EUROTRA

PRACTICAL EXPERIENCE WITH A MULTILINGUAL MACHINE TRANSLATION SYSTEM UNDER DEVELOPMENT

Giovanni B. Varile & Peter Lau
Commission of the European Communities
J. Monnet Bldg. B4/002
L-2920 Luxembourg

ABSTRACT

In this paper we will motivate the design decisions for the architecture of the Eurotra system and its grammar formalism in terms of the intrinsic constraints of multilingual machine translation and the extrinsic requirements imposed by the specific organizational structure of the project. We will give an account of the state of implementation of the system to date and assess advantages and disadvantages of its framework in the light of the implementation.

INTRODUCTION

This paper tries to give an up-to-date picture of the Eurotra project as of the third quarter of 1987, both concerning the state of the implementation of the system and a critical evaluation of the Eurotra formalism.

The paper is structured as follows : in Section 1 we briefly summarize the purposes and organization of the project and the consequences for the design of the Eurotra framework.

In Section 2 we outline and motivate the distinctive aspects of the framework in view of the extrinsic requirements outlined in Section 1 and the intrinsic requirements of multilingual machine translation (MMT).

In Section 3 we give the linguistic assumptions underlying the first small-scale implementation, which will be described in some detail in Section 6.

Section 4 contains an evaluation of the framework, both for its positive and its negative aspects, in the light of the implementation carried out to date.

In Section 5 we describe the design modifications carried out in order to overcome some of the problems spotted during the implementation and report on the recoding effort which was necessary to-date to adapt the original grammars to the modified framework.

1. GOALS AND ORGANIZATION OF THE EUROTRA PROJECT

Eurotra is the European Community machine translation (MT) R&D programme. It aims at producing by 1990 a relatively small multilingual prototype MT system able to translate between all nine official Community languages a limited number of text types in a limited domain, with a lexicon of ca. 20000 entries per language.

The text types Eurotra aims to translate may be defined as typical administrative texts (memos, minutes of meetings, communications etc.) in the domain of IT. The system will be truly multilingual in the sense that neither analysis of source languages nor synthesis of target languages will depend on information which is not exclusively pertaining to the source, respectively target languages in question. As a consequence, the addition of new languages will not imply any modifications to the already existing analysis and synthesis modules, but only the creation of the new transfer modules needed.

Eurotra is a large programme, involving over 160 people located in 20 centers spread all over the European Community. While this decentralized organizational scheme has obvious drawbacks, it is the only possible one which allows us to get all the linguistic and especially language competence necessary for a successful completion of the programme without a brain drain from national centers which would go against the second goal of the programme : spreading natural language processing (NLP) expertise across Europe.

In this respect, Eurotra has already been successful in that it has contributed to a larger awareness amongst European computational linguists of the need and possibility of a continent-wide collaboration in the field while promoting the awareness in governmental bodies in Europe of the growing economical and cultural importance of basic and applied research in NLP.

Furthermore, Eurotra contributes directly in broadening the computational linguistic competence in Europe, having to train a number of people without previous competence in NLP in order to get the manpower that the project needs for a successful completion.

Eurotra's decentralized organization had a strong influence on the design of the system, be it the linguistic metalanguage or the underlying software, because of the necessity of integrating in an easy way modules produced by people with widely differing linguistic backgrounds in geographically distant places. We will elaborate on these consequences in the next section, devoted to the motivation of the design decisions of the system.

More details on the Eurotra organizational structure can be found in [5],[8] and [9].

2. BASIC DESIGN DECISIONS UNDERLYING THE EUROTRA FRAMEWORK

The research orientation of Eurotra is motivated by the intrinsic experimental nature of MMT. On the other hand, we are committed to produce a working prototype MMT system by the end of the programme, which should serve as a basis for the development by industry of the next generation of MT systems. These facts, together with the previously mentioned size and decentralization of Eurotra, have determined the design of the system, be it the linguistic meta-language or the underlying software.

The Eurotra framework has been described in detail in previous papers ([1],[2],[12]). Here we will only briefly summarize its distinctive characteristics.

The central problem in multilingual, transfer-based MT derives from the fact that the number of transfer components of such a system grows geometrically with the number of languages covered (72 transfer components for 9 languages). In order to be feasible at all, such a system requires that the transfer components be kept small, in principle limited to the lexical component. Our problem can be formulated as follows :

what is the nature of the representation of a text guaranteeing translational adequacy ("good quality translation") while allowing simple transfer ?

