature of English verb complex derivation is the alternation of the syntactic main verb, described in fig. 1. On the

$$(\underbrace{have + -ed}_{past}\ (\underbrace{be + -ing}_{progressive}\ (\text{swim}))) \quad \longleftarrow \quad \text{`seem'}$$
$$\Downarrow$$
$$(\text{seem}\ (\text{to}\ \underbrace{have + -ed}_{past}\ (\underbrace{be + -ing}_{progressive}\ (\text{swim}))))$$

Figure 1: new verb arrival in English

contrary in Japanese, new verb elements are agglutinated at the tail of verbs, as is shown in fig. 2.

$$(\underbrace{oyoi-}_{swim}\ (\ \underbrace{-dei-}_{progressive}\ \ (\underbrace{-ta-}_{past}))) \quad \longleftarrow \quad \text{`}rashii\text{(seem)'}$$
$$\Downarrow$$
$$(\underbrace{oyoi-}_{swim}\ (\ \underbrace{-dei-}_{progressive}\ \ (\underbrace{-ta-}_{past}\ (\text{-}rashii))))$$

Figure 2: new verb arrival in Japanese

1.3 The existence of external modalities

In the history of machine translation, the analysis of a complex verb structure seems to have been rather neglected. Even in an advanced machine translation sys-

[1]The systems of tense, aspect, and modality are different from language to language, and their typology has been discussed in linguistics [1], [2], [9].

tem of CMU-CMT [8], a verb complex is represented conventionally like:

```
(vp (root ...)
    (negation +/-)
    (modality ....)
    (tense ....)
    (aspect ...) )
```

However this flat description fails in the following cases.

First, certain modalities can have their own negation. The sentence

You must play. ($\Box$[you play])

can be negated in two ways:

You must not play.
($\Box\neg$[you play] = $\Box$[you don't play])
You don't have to play.
($\neg\Box$[you play])

The former is the negation of 'must' while the latter is the negation of the sentence, where in parentheses above $\Box$ is necessity and $\neg$ is negation operator.

A similar thing can be said for tense. A certain kind of complex verbs such as 'seem to be' can have two positions to be tensed as below:

$$\left\{ \begin{array}{l} \textit{seem} \\ \textit{seemed} \end{array} \right\} \textit{to} \left\{ \begin{array}{l} \textit{be} \\ \textit{have been} \end{array} \right\}$$

It has been discussed that there are so called *external* modal expressions, that are embedded from the outside of the sentence ([3], [4], [10], [11], and [14]). The key issue is that those external ones concern only the speaker's attitude of the sentences in illocutionary situations in order to express deonticity, uncertainty, and so on, and are indifferent to the sentence subjects. Although there should be a controversial discussion as to whether a certain modal verb is connected to speakers or not [2], we will engage this matter in the following formalization of complex verb translation.

2 Formalization of complex verb translation

We will give a formal definition of modality as a function here, and discuss the strategy of verb complex generation in which each modal element is used as a function itself. In this section we use the term 'modal functions' both for modal operators and aspectual operators as long as there is no confusion.

[2]Although the concept of modality is based on the deontic distinction/epistemic between possibility and necessity such as 'can', 'may', 'must', the term is sometimes used in a broader sense inclusively 'will'(volitive) or 'can'(ability). We distinguished the externality by its independent tense and polarity (positive/negative), so that broad-sense modal verbs such as volitive and ability are regarded as actually internal while narrow-sense ones are often external.

Externality To clarify the externality of modal functions mentioned in the previous section, we put the following definition:

Definition of externality: We call those modal functions which can have their own tense and negation *external*; otherwise we call them *internal.*

Interlingua Here we propose the simplified sets of modalities, aspects, and tenses, which are recognizable both in English and in Japanese, mostly based upon the work of [7].

tense = {past, present, future}
aspect = {progressive, perfect, iterative, inchoative, terminative}
external modality
= {possible, necessary}$\otimes${epistemic, deontic}
or {hearsay, seem}
internal modality = {able}

The two dimensional analysis of epistemic and deontic is considered, reflecting the duality in meaning of 'may' and 'must'.

