I.T.L. Review 64(1984), 1-21

About the Treatment of Ambiguity in Machine Translation*

Lieven JASPAERT
K.U. Leuven

1. Synonymy and ambiguity

Notions like *synonymy* and *ambiguity* are common ones in linguistics. Traditionally, one refers to expressions of the same language as synonyms when, despite their formal differences, they mean the same[1]. Expressions of some language are said to be ambiguous when one single expression has several non-equivalent meanings. In these definitions, the crucial part contains each time the notion *meaning* (same or different meaning).

Now if meaning were describable in an absolute fashion as composed of only constant cognito-semantic atoms, meanings as they are expressed by language could with ease be compared as to their similarities and differences. But meaning is a *relative* notion : what one says to be the meaning of some expression varies according to the particular perspective one chooses to adopt for talking about meaning. In linguistics, for example, theoreticians need not mean the same thing when they discuss meaning. It then follows that, if the definition of the notion meaning is a function of the particular semantic theory advocated, synonymy and ambiguity must also be viewed as theory-dependent or relative notions. They can only be defined properly within the boundaries of the meta-language used.

Consider, by way of clarification, sentences (1-2).

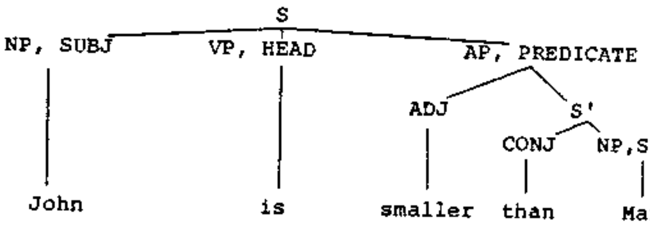
(1) John is smaller than Mary
(2) Mary is taller than John

Suppose we wish to adopt a very deep semantic representation system, which were made to include some kind of logic-inspired device yielding one single meaning representation (3) for both of the above sentences.

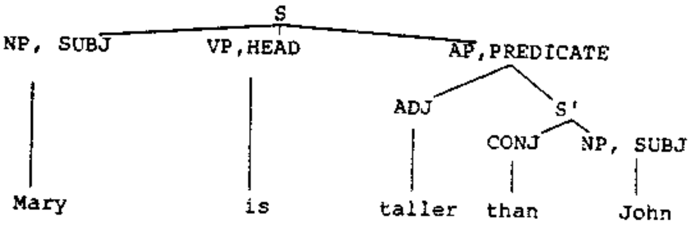
(3) SIZE (Mary) > SIZE (John)

Within this approach, (1) and (2) would be synonyms in English. But if one opted, however, for reasons that do not matter here, for a more shallow semantic description which would assign the representations (4) and (5) to the sentences (1) and (2), respectively, the latter would evidently not be synonyms in English, given our semantic theory[2].

(4)



(5)



It is not difficult to see that the same applies to ambiguity. Consider, for instance, sentence (6).

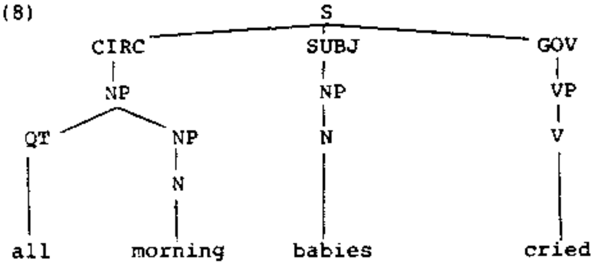
(6) All morning babies cried.

Suppose that the analysis grammar used for representing the information contained in (6) were something like (7)[3].

```
(7) V(A*) == VP(V(A*)).
    N(A*) == NP(N(A*)).
    QT(A*) + NP(U*) == NP(QT(A*), NP(U*)).
    NP(U*) + NP(V*) + VP(W*) == S(CIRC(NP(U*)), SUBJ(NP(V*)),
                                HEAD(VP(W*))).
```

This grammar will assign to (6) only one structural representation, namely (8). The obvious implication is, of course, that (6) is not ambiguous.