The answer to this question is not known, and our central effort in defining the Eurotra framework was to create an experimental environment in which research could be carried out in order to find possible answers.

This led us to the design of a system based on multiple, successive representations and mappings between them, leaving the exact definition and the number and nature of such levels to be determined experimentally.

Generators

The set of legal representations for each level is defined intensionally by a generator (in the classical sense) consisting of a set of augmented context free (CF) structure building rules, or B-rules, complemented with a set of feature percolation principles, feature rules or F-rules, and a set of filters, called killer rules or K-rules, to control overgeneration.

These rule systems have a declarative operational semantics based on unification, not unlike other current grammar formalisms ([3], [4], [7], [10], [11]).

But while some of these formalisms explicitly blur the distinction between structural information and feature information ([3], [7]), we decided to keep them strictly separated, like the formalisms described in [4] and [10], while allowing only one level of recursion in features (set valued features) for reasons of simplicity, formally not unlike the SUBCAT feature of [10].

This grammar formalism provides us with an easily understandable, powerful and uniform language and, as a consequence, the training of new members of the Eurotra groups poses no problems, an essential asset in a project with some 160 participants. Furthermore changes of the virtual machine become acceptable as long as the rule type remains essentially the same, because CF based grammars are normally easy to revise.

The current formalism includes devices such as the Kleene star, optionality marker, alternation and negation.

Figure 1 shows examples of the type of rules currently implemented, with a slightly simplified syntax.

Translators

A representation at a given level is mapped onto a representation at the adjacent level by a translation rule (T-

```
:b-rule:

pp = { cat = pp }
            [ { sf = gov cat = prep }
              { sf = compl cat = np } ]

:f-rule:

prep-feat = { pform = X  cat = pp }
                [ { lu = X  cat = prep }
                  { cat = np } ]

:k-rule:

s_mod = { cat = s }
          [ * { }
          { sf = mod pform = of } * { } ]
```

(this last rule expresses the fact that sentences cannot be modified by of-PP)

Figure 1 : Examples of rules

rule). The translation rules must satisfy two criteria :

- they must be *simple* : it must be easier to relate two adjacent levels of representation than to relate text to the semantic representation input to transfer; furthermore, a source level representation is mapped directly onto a (set of) target representations without intermediate steps or representations : we call this the *one-shot translation principle*;

- they must be *compositional* : the translation of an object must be a (simple) function of the translation of its parts; this in order to, on one hand, make the writing of translators modular, and, on the other hand, make the relation between two representational objects established by a set of translation rules understandable for the linguist.

Figure 2 shows two examples of translation rules as currently implemented. The first one is extracted from the English to German transfer module, while the second one is part of the Italian analysis component.

```
ti7 = { sf=gov cat=v arg1_feat=collective
 arg2_feat=abstr_nonhum lu=verabschieden }

     => adopt

ti5 = { cat=s coord=yes }
          [   S1{ cat=s coord=no }
            CONG{ cat=conj }
              S2{ cat=s coord=no  } ]

    => coord( CONG  S1 S2 )
```

Figure 2 : Examples of translator rules

3. EUROTRA'S LINGUISTICS

The generators for the first small implementation have been defined in such a way that they mirror traditional linguistic ideas about analytical levels : morphology, surface syntax (immediate constituents and syntactic functions), deep syntax and semantics. However, in order to offer a full treatment of all texts in our text type and domain without pre-editing, we also included generators which cater for character normalisation (in order for the dictionary to be independent of typography) and text structure (e.g. lay-out, text format, figures, footnotes).

At present the linguistic specifications define 6 levels in the Eurotra analysis and synthesis modules :

ETS (Eurotra Text Structure)
ENT (Eurotra Normalised Text)
EMS (Eurotra Morphological Structure)
ECS (Eurotra Constituent Structure)
ERS (Eurotra Relational Structure)
IS (Interface Structure)

Figures 3 and 4 show an ECS, respectively IS, representation for the sentence : "The decision adopted by the Council on 25 April 1983 was implemented by the Member States in the course of 1983".

The ECS representation of Figure 3 is reduced structurally through two steps.First it is translated into the relational representation ERS which identifies syntactic functions, elevates determiners, auxiliaries and valency bound prepositions and rearranges the constituents into a canonical order.

Then the relational representation is translated into the interface structure IS whereby passive is undone, empty elements are inserted for obligtory arguments which are absent in the surface form and semantic information is added to the feature bundles.