Verb elements as function We assume that the crude result of a parsing process for a complex verb is the list of verb elements such as:

$$past_1, seem, past_2, progressive, \texttt{swim}, \cdots \quad (2)$$

Our objective is:

- to construct the interlingua expression from these verb elements such as:
(+ed(seem))((+ed(progressive))(swim))
- and to derive a surface structure of the target language from the interlingua expression.

In order to do this, we will regard each verb element as a function. The main concern here is the domain of each function. For example, an external "past" operator +ed should operate upon external verbs, not including internal verbs. We will realize this idea, being independent of each intra-grammar in complex verbs of various languages, in the following section.

We express a verb complex as a composition of a **root-word**, **tense**, **negation**, and **verb-complement**, in which **root** and **verb-form** are included; **vcomp** may be included recursively.

Fig. 3 is an example of a λ-function of tt perfect:

$$\lambda x^{verb-structure}.\texttt{perfect}(x)$$

which takes a verb structure as a parameter, and produce a more complicated structure.

```
(vp (root be)
    (tense negation)
    (vcomp (root swim)
           (vform ing)))
           ⇓
(vp (root have)
    (tense negation)
    (vcomp (root be)
           (vform pp)
           (vcomp (root swim)
                  (vform ing))))
```

Figure 3: perfect-ization in English

3 Type inference for complex verb composition

We will make use of polymorphic type theory [6] in the further formalization. The reason we adopt the type formalism is to realize "the internality of concatenation rules"; namely each verb element should know which elements to operate upon, instead of being given a set of grammar on concatenation. Behind this purpose lie the following two issues:

- **parallel computation:** to realize a fast parallel computation on coming parallel architecture
- **partial translation:** for machine translation to be robust for ill-founded information in the lexicon

3.1 Types for verb phrase

First we will set up several mnemonic types. They are not terminal symbols of type calculus, but mnemonic identifiers of certain types.

A set of mnemonic types:

$$\{root, int, ext, tense_i, neg_i, tense_e, neg_e\}$$

For example, we can assume the following Γ as a result of our parser.

$$\Gamma = \{\text{seem} : ext, +\text{ed} : tense_i, \text{swim} : root, progressive : int\}$$

where a : α means that a is of type α. (In that parser, each syntactic category in the source language was replaced by a mnemonic type that is actually regarded as a category of the interlingua.) We use φ as a type variable denoting 'verb phrase'. Because we can regard any verb element as a modifier from a verb phrase to another phrase, those mnemonic types are always unified with a type '$\varphi \rightarrow \varphi$', except 'root' of type φ. The type composition is the process that specifies the internal variables of φ gradually. We introduce here the notion of 'most general unifier (mgu)' by Milner [6], which combines two type expressions and the result becomes a set of variable instantiations which were contained in those type expressions.

3.2 A simple example

Let us consider an example of concatenating seem with +ed(swim). First, assume that we can infer the type of +ed(swim) to be φ_1 from Γ by a variable instantiation θ, as is shown below in (3):

$$\Gamma\theta \vdash +\text{ed(swim)} : \varphi_1 \qquad (3)$$

As for seem : ext, because 'ext' was a mnemonic for a certain modifier of a verb phrase, we can put $ext = (\varphi_1 \rightarrow \varphi_2)$. Hence, for a new type variable φ_2, set η_1 as (4):

$$\eta_1 = mgu(ext\theta, \varphi_1 \rightarrow \varphi_2) \qquad (4)$$

$$(viz.\ ext\theta\eta_1 = \varphi_1\eta_1 \rightarrow \varphi_2\eta_1)$$

Note that we have not specified the contents of η_1 yet; we will give them later. We can now infer the type of combined verb phrase with (4), as is shown in fig. 4 where we can use a canonical expression (see [5]) $\lambda x.\text{seem}(x)$ instead of seem.

