PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1988,
edited by ICOT.

THE DESIGN OF POST-ANALYSIS IN THE JETS JAPANESE/ENGLISH MACHINE TRANSLATION SYSTEM

David E. Johnson

IBM Japan, Tokyo Research Laboratory

5-19 Sanbancho, Chiyoda-ku, Tokyo 102, Japan

ABSTRACT

This paper presents an overview of transfer and generation in JETS, the Japanese/English machine translation system being developed by the Japanese Processing Group at the IBM Tokyo Research Laboratory. The main goal of the JETS project is to develop a flexible, domain-independent system that can be subsequently tailored to particular applications. This focus requires careful treatment of a large variety of linguistic facts, leading to a rather abstract view of both transfer and generation. To support comparative analysis of Japanese and English, transfer has been based on the universally oriented theory of relational grammar. The generation component has two functionally independent modules -- a (syntax) grammar planner that determines which generation rules are to be invoked in various contexts, and a deterministic relational grammar developed solely in terms of English-internal facts which realizes the particular form of the English sentence.

1 INTRODUCTION

This paper discusses key aspects of the linguistic design of the transfer and generation phases of JETS, the Japanese-to-English machine translation system being developed by the Japanese Processing Group at the IBM Tokyo Research Laboratory. The analysis phase has been described in Maruyama, Morohashi, Umeda and Sumita (1988) and so will not be discussed here. The first goal of the JETS project is to develop a domain-independent kernel system that can be tailored to specific domains. The second goal is to develop an easily extendable system capable of producing high-quality output. These goals require sophisticated treatment of a large variety of linguistic facts. This, in turn, has led us to seek a design that is justified in terms of both computational and theoretical linguistics.

The conviction that sophisticated MT requires sophisticated linguistics has had a strong influence on the design of every component. However, the contrast between JETS and many other systems is perhaps most apparent on the generation side, where there is a robust generation grammar and a grammar planner, **Gramplan**. Gramplan enables the **Genie** generator to take various lexical, grammatical and stylistic factors into account in generating target-language sentences (Johnson 1988). The general need for grammar planning in MT has been recognized and discussed, e.g., by MacDonald (1987). However, the inclusion of a functionally independent, (syntax) grammar planner in an MT system is, it seems, a novel feature of JETS.

Following a design principle we call **Independent Generation**, the generation grammar has been developed independently of the transfer component, taking into consideration *only* facts about the target language (English). Besides putting generation on a firm methodological footing, independent generation provides a principled basis for the development of the transfer component -- specifically, it can be based on comparative grammatical analysis of the source and target languages.

JETS has four main components: (1) lexical analysis, (2) syntactic analysis, (3) transfer, and (4) generation. As shown in Figure 1 on page 2, the generator has two functionally independent mod-

ules: (1) the grammar planner and (2) a deterministic relational grammar.

2 THE ROLE OF RELATIONAL GRAMMAR IN TRANSFER

To support theoretically justified, yet practical, comparative analysis of Japanese and English, transfer has been based on the theory of relational grammar (RG) (Perlmutter and Postal 1974, Johnson 1974, Johnson and Postal 1980). Two of the central tenets of RG are:

- that natural-language syntax is properly characterized in terms of a universal set of primitive grammatical relations (functions) such as subject, direct object, indirect object, etc. and
- that clauses, in general, are **multi-stratal**, meaning that they have more than one level of relational structure.