Note that, although the IS representation is fairly simple from a structural point of view it still allows for modifier attachment at different levels of embedding, and thus, the two TIME constituents are attached to their proper governors without the use of complicated featurised references.

All generators being described in the same formal language, the representations of text structure, normalised text and morphology are built by augmented CF rules, just like the syntactic representations.

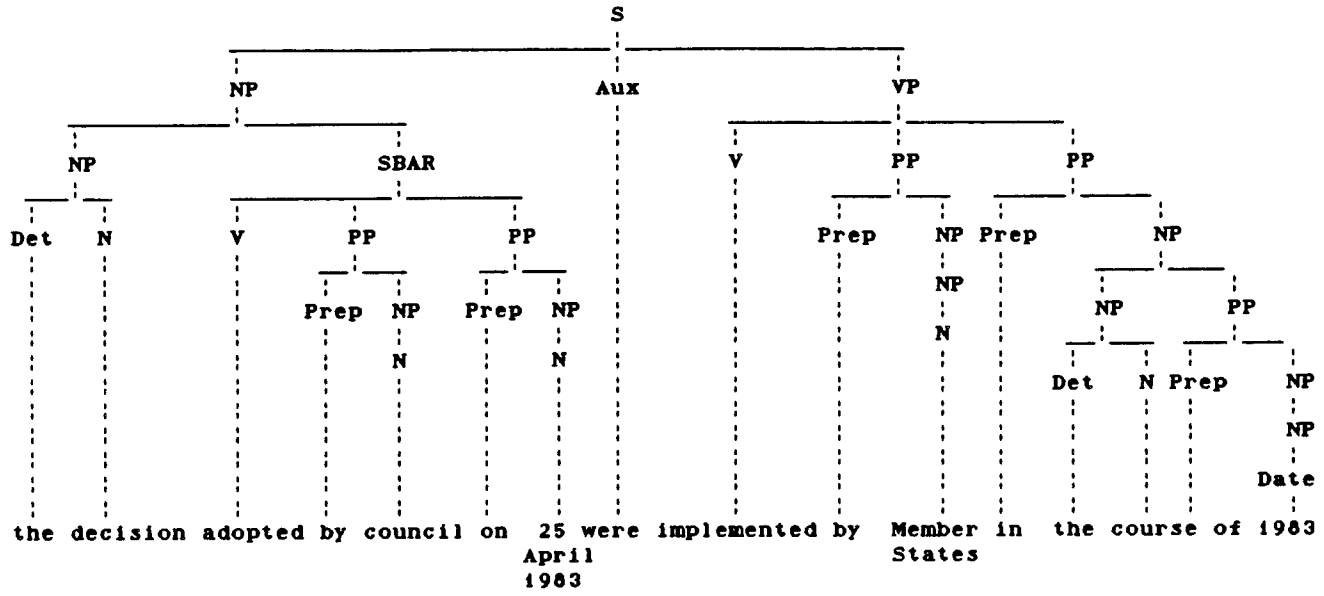

Figure 3 : Example of an ECS representation[1]

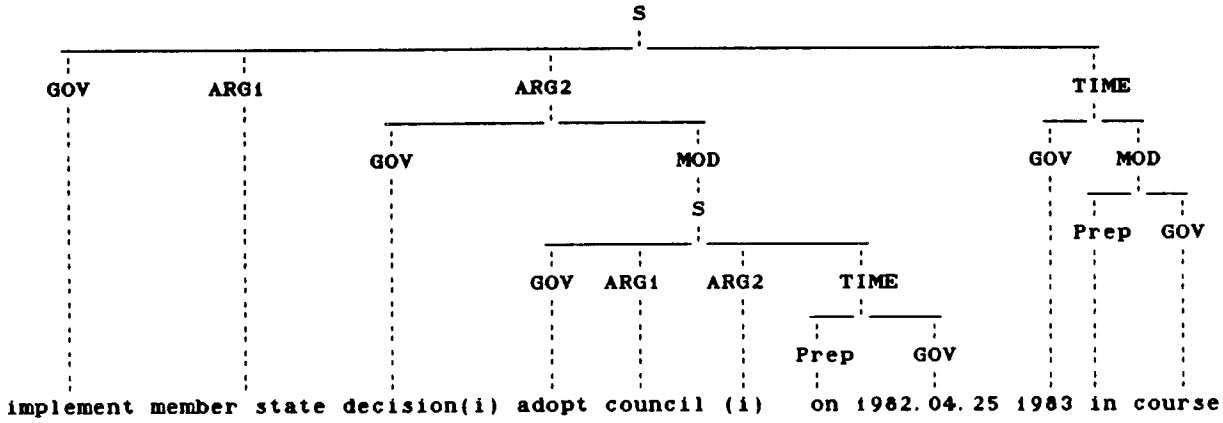

Figure 4 : Example of an IS representation[1]

The modular approach to linguistics, whereby text format, morphology, syntax and semantics are handled by separate generators, enhances the repairability and extensibility of the system. This means for instance, that changes of the ETS, ENT and EMS grammars may be made freely as long as they produce the output needed by ECS.

