Conceptual text representation for multi-lingual generation and translation

Lars Ahrenberg & Stefan Svenberg
Linköping University

This paper presents some ideas and preliminary results of a project aimed at automatic generation and translation of text from conceptual (interlingual) representations. In the first part we give some arguments for treating translation and generation as closely coupled processes and motivate the need for conceptual text representations to aid these tasks. In the second part we describe an implemented method for multi-lingual generation of sentences.

1. Introduction

During the last decade unification-based grammar formalisms have become standard tools for grammatical description within computational linguistics. A significant advantage of them is their declarativeness, something which implies that they can be used by parsers and generators with the same ease. Two developments in recent years are particularly interesting from the point of view of translation and multi-lingual generation. The first is the idea that descriptions of different languages can be related by virtually the same means as descriptions belonging to different levels of description within a single language (Kaplan et al., 1989; van Noord, 1990; Russell et al., 1990). This latter development is especially suitable for those who favour a transfer model of translation, as well-known description levels such as phrase structure, functional structure and logical form can be reused and set into correspondences.

The second idea is that parsing and generation can be seen as basically the same processes that differ only in their input (Shieber, 1988; Zajac & Emele, 1990; Emele et al., 1990). This would make it possible to handle all processing by the same mechanisms and by means of a single grammar and dictionary for each language. This idea, on the other hand, gives something to those who are interested in interlingua approaches, since, if a common representation language can be found for the description of different languages, we can use it in the different grammars and get translation systems and multi-lingual generation system almost for free.

In this paper we report on a project which is concerned with developing the second idea.[1] In the next section we describe briefly its goals and motivations. Then, in section 3, we provide some arguments in favour of conceptual text-representations and propose a way of characterizing semantic equivalence of generated texts. In section 4, we show how the conceptual representations are used for bi-lingual sentence generation. In the last section, finally, we indicate briefly our plans for the future.

Proceedings of NODALIDA 1991, pages 7-18

2. The project

The project is explorative with the overall aim to judge the possibilities of using interlingual representations in applications such as document generation, intelligent (on-line) handbooks, and translation. We do this by studying a specific text genre, the service manual, and in particular the expository sections, where an object is described and its function is explained. Our initial corpus comprises 86 sentence pairs from two service manuals issued by Volvo Truck Corporation. More specific goals of the project are to develop an interlingual representation language for this genre and algorithms for multi-lingual generation from the interlingual representations. We also investigate the possibilities of using the interlingua for automatic translation. Two languages are considered, Swedish and English.

The interlingua should cover all relevant aspects of the texts: structure, content and textual coherence, but we are only concerned with the linguistic variation that can be found within the genre. For instance, as all sentences of our corpus are declarative and present-tense, we treat such properties as structural invariants that need not be accounted for on semantic or pragmatic grounds.

A basic assumption of the project is that it is useful to consider translation and multi-lingual generation as closely related tasks. If we view the general generation problem as one of finding a text (or all texts) that satisfies a given set of constraints in a given context, it is easy to see that translation fits this definition. The source text provides one set of constraints while the grammar and genre-specific rules of the target language provide another set. On the other hand, in multilingual generation, we must somehow ensure that the texts that are generated are equivalent in important respects, e.g. as regards content, style and progression. The requirements of such an equivalence relation comes very close to what one demands of a relation between translations.

3. Towards a characterization of equivalence

The usual way to characterize the relation of translation equivalence is perhaps with reference to a number of description levels on which two texts should be identical or corresponding. For instance, Carbonell & Tomita (1987) mentions the following factors to be important for good translations: pragmatic invariance (matters of illocutionary force, style etc.), semantic invariance, structural invariance, lexical invariance and spatial invariance. Ignoring the latter, which is concerned with the external properties of the text such as length and page layout, it is interesting to note how invariance is decribed with respect to the different levels. In the case of pragmatics and semantics the notion is one of "preserving invariant" the relevant properties, while structural invariance is explained as "preserving as far as possible" the syntactic structure, and lexical invariance is explained as "preserving a one-to-one mapping of words or phrases from source to target text". We see that it is easier to imagine a common, language-independent representation for the higher levels of texts, whereas the lower levels, such as syntax and lexis, can only be brought into correspondence with each other.