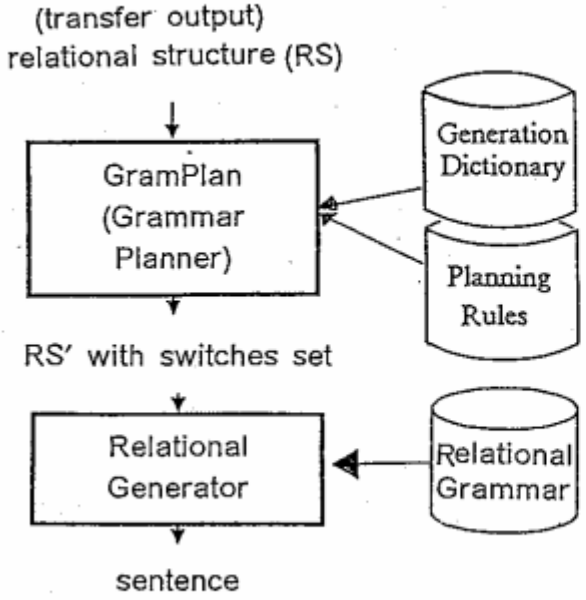


Figure 1: Genie Components

For example, "regular" passive clauses are universally taken to have the same **canonical (relational) structure** as the corresponding actives. As first proposed by Perlmutter and Postal (1974), the active/passive relation is universally characterizable in terms of a relation-changing rule that in the *generation* direction, demotes a subject to chomeur and advances a direct object to subject. Relation-changing rules in individual languages will differ in the so-called **side effects** associated with relational changes. So, in the case of English Passive, the side effects involve the introduction of the passive auxiliary **be**, the past participle verb form of the "primary" verb, and the flagging of the subject chomeur with the preposition **by**.

While lexical-functional grammar followed RG in adopting primitive grammatical relations as the basis for syntactic representation, RG has remained unique in its view that clauses are multi-stratal, entailing the recognition of clause-level, relation-changing rules such as passive, dative, clause union, subject-to-object raising, etc. It should be noted that these grammatical relations are *not* so-called deep cases in the Fillmorean sense; they are purely syntactic relations without any inherent/invariant semantic interpretation. This is one area in which JETS differs from many other systems, including the MU system, where, in theory at least, a deep-case dependency structure is assigned as a semantic representation (Nagao 1987:265). It is precisely the notions of multiple relational levels and canonical relational structure that have proven useful in the design and development of transfer and generation in JETS.

In terms of processing, the flow is basically as follows. The analysis component passes a Japanese dependency tree to the transfer component. This structure represents basic governor/dependent relations, but not grammatical relations such as subject, direct object, indirect object. The goal of transfer is to construct, in a stepwise fashion, an essentially language-neutral representation of the

sentence's canonical (relational) structure. The first major step is to identify superficial grammatical relations, converting the dependency tree into a superficial relational structure, explicitly representing relations like subject and direct object. Superficial relational structures are then successively mapped onto other, more abstract, relational structures, thus ultimately producing a canonical representation (see Figure 2). Since canonical structures abstract away from word order and morphology and involve the recovery of basic predicate/argument structure, they are essentially language-neutral. For practical reasons, they also typically contain a variety of language-specific annotations. These annotations specify such things as, e.g., that the source-language clause was passive or that a particular nominal was topic, as well as target-language lexical constraints, governed prepositions, complementizers, etc. Many of these annotations are required (hence justified) by the fact that we are applying aspects of theoretical linguistics to the MT enterprise. Annotated canonical structures are the starting point for generation.

3 POST-ANALYSIS SYNTACTIC PROCESSING

In this section, syntactic processing from the output of analysis through syntactic generation is discussed with reference to two examples. The first is

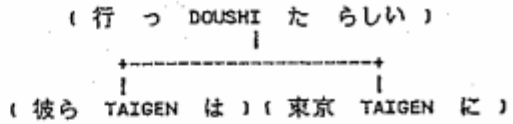
彼らは東京へ行ったらしい。
karera wa tookyoo e itta rashii
They as-for Tokyo to go-past seem
"They seem to have gone to Tokyo"

whose Japanese dependency tree is mapped by transfer into a canonical relational structure, as shown in Figure 2. This canonical structure represents the source sentence in terms of a head predicate and a set of arguments, each of which is labelled with a grammatical relation such as subject, direct object, etc. Note that the canonical structure in Figure 2 does not correspond directly to a grammatical English sentence. That is, **rashii** is interpreted as a unary predicate **seem**, even though the English correspondent requires either an extraposed clause (**It seems that ...**) or an infinitive clause (**seems to ...**). In many systems, transfer would select one or the other of these options. Further, this selection would be invariant, i.e., **seem**

