THE A T E F AND C E T A SYSTEMS

J. CHAUCHE

Mathematiques Appliquees - Informatique
Universite Scientifique et Medicale
Grenoble

SUMMARY

ATEF converts an input string into a labeled tree; the label evolves under the control of a grammar. A set of labels is associated with each segment of the string, and several functions permit control of the number of alternative labels. CETA simulates a transformational grammar. It uses a set of grammars with conditional linkages. The applicability of a transformation can be determined in part by conditions on the resulting tree.

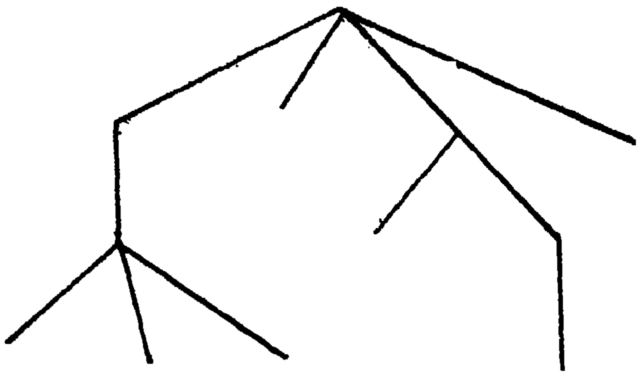
Computer processing of natural languages requires more or less polished algorithmic models. The two systems presented here represent a choice of a large class among the algorithms proposed in recent years to solve these problems. The principal choice determined by these systems lies in the formal use of labeled trees (*arborescences*). Freedom of choice of these labels and possible structures gives these systems broad fields of applications in several domains and notably in that of the automatic processing of natural languages. The ATEF system has the purpose of transforming a string of words into a tree which is manipulable by the CETA system. The definition of labeled trees determines what objects CETA can manipulate and the objectives of ATEF. This note therefore begins with the definition of labeled trees. To obtain a tree of this type beginning with an input string, ATEF uses a dictionary and a finite-state grammar. The result of this system can be manipulated by CETA in order to obtain the desired type of structure. The example of analysis given here shows the possibilities of the CETA system with two different manipulative strategies: search for constituent or dependency structure.

1. LABELED TREES

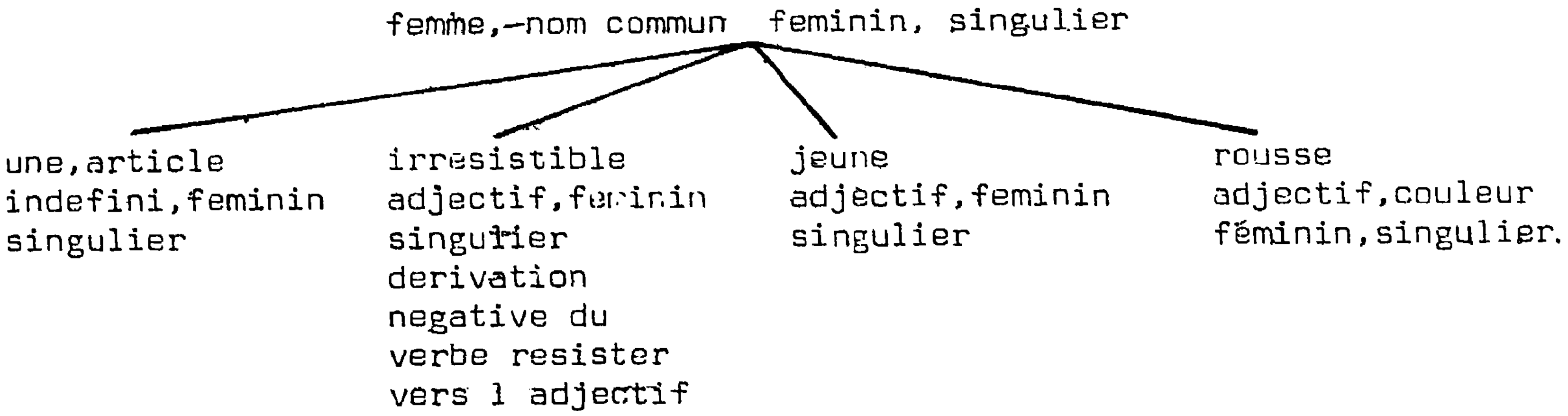
A tree is a set of points with which is associated a structure, that is to say a relation having the properties:

> The relation between two points is directed (one point depends on the other)
>
> A point cannot depend on a point belonging to its own descent set (the descent set of a point is the set of points that depend on it, the points that depend on them, etc.)
>
> A unique point descends from no other.

It is possible to draw a tree placing below a point all of its descendants, linked by lines. (See the example on the next frame.)



A labeled tree is a tree such that with each of its points is associated a label This label is formed of a set of data. The figure below represents a labeled tree.


femme,-nom commun feminin, singulier
une,article indefini,feminin singulier
irresistible adjectif,feminin singulier derivation negative du verbe resister vers 1 adjectif
jeune adjectif,feminin singulier
rousse adjectif,couleur féminin,singulier.

2: THE ATEF SYSTEM

The purpose of the ATEF system is to transform an input string of words into a labeled tree, each word in the string possibly, leading to one or several points in the final tree (ambiguity). The determination of the label originating in an input word results from its analysis. This analysis proceeds by segmentation of the input word according to elements from different dictionaries . A correct segmentation therefore gives a label for a point of the final tree . In advance of any analysis, the definition of the elements employed in the composition of different labels is required and is supplied by two files called variable declaration files.

A label will consist of a set of variables. Each variable must be defined with its set of possible values. Thus if one defines the variable "category" the set of "categories" that can be used must be specified. The set is written

category = (NOUN, ARTICLE, PRONOUN, ADJECTIVE, VERB, ...)

(A constraint requires that the name of a variable must not be longer than 7 characters. Thus the preceding variable could be written, for example, CAT = (NN, ART, PRN, ADJ, VRB, ...))

The definition of a particular label consists in an enumeration of the variables relevant to the label. A set of labels can be predefined and is collected in a so-called format file. The ATEF system analyzes the words and thus employs dictionaries A dictionary is a set of segments (character strings), with each of which is associated a label, a processing pointer, and a lexical unit pointer. The processing pointer specifies the particular process which must be associated with the segment.

The analysis of the input word by the ATEF system resides at first in a label processing, that is to say in an evolution of the empty label toward a final label characteristic of the analyzed word This evolution is controlled by the grammar, which at each moment has access to two labels. the label being developed (noted by the symbol C) and the label associated with the segment which was read in the dictionary (noted by A) The analysis of a word aims to produce a segmentation of the woru simultaneously compatible with the segments of the different dictionaries (the word must be an assembly of dictionary segments) and compatible with a correct evolution of the grammar Thus the segmentation of the input word is tighcly bound to the evolution of the grammar which controls the coherence of the segmentation In the course of a segmentation operation the state of the system takes into account for the analyzed word

the label resulting from the analysis of the segments already obtaıned for this word

the label associated with the segment found in a dictıonary

the remaining characters of the input word

the complete form of the input word

Thus for example in the course of the analysis of the word irresistible and after analysis of the segment "ible" and in the course of reading the segment 'resist" the following elements are obtained

C the label resulting from the analysis of "ible" This label contains for example the variable derivation with value verb-adj, the variable gender with value masculine and feminine, the variable number with value singular.

