Document generated on 07/15/2025 5:18 p.m.

Meta
Journal des traducteurs
Translators' Journal

META

Unification and Machine Translation

Doug Arnold and Louisa Sadler

Volume 37, Number 4, décembre 1992

Études et recherches en traductique / Studies and Researches in Machine Translation

URI: https://id.erudit.org/iderudit/001904ar
DOI: https://doi.org/10.7202/001904ar

See table of contents

Publisher(s)

Les Presses de l'Université de Montréal

ISSN

0026-0452 (print)
1492-1421 (digital)

Explore this journal

Cite this article

Arnold, D. & Sadler, L. (1992). Unification and Machine Translation. *Meta*, *37*(4), 657–680. https://doi.org/10.7202/001904ar

Article abstract

There have been important developments in computationally oriented linguistic theory in the last 10 years, in particular, the development of formalisms based on UNIFICATION. This is already important as regards Machine Translation research, and will become increasingly so as the next generation of practical MT systems evolves. One purpose of this paper is tutorial — to give an overview of these developments, to say why they are positive and to explore their application in the context of MT. The paper also has a non-tutorial aim of conceptual clarification: we think these developments contain a number of options that are interestingly different.

Tous droits réservés © Les Presses de l'Université de Montréal, 1992

This document is protected by copyright law. Use of the services of Érudit (including reproduction) is subject to its terms and conditions, which can be viewed online.

https://apropos.erudit.org/en/users/policy-on-use/

érudit

This article is disseminated and preserved by Érudit.

Érudit is a non-profit inter-university consortium of the Université de Montréal, Université Laval, and the Université du Québec à Montréal. Its mission is to promote and disseminate research.

https://www.erudit.org/en/

UNIFICATION AND MACHINE TRANSLATION

LOUISA SADLER, DOUG ARNOLD
University of Essex, Colchester, UK

Résumé

D'importants développements se sont produits ces dix dernières années en linguistique informatique, plus particulièrement dans le cas des formalismes basés sur l'UNIFICATION. Ce formalisme est déjà important dans la recherche en traduction automatique et il le sera de plus en plus au fur et à mesure que de nouveaux systèmes de TA verront le jour. Un des buts du présent article est pédagogique : il s'agit de présenter les nouveaux développements en matière de linguistique informatique, de faire valoir les aspects positifs et d'expliquer leur applicabilité dans le contexte de la TA.

Abstract

There have been important developments in computationally oriented linguistic theory in the last 10 years. In particular, the development of formalisms based on UNIFICATION. This is already important as regards Machine Translation research, and will become increasingly so as the next generation of practical MT systems evolves. One purpose of this paper is tutorial — to give an overview of these developments, to say why they are positive and to explore their application in the context of MT. The paper also has a non-tutorial aim of conceptual clarification: we think these developments contain a number of options that are interestingly different.

INTRODUCTION

There have been important developments in computationally oriented linguistic theory in the last 10 years. These developments have already had an effect on research MT systems, and will have an increasingly powerful one as the next generation of practical MT systems evolves. One purpose of this paper is tutorial — to give an overview of these developments, to say why they are positive and to explore their application in the context of MT. The non-tutorial aim is conceptual clarification: we think these developments contain a number of options that are interestingly different.

The structure of the paper is as follows. Section 1. summarizes the 'traditional' approaches to MT, noting their advantages and limitations. In section 2. we look at the linguistic background and techniques, introducing a number of key properties of contemporary formalisms. Sections 3. and 4. deal with the application of these formalisms to MT: we first consider the extension of the classical transfer approach to deal with more complex data structures and then go on to discuss a rather different approach to the statement of translational correspondences.

We are only concerned here with the core components of 'linguistically based' MT systems, and shall have nothing to say about issues such as text handling, discourse processing and the integration of real world knowledge, or about user interfaces and interaction. This is not to say we think these are unimportant, in fact the reverse is the case.

1. SETTING THE SCENE: TRANSFER AND INTERLINGUAL SYSTEMS

A traditional and very familiar distinction is made in MT between *interlingual* and *transfer* systems. In an interlingual system a mapping is defined between each natural

language and some abstract meaning representation language (the *interlingua*): whenever sentences stand in a translation relationship to each other, they will share a common representation in this language. In general, this requires the representation to be language independent, since all lexical and structural differences between source and target languages must be neutralized.

This approach has several attractions. Adding a new language to the translation system is relatively easy — all that is required is to specify the string <–> interlingua mapping. In particular, notice that there is no need to add any bilingual information. Thus an interlingual system would appear to be straightforwardly extensible. A second advantage is that in principle one can avoid 'translationese': the interlingual representation has to be so abstract that it does not reflect any structural aspects of the source string — hence there is no danger of the target text being unduly influenced by the source structure.

However, the approach also poses a number of serious challenges. It is not at all clear that the state of knowledge in linguistic semantics can support the definition of an interlingua, even for closely related languages. From a purely practical point of view, setting up the framework of conceptual primitives is a huge undertaking, even for a very limited domain (*e.g.* the components and structure of a line printer). In addition, the idea of language independence means that the interlingua must 'multiply out' the distinctions that are made in any of the languages concerned: *e.g.* since Italian distinguishes two kinds of wall (internal, and external, roughly), so must the interlingua, and similarly for any other distinction in any language. This poses a problem for analysis, since there may be no available information about which sense of a word is intended, especially if a source item is vague, rather than ambiguous. One can also see that there will be problems for synthesis, since the form of the target structure will be radically underdetermined, and one will typically have to choose between a large number of alternative realizations. If the interlingual representation is logical there will typically be a large number of equivalent representations for any source sentence, and all of these may have to be found, and tried in synthesis, because there can be no guarantee that any one of them will succeed in synthesis. There is also a problem of robustness: the difficulty of defining analysis and synthesis components for very abstract representations means that these processes are likely to fail for a significant number of cases.[1]

For these sorts of reason, most existing MT systems use the less ambitious *transfer* approach, which does not presuppose a completely language independent level of representation, or completely disambiguated representations. Instead, a level of representation is adopted which seeks to minimize, but not eliminate, structural and lexical differences between languages. Explicit mapping rules ('transfer' rules) are used to relate the representations of source and target language, typically substituting target lexemes for source lexemes, and performing some structural manipulations. While this approach promises a degree of extensibility (adding a new language to a system should be easier than building complete translation systems for each existing language pair), it is plainly not as extensible as an interlingual approach (for each new language one needs analysis and synthesis components, and transfer components into and out of each existing language). However, its chief attraction is its feasibility. While there are those who doubt whether interlingual systems are even theoretically possible in general, the compromise of a transfer approach is highly practicable.

Nevertheless, there are some serious drawbacks, in some cases similar to those of the interlingual approach, though generally less serious.

(1) There is still a great deal of linguistic semantic work involved in defining a level of representation which minimizes bilingual mappings, and a considerable effort

involved in actually specifying the bilingual mapping rules and in controlling the interaction of the rules, once written.

(2) As with the interlingual approach, there are problems of ambiguity — in general analysis, transfer, and synthesis will produce several outputs for any single input, and one is faced with the problem of choice.

(3) There is the problem (also shared by interlingual approaches) that if information is thought to be required for translation, it must be explicitly represented in the representation that is output by analysis. This poses a problem, because it is widely assumed that such information will not be conceptually homogenous, but will relate to different levels of linguistic organization. For example, one may want to refer both to surface properties of the source structure (*e.g.* what the *subject*, or *tense* is), and semantic properties (what the *agent*, or *time reference* is). This may be for reasons of descriptive convenience or robustness (surface properties can be computed more reliably and easily, and so provide a more reliable basis for transfer than more abstract properties). Creating a coherent design for such a hybrid level of representation is very problematic.

(4) Given that the input to transfer retains some aspects of source structure, the desire to keep transfer rules as simple as possible means there is a strong tendency for this source structure to be imposed on the target structure. This can produce 'translationese'.