The sequence of representations related by T-rules spans the distance between the actual text and the interface structure (IS) which is the beginning and end point of transfer.

In order to simplify transfer, IS abstracts away from surface phenomena like inflection, derivation, compounding, constituent structure etc. It contains semantically labelled arguments and modifiers related to a predicate, and the

[1]These trees are given in the form which is output by our parametrizable prettyprinter with the parameter set to the "cat" feature in Figure 3 and to the "sr" (for semantic relation) feature in Figure 4. Representations including all feature information are much more complex and impractical to read.

transfer dictionaries, ideally, just relate lexical units of one IS to lexical units of another.

In some cases, though, this does not work, because of, e.g., lexical holes like the German word "Schimmel" (meaning "white horse") which has no lexical correspondent in English, or different functions mapping onto one another as the English predicate "like" which maps onto a German adverbial modifier "gern". In these cases explicit non-lexical T-rules must be written for transfer.

The purpose of the IS experimentation in Eurotra is precisely to minimize the number of explicit transfer T-rules and the entire modular design as it has been proposed in the linguistic specifications is primarily geared towards an experimentation process aimed at making it possible to reach an optimal IS through multiple cycles of prototyping.

4. EVALUATION OF THE FRAMEWORK

The architecture of our framework has given us full satisfaction with respect to its *modularity*, *simplicity* and the ease with which we could *modify certain design characteristics* of the system, as described in Section 5.

Its *modularity* has made it possible to experiment with the interface structure quite independently from the rest of the system. For instance, research and experimentation about an adequate treatment of time and modality could be done in parallel with, and independently of, grammar implementation work concerning other levels of representation.

The *simplicity* of the generator's formalism has had positive and negative aspects. Amongst the former we can mention the fact that it was *easy to learn and teach*, *easy to modify* and a reasonably good *communication tool* for the scientists involved in the definition and implementation of the system.

On the negative side we must mention the fact that it was not expressive enough to describe in a natural way all the phenomena occurring in the languages we are treating[1]. Furthermore, for a system which requires the T-rules to be compositional, the operational semantics of generators turned out to be too poor, causing a proliferation of complex T-rules.

For the implementation work, this has caused some problems. For instance, we couldn't implement ETS, ENT and EMS in the first cycle because the prototype only allowed structure manipulation to happen at one level. This meant that we could not build text structure, morphological structure and syntactic structure independently, they all had to be built by one generator, which then became very big and difficult to manage.

Another problem was that the analytical strategy of elevating functional elements like articles and prepositions, which is motivated by the needs of MMT rather than by linguistic considerations, could not easily be reversed, because building new nodes in synthesis to represent these elements is addition of structural information.

In consequence, we had to change the specifications of the virtual machine in such a way that each generator had the power of *completing* a representational object according to its own definition of well formed representation, thus alleviating the task of the translators.

The *modifiability* of our original framework was invaluable in this redesign since it allowed us to keep the core concepts basically unchanged while extending the functionality and expressiveness of the formalism.

5. DESIGN MODIFICATIONS TO THE FRAMEWORK

As reported above, the first implementation showed that two factors were responsible for the heaviness of the translators (not to be confused with transfer) : the relatively simple operational semantics of the generators combined with the requirement that the translators be compositional.

Rather than giving up compositionality, which guarantees a well defined relation between representations belonging to adjacent levels, we increased the power of the generators by modifying their operational semantics to include a controlled form of *addition* of structural information, rather than just addition of feature information.

Translators were impoverished to the

---

[1]This is true even considering the fact that mechanisms for treating unbounded phenomena, for expressing rules in an ID/LP format and for built-in feature inheritance were given second priority, and, therefore, were not implemented in the first version of the framework, nor in the first revision of the framework reported here.

point where they can now be defined by default almost everywhere by specifying correspondences between feature theories pertaining to adjacent generators. Special, exceptional cases of T-rules have still to be specified explicitly.