$$\frac{\Gamma\eta_1 \vdash \text{seem} : ext\eta_1 \qquad \Gamma\theta\eta_1 \vdash +\text{ed(swim)} : \varphi_1\eta_1}{\Gamma\theta\eta_1 \vdash \text{seem(+ed(swim))} : \varphi_2\eta_1}(\rightarrow E)$$

Figure 4: inference of seem(+ed(swim))

We use '$\preceq$' to denote that the type of the left-hand side is more instantiated than that of the right-hand side, so that:

$$\varphi_1\theta\eta_1 \preceq \varphi_1\eta_1 \preceq \varphi_1$$

3.3 Full-fledged verb type

In order to analyze the contents of θ in (3) in detail, we need to clarify what $tense_i$ actually does. We will argue what each mnemonic means in this subsection. Concretely saying, we will specify what kind of internal variables can occur in various verb types.

Our presupposition is that a 'full-fledged' complex verb contains 'external part' as well as 'internal part'. So that a verb type φ is assumed to have two internal type variables φ_{int} and φ_{ext}. Each of them has a '**head**' verb which incurs tense or negation if exists. We call a compound of tense and negation '**cap**'. As for a cap, we assume a construction of:

$$negation(tense(X))\ ^3$$

As a result, our full-fledged verb type becomes as fig. 5.

[3]This formalization is actually after the consideration of the following generation process. When we construct a past and negation of 'walk':

walk → walked → didn't walk

is less natural than

walk → don't walk → didn't walk

because it is 'don't' which incurs the operation of past. The exactly similar thing happens also in Japanese.

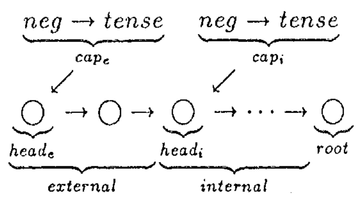


Figure 5: complex verb structure

Let us get back to the type inference of `+ed(swim)` here, to see the basis of our formalism of type unification. `+ed` is of type $tense_i$, and `swim` is of type $root$ that was a mnemonic for simple φ. θ in (3) becomes as follows:

$$\theta = mgu(tense_i, root \rightarrow \varphi_1)$$

Because $tense_i$ itself is not a function, it must be qualified as of type cap_i to act as a function by itself. We call this qualification '**promotion**', to mean that the component raises its type to connect with others. The similar thing can be said for $root$, which must be promoted to φ_{int} so as to be in the domain of cap_i. Fig. 6 depicts the type promotion.

$$\begin{tikzcd}
tense_i \arrow[d] & & root \arrow[d, "\text{promote}"] \\
cap_i[\phi/neg_i] \arrow[dr] & & \varphi_{int}[root/head_i] \arrow[dl] \\
& cap_i(\varphi_{int}) \arrow[d, "\text{promote}"] & \\
& \varphi_{int}[cap_i(root)/head_i] \arrow[d] & \\
& \varphi_{int}[root[\phi/neg, 1/tense]/head_i] \arrow[d, "\text{promote}"] & \\
& \varphi[\phi/\varphi_{ext}, root[\phi/neg, 1/tense]/head_i] &
\end{tikzcd}$$

Figure 6: type promotion

A unifier, or what we have called a set of instantiation so far, like θ or η is exactly a set of promotions where some missing verb elements (compared with full-fledged type) are ignored or replaced by other elements. For example, the contents of θ becomes as follows:

$$\begin{aligned}\theta &= [\phi/neg_i, tense_i/cap_i, root/head_i, \\ &\quad head_i/\varphi_{int}, \phi/\varphi_{ext}]\end{aligned}$$

3.4 Component embedding

In the type inference of fig. 4, we happened to choose two items of `seem` and `+ed(swim)`. Actually, we can combine any two items picked up from Γ. In this subsection we will show an example, in which we try to concatenate the consequence of fig. 4 with *progressive* : *int*. In the conventional generation, an internal verb element such as 'progressive' must be concatenated to the structure, prior to an external element such as 'seem'. However, in our type expression, 'be +ing' can join the '`seem(+ed(swim))`' structure, and also correctly choose the target element '`swim`' from the structure, instead of '`seem`' which exists most exteriorly.