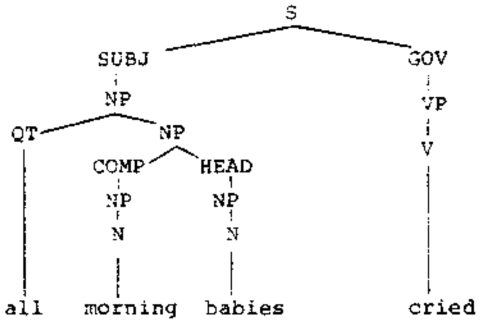
(8)



If, on the other hand, we choose to use a more refined analysis grammar such as the one given in (9), sentence (6) could indeed be represented in two non-equivalent ways, one of which would be identical to (8), another one (10). Hence, under our second grammar, (6) is ambiguous.

```
(9) V(A*) == VP(V(A*)).
    N(A*) == NP(N(A*)).
    NP(U*) + NP(W*) == NP(COMP(NP(U*)), HEAD(NP(W*)))
                       / QT -HORS- U*.
    QT(A*) + NP(U*) == NP(QT(A*),NP(U*)).
    NP(U*) + NP(V*) + VP(W*) == S(CIRC(NP(U*)),SUBJ(NP(V*)),
                                GOV(VP(W*))).
    NP(U*) + VP(W*) == S(SUBJ(NP(U*)),GOV(VP(W*))).
```

(10)



We hope that it has become clear now that there exists a tight link between the level of depth at which some semantic description mechanism operates, on the one hand, and the generalizing capacity of the definition of synonymy and ambiguity, on the other. We shall look further into this relation within the context of problems related to Machine Translation[4].

2. Machine Translation

It can be safely stated that (automatic) translation should be meaning-preserving, i.e. no semantically relevant information may be lost in the course of the operations which constitute the translation process. This implies that the target language (TL) sentence should mean the same as the source language (SL) sentence it is derived from. Whereas expressions belonging to the same language were called synonyms if they mean the same, expressions which mean the same but which belong to different languages will be called *S-equivalents*. Notice that S-equivalence is, just like its sister synonymy, a relative notion that can only be defined within some specific semantic representation theory. It goes without saying that this will be the semantic theory implemented into the Machine Translation system. We can, therefore, define now when exactly we may consider sentences of different languages as S-equivalents, namely when the structural descriptions of these sentences are *D-equivalent*. D-equivalence of structural descriptions we define as identity of the semantically relevant parts of these structural descriptions[5].

Suppose we incorporated into our MT system a universal semantic interpretation component which would be powerful enough to represent, on the basis of its finite set of semantic primitives, any possible meaning which can be expressed in any natural language. Such (still highly hypothetical) universal meaning systems are called *interlinguas*[6]. It is clear that within such a framework, sentences (11-13) would have D-equivalent structural representations, and that they would hence be synonymous (or S-equivalent, assuming we translate from and into English).

(11) John borrowed 300 dollars from Steve

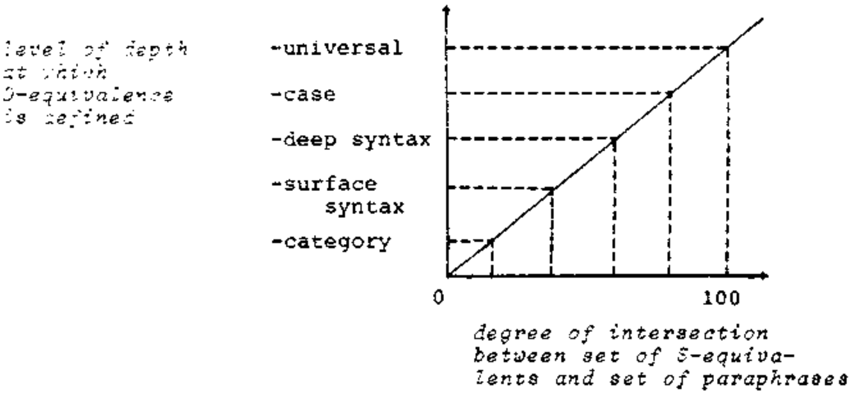
(12) Steve lent 300 dollars to John

(13) Steve gave John a 300 dollar loan

But it is obvious then that if we were to use a less pretentiously deep semantic representation system, these sentences need not be S-equivalent.

The conclusion we must draw from the above observations is that the set of all S-equivalent sentences need not coincide with the set of all paraphrases[7], in the broadest possible sense of the word. How total this intersection is depends hence on the depth of the semantic interpretation system employed for assigning structural representations to those sentences. It would be very convenient for MT, however, to be able to define D-equivalence (and hence S-equivalence) in such a way as to make the set of S-equivalent sentences as large a subset of the superset of all paraphrases of those sentences. In order to do so we have to (i) choose as deep as possible a level for semantic representation, but (ii) avoid choosing too deep a level, since the more abstract the representations one aims at are, the greater the likelihood becomes that the latter can no longer be computed in a straightforward fashion, if at all[8]. The right attitude for making this choice is hence : as deep as possible, but as shallow as necessary. Figure (14) shows neatly the relation between various options of depth of the semantic description system and the degree of intersection between the respective sets of S-equivalent sentences and of paraphrases[9]. It is worth mentioning here, as was pointed out to us by one of the readers of an early draft of this article, that we are not implying that these are *exclusive* choices : i.e. that a system