In the modified framework, the representation output by a translator is completed by the target generator. For instance, ECS will eventually expect as input from the preceding level, which is EMS, a tree with the top node T(ext) branching into C(hapter), Se(ctions) and P(aragraphs). The P nodes should then branch directly into wordform nodes which dominate tree-structure representations of the morphological structure of each wordform. The ECS generator will then complete this tree by inserting S nodes and nonterminal categorial nodes of the constituent structure representation of each sentence, resulting in a parse tree of the sentence.

From our original framework we retain the architecture, i.e. breaking up the monolingual components of the system into a sequence of generators which are related by translators. In principle the same generators are used for analysis and synthesis, but we don't know yet whether the non-default translators can also be used in both directions.

The first implemented prototype based on the revised framework has been used for experiments with rewriting the grammars of the first implementational cycle, and these experiments have confirmed the assumption that recoding of grammars does not pose special problems.

Here it must be mentioned that an alternative modified framework has been proposed to overcome the inadequacies discovered during the first cycle of implementation, which departs more radically from the original one. The testing that this alternative revised framework has undergone has been more limited for various reasons. The merits and demerits of the two proposed revisions will be assessed during the first quarter of 1988 in a controlled experiment.

7. CURRENTLY IMPLEMENTED SYSTEM

The majority of the implementation work to date has been carried out within the original Eurotra framework leading to a system covering to varying degrees the original seven languages foreseen, that is Danish, Dutch, English, French, German, Greek and Italian. Work done on Portuguese and Spanish is scheduled in a different way, given that Portugal and Spain became members of the Community only in 1986. Analysis modules exist for all languages, generation for five languages. Transfer components have been written for the following ten language pairs :

Danish - English	Danish - German
English - Danish	English - German
English - Greek	French - Greek
German - Danish	German - English
German - Greek	Greek - English

An average grammar has ca. 400 rules and 600 lexical entries (accounting for ca. 3000 full-word forms in moderately inflected languages).

Before the end of 1987 it is expected that all already implemented components will be recoded in the new formalism, that ten more language pairs will be added while the lexical and linguistic coverage will be increased.

Only the levels ECS, ERS and IS have been implemented to date. This means that in the first implementation each monolingual component works on the basis of a full-form dictionary, since the generator for morphology was given second priority, and accepts only single sentences as input.

The linguistic coverage includes main clauses and relative clauses; all types of noun phrases; simple coordination in noun phrases; all verbal tenses (excluding modal constructions); possessive, relative, reflexive and indefinite pronouns; all prepositional phrases; adverbs; numerals; particles.

Research and experimentation are still ongoing to include ellipsis; modality; negation; scope; quantification; time-tense relation and pronoun resolution, but we do not expect to be able to do pronoun resolution on the basis of real world knowledge in the near future.

The implementation of the virtual machine is done mainly in C-Prolog. A new version of the software including a relational data base for the coding and maintenance of the lexicon (to be extended to grammars) has just been released. More details on the philosophy of the software construction can be found in [6].

A fragment of a sample grammar for English is shown in Appendix A. Some examples of T-rules are given in Appendix B.

Future Work

By mid 1988 we plan to have a small scale, corpus based, prototype with a coverage of 2500 lexical entries per language for all the seven original languages together with all the 42 transfer components.

In the shorter term, several additions to the new framework are foreseen, the most important of which concern the possibility to express grammar rules and T-rules in an ID/LP type format [4] and a mechanism for treating unbounded phenomena.

Current research topics in linguistics have been mentioned above. In relation to the framework, we are currently investigating about a feature inheritance mechanism and some restricted form of complex features other than the set valued features mentioned in Section 2.

CONCLUSIONS

In this paper we have reported on the organization and current state of implementation of the Eurotra project. We have briefly summarized and motivated the design decisions underlying its formal framework, reporting on the inadequacies discovered during the first implementation work.

We have argued that the basic design was adequate in that it supported the experimentation on the interface structure. Furthermore, where the design had to be modified it could be, and the necessary recoding of grammars did not cause major problems.

Finally, we have mentioned further modifications forseen in the future and, based on our past experience, we are confident that we will be able to carry them out in a way which will not cause significant disruption of the grammar and lexicon coding work.

ACKNOWLEDGEMENTS

The work reported in this paper is the result of a collaborative effort of many people over a number of years. We cannot claim exclusive authorship of any of the ideas exposed in the paper. Any omission, imprecision or error are of course our sole responsability.