However, even if two lexemes, or two constructions, of different languages cannot be treated as having the same properties, but set into a correspondence for lack of better alternatives, we can still give them a common description. From the formal point of view there seems to be no relevant distinction between identity and correspondence, since, if two elements correspond, we can introduce a property at the interlingua level that is expressed by these two elements (and no others) in the two languages. The association between the elements and the interlingual descriptor is then made in the grammars of the two languages. Conversely, of course, a perceived identity of meaning of two elements from different languages can be represented as a trivial correspondence between the elements. Note, though, that by using an interlingua representation we can decompose a

correspondence between often, quite complex sets of properties into two simpler relations both of which relate a simple descriptor at the interlingua level to complex language-specific representations, just as a word form is associated with its morpho-syntactic and semantic properties in a dictionary. This is advantegous not least from the point of view of multi-lingual generation.

Now, if we want to get at the bottom of a generation problem, we will need to refer to complex combinations of properties. But this is the case whether we work with interlingua representations or correspondence rules. With the interlingua approach we then have the advantage that all information is accessible at a single level of description. It is often argued that in the case we deal with languages that make the same kind of distinctions and use the same kind of constructions, as is the case with Swedish and English, we need not consider the relevant pragmatic and semantic properties, but merely note the correspondences. However, in our corpus we find several pairs of sentences, such as the following, that seem difficult to describe on the basis of structurally based correspondence rules only:[2]

1. An ellipsis occurs only in one of the languages, but not in the other.
   S: Basväxeldelen manövreras mekaniskt, rangeväxeln pneumatiskt.
   E: The basic section is mechanically operated; the range gear is pneumatically operated.

2. An integrated complement corresponds to the subject head, while the subject corresponds to a subject modifier.
   S: Spärrventilen har till uppgift att förhindra växling av rangeväxeln när ...
   E: The purpose of the inhibitor valve is to prevent inadvertant shifting of the range gear when ...

3. A passive clause corresponds to an active clause with an anaphoric subject.
   S: Spärrventilens kolv trycks upp ur fördjupningen på kolvstången ...
   E: This moves the plunger of the inhibitor valve out of its dimple ...

4. A simple NP corresponds to a coordinated, disjunctive NP.
   S: Den här nedkylningen kallas laddluftkylning.
   E: This cooling process is known as charge air cooling or intercooling.

As a basis for describing the semantic equivalence of two sentences that are translation equivalents we appeal to the notion of topic, or topical question (Carlson, 1983; Ahrenberg, 1987). Speaking informally, we can say that a necessary condition for two sentences being translation equivalents is that they answer the same question by the same standards, where standards refer to such things as truthfulness, completeness, clarity and relevance.[3] This is actually also a condition that can be applied in practice; often it is not a difficult task to decide which question or questions a given text sentence attempts to answer, as evidence both from its form and its context can be used.

If we look at the four sentence pairs above, the topical questions of the first pair can be rendered in English as *How is the basic section operated, and how is the range gear operated?* At the conceptual level we may introduce a concept, Operation, with two arguments, one for the object (gear or gear set) being operated upon and one for the manner in which it is done. A question that relates to this concept, one that makes it a topical concept, can be represented as a propositional structure which is unspecified with respect to one of its arguments:

$$\begin{bmatrix} \text{aspect} & \text{Operation} \\ \text{thing} & \text{r-gear1} \\ \text{value} & [\,] \end{bmatrix}$$

As for the occurrence of the finite verb in the second conjunct of the English sentence and its absence in the corresponding Swedish sentence, it can be handled completely within the grammars of the two languages. We need not formulate a separate correspondence rule saying that a finite verb in one language can correspond to nothing in the other language under certain circumstances.