Analysis Tree

```
(SENT   (KAKUYOUSO(TAIGEN$" 彼ら ")
                (KAKUHJ " は "))
        (KAKUYOUSO(TAIGEN$" 東京 ")
                (KAKUHJ " に "))
        (DOUSHI$" 行   っ ")
        (JUTTAIJI" た   らしい "))
```

Japanese Dependency Tree



Canonical Relational Structure (Annotated)

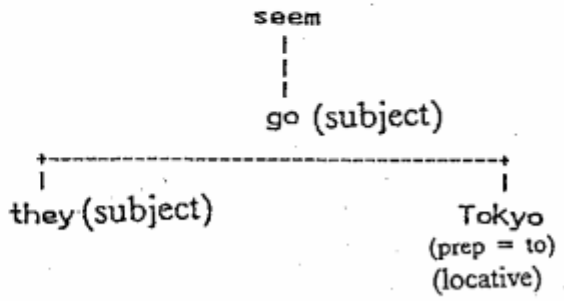


Figure 2: Analysis and Transfer Example

would always be mapped onto the same structure, independent of contextual factors. JETS, in contrast, takes a more abstract approach to generation.

The reason for taking a very abstract approach to transfer and generation is that, as generative grammarians demonstrated long ago, matters such as Extraposition versus Subject-to-Subject Raising interact in complex ways with many aspects of English grammar, including tense, modals, reflexives, dummy **there**, and verb agreement:

***They seem to can swim.**
It seems that they can swim.
They seem to be able to swim.
She seemed to him to be trying to see herself/*himself in the mirror.
There seem (*s) to have been lions in his closet.

Such interactions are quite common. To take another example, consider the contrasting behavior of **certain** and the semantically similar **sure**:

***They are certain to can swim.**
It is certain that they can swim.
They are certain to be able to swim.
***They are sure to can swim.**
***It is sure that they can swim.**
They are sure to be able to swim

As these examples show, the realization of modality in English is not straightforward. If **sure** is selected, then, since raising is obligatory, "ability" must be realized with the predicate **able**. Nor is this problem limited to the realization of "ability"; expression of "futurity", e.g., involves similar considerations:

It is certain that they will win.
They are certain to win.
***They are certain to will win.**
***It is sure that they will win.**
They are sure to win.
It seems that they will win.
?They seem to win. (non-intended meaning)

For transfer to decide properly a wide range of grammatical interactions would require, in general, that it include a complex English grammar. From a design viewpoint, this would put too big a burden on transfer, resulting in a non-modular, unmanageable program. Yet failure to incorporate such interactions must result in ungrammatical or stylistically awkward output. In JETS, such issues are addressed by the generation component's grammar planner.

We now turn to the processing of another example:
母親は子供に薬を飲ませた。
hahaoya wa kodomo ni kusuri o nomaseta.
mother as-for child to medicine dop drink-cause-past
"The mother made the child drink medicine"

Transfer first decomposes the single clause with the morphologically complex verb **nomaseta** into a bi-clausal relational structure. Subsequent lexical transfer maps the Japanese causative morpheme into the English lexical item **make**, etc. In this case, the Japanese canonical structure is isomorphic to the English one.

The general point here is the same: transfer determines neither the superficial grammatical relations nor the complement verb form of the English sentence. In this case, the generator, based on the lexical requirements of **make**, invokes the English Subject-to-Object Raising rule (Postal 1974). Subject-to-Object Raising raises up the complement subject, **child**, as the direct object of **make** (see Figure 3 on page 5). A later rule determines that the complement-verb form is the bare infinitive, rather than the **to**-infinitive required for passivization:

The mother made the child drink medicine.
***The mother made the child to drink medicine.**
The child was made to drink medicine.
***The child was made drink medicine.**

As these examples illustrate, the details of complementation cannot be determined until decisions involving such matters as Subject-to-Subject Raising, Subject-to-Object Raising, Passive, etc. have been

settled. These decisions, in turn, involve clause-level, contextual properties. Once again, the importance of treating generation as an integrated system can be seen. Attempts to build this information piecemeal into transfer must, eventually, lead to poor results.