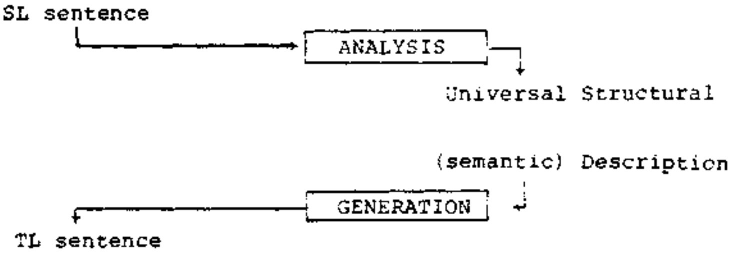
(14)



is not necessarily committed to throwing away all shallower information once a certain level of depth is attained in the course of analysis, nor to using only the deeper level information. Our purpose here is solely to point out that, whatever level(s) of information is (are) taken into consideration for defining D-equivalence, this choice has immediate consequences on the nature of the tasks to be performed by the Machine Translation system. Since this is a complex problem, we shall devote an entire section to it.

3. D-equivalence and its relation to the design of MT systems. Our definitions of S-equivalence and D-equivalence, on the one hand, and the phenomena depicted in (14), on the other, have important consequences on the design of MT systems. First, one can see clearly that, if the semantic description system used were of the interlingua type, the translation process would mainly consist of the two components shown in (15).

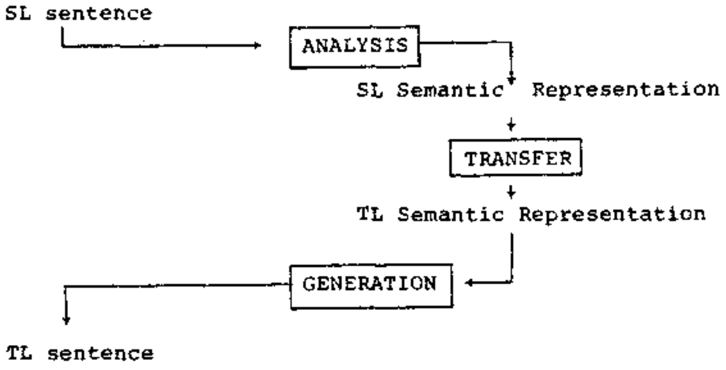
(15)



At this moment, interlinguas which could handle the immense job of translating natural language into universal semantics do not exist. Current MT systems use a more shallow semantic descriptive mechanism, such as Case Grammar-inspired semantic systems (EUROTRA)[13]. The logical consequence of such a choice is that, as predicted by (14), the translation process looks quite different from the one outlined in (15). (16) shows this diagrammatically.

This translation process has properties which are quite distinct from the ones exhibited by the previous one. Its analysis and generation components are solely monolingual, in that they

(16)



do not use information belonging to the target language or the source language, respectively. Transfer is the only component which makes use of information of both languages : it is bilingual. Notice also that, whereas in (15) lexical items will be decomposed into their constitutive semantic primitives, (16) leaves them untouched[11]. It is the task, amongst others, of transfer to perform the correct lexical substitution on the basis of information contained in both the bilingual dictionaries it uses and the structural descriptions of the SL sentence it operates on. But lexical transfer is not the only task of the transfer component. In section 2 we have shown that the set of S-equivalents is only a subset of all paraphrases, if no universal semantics were used. From this it follows that the source language will contain a number of constructions (sentences) for which it is not the case that their structural description is D-equivalent to the structural description of the TL sentences one deems to be the appropriate translations of these SL sentences. Consider, for instance, the following Dutch sentence.

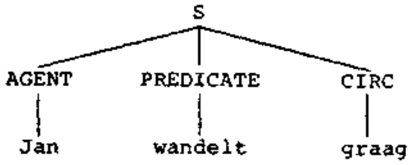
(17a) Jan wandelt graag

(John-walks-*pleasurably) = John likes to walk

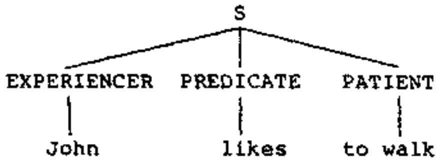
(17b) and (17c), respectively, give the structural description of the Dutch (SL) sentence and the appropriate English (TL) sentence.