As for the second pair of sentences we are faced with a correspondence pattern which is even more involved than the splitting/fusing examples discussed by Kaplan et al. (1989) and Sadler & Thompson (1991). We can avoid introducing explicit correspondence rules, if we state the rules in terms of relations between interlingua descriptions and language-specific grammatical descriptions, however. Moreover, they become quite simple because the interlingua description is a simple one. The topical question is *What is the purpose of the inhibitor valve?* with the interlingua description

$$\begin{bmatrix} \text{aspect} & \text{Purpose} \\ \text{thing} & \text{inh-valve2} \\ \text{value} & [] \end{bmatrix}$$

The rules we need are associated with the concept Purpose in the knowledge-base as explained in section 4.4.

The third pair illustrates the importance of co-text. The topical question may be rendered as *What happens in connection with this?*, where *this* refers to an event of shifting the range gear described in the previous sentence. While the event is explicitly referred to in the English sentence, it is not so in the Swedish sentence, illustrating the common property of texts that causal relationships between events are often not given explicit expression. At present we have not defined text-level rules, so this sentence-pair cannot be handled by our generator, but it seems clear that a correspondence rule using only structural information is not sufficient for the purpose.

The fourth pair of sentences, finally, addresses the topical question of what a certain process is called. As it happens two terms are used for it in English while only one is used in Swedish. The result is that a disjunction is used in English – probably in response to some standard of completeness – with no corresponding disjunction in the Swedish sentence. The fact that a simple NP in one language in some cases can correspond to a disjunctive NP in another language is for obvious reasons not something that one would like to express as a general possibility of structural correspondence. However, without access to a semantic/pragmatic level of description one cannot express the appropriate constraints.

4. The unification-based generator

4.1 Overview

In this section we show how sentences can be generated from conceptual representations. We refer to these representations as content descriptions as they mainly contain semantic information. The generation process roughly have the stages illustrated in figure 1.

The first module of the generator constructs language specific grammatical descriptions using relational grammar rules and information in the common knowledge-base (KB). These will then be fed into the surface string generator, which has its own grammar. Between these two main modules there is also an interface whose purpose is to fine-tune the incoming grammatical descriptions so that they conform to the demands of the surface generator.

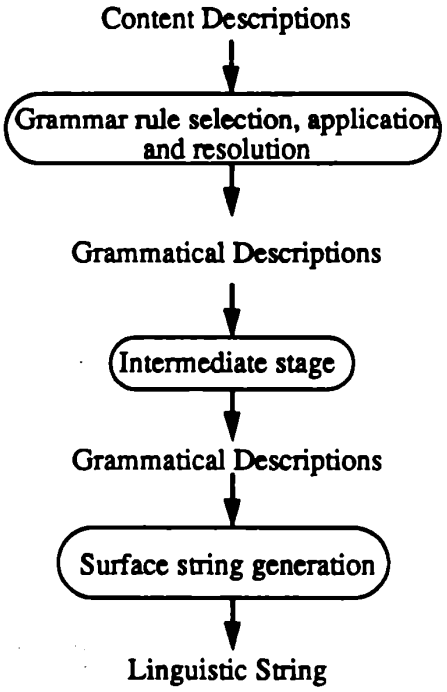


Figure 1: Basic architecture of the generator

An editor is built on top of the KB enabling content descriptions, grammatical descriptions, grammar rules, domain and linguistic knowledge to be interactively and incrementally added or modified. All information is coded as feature-value matrices with unification as the sole method of combining.

The generation process is not language specific. The KB can support knowledge for several languages. Given a content description the system concurrently generates the corresponding strings for all the supported languages. In the next subsections we will describe content descriptions, grammar rules, and KB for the first stage of generation. The surface string generation module is modeled on Shieber (1988) and uses grammars written in the PATR-II-format.

4.2 Content descriptions

In order to achieve multi-lingual generation we want to work with language independent chunks of information. The traditional approach is to use logical propositions, such as a conjunction of facts, which the text is supposed to express. The content descriptions in the system contain logical propositions, but also distinguish them on the basis of pragmatic function. Moreover the content descriptions need not be fully specified as some of the content may be retrieved from the KB.