REFERENCES

[1] Arnold D. (1986)
Eurotra : a European Perspective on MT
in: Proceedings of the IEEE, Vol 74, No 7, Special Issue on Natural Language Processing, pp. 979-992

[2] Arnold D. J., Krauwer S., Rosner M., des Tombe L., Varile G. B. (1986a)
The <C, A>, T Framework in Eurotra : a Theoretically Committed Notation for MT
in : Proceedings of COLING86, ICCL, pp. 297-303

[3] J. Bresnan (Ed.)(1982)
The Mental Representation of Grammatical Relations
M. I. T. Press, Cambridge, MA

[4] G. Gazdar, E. Klein, G. K. Pullum and I. Sag (1985)
Generalized Phrase Structure Grammar
Harvard University Press
Cambridge, MA

[5] Johnson R., King M., des Tombe L. (1985)
EUROTRA : a Multilingual System under Development
in : Computational Linguistics, Vol 11, Number 2-3, April-September, pp. 155-169

[6] R. L. Johnson, S. Krauwer, M. A. Rosner, G. B. Varile (1984)
The design of the Kernel Architecture for the Eurotra Software
in : Proceedings of COLING84, ACL, pp. 226-235

[7] M. Kay (1983)
Unification Grammar
Technical Report
Xerox PARC
Palo Alto, CA

[8] Perschke S. (1986)
Eurotra : General overview
in : Multilingua, Vol 5, No 3
Mouton De Gruyter, 1986, pp. 134-135

[9] Perschke S. (1987)
Eurotra : The EEC R&D programme for the creation of a machine translation system of advanced design
in : Proceedings of the Hakone MT Summit

[10] C. J. Pollard (1984)
Generalized Phrase Structure Grammars Head Grammars and Natural Language
Ph. Dissertation
Department of Linguistics, Stanford University

[11] S.M.Shieber, H.Uszkoreit, F.C.N.Pereira, J.J.Robinson and M.Tyson (1983)
The Formalism and Implementation of PATR-II
in : Research on Interactive Acquisition and Use of Knowledge
A.I. Center, S.R.I. International, Palo Alto, CA

[12] des Tombe L., Arnold D.J., Jaspaert L., Johnson R., Krauwer S., Rosner M., Varile G.B. & Warwick S. (1985)
A Preliminary Linguistic Framework for Eurotra
in : Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages
Colgate University, Hamilton, N.Y., pp. 283-288

APPENDIX A Sample Grammar Fragment

```
%this is a sample fragment of a ECS grammar for English as implemented
%comment lines start with a "%"

:level:ecsEn

%this declaration states that the grammar is a fragment of an English ECS grammar

:b:

%this declaration starts the b-rule section of the ECS grammar

s1 = {cat=s type=main tense=T voice=V mode=M}
        [ {cat=np} {cat=vp tense=T voice=V mode=M} ^{cat=np} ^{cat=pp} ]

vp1 = {cat=vp tense=T voice=V mode=M}
        [ {cat=v tense=T voice=V mode=M} ^{cat=np} ^{cat=pp} ]

vp2 = {cat=vp tense=T voice=pass mode=M}
        [ {cat=aux v_type=finite tense=T mode=M lu=be}
          {cat=v v_type=ptc} ^{cat=np} ^{cat=pp} ]

np1 = {cat=np n_type=T pers=P num=N case=C def=D}
        [ ^{cat=detp def=D} *{cat=ap def=D} {cat=n n_type=T pers=P num=N case=C} ]

pp1 = {cat=pp prep=L}
        [ {cat=prep lu=L} {cat=np} ]

ap1 = {cat=ap degree=D}
        [ {cat=adj degree=D} ]

detp = {cat=detp def=D}
        [ {cat=det def=D} ]
```

APPENDIX B Sample T-rules

```
%T-rule between ECS and ERS for elevating a determiner;
%capital letter symbols in front of feature bundles are indices (names) which can be
%referred to in the right hand side of a T-rule for coding economy

tnp1 = NP {cat=np}
          [ {cat=detp} A*{cat=ap} N{cat=n} ]

    => NP [ N A ]

%T-rules between ERS and ECS for eliminating a VP node

ts1 = S {cat=s type=main}
        [ NP1{cat=np} {cat=vp}
                      [ V{cat=v} NP2 ^{cat=np} PP ^{cat=pp} ]]

   => S [ V NP1 NP2 PP ]
```