(5) There is a problem of redundancy: transfer rules and the rules of the target grammar are both involved in characterizing the target structures. Ideally, one would like to divide work between them in some principled way. It is a serious waste of effort to describe the same facts twice (the same issue arises with respect to transfer and the source grammar, and between the grammars of each language when it is considered as source and target). Similarly, there will typically be a considerable overlap between the transfer component from (say) English to French, and that from French to English.

Dealing with these issues in a satisfactory manner is a daunting task. In recent years a good deal of attention has been focussed on the problem of the acquisition and maintenance of bilingual knowledge for MT and the considerable difficulty of capturing explicitly the precise conditions under which elements stand in a translation relation. A particularly radical approach starts from the observation that much of the difficulty arises from the need to produce *rules* describing the various relations between representations. If one could simply pair up the corresponding parts of a bilingual corpus and use this directly, one would have a translation system for everything in the corpus, without the need for any rule components. Of course, such a system would not be able to handle novel inputs. Other less extreme approaches seek to use such a corpus as a bilingual knowledge bank to guide lexical choice, bypassing the need for explicit statements of complex conditions on bilingual pairings (for the discrimination of readings, for example). With a sufficiently large corpus, one can extract statistical information about likely translations in contexts of various sizes, and on this basis derive probable translations for the input. Alternatively, one could try to find material that shares certain similarities (*e.g.* being in the same thesaural area for certain items), and thus compute a best match for translation (translate 'by analogy'). Several refinements are possible — for example the items in the database of translations derived from the original corpus can be linguistically analyzed (parsed) in some way. Taking this a step further, adding some statistical or analogical techniques would alleviate some of the difficulties of constructing rule based systems.

While statistical and analogical techniques will clearly be important in the next generation of MT systems, we do not believe they amount to a complete solution, even in

combination with more traditional techniques. Moreover, whatever their attraction in terms of the practical goal of building MT systems, they are unrevealing in relation to the 'scientific' goal of MT research, which is to do with casting light on formal characteristics of the relations between translations, and other standard questions of translation theory. This is because addressing these questions presupposes explicit descriptions of the properties of, and relations between, linguistic expressions, which these techniques are designed to avoid.

In this paper we will explore some less radical approaches to the classical problems of MT design, focussing on the use of contemporary linguistic theory in transfer-based MT. Because the knowledge acquisition task is so huge in MT, we believe that MT formalisms should fulfill a number of basic criteria which to a certain extent alleviate the problems noted above.

For example, the need for the representations to be suitably expressive, with a well-defined theoretical and conceptual basis, is partly met if one stays close to established mainstream computational linguistic theories, and does not tailor one's descriptions too closely to the intended application (*e.g.* the grammars should not be tailored to any particular language pair, and ideally the monolingual parts of the system should not even be specifically designed for MT). This also reduces the size of the descriptive problem (*i.e.* the number and complexity of the rules to be written if a system is to operate over a significant domain), since one can exploit descriptions developed for such theories, and for other applications. The descriptive problem is further reduced by stressing a number of formal properties, all of which make systems easier to construct, modify and extend: *modularity* (separating algorithmic from linguistic information, and separating information about different languages — ideally individual rules should be modular, in the sense of describing complete and coherent pieces of linguistic knowledge); *declarativity* (so the behaviour of the system can be described, and understood independent of the procedures used); *reversibility* (the same components and rules should be usable for analysis and synthesis, for either direction of translation); and *monotonicity* (the addition of new statements only adds information, never changing or removing existing information — modification and extension of monotonic systems is much easier than for non-monotonic ones, since the interaction of components is more transparent).

Recent developments in computational linguistics, and computationally oriented linguistics promise these benefits, as well as solutions to other problems, in particular, the problems of *redundancy*, and the need for 'hybrid' representations to permit information at different linguistic levels to influence translation.

By way of introducing these developments, in the next section we will look in more detail at the traditional approach to a number of commonplace descriptive problems, and show how more attractive solutions are now available.

2. LINGUISTIC TECHNIQUES: A CLOSER LOOK

In 'traditional' MT, as in much modern linguistics, the standard data structure is the *tree*, whose nodes are labeled with category information of various kinds, and the standard operations are various kinds of tree-to-tree transduction — operations (*e.g.* rules) which take one tree as input, and give another as output. More recent work in MT, and some contemporary linguistics, has replaced trees with the slightly more general notion of a *feature structure*, and replaced tree-to-tree transductions with a collection of operations based on the idea of *unification*. To give an idea of what this means, and why it seems like a positive development, we will sketch how some simple monolingual and translational phenomena can be handled in traditional 'tree-based' and 'unification-based' approaches.

2.1. TREE-BASED APPROACHES

The basic idea of 'tree-based' formalisms is that there are labeled trees — that is graphs where nodes are labeled, and where each node has a single mother (a labeled bracketing is exactly equivalent) — and there are rules that match an input tree, and which may transform it in various ways to produce an output tree, deriving one tree from another. We will now look at how three relatively commonplace phenomena can be handled: control, unbounded dependencies, and agreement.

The 'control' relation is the relation of referential dependence that exists between the subject of *like* and the empty subject of *swim* in (1a).

(1)
a. Sam_i likes [$_S$ e_i to swim everyday].
b. Sam likes [$_S$ e to swim everyday].

The normal tree-based representation of this dependence involves an empty element (*e*) in the embedded subject position and co-subscripting between *Sam* and *e*, which would be achieved by means of a rule which transforms a structure without the subscripting (1b) into one that has it (1a).

A similar representation is standardly involved in the treatment of unbounded dependencies, as in (2a), with the difference that here the rule is sometimes thought of as involving the movement of *Which boy* from an underlying position in (2b) to a structure like (2a). Where the dependency in control cases is restricted to items in neighbouring clauses, here it can hold over any number of clauses (hence it is 'unbounded').

(2)
a. Which boy_i do you think [$_S$ Kim said [$_S$ Sam saw e_i yesterday]]?
b. Do you think [$_S$ Kim said [$_S$ Sam saw which boy yesterday]]?

Agreement in English is very simple: subjects and verbs, and determiners and nouns agree for number. However, for generality, assume that there is a collection of agreement properties ('AGR') which must be common to subject and verb. Agreement is a special case of the very general phenomenon of the transmission of information from one linguistic structure to another.

(3)
a. The boys admire honesty.
b. *The boys admires honesty.

Looking at an example like (3a), where the agreement properties of the verb in isolation are not apparent (*cf. I admire honesty*), in a tree-based approach the natural thing to do is to copy the agreement properties of the subject onto the verb, perhaps via some other nodes (*e.g.* an abstract INFL node, and the VP); (3b) would be ruled out because once *admire* is marked plural, it cannot be realized as *admires*.

Applying this tree transduction approach computationally, as in MT, involves providing precise definitions of a number of notions, such as what it means for a rule to 'match' a structure, and what range of possible transformations are possible. The details of these definitions varies, but the general idea is that a rule like (4a) might match an SVO structure for *Sam saw Kim*, and derive an SOV structure such as that for *Sam wa Kim wo mitta*, which is the corresponding Japanese. A rule like (4b) would match a tree consisting of just the verb *see*, and derive a tree consisting of its translation *miru*.

(4)
a. [$_S$ 1:NP 2:V 3:NP] –> [$_S$ 1:NP 3:NP 2:V]
b. [$_V$ see] –> [$_V$ miru]

Probably the simplest way of using such rules for translation is to apply them recursively to a source tree: one begins at the root node, and finds a rule whose left-hand-side matches that node: (*e.g.* (4a) would match an S-node with three daughters NP, V, and NP). The right-hand-side of the rule is put on hold until all the subtrees have been translated, after which it is interpreted as an instruction to combine the translations of these sub-trees. One can think of this as a process of 'decomposing' the source tree and recombining the translations. The same process is used to translate the subtrees (hence it is 'recursive'); the process 'bottoms out' with rules like (4b), where there are no further sub-trees to translate. Many variations on this sort of approach (in terms of the content of the representations, what other operations are allowed, and the precise processing strategy) can be found. However, the essential outline is the same, and can be applied with slight variations to all stages of analysis (once a surface parse tree has been produced), transfer and synthesis.