Propositions can be of a number of different types. They are expressed differently depending on the type. The types can be grouped into two main categories. First those that have a purely logical meaning, currently being the simple type thing-aspect-value (tav) and the conjunction (and). Secondly, we have a number of types that besides having a logical interpretation also have a non-logical function. Below we will discuss instances of both types and give small examples of their use.

- Thing-aspect-value, which is used to express a value of an aspect of a certain thing. The example below shows a simple fact written as an attribute-value matrix which says that a thing called r1000 has the color green.

$$\begin{bmatrix} \text{type} & \text{tav} \\ \text{thing} & \text{r1000} \\ \text{aspect} & \text{color} \\ \text{value} & \text{green} \end{bmatrix}$$

- And, a logical conjunction of two other propositions:

$$\begin{bmatrix} \text{type} & \text{and} \\ \text{a1} & [\,] \\ \text{a2} & [\,] \end{bmatrix}$$

Most often the a1 value is of type tav while the a2 is either a tav or an and.

- Reference-scope, which logically is a conjunction. It has two attributes, called ref and scope each containing propositions about a given object (or set). The non-logical aspect of it lies in the way the two propositions have been divided. The first argument specifies a proposition which is used for making reference to the object. The intention is that the object and proposition must be known to the reader at this point, either by having been introduced at an earlier point in the text or included in the reader's background knowledge. The proposition will determine how reference to the object will be achieved at the surface level; e.g. by using its proper name, a pronoun, or a definite description, possibly including a number of adjectives and so on. The scope value contains information that is new in the current context. It identifies the proposition that is asserted about the object.The reference-scope type is used to express the contents of simple clauses. The first argument will form the subject of that clause while the scope value is used to build the predicate, including verb and complements.

Below is a simple example of a content description of the clause *The basic section is mechanically operated.* "The basic section" is known while "mechanically operated" is new information which is asserted in the sentence.

$$\begin{bmatrix} \text{type} & \text{rs} \\ \text{ref} & \begin{bmatrix} \text{type} & \text{tav} \\ \text{thing} & {}^{1}[\text{r1000bs}] \\ \text{aspect} & \text{isa} \\ \text{value} & \text{b-section} \end{bmatrix} \\ \text{scope} & \begin{bmatrix} \text{type} & \text{tav} \\ \text{thing} & {}^{1}[\,] \\ \text{aspect} & \text{Operation} \\ \text{value} & [\,] \end{bmatrix} \end{bmatrix}$$

The value r1000bs means the basic section of the gearbox r1000. The aspect *isa* gets the reflexive and transitive closure of the classes r1000 belongs to. From these, the concept b-section has been chosen. It has the name "basic section" attached to it.

The scope value is an unspecified proposition, i.e. one corresponding to a topical question as explained above. The unspecified value will be retrieved from the KB during the generation process.

4.3 Mapping content descriptions onto grammatical descriptions

A content description represents properties that are common to equivalent sentences of different languages. In addition, each sentence in the equivalence class satisfies a language-specific grammatical description. Grammatical descriptions and content descriptions are related roughly as the two sides of the classical Saussurean linguistic sign. Thus, it is natural to express possible relations between grammatical features and content features as a relation between partial structures. In the current grammar we actually use a number of different relations, which encode both the language and the textual function of the structures being related, as in the following examples where cd and gd stand for content and grammatical descriptions respectively:

1. inform-sw(cd, gd), refer-eng(cd, gd), describe-eng(cd, gd)

Another possibility would be to encode the function and language as arguments:

2. signify(cd, languge, function, gd)

A specification of the relation under a more complete generation scheme would have to take many other aspects into account, e.g. as follows:

3. signify(cd, language, function, type, user-model-in, context-in, context-out, user-model-out, gd)

where 'type' is the type of text object; such as clause or sentence. 'user-model-in' is a model of what background knowledge the reader posesses at this point, 'context-in' records relevant objects mentioned in the text so far to aid pronoun generation and ellipsis. The user-model-out will reflect the fact that the reader has accomodated the new information in the content description. This can be used to ensure that the new information indeed is relevant, coherent and consistent with what has been said and with what the reader already knows.