The translation from Dutch to English in this case cannot be done in a straightforward way, since the structural descriptions

(17b)



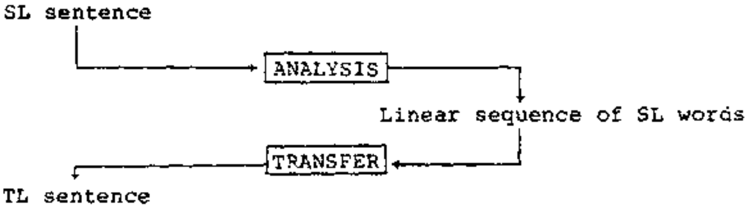
(17c)



of the respective sentences are not D-equivalent. Here then lies another task for the transfer component : if it receives from analysis a structural description for which the target language possesses no direct D-equivalent, transfer will have to modify the input structure in order to make it D-equivalent to the corresponding target language structure. This operation is known as *structural transfer*. From what precedes it will be obvious that, in order to determine which kinds of structural transfer must be performed, some rigid definition of D-equivalence (and hence of S-equivalence) must be wired into the design of the MT system.

Let us now make one final step further by supposing that we would dispense with any kind of semantic analysis whatsoever. Analysis would then consist of the trivial operation of mapping SL sentences onto some linear sequence of lexical items. This is an extreme case, since one would certainly want to perform some preliminary morphological analysis, be it only to alleviate the lot of the poor dictionary writer. But for the sake of the argument we shall assume that morphological information is contained in the monolingual dictionary. The translation process would now be the one shown in (18).

(18)



As (18) points out, no generation would remain. Transfer, in such a system, would consist of a whole bag of extremely ad hoc rules for choosing correct word translations. It needs no argument that such MT systems would produce very trivial and bad translations. Incidentally, the early MT systems designed in the fifties were based precisely on the idea that translation could be performed without any syntactic, let alone semantic, analysis of the SL sentences.
We hope to have demonstrated by now the relevance of notions like D-equivalence and S-equivalence for MT. Let us now come back to the problem of ambiguity.

4. D-equivalence, S-equivalence and ambiguity

In the previous sections, we have introduced the notions of D-equivalence and S-equivalence. We have shown that the definitions of these notions have bearing on the divison of tasks between the various components of an MT system. The existence and the actual representation of ambiguity has been deliberately left out of the discussion. In this section we will examine to what extent the definitions given are affected by ambiguity.
Let us first set the terminological scene. Consider an ambiguous SL sentence S_{SL}. We shall assume throughout the present discussion that the SL analysis component will possess a restricted disambiguating power : it will be capable of choosing between ambiguous lexical items in a number of cases[12] or resolve structural ambiguity in those cases for which it possesses the information necessary for computing the 'right' alternative. However, it will happen that analysis will not be able to eliminate all competing alternatives and to decide in favor of one of them. If this happens, the SL analysis component will construct for each one of the alternative readings of S_{SL} a structural description which is itself not ambiguous. From what follows it will become clear that, if we were to give analysis disambiguating power even in these cases, such disambiguation could only be performed on arbitrary grounds and would lead to a loss of information and evidently bad translations. A possible way of escape would be

to equip analysis with a preference assignment device which would associate a preference weight to each alternative analysis tree. The computation of these preference values is by no means an easy undertaking. Very subtle semantic and even world knowledge information is required for performing this task. We shall not deal with the modalities of the incorporation of such a mechanism in MT systems, but simply consider its theoretical implications for the treatment of ambiguity.

It may finally be instructive to remind the reader of the traditional properties of an equivalent relation. In order to be equivalent, a relation must be *symmetric, reflexive* and *transitive.* Symmetry means that it follows from

(i) A R B (A stands in some relation R to B)

that

(ii) B R A

A relation is reflexive when it is the case that if

(i) A R B

then it is also the case that

(ii) A R A & B R B

Finally, a relation is said to be transitive when it follows from

(i) A R B & B R C

that

(ii) A R C

We also should warn the reader here that, whereas D-equivalence is a genuine equivalent relation, i.e. has all three properties defined in the above, S-equivalence is not. The reason why this is the case is the existence of ambiguity.

4.1. Types of ambiguity

We now invite the reader to perform some mental gymnastics. We will consider, one after the other, three hypothetical languages : language L in which no ambiguity occurs, language M which has only *correlating ambiguity* and language N which has *wild ambiguity.*

Imagine three sentences of language L, S_1^L, S_2^L and S_3^L, respectively. Since language L has only non-ambiguous sentences, the analysis component of L will assign one analysis tree to each respective sentence : $A(S_1^L)$, $A(S_2^L)$ and $A(S_3^L)$, respectively. Suppose now that

(19) $A(S_1^L)$ *D-equivalent-to* $A(S_2^L)$
& $A(S_2^L)$ *D-equivalent-to* $A(S_3^L)$

According to our definition of S-equivalence in terms of D-equivalence, (20) must follow logically from (19).