Now, one can begin to see some of the problems with the approach. For example, in the case of agreement, the problem is that there are examples like (5a, b) where the copying seems to have to go not from subject to verb, but the other way, since here it is the agreement properties of the subject *the committee* which cannot be decided in isolation (*the committee* can be singular, if considered as a single collective entity, or plural, if considered in terms of its individual members).

(5)
a. The committee have (all) gone home.
b. The committee has decided.

The point is that in this kind of approach, one is led naturally to think of taking a tree where the agreement information is in one place to a tree where it is copied elsewhere, implying a directionality which does not exist in fact, and causing problems as one tries to cover all the different cases.

Examples like (6a, b), where the English example involves control, represent one of the most familiar problem cases of MT, but here we will focus on an aspect of the problem that is not usually discussed.

(6)
a. Sam_i likes [$_S$ e_i to swim everyday].
b. Sam zwemt graag dagelijks.
Sam swims 'likingly' daily

A very natural way of thinking about this translation into Dutch is that it is just like the 'normal' translation of *Sam zwemt dagelijks*, (giving *Sam swims everyday*), and that *graag* translates as *like to*. This would give (7a) as the translation, where *e* is the empty subject of *likes*.

(7)
a. e likes [$_S$ Sam to swim everyday].
b. e_i likes [$_S$ Sam_i to swim everyday].

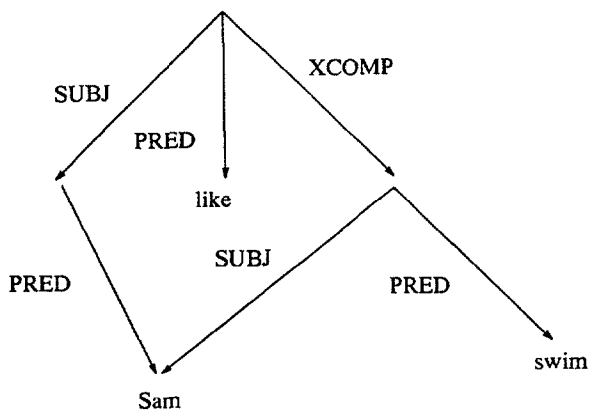
One problem here is that the co-indexation/control rule will probably not have been set up to handle this configuration of *e* and *Sam*. Assuming this can be overcome, we will get (7b). However, the problem remains that the co-indexation does not entail what we would like, namely that *e* and *Sam* should swap places. This can be a real embarrassment for tree based approaches, since it is very difficult to define rules which will perform this kind of manipulation correctly, and the alternative is a serious complication of the translation rules.

One of the problems with the treatment of unbounded dependencies is similar — there is nothing inherent in the notion of co-indexation which means that, for example, certain properties of the displaced item *Which boy* are available in the lower clause — if ever they are required (*e.g.* for agreement), a rule will have to be formulated which accesses them explicitly. (Other difficulties arise in establishing the relation between surface and 'underlying' positions in analysis and synthesis with this sort of tree-transduction formalism).

2.2. UNIFICATION-BASED APPROACHES

In most unification-based approaches, the basic representation is a *feature structure*, FS, (a form of Directed Acyclic Graph, DAG),[2] the basic operation is *unification* (of FSs), and matching involves *subsumption*.[3] An FS is an unordered collection of *attributes* and *values*, some of which may themselves be FSs. One graphic realization makes them look like trees. The critical differences are: (i) that branches are labeled; (ii) that branches are not ordered — this is an immediate advantage in a translation context, since one does not have to write rules that explicitly *change* order, either to produce a canonical 'underlying' order, or for a specific target language; (iii) and *there is no single mother condition*, that is, a single node can have two mothers; to put it another way, a single structure can appear in two places in an FS. Such structures are said to be *re-entrant*. This provides a natural representation of the control construction above:[4]

(8)



However, the more normal (but fully equivalent) representation of such structures is as a matrix of attributes (labels on branches) and values (labels on nodes).

(9)

$$\begin{bmatrix} \text{PRED} & \text{'like'} \\ \text{SUBJ} & [\text{i}]\ \begin{bmatrix} \text{PRED} & \text{'Sam'} \end{bmatrix} \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'swim'} \\ \text{SUBJ} & [\text{i}] \end{bmatrix} \end{bmatrix}$$

Notice that the re-entrance is represented as a subscript in the matrix, just as in the tree representation. However, there is a crucial difference of interpretation: in the tree, co-subscripting represents referential dependence; in the FS matrix, it indicates that the SUBJ of *like* and the SUBJ of the XCOMP of *like* are *the same structure* — any operation on one is automatically, and necessarily an operation on the other; the single structure [PRED Sam] is present in both places. Notice that if we produce such a structure as the translation of (1a), there will be no need for rules that move structure around.[5]

Unification is an operation that takes two information bearing objects (*i.e.* descriptions), and combines the information in both, providing it is consistent. For example, consider the FSs in (10).

(10)

a. $\begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUMBER SING} \end{bmatrix} \end{bmatrix}$

b. $\begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{PERSON THIRD} \\ \text{GENDER MASC} \end{bmatrix} \end{bmatrix}$

c. $\begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{NUMBER SING} \\ \text{PERSON THIRD} \\ \text{GENDER MASC} \end{bmatrix} \end{bmatrix}$

(10a) unifies with (10b) to give (10c); unification of (10c) and (10d) fails because they have contradictory values for the NUMBER attribute. In terms of the objects described, the FS obtained by unifying FSs F_1 and F_2, is a description of those objects that satisfy both the description F_1 and the description F_2 (as the descriptions get larger, so the set of objects described gets smaller, of course). *Subsumption* is the unification-related notion of matching: F_1 subsumes F_2 just in case F_2 contains at least as much information as F_1; that is, if F_1 describes every object that F_2 describes (and perhaps some more as well). (10a) and (10b) both subsume (10c), but not vice versa. Notice that unifying two FSs produces a FS that both subsume; that if F_1 subsumes F_2, then unifying F_1 and F_2 will produce F_2; and that every object subsumes itself.

FSs themselves are described by sets of constraints. A standard approach uses an abstraction of a context free phrase structure rule to describe the requisite string concatenation and a set of identities holding over (parts of) the associated FSs. Solving these constraints will involve unifying structures (and substructures) that are constrained to be identical.[6]

(11)

a. $X_0 \longrightarrow X_1, X_2$
X_0 (CAT) = S
X_1 (CAT) = NP
X_2 (CAT) = VP
X_0 (GS) = X_2(GS)
X_0 (HEAD) = X_2 (HEAD)
X_0 (GS SUBJ) = X_1
X_0 (HEAD AGR) = X_1 (AGR)

b. $X_0 \longrightarrow X_1, X_2$
X_0 (CAT) = VP
X_1 (CAT) = V
X_2 (CAT) = NP
X_0 (GS) = X_2 (GS)
X_0 (HEAD) = X_1 (HEAD)
X_0 (GS SUBJ) = X_1
X_0 (HEAD AGR) = X_1 (AGR)
X_1 (SUBCAT) = NP
X_0 (GS) = X_1 (GS)
X_0 (GS OBJ) = X_2

The constraints in (11a) state certain requirements on three FSs — the CAT values are required to be S, NP and VP respectively, and re-entrances are required, as shown in (12).

(12)

$$X_0 = \begin{bmatrix} \text{CAT S} \\ \text{GS [1]} \begin{bmatrix} \text{SUBJ [3]} \end{bmatrix} \\ \text{HEAD [2]} \begin{bmatrix} \text{AGR [4]} \end{bmatrix} \end{bmatrix}$$

$$X_1 = \begin{bmatrix} \text{CAT NP} \\ \text{AGR [4]} \end{bmatrix}$$

$$X_2 = \begin{bmatrix} \text{CAT VP} \\ \text{GS [1]} \\ \text{HEAD [2]} \end{bmatrix}$$

The constraints require the NP's AGR FS be identical to the S HEAD AGR and the S HEAD FS be identical to the VP HEAD feature, enforcing subject-verb agreement. Notice that the 'directionality' issue that arose with the tree-based account does not arise here — it does not matter where the information about agreement features comes from, unification and the identities declared between structures ensure that it is combined correctly and passed to all the relevant structures.