From now on version 1 of the relation specification will be considered since it is the one has been implemented.

Given a content description and a grammatical description the generator will try to prove the relation between them. Either description may be partially specified. The generator will during the proof procedure suggest instantiations to complete the specifications. If we would like to generate a grammatical description that argument should initially be uninstantiated. If the process was successful the generator returns the answers one by one even if there are an infinite number of them. In principle, we could also parse a grammatical description which would lead to a content description. At the time of writing this is not yet practical for efficiency reasons. The generator works according to the principle known as SLD-resolution (see e.g. Nilsson & Maluszynski, 1990). The attribute-value matrices are coded as directed acyclic graphs. The selection function is graph unification. This is similar to, but not necessarily limited to, the way logic programming languages work.

The grammar is a rule base, where each rule generally takes the following form:

$rel_0(cd_0, gd_0) \leftarrow rel_1(cd_1, gd_1), ..., rel_n(cd_n, gd_n).$

We refer to the left-hand side as the head and the right-hand side as the body of the rule. The head consists of a single term, while the body may contain any number of terms. The rules currently in

use actually contain terms with additional arguments, but we ignore these here.

Given a content description for a sentence, we will first construct a term having inform-sw or inform-eng as a functor. The content description will become the first argument and the second argument will be initialized to the null dag, giving, say, inform-sw(cd, []). The generation of the grammatical description will start by picking out those rules which have inform-sw as the functor of their heads. They will then be tried out one by one. The arguments of the term will be unified with the corresponding argument of the rule head. If that succeeds all subgoals appearing on the right hand side of the rule must hold. These are tried out recursively. The arguments of the subgoals share material with each other and particularly with the head arguments. As new material is unified into the structures,the changes become immediately visible everywhere. When all the subgoals have successfully been proved the process suspends in that state. The arguments of the initial term have been fully instantiated and can be picked out. After that the process resumes by backtracking to choice points in the SLD-tree where alternative instantiations can be found.

The rules are written in such a way that a structural depth-first analysis will be performed on the content description. This also means that the grammatical description will be built top-down. The subgoals take care of the substructures. The recursion is stopped, aside from failing unifications, by rules having no right hand side or by built-in rules accessing the KB.

The far most important built-in rule is the the primitive retrieval operation, iget. It has three arguments: carrier, indicator, and value. The carrier denotes an object that has a value stored under a certain indicator. It can, from the generator's point of view, be regarded as a simple property. The iget is used for two different purposes:

1. Checking the validity of the content description. The world modelled in the KB must sanction the information expressed there.
2. Retrieving linguistic information from domain objects. All domain concepts mentioned in the content description contribute fragments of grammatical descriptions. The rules glue these fragments together to form the gd.

We end this subsection with two rules that handles reference-scope descriptions. The first rule says that in order to relate a cd of this type and a corresponding gd, besides the condition that they shall unify with the argument matrices, the reference information must be possible to express as part of a grammatical subject and the scope information must be expressed as part of a grammatical predicate. In addition the rule adds more features to the gd, in this case tense information.

$$\text{inform-sw}\left(\begin{bmatrix}\text{type} & \text{rs}\\ \text{ref} & {}^{1}[]\\ \text{scope} & {}^{3}[]\end{bmatrix}, \begin{bmatrix}\text{subject} & {}^{2}[]\\ \text{predicate} & {}^{4}[\text{vform pres}]\end{bmatrix}\right) \leftarrow$$

$$\text{refer-sw}\,({}^{1}[], {}^{2}[]), \text{describe-sw}\,({}^{3}[], {}^{4}[])$$