(20) S_1^L *S-equivalent-to* S_2^L
& S_2^L *S-equivalent-to* S_3^L

But what is the nature of the S-equivalence relation in language L ? It is clear that S-equivalence in L is a reflexive relation : S_1^L is S-equivalent to itself. It also is a symmetric relation : if S_1^L is S-equivalent to S_2^L, then the converse must also be the case since both sentences have analysis trees which are D-equivalent. But in language L S-equivalence is also transitive : if (19) is true in language L, then surely so must be (21), and so will be (22).

(21) $A(S_1^L)$ *D-equivalent-to* $A(S_3^L)$

(22) S_1^L *S-equivalent-to* S_3^L

The step from (21) to (22) can be made due to the definition of S-equivalence in terms of D-equivalence.
In a language with no ambiguity, such as our language L, both D-equivalence and S-equivalence are genuine equivalent relations in the mathematical sense of the term. As a matter of fact, in language L we could dispense with the distinction between S-equivalence and D-equivalence entirely, and solely speak of an equivalence relation between sentences and/or analysis trees of language L. But, (un)fortunately, natural languages are not as univocal as our imaginary language L.
Imagine then some language M which knows ambiguity, but only of a very particular kind. We refer to (23) as to a case of *correlating ambiguity*.

$$(23)\quad \begin{array}{lclclcl} S_1^M & \textit{S-equivalent-to} & S_2^M & \& & S_2^M & \textit{S-equivalent-to} & S_3^M \\ & \uparrow & & & & \uparrow & \\ A_1(S_1^M) & \textit{D-equivalent-to} & A_1(S_2^M) & & A_1(S_2^M) & \textit{D-equivalent-to} & A_1(S_3^M) \\ & \& & & & & \& & \\ A_2(S_1^M) & \textit{D-equivalent-to} & A_2(S_2^M) & & A_2(S_2^M) & \textit{D-equivalent-to} & A_2(S_3^M) \end{array}$$

In language M, S-equivalence is defined in terms of the resemblances between the sets of alternative structural descriptions for sentences of M. Ambiguous sentences of M are S-equivalent if they have exactly the same number of alternative analysis trees and iff for each analysis tree for S^M_i there exists a D-equivalent S^M_j. It is furthermore clear that S-equivalence in language M is a genuine equivalence relation, despite the existence of some kind of ambiguity. Natural languages sometimes display this type of ambiguity, but it is far from being the most common type. E.g. the S-equivalence between the following Dutch and German sentences, which have the same ambiguity properties. Both sentences can be understood

(24) Hij heeft het paard genomen
(he-has-the horse/the knight-taken)

(25) Er hat das Pferd genommen
(he-has-the horse/the knight-taken)

in two ways depending on whether one refers to an animal or rather to a chess-piece.

Consider, finally, language N which exhibits *wild ambiguity*. By this term we understand the possibility that sentences of N have a different number of analysis trees of which only a subset (which may even be empty) are D-equivalent. Such a case is illustrated in (26).

(26)

$$\begin{array}{cccccc} S^N_1 & \overset{?}{\uparrow} & S^N_2 \ \& & S^N_2 & \overset{?}{\nuparrow} & S^N_3 \\ A(S^N_1) & \textit{D-equivalent-to} & A_1(S^N_2) & A_1(S^N_2) & \textit{D-equivalent-to} & * \\ & \& & & & \& & \\ * & \textit{D-equivalent-to} & A_2(S^N_2) & A_2(S^N_2) & \textit{D-equivalent-to} & A(S^N_3) \end{array}$$

In (26), S^N_2 has two analysis trees of which the first is D-equivalent to the only analysis tree of S^N_1 (but not of S^N_3), and of which the second is D-equivalent to the analysis tree of S^N_3 (but not of S^N_1). This type of ambiguity occurs in natural language. Consider, for instance, the following sentences (27-29).

(27) Morning babies all cried

(28) All morning babies cried

(29) Babies cried all morning

The analysis tree of the non-ambiguous sentence (27) is D-equivalent to one of the two analysis trees for sentence (28), but surely not to the analysis tree of (29). Conversely, sentence (29) has an analysis tree which is D-equivalent to the second analysis tree of (28), but not to the first one, nor to the analysis tree of (27).

The problem we are now facing is to define the nature of the relation between (27) and (28), and between (28) and (29), respectively -or, in a general way, between S_1^N and S_2^N, and between S_2^N and S_3^N. It is obvious that we are no longer dealing with a transitive relation, for S_1^N will certainly not be S-equivalent to S_3^N: indeed, as can be seen in (26), they have no analysis trees which are D-equivalent. Nor will (27) be S-equivalent to (29), for the same reason. We can see now that S-equivalence in natural language and in language N is not an equivalence relation in the mathematical sense. We shall have to define it in another way.