A the label associated with the segment "resist This label contains notably the lexical unit "resister", the varıable category with value verb

The characters IR

The complete form IRRESISTIBLE

The purpose of the grammar is to permit or prevent the evolution of label C starting with label A Here, the label will evolve and obtain the variable category with value adjective. A rule associated with the segment "resist" by means of its pointer will therefore describe this evolution of the label C. When no evolution of the label C is possible, the corresponding segmentation is blocked and considered nonsignificant.

The set of labels plays a fundamental role in this system and forms the set of states of the finite state transducer corresponding to the logical model of the system Each coherent segmentation of a word (a word can have several coherent segmentations leading to ambiguities) provides a labeled point in the final tree Three elements are fundamental to the system

the choice and evolution of the segmentation

the calculation of the set of labels associated with a word

the positioning of the labeled points created by the analysis of a word in the final tree

The choice and evolution of the segmentation has to do with the sequence of input characters. The segmentation forces, above all, a prior linguistic choice. Thus with the segment "UN" two possibilities can be conceived

either accept "UN E" as a coherent segmentation

or have the segment "UNE" in the dictionary and refuse the segmentation "UN E"

For each initial form several segmentations are possible to arrive at the same results and only a linguistic study of the phenomena permits a decision on the strategy to be adopted. In any event, this strategy is left to the user of the system In the course of a segmentation the system can operate directly on the nonsegmented characters in order to force them into a "canonical" form. Thus in the case of the word reel several possibilities arise to accept a word like realite

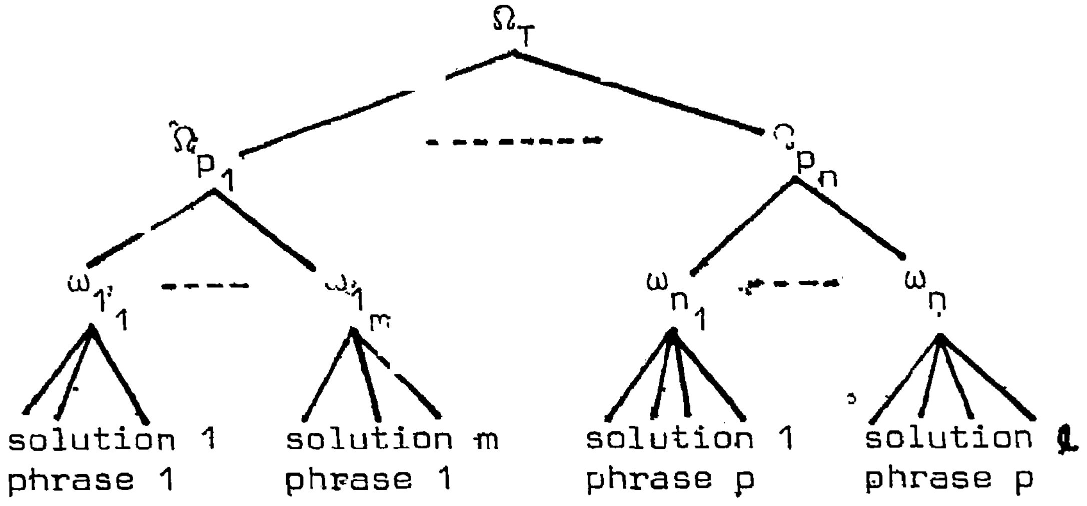
> put the segment "real" in a dictionary as well as the segment "reel", the former will generate words like réalité, irréalité, etc.
>
> put the single segment "réel" in the dictionary and the analysis of the word réalite will follow the schema
>
> réalité => 1st segment found "ité", remainder "réal"
> modification réal ->réel => 2nd segment found "réel"
> segmentation "réel ité"
>
> N B In this analysis, it is to be noted that the search for successive segments is performed from left to right for the input word. This depends on the strategy adopted and, for a given use, the direction of the segmentation of a word can be either left to right or right to left.

To avoid a proliferation of possible segmentations and therefore of possible solutions, several functions provide for intervention in the segmentation A first possibility is offered by the management of the dictionaries In fact, the system includes several dictionaries and after isolation of each segment the system can "open" or "close" a dictionary

This method makes it easy to avoid, for example, looking for two consecutive prefixes. Another mode of intervention which is more direct, is provided by the presence of functions acting on the enumeration procedures by which the system counts off solutions. For example, the system analyzes all possible segmentations starting with a given segment beginning with the segmentations containing most characters. An intervention at this level makes it possible not to analyze but to reject subsegmentations of a segment. The analysis of the segment "UNE" can, for example, reject the analysis of the subsegment "UN E" (Observe that the segmentation of the word "chacune" will then be obtained as CHAC UNE because the segmentation CHACUN E will be rejected as a subsegmentation of "UNE" This problem can easily be resolved because these functions appear in the rules of the grammar and are consequently conditional. One can at the same time forbid the subsegmentation "UN E" in the word "UNE" and authorize this segmentation in the word "CHACUNE")

The calculation of the set of labels associated with a word is produced and controled by the grammar. This calculation corresponds above all with a conditional modification of the label C or current state starting from the label A or argument state. The condition for the evolution of this label is such that if no evolution is possible then the corresponding segmentation is rejected. This condition can refer to the labels of the preceding analyzed words and can condition its result on the analysis of the following form. Thus for example in the course of the analysis of the word "LA" in the sequence "il la voit", the segmentation taking "la" as article can be rejected. The transfer of information to different labels can be realized through assignment to the following label S. When this label has been assigned in the course of the analysis of a word the analysis of the following word will begin with the assigned label instead of the null label.

The final result of the system is a labeled tree. With no supplementary specification in the course of analysis, this tree appears in the following form:



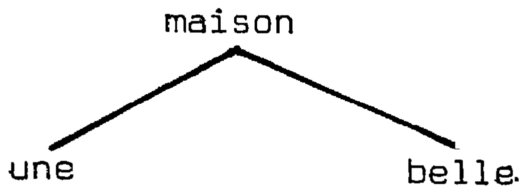
The solution for a sentence (*phrase*) consists of a string of labels (one for each word of the sentence), each of which represents an interpretation of a word of this sentence. In this case, the sentence is not structured; simply the ambiguities are separated. In the course of the analysis of the words, a first sketch of a construction can be made and give as result a more developed tree. These functions specify the position that the point to which the calculated mask applies must take in the final tree. This position is determined in all cases below a point ω_i and is relative to the root (first point on the left below ω_i) and to the rightmost point of the tree already constructed. Thus this point can become itself the root, the rightmost leaf, etc.