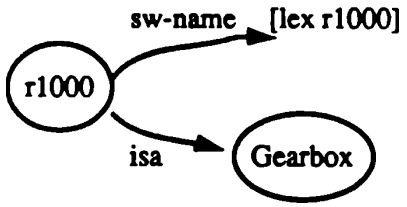
$$\text{describe-sw}\left(\begin{bmatrix}\text{thing} & {}^{1}[]\\ \text{aspect} & {}^{2}[]\\ \text{value} & {}^{3}[]\end{bmatrix}, {}^{4}[]\right) \leftarrow$$

$$\text{iget}({}^{1}[], {}^{2}[], {}^{3}[]), \text{iget}\left({}^{2}[], \text{sw-epred}, \begin{bmatrix}\text{arg} & {}^{5}[]\\ \text{body} & {}^{4}[]\end{bmatrix}\right), \text{iget}({}^{3}[], \text{sw-name}, {}^{5}[])$$

The predicate is obtained by applying a rule for the functor describe-sw. The right-hand side of this rule consists of three calls to the KB, where the first answers the topical question and the other two retrieves linguistic information as explained in the next section.

4.4 The knowledge base

The knowledge necessary for the first stage of the generation process is kept in a single knowledge base. The primary aim of the knowledge base is to store information about the domain objects. It is organized as an inheritance network, allowing instances to inherit properties from their concepts. The properties can be divided in two classes, domain related and language related. In the small example below the r1000, called the carrier, is an instance of the concept Gearbox, and it has the property sw-name, also called the indicator, defined to be the value [lex r1000].



Associated with each indicator, there is a method which knows how the value is to be retrieved. The value can either be cached in the carrier node directly, inherited from a concept through isa-links, or otherwise computed. It is the fact that the knowledge is only accessible through the domain objects that makes generation much easier than parsing. In order to make the system bidirectional we would also have to make domain objects accessible from their property values, especially the linguistic properties.

Linguistic information can be associated with domain concepts in different ways. Nouns are stored under an indicator name which is then differentiated for the two languages as in the example above. Information relevant for predicates is stored under the indicator epred which is differentiated in the same way. In the case of the concept Operation, the Swedish epred-value is a structure, which says that the predicate should contain the verb *manövrera* in the passive voice, and a manner adverbial expressing the manner of operation:

$$\begin{bmatrix} \text{arg} & {}^{1}[\,] \\ \text{body} & \begin{bmatrix} \text{passive} & \text{yes} \\ \text{compl1} & \begin{bmatrix} \text{manner} & {}^{1}\begin{bmatrix} \text{aform} & \text{t} \end{bmatrix} \end{bmatrix} \\ \text{lex} & \text{manövrera} \end{bmatrix} \end{bmatrix}$$

In section 3 we considered the following sentence pair, and introduced a concept Purpose for its description:

S: Spärrventilen har till uppgift att förhindra växling av rangeväxeln när ...
E: The purpose of the inhibitor valve is to prevent inadvertant shifting of the range gear when ...

The Swedish sentence can be handled by the previous rule for reference-scope propositions.[4] The structure for the Swedish predicate information associated with Purpose would be as follows:

$$\begin{bmatrix} \text{arg} & {}^{1}[\,] \\ \text{body} & \begin{bmatrix} \text{compl1} & \begin{bmatrix} \text{pobj} & \begin{bmatrix} \text{prep} & \text{till} \\ \text{obj} & \begin{bmatrix} \text{lex} & \text{uppgift} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{lex} & \text{ha} \\ \text{compl2} & \begin{bmatrix} \text{infcomp} & {}^{1}[\,] \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