Given the following rules for V and NP (13b), we will get the FS in (13c) for the sentence (13a).[7]

(13)

a. Sam saw Kim

b. X_0 —> [sees]
X_0 (CAT) = v
X_0 (GS PRED) = 'see'
X_0 (SUBCAT) = NP
X_0 (HEAD TENSE) = PRES
X_0 (HEAD AGR PER) = THIRD
X_0 (HEAD AGR NUM) = SING

c. X_0 (CAT) = NP
X_0 (GS PRED) = 'Sam'
X_0 (AGR PER) = THIRD
X_0 (AGR NUM) = SING

d. X_0 —> [Kim]
X_0 (CAT) = NP
X_0 (GS PRED) = 'Kim'
X_0 (AGR PER) = THIRD
X_0 (AGR NUM) = SING

(13e)

$$\begin{bmatrix} \text{CAT S} \\ \text{GS} \begin{bmatrix} \text{PRED 'see'} \\ \text{SUBJ} \begin{bmatrix} \text{PRED 'Sam'} \\ \text{AGR [i]} \begin{bmatrix} \text{NUM SING} \\ \text{PER THIRD} \end{bmatrix} \end{bmatrix} \\ \text{OBJ} \begin{bmatrix} \text{PRED 'Kim'} \\ \text{AGR} \begin{bmatrix} \text{NUM SING} \\ \text{PER THIRD} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{HEAD} \begin{bmatrix} \text{TENSE PRES} \\ \text{AGR [i]} \end{bmatrix} \end{bmatrix}$$

Before looking at some refinements, it is worth noting some other useful properties. The *order* in which these constraints are evaluated is irrelevant: unifying F_1 and F_2 is the same as unifying F_2 and F_1, and unifying the result of this with F_3, is the same as unifying F_1 with the unification of F_2 and F_3 (unification is commutative and associative). Thus, unification based formalisms are (potentially) *declarative*. Unification is also *monotonic* since it can only add information. Although formalisms making heavy use of unification are always computationally expensive, there is a great deal of active research in this area and good algorithms are available.

With this background, one can also see how a number of other phenomena can be dealt with with this apparatus. For example, in a unification based approach, there will be a re-entrance between the position associated with *Which boy*, and the position marked with *e* in (14).

(14)
Which boy_i do you think $[_S$ Kim said $[_S$ Sam saw e_i yesterday]]?

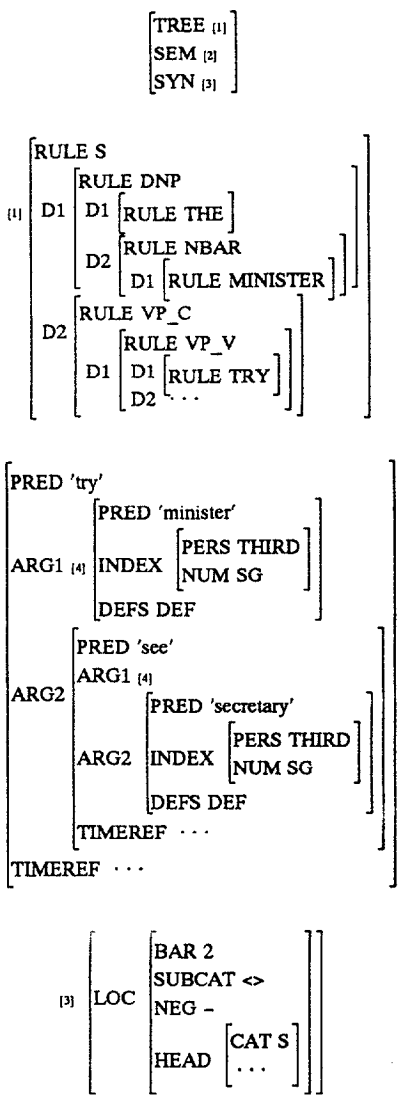
This will ensure that information about the displaced material that is needed in the lower sentence is available there (*e.g.* its semantics). Moreover, we can establish the link between the positions indirectly. Suppose we associate an attribute (*e.g.* GAP) with each sentence and VP node, whose value is re-entrant for each (note this already follows from the rules already given, which make these FSs identical in *all* respects), and ensure that the value of this attribute in the matrix sentence is the FS associated with *Which boy*. It only remains to allow the value of this feature to fill the OBJ role of *see* to get the desired effect without recourse to movement rules of any kind.

The examples considered so far have been concerned with syntactic information in the main, but there is no reason why relations between different levels of linguistic representation, which are traditionally related by tree transductions, should not be treated in the same sort of way. Using a unification based grammatical formalism it is possible to avoid stating derivational relations between representations. Instead, one can exploit the information sharing capacity of re-entrances in complex data structures to classify linguistic objects on a number of levels simultaneously.

For example, a unification-based grammar might describe a linguistic object along three dimensions — in terms of: (i) the phrase structure rules used (that is, encoding the parse tree in a feature), (ii) the intrinsic syntactic properties; and (iii) its semantics.[8] Such a (complex) structure is indicated schematically in the following FSs, for the sentence (15).

(15)
a. The minister tries to see the secretary.

(16)

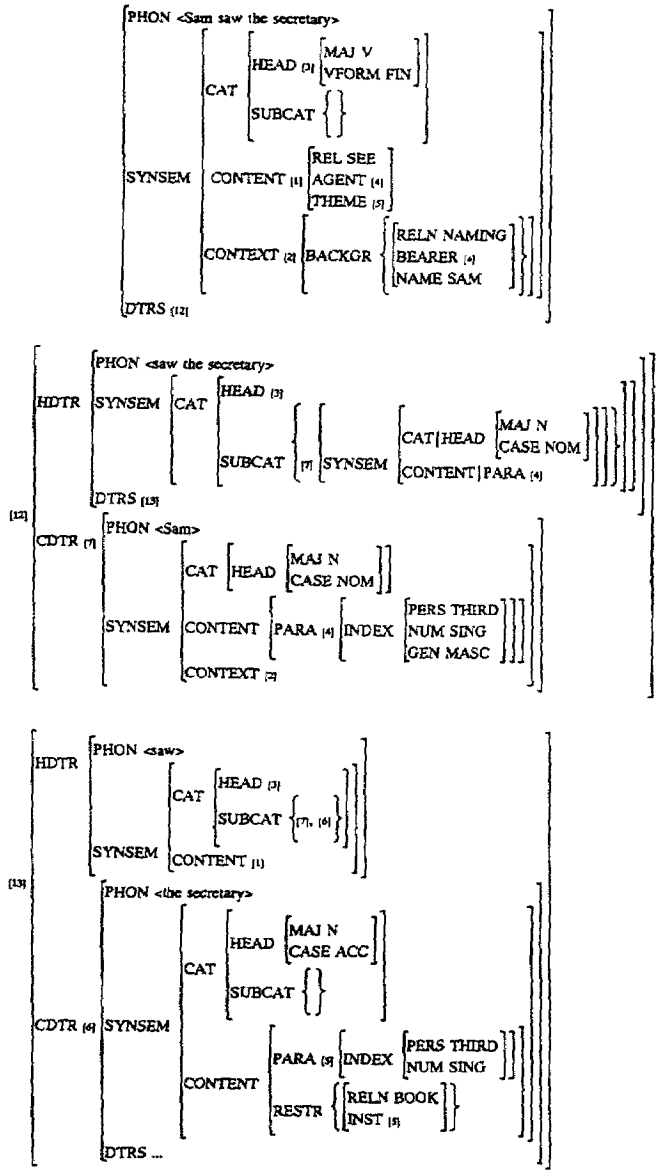


Linguistic descriptions which simultaneously express information pertaining to a number of different levels of linguistic description in one complex FSs, such as (16), are generally known as *sign-based*. HPSG (Pollard and Sag 1987) is the best known sign-based formalism, and there are also sign-based variants of categorial grammar. Although there is generally a rule skeleton, such formalisms are lexical in the sense that nearly all of the richly structured information originates in the lexicon and lexical FSs define the domain for the application of constraints.