With, for example, the analysis of the string "une belle maison", we can have

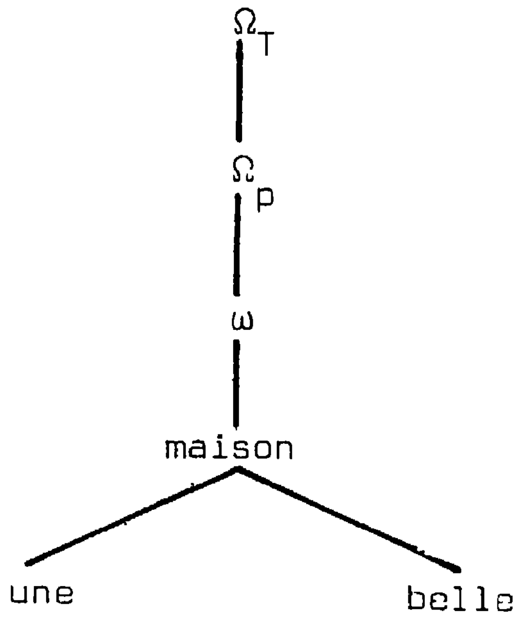
during the analysis of "une", no tree

during the analysis of "belle", the tree contains the single point "une". A function can render the point "belle" as root and give belle
une

during the analysis of "maison", if the constructed tree is belle,. a function can provide for swapping
une
the root with the occurrence in work and give the tree



In this case, the result for the system will be



5. THE CETA SYSTEM

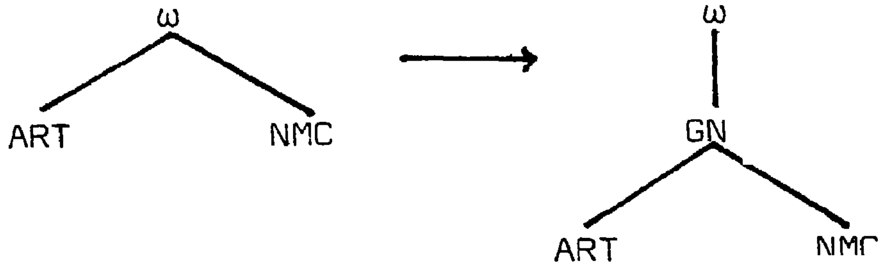
The CETA system provides for writing and simulating a transformational grammar. This system manipulates labeled trees of the type described above (labeled trees produced by the ATEF or other system). To construct a transformational grammar with this system two complementary elements are necessary:

the set of rules used defines the set of primitives of the system for a given application

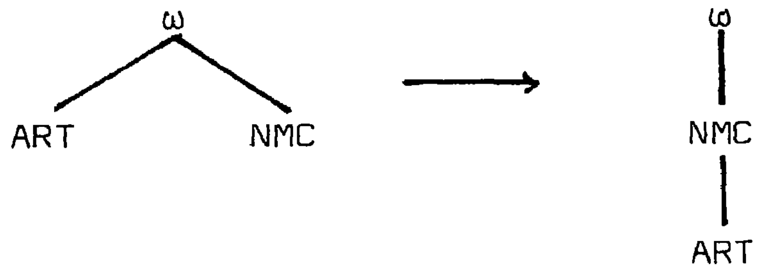
The set of grammars and the definition of their linkage defines the mode of use of the primitives

The definition of a transformation rule defines a mode of potential transformation of the tree considered. A rule is defined by a lefthand part representing the subtree to be modified and a righthand part defining the resulting subtree. For example, let the following be two rules:

R1 :

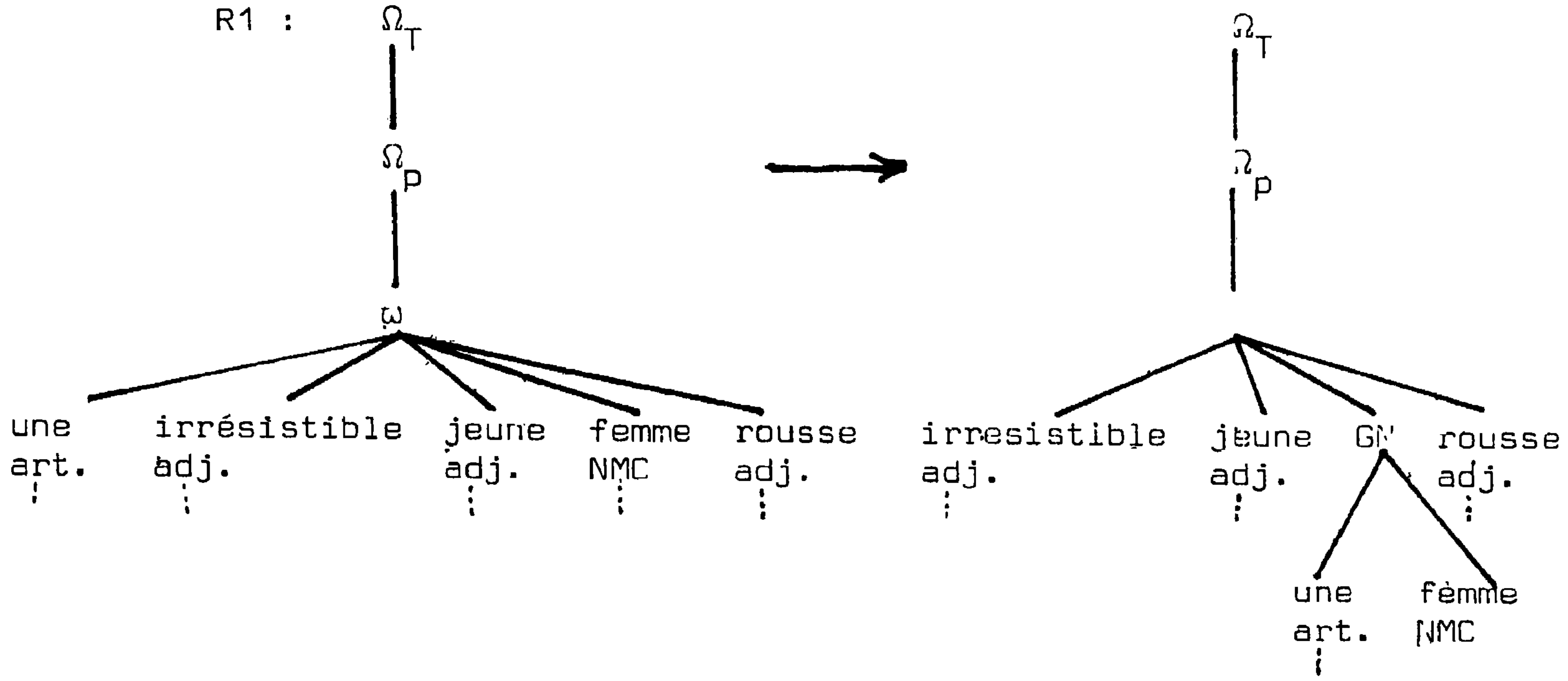


R2 :