We shall propose and discuss three possibilities for defining S-equivalence in natural language : a strong definition, a weak one and an intermediate one. According to the strong definition, sentences are S-equivalent when their respective sets of alternative analysis trees resemble each other in the following way : they must have the same number of analysis trees and the latter must correlate over the respective sets, i.e. for each analysis tree for sentence_i there must be exactly one D-equivalent analysis tree for sentence_j. Notice that this is precisely the definition of S-equivalence described for language M. Applied to language N, or to natural language in general, this strong formulation implies that the relation between S_1^N and S_2^N, on the one hand, and between S_2^N and S_3^N, on the other, would not be an S-equivalence relation. What are the consequences of such a definition for the processes involved in MT ? During the discussion in our previous sections we made it clear that an MT system must make sure that

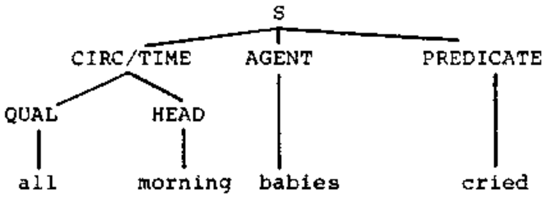
the SL sentence is S-equivalent to the TL sentence derived, i.e. both sentences must have analysis trees which are D-equivalent[13]. It is not hard to see that if, a non-ambiguous SL sentence were translated into an ambiguous TL sentence, these sentences would, under our strong definition, not be S-equivalent. Generation is not allowed to 'ambiguate'.
But this observation has drastic consequences for the task of generation : if this component may not ambiguate, it will have to check itself whether the TL sentences it produces have exactly the same (congruent) set of analysis trees as the original SL sentences. This can, in our opinion, not be achieved unless generation is made to analyse post-factum its own output. This is clearly a very burdensome procedure which could be avoided by defining S-equivalence in a less stringent way.
The weak definition goes as follows : for sentences S_i and S_j to be S-equivalent, the set of analysis trees for S_i must contain at least one analysis tree which is D-equivalent to an analysis tree of S_j. Phrased in a different way, S-equivalence holds between sentences which have a non-empty intersection of D-equivalent analysis trees. We shall discuss the implications of this definition in the next section.
In fact, there exists a way to strengthen this weak definition by demanding that S-equivalent sentences have sets of analysis trees which intersect in those D-equivalent analysis trees which have, for each respective sentence, been assigned the highest preference value. So, the intersection cannot just be non-empty; the D-equivalent analysis trees should also be the ones which were regarded by the preference device as the most likely reading for some particular sentence. We shall not push this issue further, since the description of such a preference device is a difficult task which falls beyond the scope of this paper.

. The weak definition and its consequences for an MT system.

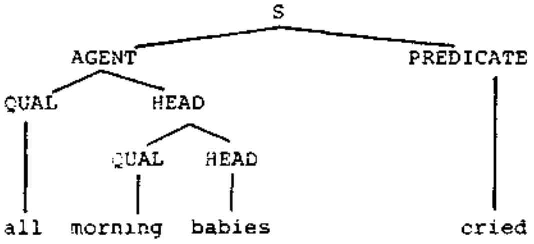
In this section, we will investigate in what ways ambiguity is dealt with by a translation system built around the notions of S-equivalence and D-equivalence defined in the weak way. In a way, this section will serve as a check on the observa-

tions made in the previous sections.
We shall consider two cases, one in which a non-ambiguous sentence is tranlated into an ambiguous one (ambiguation) and another where an ambiquous sentence has to be translated.
In order to make things as clear as possible, we assume that we are translating from English back into English.
Let us first check in what way the English generation will cope with the crying-babies examples. Given source language sentence (30), the English analysis component will construct two different analysis trees, A_1 (30) and A_2 (30), respectively[14].

A_1 (30)



A_2 (30)



Since vertical geometry (expressing dependency relations) is to be considered a semantically relevant element of the structure, A_1 (30) and A_2 (30) are not D-equivalent to each other, so (30) is genuinely ambiguous.
Since we translate back into English, A_1 (30) and A_2 (30) will be the input to English generation. This component must derive from each of these analysis trees some TL sentence which is S-equivalent to our original sentence (30). Notice that both alternative trees are fed to generation, since in this case neither analysis nor transfer could have disambiguated (30). According to our definition of S-equivalence, generation can derive from these two analysis trees the two respective sets of sentence which follow here.