In order to facilitate our later discussion of translation with these formalisms, we give an example also (again schematically) of a sign for the string *Sam saw the secretary* in HPSG style:[9]

(17a) Sam saw the secretary.

(17b)



Notice such structures encode information from a number of different levels of representation, but without losing linguistic and conceptual coherence — without becoming 'hybrid'.

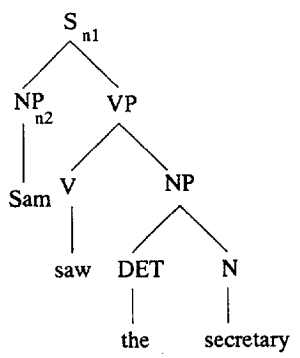
However, it is not necessary to encode information about different levels within a single structure to exploit the information sharing possibilities of unification. One of the earliest formalisms to exploit unification for information-passing was Lexical Functional Grammar, LFG (Bresnan and Kaplan 1982). LFG syntax involves two distinct levels of representation: (i) phrase, or constituent structure trees, called *c-structures*, and (ii) a complex FS, encoding surface grammatical functions (f-structure). These are related by means of a function ϕ from c-structure nodes to f-structures. As in our example (11) above, constraints over FSs are expressed as annotations to phrase structure rules. Much of the information originates in the lexicon. The symbol $\uparrow$ in (18) can be read as "the f-structure associated with the mother node" and $\downarrow$ as "the f-structure associated with this node" and are simply convenient notational shorthands for, respectively, ϕ of the mother of the current node, and ϕ of this node.

(18)

$$\begin{array}{lcll} S & \longrightarrow & NP & VP \\ & & (\uparrow \text{SUBJ}) = \downarrow & \uparrow = \downarrow \end{array}$$

Viewed in this light, it is easy to see that any number of functions between different description languages can be defined with the effect of simultaneously classifying a linguistic object along a number of dimensions. This is shown in (19), in which σ is a function which projects semantic structures from f-structures (linguistically significant mappings between levels are known as *projections*). For example, f_1 is $\phi(n1)$, s_1 is $\sigma(f_1)$, and hence also $\sigma(\phi(n_1))$.

(19)



$$f_1 \begin{bmatrix} \text{SUBJ } f_2 \begin{bmatrix} \text{PRED 'Sam'} \\ \text{NUM SG} \end{bmatrix} \\ \text{OBJ } \begin{bmatrix} \text{PRED 'secretary'} \\ \text{NUM SG} \\ \text{SPEC } \begin{bmatrix} \text{DEF +} \\ \text{PRED 'the'} \end{bmatrix} \end{bmatrix} \\ \text{PRED 'see<SUBJ,OBJ>'} \\ \text{TENSE PAST} \end{bmatrix}$$

(20)
see V
(↑ PRED) = 'see'
(σ ↑ REL) = SEE
(σ ↑ ARG1) = σ(↑ SUBJ)
(σ ↑ ARG2) = σ(↑ OBJ)

One can already see some reasons why an approach based on FSs and unification is attractive for MT. Four reasons are immediately apparent: (i) FSs are unordered, so there is no need to write rules to 'canonicalize' or normalize the order of structures, as there is with tree-based representations; (ii) unification provides a simple mechanism for combining information from a variety of sources, as in subject-verb agreement; (iii) re-entrance provides a simple implementation of the intuition behind co-subscripting, without the linguist needing to write rules to enforce it; (iv) because linguistic representations can be multidimensional, it is possible in principle to maintain a clear separation between different linguistic levels but still refer directly to different sorts of linguistic information in stating translational correspondences.

In the following section, we will explore some of the ways these ideas can be exploited for describing translation relations.

3. UNIFICATION BASED FORMALISMS AND TRANSFER

In section 2.2. above, we suggested that there were a number of reasons for believing that it would be fruitful to use unification-based formalisms for transfer-based MT. In this section we will discuss how such formalisms can best be exploited for describing translation relations. To begin with, there are a number of approaches which essentially augment a fairly standard unification-based formalism for monolingual analysis and generation with a component for relating attribute-value structures, which we will discuss in outline.

Perhaps the simplest way in which ideas about unification can be exploited in translation is to take a standard tree based formalism, but use unification to pass information between the nodes of the trees, and use unification and subsumption for matching of translation rules. This is essentially the way the central parts of Eurotra formalisms such as CAT-2 (Sharp 1988, and Sharp this volume), ETF (Bech 1991), and MiMo (Arnold and Sadler 1991) operate. Of course, such formalisms do not fully exploit the expressive possibilities of unification, since the basic data structures are trees and not re-entrant graphs.

Extending the traditional transfer method to deal with FSs is in many respects straightforward. The monolingual (analysis) component of a system might produce (perhaps by traditional mapping between levels of representation of increasing abstractness) a level of semantic representation to serve as input to transfer. This source FS is mapped to a target FS by means of a bilingual grammar of transfer rules whose left-hand-sides are matched against the source structure and whose right-hand-sides indicate the content of the corresponding target FS. For concreteness, we can assume the very simple representation given in (21b) is input to transfer.

(21)
a. Sam saw Kim.

(21b)

$$\begin{bmatrix} \text{PRED 'see'} \\ \text{ARG1} \ \begin{bmatrix} \text{PRED 'Sam'} \end{bmatrix} \\ \text{ARG2} \ \begin{bmatrix} \text{PRED 'Kim'} \end{bmatrix} \end{bmatrix}$$

A transfer rule for this structure might then look like (22), which states a bidirectional correspondence between a FS containing the PRED *like* and a FS containing the PRED *aimer*, and furthermore states two conditions, that the ARG1 of the English structure and the ARG1 of the French structure must be related by further transfer rules, and likewise for the ARG2s.

(22)

EN:	PRED = like	FR: PRED = aimer
	ARG1 = E1	ARG1 = F1
	ARG2 = E2	ARG2 = F2

CORRES: E1 <–> F1 and E2 <–> F2

For a rule like this to apply to a FS, it must match that structure (that is, it must not add any information not already in the structure and it must contain all the information in the structure it matches). Such rules are recursively applied to successively smaller collections of source language attribute-values pairs (in a top-down approach — a bottom-up approach is also possible). Like classical tree-based transfer systems, this is 'structural' or 'derivational' in involving the *decomposition* of source structures on the left-hand-side and the actual construction of target structures on the right-hand-side, and differs only in that it is applied to FSs, not trees. This is essentially the approach employed in ELU (Estival *et al.* 1990) and MiMo2 (van Noord *et al.* 1990). We will return below to the question of 'structural' approaches to transfer involving recursive rule application.

We have stressed that one of the advantages of unification-based formalisms is that they permit one to simultaneously classify an object along different dimensions. The monolingual component of an MT system therefore may be sign-based, perhaps classifying the string in terms of its orthographic, surface syntactic, deep syntactic, semantic and discourse-oriented properties, each associated with a different feature. At first glance, such formalisms would seem to allow for the direct use of information associated with these different features in transfer without falling into the problems associated with the 'hybrid' (potentially linguistically incoherent) but abstract level of representation of standard approaches to transfer. We will now consider a number of ways in which such sign-based formalisms can be used for transfer.

Before proceeding, it is necessary to distinguish between sign-based transfer and the use of a sign-based monolingual component. Adopting a sign-based monolingual grammar does not necessarily commit one to sign-based transfer. A system like MiMo2, for example, uses a sign-based monolingual grammar ((16) is a simplified version of the output of this system, in fact), but only uses the information associated with the distinguished SEM feature in transfer. Such a system is also simply a classical transfer system transferring between FSs rather than tree structures.

Notice also that in the approaches described so far the transfer algorithm does not differ in any essential way from that associated with the standard tree-based transfer model. The precise mechanisms by means of which rules are matched against source representations and target representations built up in transfer differ but the general

characteristics are the same. What they have in common is that they are structure-based or representation-based.