If there is such a set of instantiation η_2 that:

$$\eta_2 = mgu(int\theta\eta_1, \varphi_2\eta_1 \rightarrow \varphi_3) \qquad (5)$$

then we can validate the inference in fig. 7. However, because the domain of int must be 'cap'-less φ_{int}, we cannot legalize the type inference of fig. 7 immediately.

What we are required to do now is a type '**demotion**', as opposed to the promotion. Roughly saying, a verb type of `seem(+ed(swim))` is regarded as below:

$$\varphi \equiv \varphi_{ext}(tense_i(root)) \qquad (6)$$

though each of the right hand side of (6) is promoted to some qualified type. This means that the history of promotion must be included in the unifier θ. Hence, we can scrutinize the contents of θ so that we may find where int can be embeddable. In this case,

$$\{root/head_i, head_i/\varphi_{int}\}$$

in θ (viz. $\varphi_{int} \leftarrow int$) should be demoted, and we can redefine a verb type of (6) as:

$$\varphi_{ext}(tense_i(root)) \rightarrow \lambda x^{int}.\varphi_{ext}(tense_i(x^{int}(root)))$$

Suppose that η_3 replaces the history of promotions and demotions as below:

$$(\Gamma \cup \{\varphi : int\})\theta\eta_1\eta_3 \vdash \texttt{seem(+ed(swim))} : \varphi_4$$

then we can make the inference in fig. 8. In this case,

$$\frac{\{\Gamma \cup \varphi : int\}\theta\eta_1\eta_3 \vdash \texttt{seem(+ed(swim))} : \varphi_4}{\Gamma\theta\eta_1\eta_3 \vdash \lambda\varphi.\texttt{seem(+ed(}\varphi\texttt{(swim)))} : int\eta_3 \rightarrow \varphi_4}(\rightarrow I)$$

Figure 8: inference of λ-abstraction

there is only a place for int to be embedded in between `+ed` and `swim`, and int operates upon a $root$.

This time, we can make an inference from abstracted verb structure, as is shown in fig. 9.

3.5 Head shifting

We mentioned that another superiority of type formalism is its partiality. Actually we can compose a verb structure from any part of given interlingua set, and this feature is realized dynamically. The computation of verb complex generation may be stopped any time by ill-foundedness of machine translation system. Even in such a case, our formalism can offer a part of surface structure which had been partially completed so far. The type definition above gives us an important clue to how to compose verb elements. The algorithm necessarily becomes as follows:

1. Pick up an original meaning base.

$$\frac{\Gamma\theta\eta_1 \vdash \texttt{seem(+ed(swim))} : \varphi_2\eta_1 \qquad \Gamma \vdash \mathit{progressive} : \mathit{int}}{\Gamma\theta\eta_1\eta_2 \vdash \mathit{progressive}(\texttt{seem(+ed(swim))}) : \varphi_3\eta_2}(\to E)$$