(31) a. All morning babies cried
b. *Morning babies all cried
c. Babies cried all morning

(32) a. All morning babies cried
b. Morning babies all cried
c. *Babies cried all morning

TL sentence (32 c) cannot be derived from A_2 (30), since its own TL analysis tree is not D-equivalent to A_2 (30). This is the condition embodied in our definition of the task of the generation component : for the TL sentence to be S-equivalent to the SL sentence, its own TL analysis tree must be D-equivalent to the SL sentence's analysis tree[15]. But both (32 a,b) are derivable from A_2 (30), given the fact that both these TL sentences have at least one analysis tree which is D-equivalent to A_2 (30), and are hence both S-equivalent to (30). The same kind of observation can be made for the TL sentences given in (31), all derived from A_1 (30). (31 b) is not derivable, since its own analysis tree is not D-equivalent to A_1 (30); (31 a,c) are derivable by the generation component since both are S-equivalent to (30) and have each at least one analysis tree which is D-equivalent to A_1 (30)[16].
But an important question remains unanswered : which will be the ultimate translation of our ambiguous sentence (30). As we can see from the two sets of possible TL sentences, derived from the two respective alternative analysis of (30), (30) could possibly be translated into two alternative TL sentences, in our case (32 b) and (31 c). Where then must the choice be made as to the most appropriate translation ? It is clear that, unless either preference weights are implemented into the system, or analysis, transfer or generation are given the power to disambiguate in all cases of ambiguity - on what grounds this disambiguation will be done is not immediately obvious -, the ambiguity will remain unsolved throughout the entire translation process. The only way out which is left is to have the post-editor make the ultimate choice. However, we have to keep in mind the fact that we are trans-

lating from and into English.

This of course makes it possible for the MT system to end up with the same sentence as the one it started out from.

In the majority of cases, post-editing will be required.

This necessity of post-editing seems, however, preferable to the situation where the machine itself would make the ultimate choice on completely arbitrary grounds.

The second case we wanted to investigate is one where a non-ambiguous sentence is to be translated. Suppose the TL sentence produced by generation were itself ambiguous. Such a case can be exemplified by the SL sentence (33) and its ambiguous translation (34).

(33) SL : Babies cried all morning

(34) TL : All morning babies cried

Notice that such a case may occur, given our weak definition of S-equivalence. As a matter of fact, (34) has amongst its two analysis trees one tree which is D-equivalent to the analysis tree of (33). If we assume the weak definition of S-equivalence, there seems to be no way to avoid cases of ambiguation described here[17]. Even the assignment of preference weights cannot straightforwardly solve the problem : since the ambiguity only occurs at the end of generation, it would be necessary for generation to re-analyse its own output. We have already pointed out that such a procedure would enormously complicate the translation process.

Do we have to drop then our weak definition of S-equivalence ? It goes without saying that we cannot choose the stronger definition because that would only make things worse. Indeed, if S-equivalence were defined in terms of a correlation between the alternative sets of analysis trees, generation would again have to check its own output, and this time in a more severe way. This time it would not only have to check for ambiguation, but also for those cases where not all analysis trees of the SL and the TL sentences, respectively, are D-equivalent to each other on a one-to-one basis. Do we then have to weaken even more our definition of S-equivalence, accepting that generation produces translations which have no analysis trees which are D-equivalent to some analysis tree of the SL

and the TL sentences, respectively, are D-equivalent to each other on a one-to-one basis ?
This would be going too far : indeed, if such were the option made, it would become impossible to foresee and control the translation of ambiguous sentences.

6. Conclusions

In this paper we have proposed a definition of the relation that binds SL and TL sentences in an MT system. To do so we let ourselves be inspired by the inherently relative notions of synonymy and ambiguity. We have also shown the link between some arbitrary depth of analysis and the question of which tasks are performed in what component of the MT system. The over-all design of such a system will be influenced by whatever definition of S-equivalence (and D-equivalence) is chosen.
The main conclusion to draw from this paper is that ambiguity remains an obstinate problem, despite the attempt to formalize to some extent the meaning-preserving characteristic of Machine Translation systems. By adopting the weakest definition of S-equivalence, we were able to suggest a plausible solution to the problem of translating ambiguous sentences correctly. What we did not solve at all, however, is the case where a non-ambiguous sentence is translated into an ambiguous one. We do not even know whether this will happen in a sufficiently big number of cases in order for us to worry about the problem. If it does, maybe one should not rule out completely the possibility of some re-analysis after generation, checking for unwanted ambiguities. But, in any case, we have certainly made the point clear that whatever solution one cares to propose for resolving the ambiguity problem in MT must be based on some definition of S-equivalence.

NOTES

[*] We are endebted to Maghi King, F. van Eynde and K. van den Eynde for many useful comments on earlier drafts of this paper. The research which has led to the present article was performed partly within the framework of the Machine Translation project of the European Communities, EUROTRA. A number of the ideas presentend here emane from one of the conjoined Belgo-Dutch reports, presented to the E.E.C., in casu Van Eynde, Honig, Jaspaert, Krauwer, Neijt, des Tombe, Vaassen, *The Task of Transfer vis-à-vis Analysis and Generation*, ET-10/B-NL, 1982.