Given this approach, the only straightforward way in which the transfer algorithm can apply rules to *signs* is by recursing through the structure of a feature or attribute which encodes the derivational history of the sign itself.[10]

Such a feature is not present in the MiMo2 style representation of (16), but is encoded in the DTRS ('daughters') attribute in the HPSG formalism (see (17)). In HPSG, rules apply in analysis to combine signs, and the derivation is itself encoded in the structure of DTRS. A moment's inspection of the representation should make it clear that the choice of any other attribute as the structure to which to apply transfer rules would not make the multidimensional classification of linguistic objects available to transfer.

Of course, writing transfer rules on the basis of what is in effect the derivational history or parse tree of the final sign is counterintuitive at best. For example, this means that the set of transfer rules will be unnecessarily complicated (the whole point of normalization and use of abstract representations in standard transfer systems is precisely to minimize the differences between languages and therefore the complexity of transfer).

In fact, it is the use of a recursive algorithm for transfer which effectively limits the possibility, in principle opened up by the use of sign-based representations, for the expression of correspondences along different dimensions. We will discuss two further problems in this section and present some alternatives in the following section.

As a consequence of the structural approach, structures to which transfer rules apply must be exhaustively decomposed in order to ensure completeness of translation (in FSs this is usually ensured by carrying out a mutual subsumption check between the FS induced by rule application and the source (input) FS). This means that when material has no translation, it must be explicitly translated as 'nil' or some such (*e.g.* pleonastic elements such as *it* in *It is likely that he will come*).

A key advantage of FSs is that they permit re-entrance or sharing. In structure-based transfer it is fairly straightforward to deal with 'local' re-entrances (such as that occurring in control constructions), which fall within the scope of one transfer rule. On one approach (taken in ELU), re-entrances which are within the structure described by one transfer rule can be translated by binding the re-entrant paths within the (input) structure to the same variable and stating a correspondence between the relevant source side and the target side variables. In this way the re-entrance is translated as one structure. Re-entrancies can also be created and destroyed in transfer in this way. An alternative (taken in MiMo2) is that the re-entrant paths are separately translated, but the re-entrance can be explicitly mentioned on source and target sides, requiring token-identity (*i.e.* re-entrance) between the results of the separate translations.

However neither approach can be simply generalized to 'long-distance' re-entrances, which are typically used to encode long-distance dependencies, and for such phenomena, these approaches can provide no general treatment.

Of course, there are a number of ways in which one might try to remedy this inadequacy. For example, one could unfold the re-entrances as type identities (*i.e.* reinterpret the FS as a tree), or 'thread' shared values through the structure, in such a way that they become local (this is in fact a standard technique for reducing unbounded dependencies to local ones). However, none are satisfactory. The former loses information, so that source FS and target FS are no longer equivalent, and causes problems in generation, where some method must be found for ensuring that lexical content is not duplicated and appears in the right place. Threading techniques are unattractive because of the (often extreme) complication they introduce in grammars and representations.

4. CONSTRAINT-BASED TRANSLATION

In this section we will sketch out in outline two approaches to transfer which are not structural or representation-based, in the sense of involving recursion through all or part of the source structure. These approaches directly exploit the interpretation of equalities as separate constraints and permit one to mix information pertaining to different levels.

One very simple way of avoiding having to recurse through a source structure is possible in a sign-based approach because of the way unification causes information to be shared. Normally, one thinks of rules such as those in section 2. as passing information from smaller constituents to larger ones (from daughters to mothers). However, in fact, what happens when structures are unified is that information is passed in both directions — constraints that identify the semantics of a sentence with those of its VP, and identify the semantics of VP with those of the head V not only pass the semantics of the V up to S, *they also pass the semantics of the S down to the V* (this is illustrated in the HPSG style sign in (17)). In parsing with a sign-based formalism, analysis begins with a string of underspecified signs from the lexicon (retrieved after morphological analysis). Parsing a source string produces successively larger structures, unifying information in the component signs in various ways, and simultaneously further specifying the content of the lexical signs. Now one way of avoiding recursion through a source structure, often referred to as the 'shake-and-bake' approach (Whitelock 1991) is to take just the string of lexical signs (which have become instantiated by the parsing process) as the input to transfer, and map them to their target equivalents, preserving certain 'transfer' properties (*e.g.* their semantics). Now, the idea is that there are very few ways that these signs can be combined by the target grammar to produce a single sign — and the way the target grammar operates will mean that this sign will necessarily have essentially the same transfer properties as the source sign. Thus, all one needs to do is to process the 'bag'[11] of target lexical signs with the target grammar, and one will produce a target sign which is equivalent to the source sign. The obvious way to do this is just to parse the target signs.

For example, parsing the string of English lexical signs *Sam*, *sees*, *the*, and *secretary* will produce a sign for the whole sentence, and instantiate the semantics on each lexical sign. These signs can then be looked up in a bilingual lexicon, giving the bag of corresponding French signs {*Sam*, *voir*, *le*, *secrétaire*}, with semantics unified with that of the corresponding source items (since *secrétaire* can be masculine or feminine in French, one would also have to consider the bag containing *la*, but we will ignore this here). Normally in parsing, one has an ordered list of signs, whose syntactic and semantic relations are underdetermined. In this case, one has an unordered bag of signs, whose semantics are determined. However, normal parsing techniques are still applicable — crudely, one simply tries all possible orders (one 'shakes' the 'bag' of target lexical signs to obtain alternative orders, and tries to 'bake' to produce a single target sign). This process will produce *Sam voit le secrétaire*. The identity between the semantics of *see* and *voir* that was established in transfer means that *Le secrétaire voit Sam* is not produced (which would require a different semantics). **voit Sam le secrétaire* is not produced because the target grammar rules do not permit this, and agreement between subject and verb is achieved by the target grammar rules in the same way.

Since this approach does not recurse through the source structure, there is little danger of 'translationese' — the source structure is simply ignored for translation. There is the possibility of a neat division of labour between 'transfer' and synthesis, avoiding redundancy — transfer stipulates the semantics, and the lexical signs to be used, but all other decisions are left to the target grammar. Notice also that this approach allows information from different linguistic levels to influence translation — in principle, any properties of the source sign at all can be used in choosing the target lexical signs. The approach

should also be reversible in principle. As we have described it, the requirement that source and target items have the *same* semantics makes it appear 'interlingual', and subject to many of the objections raised in section 1. However, this is not essential — all that is necessary is that the semantics instantiated on the target items is sufficient to constrain the synthesis process (target language parsing) to producing expressions which are equivalent to the source expressions. Identifying source and target semantics is one approach, but it would also be possible to manage with parts of the semantics, and parts of the syntax, for example.

There are a number of open issues with this approach. In particular, a critical feature is that all information about the source text necessary for generating the correct target expressions has to be present in a target lexical sign. This poses a problem with properties that are not realized lexically, but are expressed by word order, for example (the distinction between yes-no questions, and declaratives is expressed this way in English). Such properties must either be encoded in the 'transfer properties', or abstract lexical items must be introduced to carry the information. The latter option is unnatural, and unattractive (*e.g.* a description written in this way will not be immediately useful in other applications), and it is not clear that the former option is always possible. A similar problem arises with respect to items in the target language that correspond to no lexical source — for example, the auxiliary verb in the French (23b) corresponding to the English (23a).

(23)
a. Sam saw the secretary.
b. Sam *a* vu le secrétaire.
   Sam has seen the secretary

One possibility here is to introduce an abstract lexeme ('*past*', say) into English, to provide a source for this auxiliary. We have already looked at the disadvantages of this. An alternative is to have the past tense of *see* introduce two items into the target bag — *avoir*, and *voir*. However, this is massively redundant, since one will need to do this for every past tense verb, and it undermines the neat division of labour between transfer and synthesis that we noted earlier, since here the transfer component appears to be supplying information (the information that certain kinds of past tense require an auxiliary) which is predictable from the target grammar. A further alternative is to allow a class of words which can appear in any target structure (they correspond to the ε or zero source string) — *avoir* would be one such, so in the case of (23a) French synthesis would begin with both {*Sam*, *avoir*, *voir*, *le*, *secrétaire*}, and {*Sam*, *voir*, *le*, *secrétaire*} — only the former would result in a complete target sign, given the semantics associated with the main verb, of course. The problem here is that one may be faced with a combinatorial explosion of alternative target bags to be considered (if there are only 10 such items in a language, then one will have 10! alternative target bags to consider for *every* source sentence, no matter how simple, or unambiguous. This is far too large a number for reasonable processing).