Subject-to-Object Raising is highly motivated for English and is often called for in Japanese-English MT. For example, consider clauses involving the verb **prevent**, as in **She prevented him from going**. Postal (1974) argued that verbs like **prevent** involve Raising. And this is just what is needed for the task at hand. Some possible Japanese sources for the above sentence include:

彼女は彼が行くのを阻んだ／邪魔した／妨げた。

kanojo wa (kare ga iku no) o habanda/ jamasita/samatageta

she as-for (he subp go nmnl) dop prevent (subp = subject particle dop = direct object particle, nmnl = nominalizer)

The Japanese structure clearly contains a nominalized clause as direct object. Generating the English correspondent involves little work for transfer. The dictionary entry for **prevent** would specify that it governs the preposition **from** and, oversimplifying somewhat, takes Subject-to-Object Raising. This lexically specified information is sufficient for general generation rules to determine the correct superficial form.

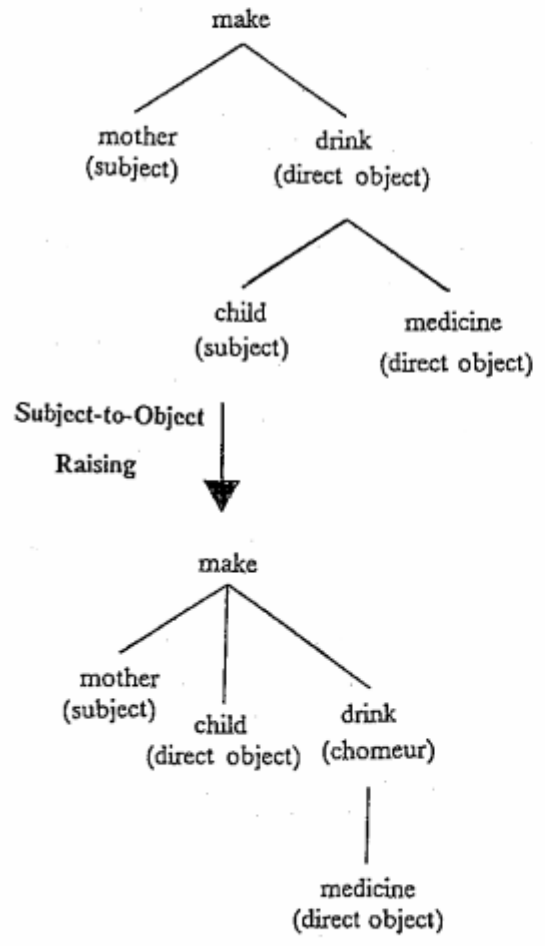


Figure 3: Causative Relational Structure for English

4 THE GRAMMAR PLANNER

The next stage is to submit the canonical relational structure to Gramplan, which controls which generation rules will apply by setting various so-called **rule switches** in accordance with lexical, syntactic and stylistic requirements (Johnson 1988). Rule switches are simply features which name rules in the deterministic generation grammar. Each rule switch may be set either to **Yes** or to **No**, with the obvious interpretation. Switch setting must obey certain principles, chief among which is the principle that *no lexically specified rule switch setting may be altered.*

In the case of **make**, for example, the lexical requirement that Subject-to-Object Raising apply is invoked by simply adding the switch **(B-Raise = Yes)** to the lexical entry. (B-Raising is the name given to Subject-to-Object Raising by Postal 1974.) As futher illustration, we mention two other lexical entries involving B-Raising: **believe** and **want**. To generate **I want him to swim well** and **I want to swim well.** requires adding to the entry for **want**

two specifications, **(Equi = Yes)** and **(B-Raise = Yes)**. (Equi is the governed rule that removes a subordinate subject if it is co-referential to a direct object or subject in the next higher clause.) Note that in the generation grammar, Equi precedes B-Raising, hence, **I want me/myself to swim well** is not generated. (This rule ordering dependency can be removed, but discussion is beyond the scope of this paper.) To generate **I believe that he swims well** and **I believe that I swim well** but not ***I believe to swim well** involves simply adding the switch **(Equi = No)** to the entry for **believe**.