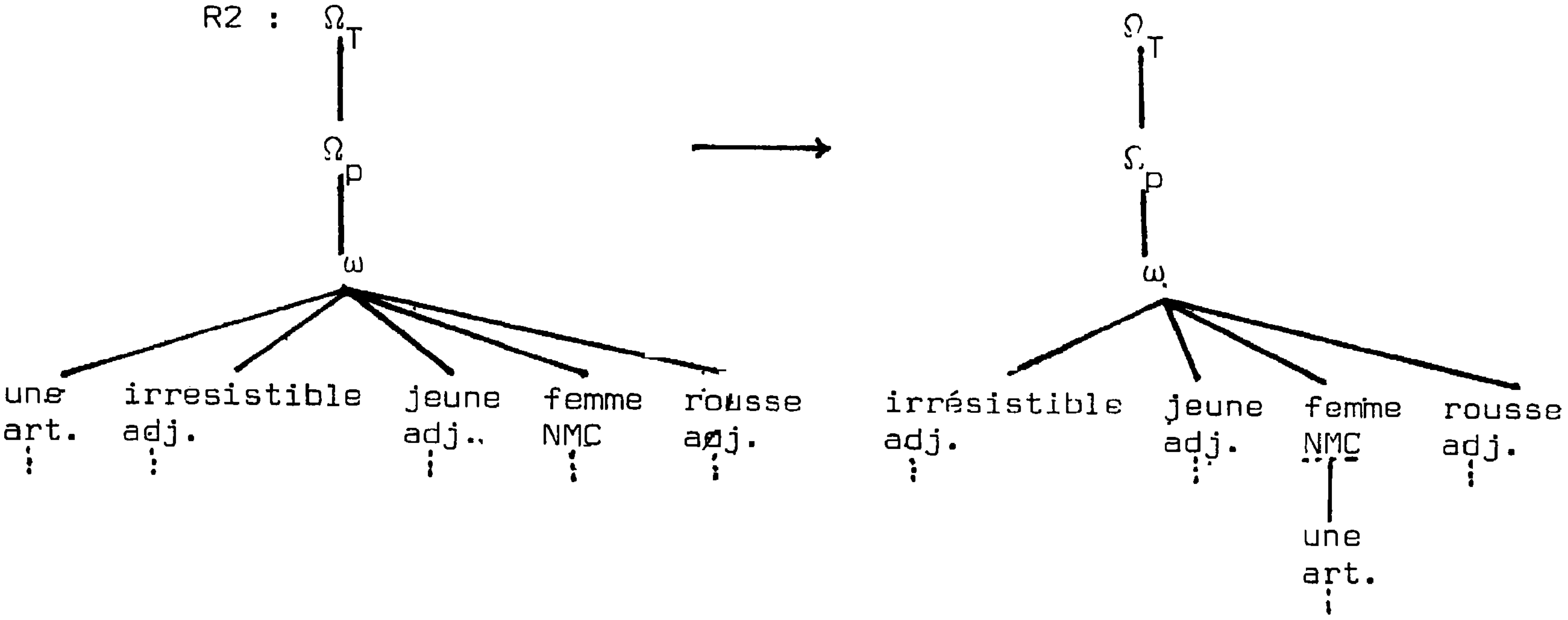


On the tree resulting from the analysis by the ATEF system of the sentence "une irrésistible jeune femme rousse", we will have the following applications:

R1 :

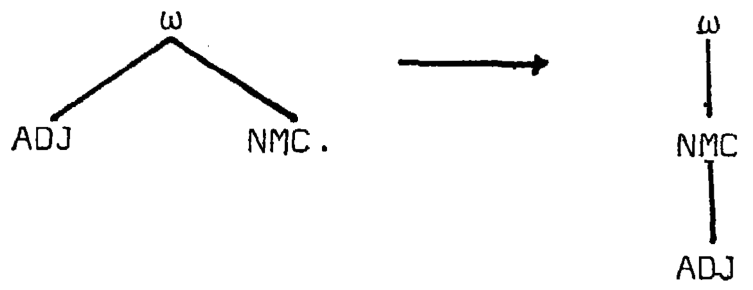


R2 :