The essence of constraint-based translation is the very simple idea that specifying transfer should simply be the statement of (local) equalities which are interpreted as constraints over the target structure(s). Our second example involves the multi-dimensional but not sign-based theory of LFG. In our brief discussion of this theory in section 2. we noted that levels of linguistic representation are related by means of *projections* or mapping functions. Since projections are functions, they may be composed, opening up wide descriptive possibilities for relating levels. The theory also allows for the use of inverse functions (*e.g.* ϕ^{-1} from f-structure to the associated c-structure nodes).

Kaplan *et al.* (1989) show how the LFG constraint language can be used to state bilingual correspondences. In the remainder of this section, we will outline their proposal. Kaplan *et al.* define two translation functions τ (between f-structures) and τ' (between semantic structures). (Semantic structures themselves are projected from f-structures by means of the mapping function σ). By means of these functions, one can 'co-describe' elements of source and target f-structures and s-structures respectively. Achieving translation can be thought of in terms of specifying and resolving a set of constraints on target structures, constraints which are expressed by means of the τ and τ' functions.

Of course the availability of function composition opens up some rich expressive possibilities for stating bilingual correspondences: τ and ϕ can be composed, as can τ' and σ. For ease of exposition, we will initially limit attention to τ, the projection from f-structure to f-structure. The basic idea of this approach to translation is as follows. A bilingual constraint is exactly like a monolingual constraint, except that it makes reference to structures in both source and target language. For example:

(24)
$(\tau(\uparrow SUBJ)) = ((\tau\uparrow)SUBJ)$

which composes τ and ϕ, states a (target side) equality between the τ of the source SUBJ f-structure and the value of the SUBJ attribute of the τ of the source f-structure containing the subject. Each side of this equation picks out a piece of target structure, hence the equation simply states two paths to the same object. We can view (24) as saying that the translation of the value of the SUBJ slot in a source f-structure fills the SUBJ slot in the f-structure which is the translation of the source f-structure which immediately contains that SUBJ slot. We can also directly assign values in the target structure, for example, (25) says that the value of the PRED attribute in the target f-structure is *voir*:

(25)
$((\tau\uparrow)PRED\ FN) =$ 'voir'

Constraints such as these are added to the lexicon and c-structure rules alongside the monolingual constraints. In parsing the source language string one gathers a set of constraints describing the source language f-structure and another set of constraints describing the target language f-structure. The solution of this latter set is a (probably incomplete) target f-structure which must then be completed and validated by the target grammar.

For concreteness, we give the set of equations for (26):

(26)
Sam saw Kim

see, V
$(\uparrow PRED) =$ 'see'
$((\tau\uparrow)PRED\ FN) =$ 'voir'
$(\tau(\uparrow SUBJ)) = ((\tau\uparrow)\ SUBJ)$
$(\tau(\uparrow OBJ)) = ((\tau\uparrow)\ OBJ)$

kim, N
$(\uparrow PRED) =$ 'Kim'
$((\tau\uparrow)PRED\ FN) =$ 'Kim'

Sam, N
$(\uparrow PRED) =$ 'Sam'
$((\tau\uparrow)PRED\ FN) =$ 'Sam'

These constraints co-describe the structures in (27):

(27)

$$e_1\begin{bmatrix}\text{SUBJ } e_2\begin{bmatrix}\text{PRED 'Sam'}\\ \text{NUM SG}\end{bmatrix}\\ \text{OBJ } e_3\begin{bmatrix}\text{PRED 'Kim'}\\ \text{NUM SG}\end{bmatrix}\\ \text{PRED 'see<SUBJ,OBJ>'}\\ \text{TENSE PAST}\end{bmatrix}$$

$$f_1\begin{bmatrix}\text{SUBJ } f_2\begin{bmatrix}\text{PRED 'Sam'}\end{bmatrix}\\ \text{OBJ } f_3\begin{bmatrix}\text{PRED 'Kim'}\end{bmatrix}\\ \text{PRED 'voir<SUBJ,OBJ>'}\end{bmatrix}$$

Notice that there is no separate recursive application of a set of transfer rules to a source f-structure — the constraints stated simply co-describe the source and target structures. Although one could first build the source FS and then the (partially described) target FS, there is also no necessary commitment to any one order of evaluation for the two sets of constraints.

The fact that there is no recursive decomposition of a source FS makes it possible to deal with the translation of long-distance re-entrances without difficulty. In the case of a Wh-question or relative clause, the (source side) Wh-element is introduced by a c-structure rule which also associates it with an attribute (either FOCUS or TOPIC) in the f-structure. The value of this attribute is re-entrant with the value of some other attribute (*e.g.* with the value of OBJ). A translation correspondence for this attribute will (typically) be given in an equation in the lexical entry for the PRED of that f-structure (exactly as previously described). By annotating the relevant c-structure rule with a bilingual correspondence, we specify a constraint over the translation of the source FOCUS or TOPIC (for example, by means of the annotation $(\tau(\downarrow\text{TOPIC})) = ((\tau\downarrow)\text{TOPIC})$ on the S' which introduces a relative clause). Notice that since τ is a function and the same source language structure is the argument of τ in these two equations, the target f-structure is required to contain a re-entrance between the two attributes specified in that target f-structure. Thus long-distance re-entrance can be simply treated.[12]

Again because there is no recursive application of transfer rules to source FSs, there is no requirement that every part of the FS be either translated or explicitly left untranslated (as there was for the approaches discussed in the previous section). Consider the translation of the French *Il est probable que Kim viendra* into the English *Kim is likely to come* (we ignore the translation of tense here, for simplicity, and also the possibility of other translations). The lexical entry for *probable* will contain equations stating a correspondence for the sentential argument, but no correspondence for the pleonastic SUBJ. The lexical entry for *venir* will state a correspondence for its SUBJ. The French SUBJ *il* is thus left untranslated.

(28a)
probable, A
(↑PRED) = 'probable<COMP>SUBJ'
(↑SUBJ FORM) = 'il'
((τ↑(PRED FN) = 'likely'
(τ(↑COMP)) = ((τ↑)XCOMP)

(28b)
venir, V
(↑PRED) = 'venir<SUBJ>
((τ↑)PRED FN) = 'come'
(τ(↑SUBJ)) = ((τ↑)SUBJ)

The English f-structure described by these τ equations (and further equations from the lexical entry for *Kim*) is incomplete, since there is no value assigned for the SUBJ of *likely*. The English lexicon contains a further monolingual equation stating a re-entrance between the SUBJ and the XCOMP subject and this constraint is added as the incomplete f-structure is completed and validated by the target grammar. The approach therefore provides a solution to the problem of redundancy in terms of a nice division of labour between target grammar and bilingual statements.

As we have seen, the descriptive apparatus of projections allows for multiple levels of structure to be related by separate correspondences. Kaplan *et al.* define τ' between semantic structures, where the σ correspondence maps from f-structures to semantic structures. For example, τ'(σ↑ARG1) = (σ↑ARG1) asserts an identity between the values of ARG1 in source and target semantics. This would be appropriate if the values were, for example, semantic indices. An equation τ'σ(↑ARG1) = σ(τ↑TOPIC) states an identity between the translation of the ARG1 in the source semantics and the semantics associated with the TOPIC of the target f-structure. Constraints such as these can also be given as further annotations to the c-structure rules and within the source lexicon, making possible the statement of constraints over multiple levels of both source and target structure, whilst still maintaining the coherence of the levels of representation in question. There are various ways in which one might want to use this facility for 'multilevel' transfer, depending in part on the linguistic content of the various levels of representation related by different bilingual mapping functions. For example, one can imagine discourse-oriented information being used for certain types of disambiguation. Or transfer could be attempted at some level of semantic structure, with f-structure correspondences being used as a fall-back position, for reasons of robustness. Phenomena for which there are (at least the beginnings of) an adequate interlingual treatment (time and aspect, for example) could be factored out and dealt with interlingually, with time and aspect information in the other levels for which bilingual correspondences are stated simply ignored. In these ways, then, this constraint-based approach seems to provide an interesting response to the need for multilevel transfer.