Gramplan also contains non-lexical planning rules. So, e.g., there is a planning rule that sets **(Passive = Yes)** if the subject is unspecified and there is no lexical setting **(Passive = No)**. Alone this would result in the generation of, e.g., **That he swims well is believed**. However, coupling this with Extraposition results in the more felicitous sentence **It is believed that he swims well**. This coupling is accomplished by a default planning rule that sets the Extraposition switch to Yes if certain properties hold.

Gramplan not only sets rule-switch values but also has the power to alter structure. This power is used, *inter alia*, for various relexicalizations. One case involves canonical structures arising from Japanese clauses containing the adjectival suffix **yasui** (**easy**). Such canonical structures are examined to see if the complement contains an intransitive verb. If so, the adjectival clause is adverbialized. This results in the generation of, e.g., **This book is easy to read** from

この本は読みやすい。
kono hon wa yomiyasui
this book as-for read-easy

and the adverbially modified **This food spoils easily** from

この食べ物は腐りやすい。
kono tabemono wa kusariyasui
this food as-for spoil-easy

rather than either ***This food is easy to spoil** or **%It is easy for this food to spoil**. Note in passing that the latter is perfectly acceptable to some speakers and strange to others (hence the **%**). However, if the embedded subject is questioned, then the adverbialized form seems clearly preferable (cf. ***(For) what kind of food is it easy to spoil?** and **What kind of food spoils easily?**). Hence, since adverbialization is always felicitous, it seems justified to do it quite generally.

Certain causative structures are also relexicalized. For example,

母親は子供にスプーンで食べさせた。
hahaoya wa kodomo ni supuun de tabe-sase-ta
mother as-for child to spoon with eat-cause-past
"The mother fed the child with a spoon."

The canonical structure is bi-clausal with matrix predicate **make** and subordinate predicate **eat**. The planner inspects causative constructions to determine whether the embedded predicate (here **eat**) has a related lexical item with causative meaning (here **feed**). If so, a new relational structure is built, where, *inter alia*, subordinate-clause argument relations are re-assigned by the general, relational rule: subordinate subject becomes indirect object of the introduced predicate (**feed**) if the subordinate predicate is transitive (**eat**), direct object if intransitive, and all other relations are reassigned as themselves. A key point is that the newly introduced predicate brings with it *whatever lexical information it has in the generation dictionary*. In the case of **feed**, there is a planning rule that determines that the indirect object will be advanced to direct object by the relational generation rule **Indo-to-Do**, *if there is no direct object present in the newly built clause*. This insures that, e.g., **The mother fed the baby with a spoon** is generated and not ***The mother fed to the baby with a spoon** (cf. **The mother fed cereal to the baby**).

The question arises: why also alter structure at this stage in processing? The answer is that the grammar planner contains sophisticated, often heuristic, knowledge about English grammar and style. It seems unreasonable to attempt to embed this knowledge in the transfer logic. It also seems ill advised to attempt to build these discriminations into the generation grammar. Overall, this highly modular approach seems well justified.

The case of **seem** is of interest in that the lexical requirement is disjunctive: either Extraposition or Subject Raising must apply. Hence, the output of the planner must place either **(Extraposition = Yes)** or **(A-Raise = Yes)** on the verb seem. (A-Raising is Postal's (1974) name for Subject-to-Subject Raising.) This is achieved by adding to the lexical entry for **seem** a specification that tells the planner to execute a specific planning-rule bundle which determines whether Subject Raising or Extraposition is preferred in the given context. Rule bundles typically contain defaults to insure some choice is made, here to insure that either **They seem to have gone to Tokyo** or **It seems that they have gone to Tokyo** will result, and not ***That they went to Tokyo seems**. With **seem**, if the subordinate clause contains a modal, then **A-Raise** is set to **No** and **Extraposition** to **Yes**. With **sure**, which requires Subject Raising, the presence of **can** in the subordinate clause will trigger a restructuring operation; resulting in a **be able** structure **(He is sure to be able to go)**.