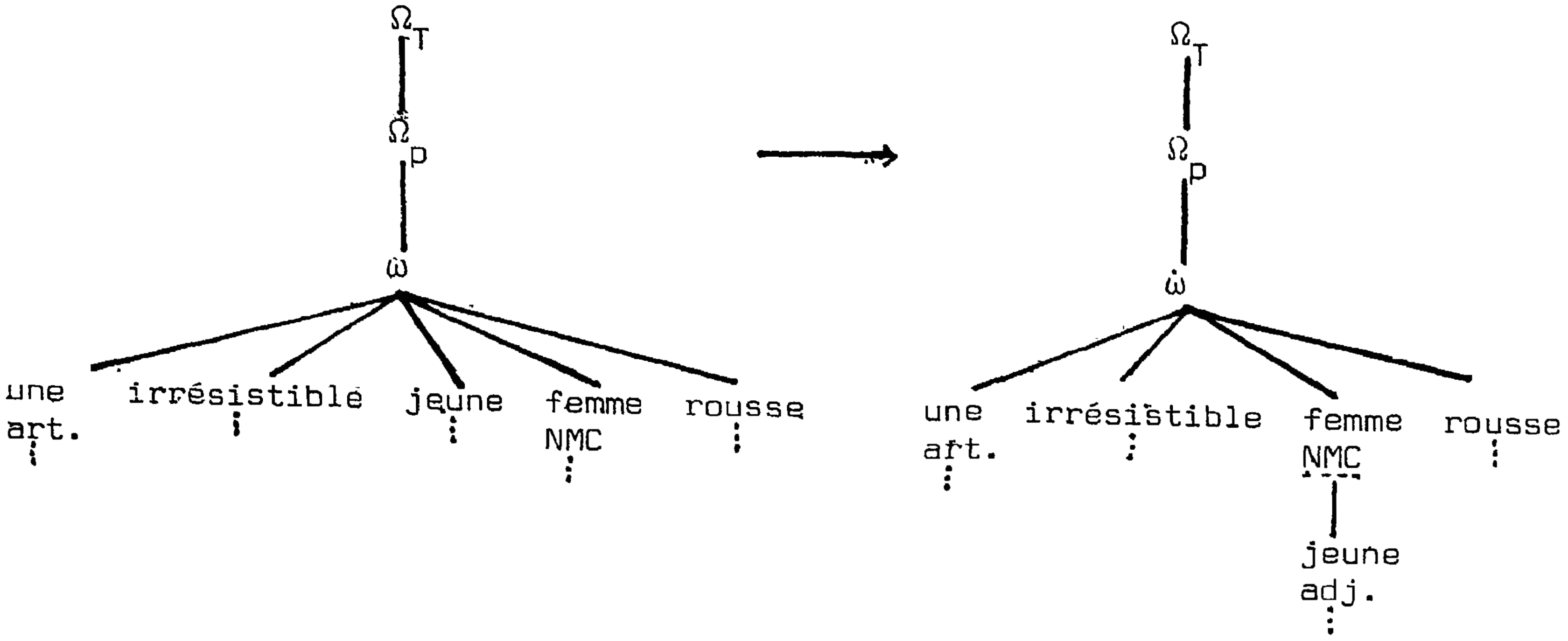


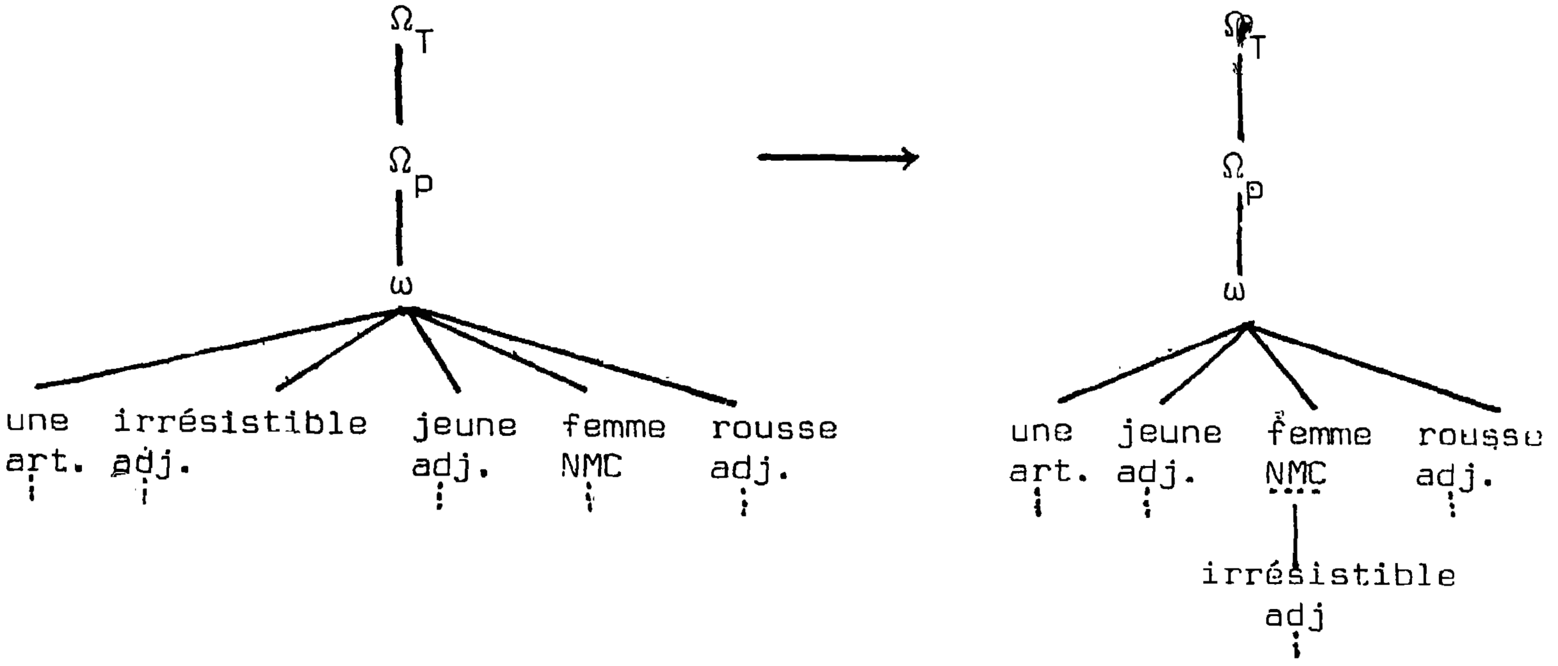
In fact, the definition of a transformation can call on a hierarchical set of subtrees. In the example taken here, the input tree is not very "deep" and most often only one-level trees are applicable. However, in the course of development of a complete structure, the considered tree is arbitrary and the definition of a complex transformation constructed beginning with several subtrees is very refined. The subtrees defined in a rule can likewise be considered ordered or unordered Let rule R3 below be considered as unordered:

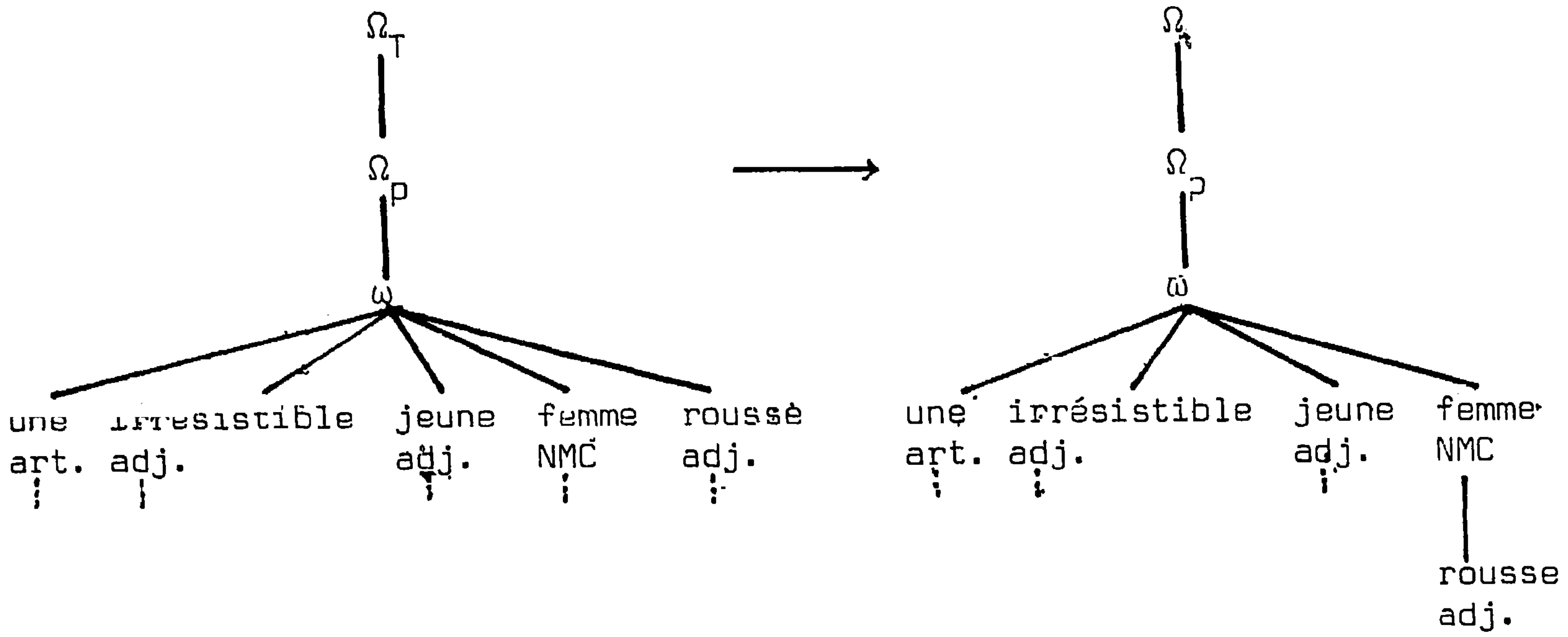
R3 :



Several applications of this rule are possible on the same input as before.