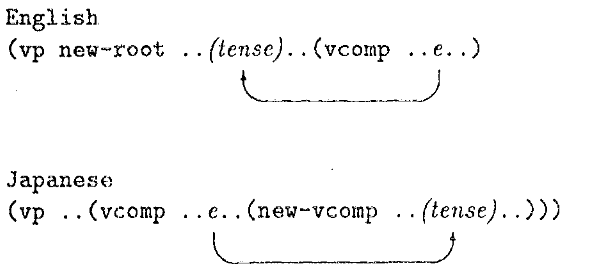
Figure 7: ill-founded type inference

$$\frac{\dfrac{\dfrac{\dfrac{\Gamma \vdash \texttt{+ed} : \mathit{tense}_i \qquad \Gamma \vdash \texttt{swim} : \mathit{root}}{\Gamma\theta \vdash \texttt{+ed(swim)} : \varphi_1} \qquad \dfrac{\Gamma \vdash \texttt{seem} : \mathit{ext}}{\Gamma \vdash \lambda\varphi.\texttt{seem}(\varphi) : \varphi_1 \to \varphi_2}}{\Gamma\theta\eta_1 \vdash \texttt{seem(+ed(swim))} : \varphi_2\eta_1}}{\Gamma\theta\eta_1\eta_3 \vdash \lambda\varphi^{int}.\texttt{seem(+ed(}\varphi\texttt{(swim)))} : \mathit{int}\eta_3 \to \varphi_4} \qquad \Gamma \vdash \mathit{prog} : \mathit{int}}{\Gamma\theta\eta_1\eta_3 \vdash \texttt{seem(+ed(}\mathit{prog}\texttt{(swim)))} : \varphi_4}$$

Figure 9: inference with λ-calculus

2. Apply internal modal functions, while ..
3. Shift the internal tense and negation to a newly applied modal function.
4. Apply external modal functions, while ..
5. Shift the external tense and negation to a newly applied modal function.

The position shift of tense, caused by the arrival of a new internal verb, is diagramed in fig. 10. Fig. 11


```
English
(vp new-root ..(tense)..(vcomp ..e..)

Japanese
(vp ..(vcomp ..e..(new-vcomp ..(tense)..)))
```

Figure 10: verb position shift

shows that every step in the derivation process can offer the partial syntactic tree.

4 Discussion

We have shown a model of complex verb translation based upon type theory, in which verb elements in the interlingua are regarded as generation functions whose domain and range are elucidated so that each verb element is certified to acquire a correct position in the target structure. Furthermore, the flexibility of type calculus was shown on the following two points.

1. We don't need to specify all the type variables every time, so that all the information a type owns can always be regarded as partial. This means that we can translate partially what can be done.
2. Because the order of calculation is not specified in the type expression, the verb structure can be composed in a way of self-organization, in the meaning that the structure is able to be decomposed and to be reorganized in the process.

In the case of complex verb translation, rephrasing to another part of speech is rather easier; we only need to 'kick out' the functional expression from the verb structure to point to another lexical item. Some external expressions must be translated into a complex sentence as the feature of externality, as we have discussed in section 1.3. We give here a definition of the identity in meaning between an external verb and its corresponding complex sentence in the type-theoretical view as (7):

$$\frac{<\mathit{agent}, \varphi(v)>: \sigma}{<<\mathit{agent}, v>, \overline{\varphi}> : \sigma_{comp}} \qquad (7)$$

where we denote a sentence by $<\mathit{agent}, \mathit{action}(\mathit{state})>$ informally, and σ is a type of sentence and σ_{comp} is a type of complex sentence. We can adopt (7) as the definition of an external verb; namely, we call such a verb φ *external* when, for φ, there is another verb expression $\overline{\varphi}$ which enables the type inference (7).

The recent study of categorial grammar such as [13], as well as the historical feat of Montague semantics, claims the efficacy of type expression. The type calculus is not specific to the complex verb nor the generation in machine translation, so that it is applicable to any generation process of natural languages. Our next goal in due course is to apply this generation mechanism to the whole categorial grammar.

5 Acknowledgement

The work was originally done as a part of machine translation system in Center for Machine Translation of Carnegie-Mellon University; an experimentation system was developed on IBM-RT with Common-Lisp [12], based upon the interlingua mentioned in section 2. The author would like to thank the members