Since Gramplan is responsible for determining the idiosyncratic grammatical effects of lexical items in particular environments, the generation rules can be stated in general terms. Those familiar with the history of transformational grammar will recall that the main stumbling block for generation was the formal statement of rule conditions and interactions. By and large, these constraints never received formal treatment, and generative grammarians finally gave up the effort to specify them. High-quality generation, however, whether in the MT domain or elsewhere, will ultimately require a solution to many of these problems. In JETS this problem has been attacked by factoring generation into two modules, as described. This has permitted us to bring various techniques to bear on grammatical/stylistic issues without losing the benefits of employing a deterministic generation grammar.

5 THE RELATIONAL GENERATION GRAMMAR

As mentioned, generation employs a deterministic relational grammar. Following the spirit of the earlier, derivational models of relational grammar proposed in Perlmutter and Postal (1974) and Johnson (1974), generation is bottom-up and distinguishes cyclical rules like Passive, Dative, Subject-to-Subject Raising and Subject-to-Object Raising from post-cyclic (relational) rules like Wh-Question Formation and Relative Clause Formation. Application of the relation-changing rules results in an *unordered*, surface relational-structure. Unlike typical theoretical models of relational grammar, the current computational model is dependency-oriented. Moreover, the surface relational-structure is "node sparse", in the sense that minor terminal elements such as inflections and prepositions are represented as feature values which are "spelled out" during linearization. We prefer to use dependency structures wherever possible because they directly represent the crucial notion **head**, and this is computationally quite convenient.

Linearization is accomplished by a top-down, recursive pass over the superficial relational-structure. Linearization rules are written in the same language as the relation-changing rules. As an example, the linearization rule for verbs states, very roughly and informally, that, given a verb head, V, output the sentence in the order: **Complement-Preposition, Complementizer, Subject, Realize-Verb(V), Particle, Direct-Object, Indirect-Object, Direct-Object-Chomeur, Subject-Chomeur, Locative, Temporal, Other-PP-Modifiers, Clausal-Complements.** Each item (other than the head) in a linearization rule is optional. Additionally, the lexical feature of any item may be empty. For example, **those in the closet** is regarded as structurally isomorphic to **those boots in the closet**, differing only in that, in the former, the noun's lexical feature is empty. Linearization treats lexically empty nodes the same as lexically specified ones, except that no word is printed. This permits the generation

of, e.g., **He can be depended on** to be done unexceptionally, namely, by means of a copy-passive rule which leaves a lexically unspecified, direct-object "trace" to carry the preposition **on**. Linearization rules can refer not only to grammatical relations but also to properties such as part-of-speech and prepositions. Since arbitrary properties can be referenced, fine-grained discriminations can be made in ordering statements. A final point to note is that functions such as **Realize-Verb**, which handles verbal morphology, can be freely embedded in the ordering statements. See Figure 4 for a schematic example of the relational generation of **They seem to have gone to Tokyo.**

6 CONCLUDING REMARKS

JETS has the merit that analysis, transfer and generation can all be developed in a highly independent fashion. To address the so-called control problem, Gramplan has been interposed between transfer and the generation grammar. Grammar planning rules can be developed and tested without altering either the generation grammar or transfer module. On the one hand, the planner removes from transfer the burden of making decisions about grammmatical structure properly considered part of the target language grammar. On the other, it enables the development of a clean, yet robust generation grammar. Furthermore, not only has the relational generation-grammar proven to be quite easy to develop, but the kind of multi-stratal representation postulated within relational grammar provides a theoretically motivated and practical bridge between Japanese and English. Finally, since the generator is being developed solely on the bases of English-internal facts, it can be used both for other MT systems and for other applications altogether.