[1] See e.g. the *Dictionnaire de Linguistique*, Larousse, 1973, p. 476 : *'Sont synonymes des mots de même sens, ou approximativement de même sens, et de formes différentes'*. It is useful to point out here that we use the notion 'synonymy' also for sentences which mean the same but are formally different. Traditionally, one finds the notion paraphrase used with respect to synonymous sentences; we shall, however, use the notion paraphrase in another acception.

[2] The tree structures given throughout the article have only an illustrative function. They do not form part of the points argued in this paper.

[3] The analysis grammar rule syntax used is the one described in Colmerauer, *Les Systemes-Q ou un formalisme pour analyser et synthetiser des phrases sur ordinateur*, publication interne n° 43, 1970. For clarity reasons, we have not incorporated the dictionaries into our small grammars.

[4] For a more complete exposition concerning Machine Translation systems in general, see Jaspaert, L., *Machinevertaling. 'Out of sight, out of mind'* of *'the invisible lunatic'*, Romaneske VII. 1, 33-43, 1982.

[5] The precise definition of D-equivalence is also relative : any choice as to include a number of semantically relevant elements of the structural descriptions of sentences has immediate consequences on the over-all design of an MT system. Let us assume here that with semantically relevant elements we mean (i) the vertical geometry of the analysis tree - i.e. surface word-order is semantically irrelvant in as much as it is not already represented vertically -, (ii) semantic labels on nodes of the tree, (iii) some category information and (iv) dictionary information about lexical units.

[6] An example of an interlingua is presented in Schank, R., *Identification of Conceptualization Underlying Natural Language*, in : Schank & Colby (eds), *Computer Models of Thought and Language*, 1973, 187-247.

7. By paraphrases we mean the set of all sentences which can be substituted in some particular context. Notice that context is intended here to cover also the extra-linguistic context : as such, paraphrase is not a notion relating to the competence of language, but rather to performance.
8. It is a well-known fact that the more abstract a level of description is, the more difficult computation of values at that level becomes and the more errors become probable.
9. It is important that levels of description be defined in terms of their distinctive properties vis-à-vis all other levels, and that for each level one define some unifying principle and a set of distinguishable, non-overlapping values.
10. Actually, EUROTRA uses a more complex semantic representation system, including, however, also case information about sentence constitutents. We have chosen not to burden the reader with facts which do not hinge on any of the point made in this paper.
11. Lexical items, in such systems, remain in the structural description, but get their semantic properties attached to them. Interlingua-based systems delete the words altogether in favor of some structured list of semantic primitives.
12. In many cases the analysis component must choose between alternative meanings of SL lexical items. The reason for this is that it would otherwise fail to compute values for other constituents which depend on the constituent which needs disambiguating.
13. One has, however, to take into account the fact that for a number of SL analysis trees the target language has no ready-made equivalent. Transfer will then have to modify the analysis tree of the source language in order to make it D-equivalent to some possible TL analysis tree and hence make it 'processable' for the TL generation component. We could add to the text '... both sentences must have analysis trees which are D-equivalent, *or made so by the transfer component*'.
14. We have made it clear in section 1 that English analysis can only detect ambiguity if it is powerful enough to do so, i.e. if it contains an analysis grammar which is fine-grained enough to capture ambiguity. We will assume that our English analysis can deal with the examples given.
15. Cf. footnote 12.
16. Compare e.g. with what would happen if we were to translate to French, rather than back to English. From A_1 (30) the French generation would be capable of deriving

(31') a. Toute la matinée les bébés pleuraient

b. *Les bébés matinaux pleuraient tous

c. Des bébés pleuraient toute la matinée

and from A_2 (30) the sentences in (32') could be generated.

(32') a. *Toute la matinée des bébés pleuraient

b. Les bébés matinaux pleuraient tous

c. *Des bébés pleuraient tout la matinée

We can see clearly now that (i) (31' a) is not ambiguous, whereas its English counterpart was, and (ii) (32' a) is no longer derivable from A_2 (30) since it has no D-equivalent analysis tree with respect to A_2 (30).

17. It is true that we could avoid ambiguation in this particular case, by making sure that generation respects as much as possible the surface word-order of the SL sentence. The implication of this addition is that surface word-order of the SL sentence must be preserved during analysis and transfer performed on the SL sentence. The consequences of this necessity are far-reaching. One will no longer be permitted to move constituents back to their original place in the structure, unless a powerful device of tracing is also implemented into the system. As a matter of fact, only then can the surface word-order be recoverable from the analysis structure. But, in any case, such modifications of generation component as proposed here will surely not avoid all cases of ambiguation, but merely those which are caused by changes in word-order