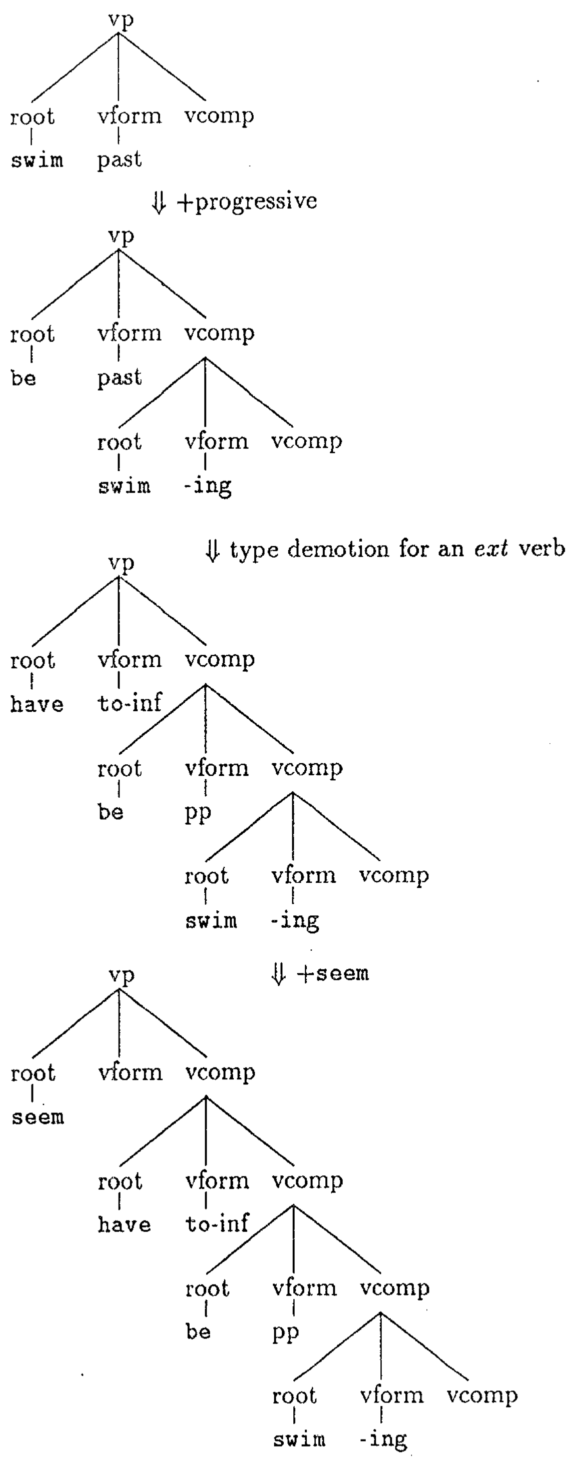


Figure 11: trees in derivation

of CMU-CMT and Institute for New Generation Computer Technology (ICOT) for many significant comments.

References

[1] B. Comrie. *Aspect.* Cambridge University Press, 1976.

[2] B. Comrie. *Tense.* Cambridge University Press, 1985.

[3] O. Jesperson. *The Philosophy of Grammar.* Allen and Unwin, London, 1924.

[4] J. Lyons. *Semantics vol.1,2.* Cambridge University Press, 1977.

[5] P. Martin-Loef. *Intuitionistic Type Theory.* Bibliopolis, 1984.

[6] R. Milner. A theory of type polymorphism in programming. *JCSS*, 17(3), 1978.

[7] S. Naito, A. Shimazu, and H. Nomura. Classification of modality function and its application to japanese language analysis. *ACL Annual Conference Proceedings*, 1985.

[8] S. Nirenburg, editor. *Machine Translation*, chapter Knowledge-based machine translation, the CMU approach, pages 68–89. Cambridge University Press, 1987.

[9] F. R. Palmer. *Mood and Modality.* Cambridge University Press, 1986.

[10] N. Rescher. *Topics in Philosophical Logic.* Dordrecht: Reidel, 1968.

[11] J.R. Searle. *Expression and Meaning: studies in the theory of speech acts.* Cambridge University Press, 1979.

[12] S. Tojo. A computational model of verb complex translation. Technical Report CMU-CMT-88-110, Center for Machine Translation, Carnegie-Mellon University, August 1988.

[13] J. van Bentham. Categorial grammar and type theory. *Journal of Philosophical Logic, to appear*, 1989.

[14] E.H. von Wright. *An Essay in Modal Logic.* North Holland, Amsterdam, 1951.