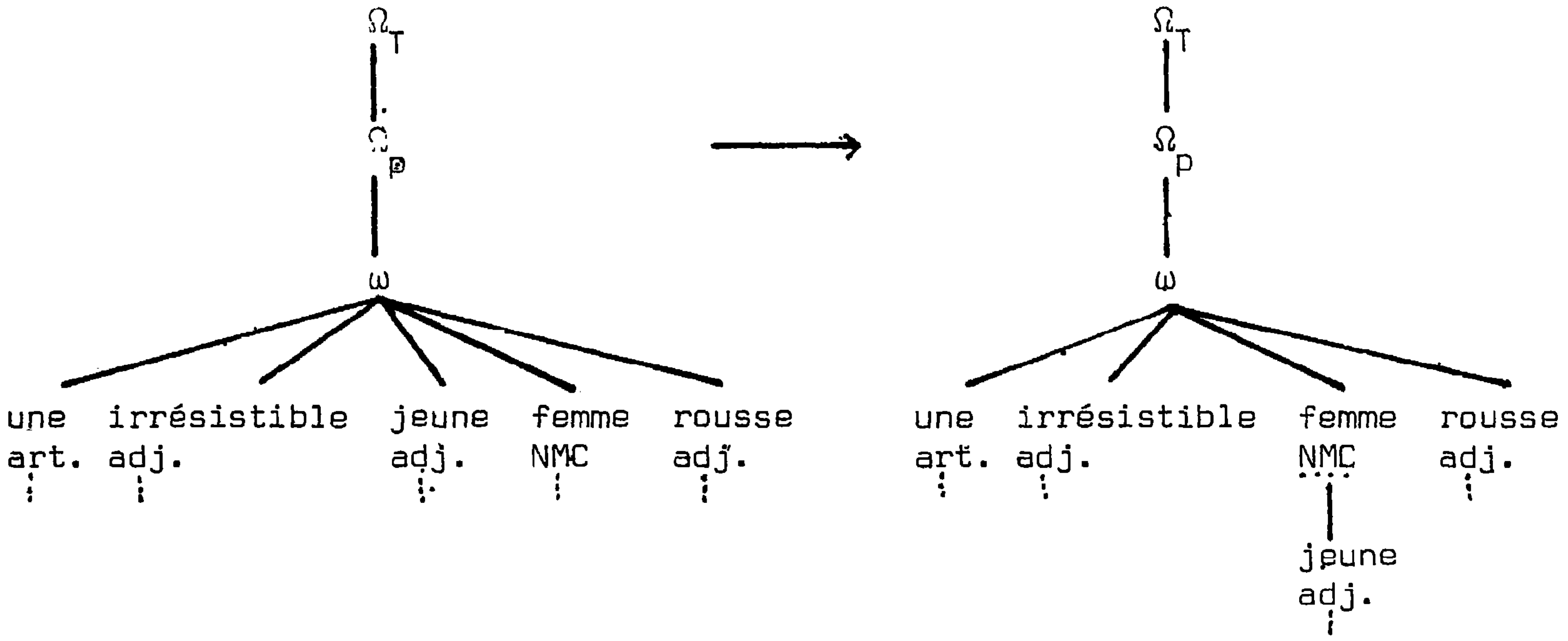




The linkage of the different rules previously described is defined by the set of elementary grammars. An elementary grammar consists of ordered rules. A rule R_i will be applied prior to an R_j if the order of R_i is less than the order of R_j An elementary grammar has furthermore a mode of execution. An elementary grammmar unitarily executable is such that its result will be obtained after an application of a part of the set of rules mentioned. (An application of the rules mentioned can cause to appear new possible applications which will not be performed in this case.) Another mode of application of an elementary grammar is exhaustive. In this mode, the set of rules of the grammar will be applied up to the maximum but the application of a given rule has the effect of eliminating it from this elementary grammar. (That is, for a given point.) With this second mode of application, the number of possible steps for a given tree is always finite. Within an elementary grammar which is unitarily or exhaustively executable, the presence of recursive rules makes it possible to obtain complex constructions by simulating repetitive procedures. A recursive rule is characterized by a call to a new grammar (which can obviously be the same as that in which the recursive rule is found). The result of the application of a recursive rule consists of the tree obtained after transformation by the called grammar of the tree transformed by the rule in question.

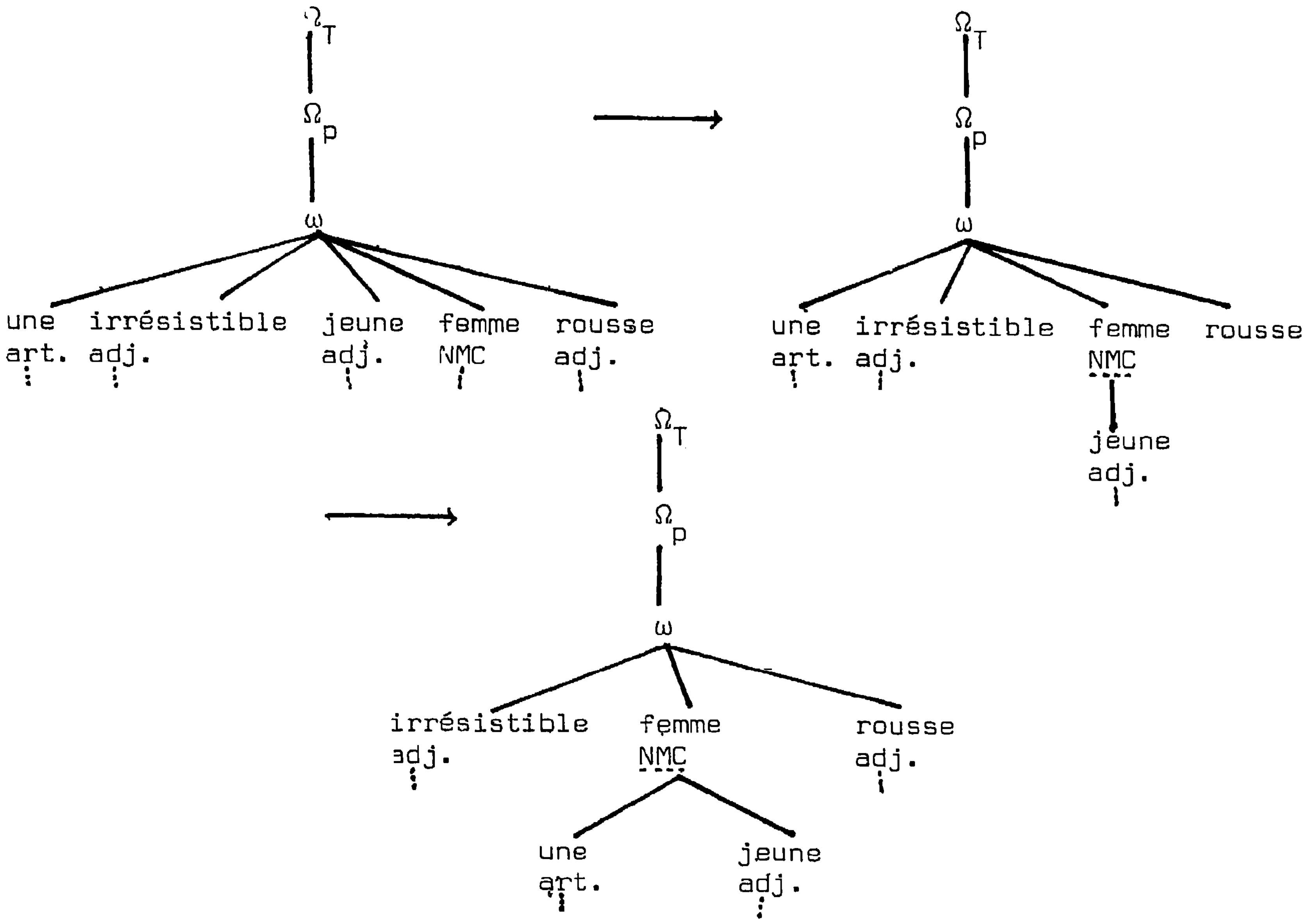
For example, let R3 and R2 be the rules previously described. The elementary grammar G consisting of these two rules will furnish as result,

In unitary mode: application of R3. (Priority is given by the order of enumeration of the rules.)