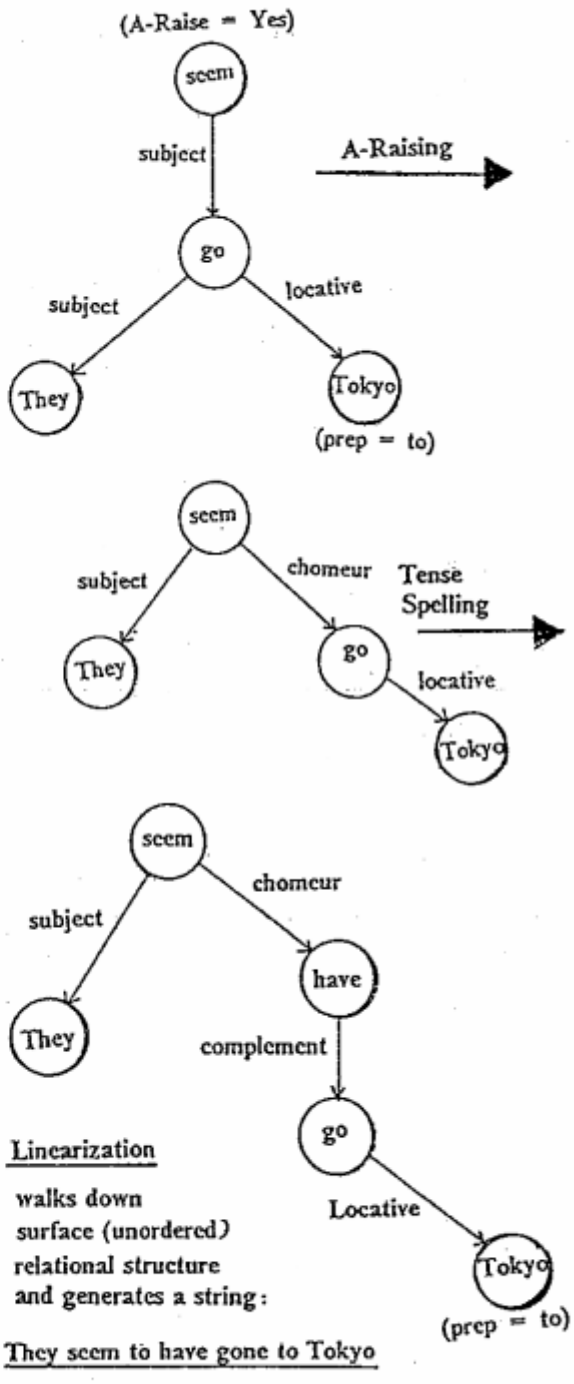


Figure 4: Generation Example

7 ACKNOWLEDGEMENTS

I would like to thank Peter A. Schindler for his major contribution to the development of the Genie generator (cf. Schindler, in prep.). I am also grateful to Paul S. Cohen and John F. Sowa for their very helpful comments on an earlier version of this paper.

8 REFERENCES

Johnson, D. E. 1974. **Toward a Theory of Relationally Based Grammar**. PhD Thesis, University of Illinois. Published by Garland Publishing Co., New York, 1979.

Johnson, D. E. 1988. "A Structured, Universal Natural Language Generator for Sophisticated Machine Translation," **IBM Technical Disclosure Bulletin**, to be issued.

Johnson, D. E. and P. M. Postal. 1980. **Arc Pair Grammar**. Princeton University Press, Princeton, NJ.

MacDonald, D. D. 1987. "Natural Language Techniques: Complexities and Techniques," in S. Nirenburg (ed.), **Machine Translation Theoretical and Methodological Issues**, Cambridge University Press, Cambridge, England.

Maruyama, N., M. Morohashi, S. Umeda and E. Sumita. 1988. "A Japanese Sentence Analyzer," **IBM Journal of Research and Development**, 32.2, 238-250.

Nagao, M. and J. Tsujii. 1986. "The Transfer Phase of the Mu Machine Translation System, **Coling '86, 97-103.**

Perlmutter, D. M. and P. M. Postal. 1974. **Lectures on Relational Grammar**. LSA Linguistic Institute, University of Massachusetts, Amherst.

Postal, P. M. 1974. **On Raising**, MIT Press, Cambridge, Mass.

Schindler, P. A. In prep. **General: an Object-Oriented Approach to Relationally-based Natural Language Processing**. M.S. Thesis, Department of Electrical Engineering and Computer Science, MIT.