One drawback of the LFG approach, however, is the fact that projections are directional. As a consequence, translation correspondences are directional (from a given source to a given target). In the worst case, this means that a separate projection will have to be defined for each direction, though the extent to which the inverse of the mapping function can be used for this purpose is an open question.

5. CONCLUSION

We began this paper by characterizing the traditional approaches to the linguistic core of MT, pointing out some of the difficulties and challenges that such approaches

face. We hope to have given some idea of how adequate contemporary linguistic formalisms are for the task of MT. In general, they are good candidates because of their intrinsic properties — they are well-understood, declarative, monotonic and computationally oriented. Linguistically, they are expressive enough to permit the sorts of analyses which are likely to be useful in MT, allowing one to express a fully explicit syntax and semantics in a modular way. Because they are mainstream formalisms, they provide a partial solution to the problem of 'effort' because at least the monolingual components of a system will be reusable in other applications. In particular, however, we have tried to show how problems such as redundancy, the need to avoid 'translationese' and the use of multilevel information are addressed by these formalisms. They permit an interesting new view of translation in which the focus of transfer is not on rules and representations but on defining constraints on the relation between source and target structures. Many issues, of course, remain open.

Notes

1. Notice also the presupposition that representations will become more similar as they become more abstract, more distant from the linguistic surface. But this is by no means obviously correct. There can be no guarantee that the conceptualizations of different speakers will necessarily converge; and if they do not, then interlingual representations may make translation harder, rather than easier.
2. 'Directed' because the lines go from one node to another (mother to daughter), and 'acyclic' because the lines do not loop back on themselves (in fact, some unification based formalisms employ cyclic graphs).
3. Though formalisms that use FSs are normally unification based, there is no necessary connection between DAGs and unification. There are linguistic theories, such as Relational Grammar (Perlmutter 1973), which use DAGs, but do not use unification. The variant of unification that deals with *logical terms* rather than FSs is also common, especially in approaches based on Prolog. The most important difference between terms and FSs is that terms contain a fixed number of ordered slots, which are distinguished by position. In FSs, the slots (attributes) are not ordered, their number is not fixed, and they are identified by label, not position. This makes FSs rather easier to deal with in practice.
4. From now on we simplify by omitting the adverb *everyday / dagelijks* from structures.
5. What we will need is a generation algorithm that makes sure that the phonetic content is realized in the right place.
6. X_0, X_1 and X_2 are the names of FSs, so a constraint like X_0 (CAT)=S can be read as saying that the attribute CAT has the value S in the structure X_0. The reason for this notation is that FSs are mathematically functions from attributes to values, so an equivalent reading would be that the function X_0 returns the value S when applied to CAT.
7. This is an appropriate place to point out the difference between the AGR value marked with the subscript [i] in (13c), which is re-entrant, and the similar value that appears in the OBJ, which happens to have the same attributes, with the same values. One speaks of *type identity* in the latter case, and *token identity* in the former case. In other words, cases of token identity occur when different paths through an FS are pointers to, or lead to a single object or value; type identity occurs when two separate objects happen to have the same composition in terms of attributes and values.
8. One could also imagine other dimensions, such as the associated string, *i.e.* the phonological or textual forms. In the 'tree' part of (16), the RULE feature encodes the name of the grammar rule used, and the features D1 and D2 encode the daughter constituents combined by the rule.
9. For simplicity, we have suppressed a number of details here. The vertical bar '|' is used to abbreviated long paths of attribute names in these FSs.
10. An alternative would involve the use of explicit mechanisms for passing information about other levels around the FSs in the derivation path as the recursive transfer algorithm is applied to one dimension. But this is complicated in a number of ways.
11. A 'bag' is a multiset, that is, a set which may contain multiple instances of objects (this is not the case with sets: {a,a,b}, and {a,b} are the same set, but different multisets).
12. There are some complications where the languages differ in the re-entrances they permit. In these cases, the possibility of underspecification of the target f-structure must be exploited. This issue is discussed in Arnold and Sadler, 1992, and Sadler and Arnold 1992.

BIBLIOGRAPHY

ARNOLD, D. J. and L. SADLER (1990): "The Theoretical Basis of MiMo", *Machine Translation*, vol. 5, pp. 195-222.

ARNOLD, D. J. and L. SADLER (1992): "Rationalism and the Treatment of Referential Dependencies", *Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, 25-27 June 1992, CCRIT-CWARC Montreal, pp. 195-204.

BECH, A. (1991): "Description of the Eurotra Framework", C. Copeland, J. Durand, S. Krauwer, B. Maegaard (Eds.), *The Eurotra Formal Specifications*, Office for Official Publications of the Commission of the European Community, Studies in Machine Translation and Natural Language Processing, vol. 2, pp. 7-40.

ESTIVAL, D., BALLIM, A., RUSSELL, G. and S. WARWICK (1990): "A Syntax and Semantics for Feature-Structure Transfer", *Third International Conference on Theoretical and Methodological Issues in Machine Translation*, 11-13 June 1990, Linguistics Research Center, Austin, Texas, pp. 131-144.

KAPLAN, R. M. and J. BRESNAN (1982): "Lexical Functional Grammar: a Formal System for Grammatical Representation", J. Bresnan (Ed.), *The Mental Representation of Grammatical Relations*, MIT Press, Cambridge, Mass., pp. 173-282.

KAPLAN, R. M., NETTER, K., WEDEKIND, J. and A. ZAENEN (1989): "Translation by Structural Correspondences", *EACL-4*, p. 272-281.

PERLMUTTER, D. M. (Ed.) (1983): *Studies in Relational Grammar 1*, Chicago, University of Chicago Press.

POLLARD, C. and I. SAG (1987): *Information Based Syntax and Semantics, vol. 1 Fundamentals*, Chicago, CSLI Lecture Notes 13, Chicago University Press.

SADLER. L. (1991): "Structural Transfer and Unification Formalisms", *Applied Computer Translation*, vol. 1, n° 4, pp. 5-21.

SADLER, L. (forthcoming): "Co-description and Translation", Frank van Eynde (Ed.), *Linguistics and Machine Translation*, London, Pinter Publishers.

SADLER, L. and D. ARNOLD (1992): "A Constraint-Based Approach to Translating Anaphoric Dependencies", *Proceedings of COLING-92*, Nantes, (Also, Working Papers in Language Processing 29, Department of Language and Linguistics, University of Essex).

SADLER, L., I. CROOKSTON, D. ARNOLD and A. WAY (1990): "LFG and Translation", *Third International Conference on Theoretical and Methodological Issues in Machine Translation*, 11-13 June 1990, Linguistics Research Center, Austin, Texas, pp. 121-130.

SADLER, L. and H.S. THOMPSON (1991): "Structural Non-Correspondence in Translation", *Proceedings of EACL-91*, Berlin, pp. 293-298.

SHARP, R. (1988): "CAT2 — Implementing a formalism for Multi-lingual MT", *Proceedings of the Second International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Pittsburgh, Carnegie Mellon University.

SHIEBER, S. (1986): *An Introduction to Unification Based Approaches to Grammar*, CSLI Lecture Notes n° 4, Chicago, University of Chicago Press.

NOORD, G. VAN, J. DORREPAAL, P. VAN DER EIJK, M. FLORENZA and L. DES TOMBE (1990): "The MiMo2 Research System", *Third International Conference on Theoretical and Methodological Issues in Machine Translation*, 11-13 June 1990, Linguistics Research Center, Austin, Texas, pp. 213-124.

WHITELOCK, P. (1991): "Shake and Bake Translation", Sharp Labs. of Europe, Abingdon, Oxford, April 1991.