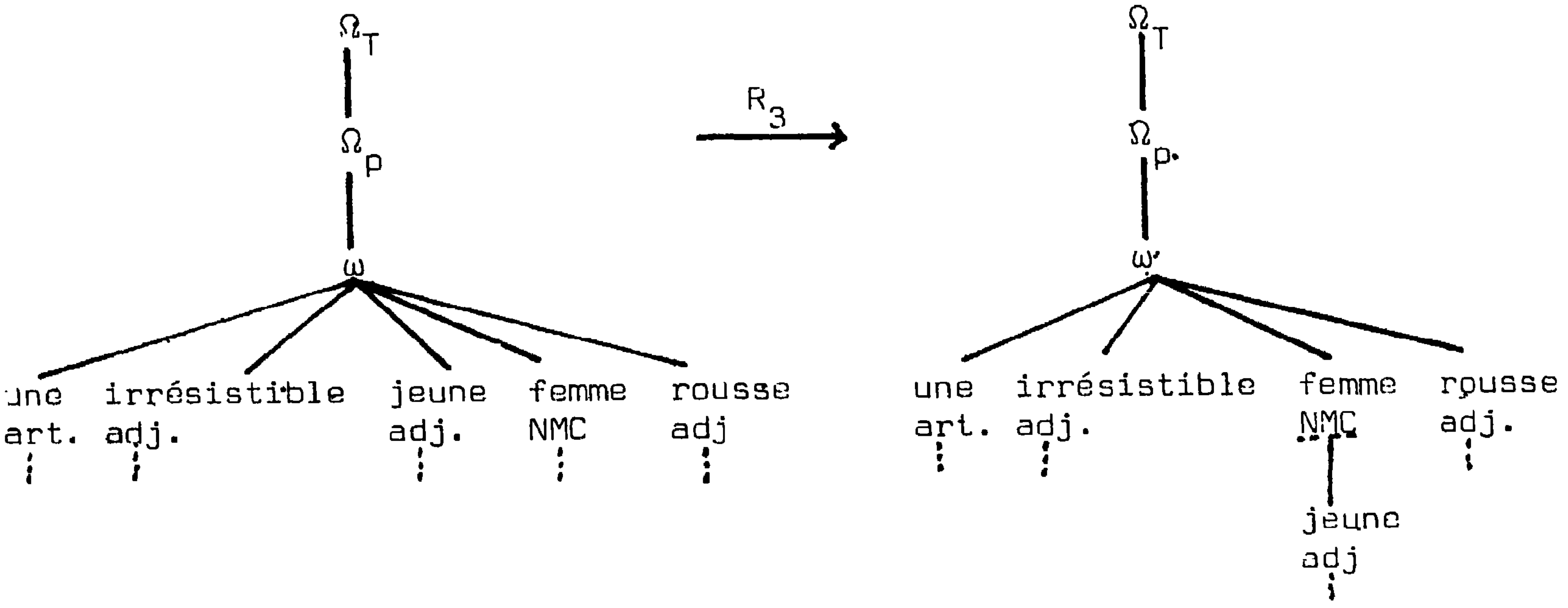


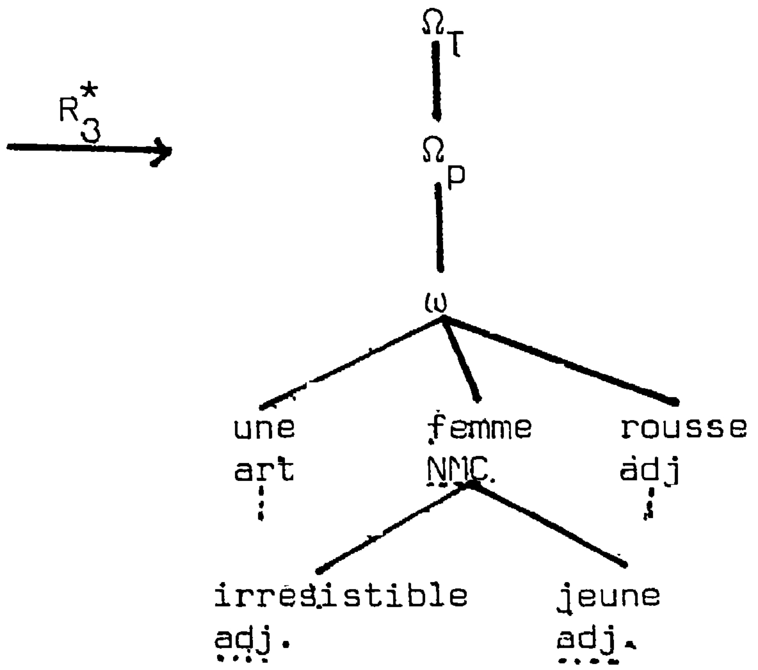
In exhaustive mode: application of R3 then R2·

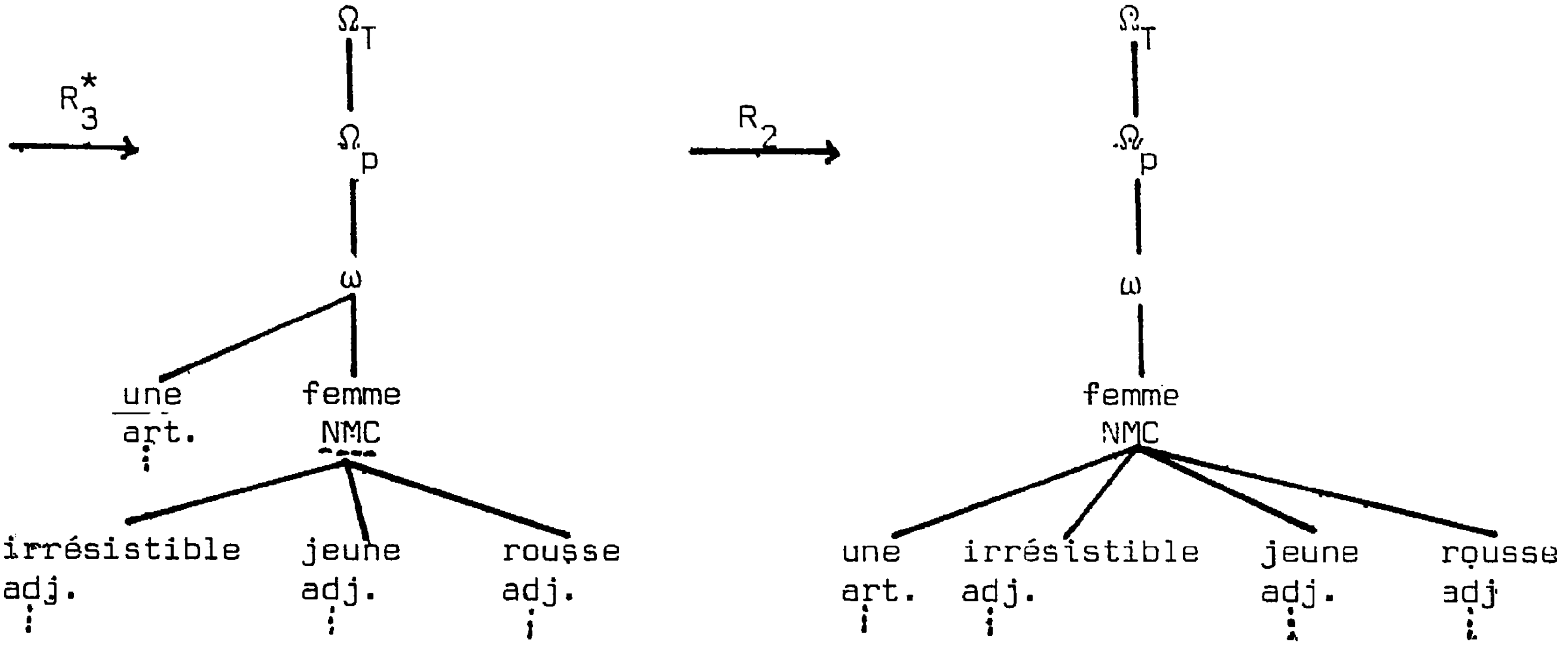
(figure at top of next frame)



With Grammar G' containing rules R3 and R2, but specifying that R3 must be recursive and call G', we have:



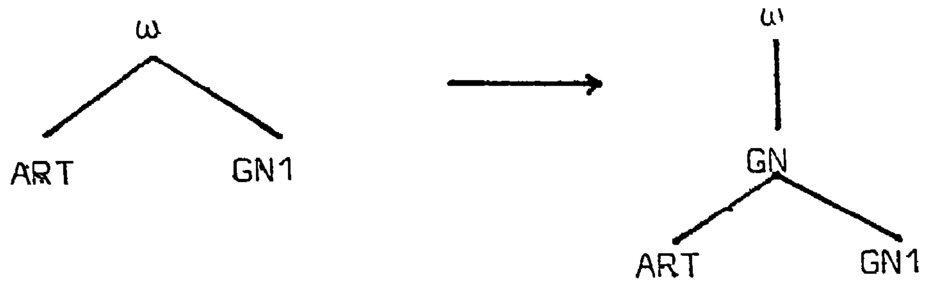

R*3
ΩT
Ωp
ω
une
art
femme
NMC.
rousse
adj
irresistible
adj.
jeune
adj.