For the English sentence we actually have to use a different clause-level rule, introducing relations of the asserted proposition both with the subject and the predicate. The first association will use the English name of the concept, while the second will use the predicate information. In conjunction these rules will produce a partial grammatical description corresponding to the pattern *the purpose of ... is ... .*

$$\text{inform-eng}\left(\begin{bmatrix} \text{type} & \text{rs} \\ \text{ref} & {}^{1}[\,] \\ \text{scope} & {}^{3}[\,] \end{bmatrix}, \begin{bmatrix} \text{subject} & {}^{2}\begin{bmatrix} \text{pmod} & \begin{bmatrix} \text{prep} & \text{of} \\ \text{obj} & {}^{4}[\,] \end{bmatrix} \end{bmatrix} \\ \text{predicate} & {}^{5}\begin{bmatrix} \text{lex} & \text{be} \\ \text{vform} & \text{pres} \end{bmatrix} \end{bmatrix}\right) \leftarrow$$

$$\text{refer-eng}\,({}^{1}[\,], {}^{4}[\,]), \text{describe-eng}\,({}^{3}[\,], {}^{5}[\,]), \text{identify-eng}\,({}^{3}[\,] {}^{2}[\,])$$

5. Status and future work

We believe that the generation procedure as far as sentences are concerned is powerful and flexible enough to handle most of the sentence-level phenomena of our corpus. However, this remains to be proved, as the current grammar only covers a fraction of the corpus. Moreover, we will extend the rule base to handle paragraph-level phenomena such as coherence relations and anaphoric dependencies as well.

If anything, the generation procedure is at present rather too powerful and unconstrained, so we want to investigate further what constraints to put on it. As for speed, the bottleneck of the generation process is the surface generator. Ideally we would like to eliminate it completely and work with a single grammar incorporating phrase structure as well as functional grammatical information.

Notes

1. The project, Konceptuell textrepresentation för automatisk generering och översättning, is funded by the Centre for Industrial Information Technology (CENIIT) at the Linköping Institute of Technology.
2. We take the sentence pairs in the corpus as *prima facie* instances of translation equivalents and thus necessary to account for. This may be questioned in a few cases, e.g. where one sentence contains a modifier having no counterpart at all in the other sentence, but all such exceptions need careful motivation.
3. Two paragraphs may be considered equivalent if they answer the same questions in the same order.
4. It needs a different rule for describe-sw, however, as the value of the asserted proposition need not be a simple concept.

References

Ahrenberg, L. (1987): Interrogative structures of Swedish: aspects of the relation between grammar and speech-acts. RUUL 14 (Doct. diss.), Uppsala University, Department of Linguistics.

Carbonell, J.G. and Tomita, M. Knowledge-based machine translation, the CMU approach. In S. Nirenburg (ed.) *Machine Translation.* Cambridge University Press, pp. 68-89.

Carlson, L. (1983): *Dialogue Games*. Dordrecht, Reidel.

Emele, M., Heid, U., Momma, S. and Zajac, R. (1990): Organizing linguistic knowledge for multi-lingual generation. *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki, 20-24 August, 1990, pp. 102-107.

Kaplan, R.M., Netter, K., Wedekind, J., Zaenen, A. (1989): Translation by structural correspondences. *Proceedings of the 4th EACL Conference*, Manchester 10-12 April, 1989, pp. 272-281.

Nilsson, U. and Maluszynski, J. (1990): *Logic, Programming and Prolog*. John Wiley & Sons.

Russell, G., Ballim, A., Estival, D., Warwick-Armstrong, S. (1991): A language for the statement of binary relations over feature structures. *Proceedings of the 5th EACL Conference*, Berlin 9-11 April, 1991, pp. 287-292.

Sadler, L. and Thompson, H. (1991): Structural non-correspondence in translation. *Proceedings of the 5th EACL Conference*, Berlin 9-11 April, 1991, pp. 293-298.

Shieber, S.M. (1988): A uniform architecture for parsing and generation. *Proceedings of the 12th*

International Conference on Computational Linguistics, Budapest, 2-27 August 1988, pp. 614-619.

van Noord, G. (1990): Reversible unification based machine translation. *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki 20-24 August, 1990, pp. 299-304.

Zajac, R. and Emele, M.(1990): Typed unification grammars. *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki, 20-24 August, 1990, Vol. 3 pp. 293-298.

Lars Ahrenberg & Stefan Svenberg
Department of Computer and Information Science
Linköping University
S - 581 83 Linköping
Email: {lah,ssv}@ida.liu.se