R*3
ΩT
Ωp
ω
une
art.
femme
NMC
irrésistible
adj.
jeune
adj.
rousse
adj.
R2
ΩT
Ωp
ω
femme
NMC
une
art.
irrésistible
adj.
jeune
adj.
rousse
adj

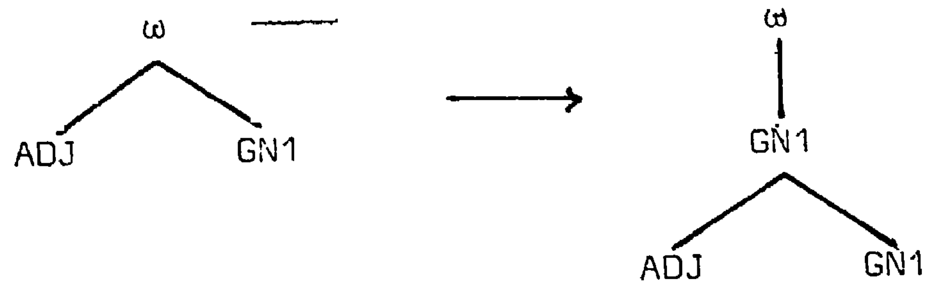
Application of rule R3* corresponds to the recursive call of this rule, terminating when the rule is no longer applicable.

With rules R4, R5, and R6, the construction is

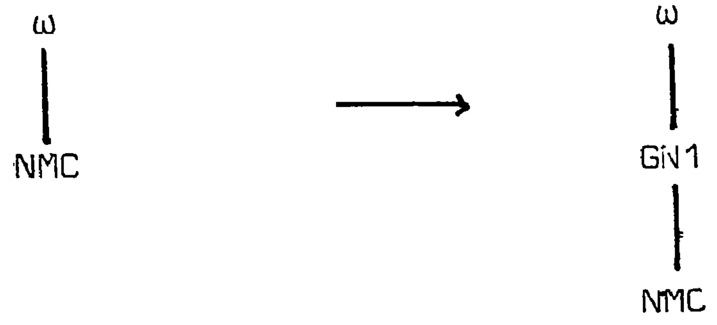
R4 :

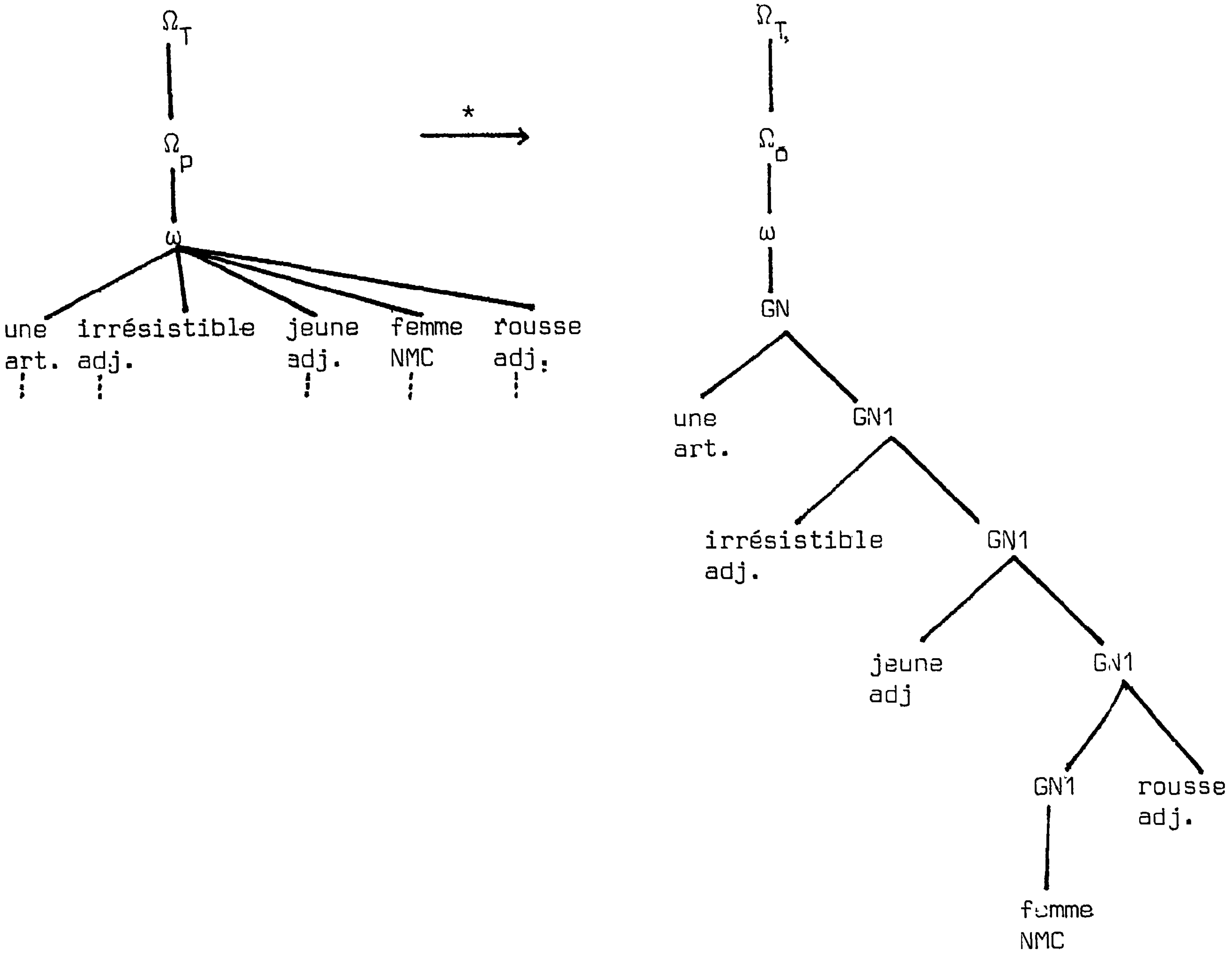


R5 :



R6 :





The definition of a CETA grammar consists of a set of elementary grammars and a conditional linking procedure over them. The linking must be such that the corresponding graph is loop-free. An elementary grammar from which no linking is possible yields as result the input tree in place of the transformed tree. This procedure permits one to obtain a method of analysis involving several criteria of acceptance, each consisting in the presence of a tree schema in the terminal tree.

REFERENCES

Chauche, J. P. Guillaume, and M. Quezel-Ambrunaz. Le systeme A.T.E.F. Internal document, G.E.T.A. December 1972.

Chauche, J. Arborescence et transformation. Thesis, Grenoble. December 1974

Chauche, J. Presentation du systeme C.E.T.A. Internal document, C.E.T.A. January 1975.

Translation of a text prepared for the First National Conference on Computational Linguistics, Varna, May, 1975.