RUSSIAN-FRENCH AT GETA : OUTLINE OF THE METHOD AND DETAILED EXAMPLE

Ch. BOITET and N. NEDOBEJKINE
GETA, UNIVERSITY OF GRENOBLE
F-38041 GRENOBLE-CEDEX 53

Introduction

The original version of this paper is very detailed. Space limitations for publication in COLING's proceedings have forced us to reduce it by a factor of five. The more detailed version has been proposed for publication in "Linguistics".

This paper is an attempt to present the computer models and linguistic strategies used in the current version of the Russian-French translation system developed at GETA, within the framework of several other applications which are developed in a parallel way, using the same computer system. This computer system, called ARIANE-78, offers to linguists not trained in programming an interactive environment, together with specialized metalanguages in which they write linguistic data and procedures (essentially, dictionaries and grammars) used to build translation systems. In ARIANE-78, translation of a text occurs in six steps : morphological analysis, multilevel analysis, lexical transfer, structural transfer, syntactic generation, morphological generation. To each such step corresponds a computer model (non-deterministic finite-state string to tree transducer, tree to tree transducer,...), a metalanguage, a compiler and execution programs. The units of translation are not sentences, but rather one or several paragraphs, so that the context usable, for instance to resolve anaphores, is larger than in other second-generation systems.

As ARIANE-78 is independent of any particular application, we begin by presenting its main features in Part I. Some of them are standard in second-generation systems, while others are original. Among these, we insist on the multilingual aspect of the system, which is quite unique, on the very powerful control structures embodied in the supported computer models (non-determinism, parallelism, heuristic programming), and on its interactive data-base aspect.

In the second and larger part, we successively describe each step of this Russian-French application. We first present the underlying computer model (there are 4 of them, as the second, third and fourth step use the same one), then the organization of the linguistic data. A small text is used throughout the text as a standard example. Examples of translations of larger texts appear at the end.

I - Current GETA translation system

The computer system ARIANE-78, together with appropriate linguistic data, constitutes a multilingual automatized translation system.

The system is a rather sophisticated second generation system. It relies on classical as well as more original principles.

1. Classical second-generation principles

Intermediate structures

The process of translation of a text from a "source" language in a "target" language is split up into *three main logical steps*, as illustrated below : *analysis, transfer and generation*. The output of the analysis is a "*structural descriptor*" of the input text, which is transformed in an equivalent structural descriptor in the target language by the transfer phase. This target structural descriptor is then transformed into the output text by the generation phase. Essential in our conception is the fact that analysis is performed independently of the target language(s). The "deeper" the analysis, the shorter the distance between the two structural descriptors. Ideally, one could imagine a "pivotal" level, at which they would be the same.

In the past, Pr. Vauquois' team tried a slightly less ambitious possibility [Vauquois, 1975], namely to use an "hybrid" (Shaumjan) pivot language, where the lexical units are taken from a natural language, so that the transfer phase is reduced to a lexical transfer, without any structural change. As it is not always possible, or even desirable, to reach this very abstract level, one may choose not to go all the way up the mountain and to stop somewhere in the middle. This is why we call our structural descriptors "*intermediate structures*". Note that ARIANE-78 imposes nothing of that kind, both extremes are still possible, and in fact the linguistic teams have agreed on "multilevel" intermediate structures which contain very deep as well as low level types of information, ranging from logical relations to traces (see details below).

Separation of programs and linguistic data

The second classical principle is to offer metalanguages, in order to keep the particular linguistic data (grammars, dictionaries) separated from the programs.

For instance, dictionary look-up is a standard function, which should not be modified in any way when a new language is introduced in the system. This separation also corresponds to a division of work and enhances transparency : dictionary look-up may be optimized by the programmers without the linguistic users ever being aware of it. The same goes for more complex functions, like pattern-matching in tree manipulating systems. In these metalanguages, linguists work directly with familiar concepts, like grammatical variables, classes, dictionaries and grammars. The grammar rules are rules of some formal model (context free, context sensitive, transduction rules). That is, one may also consider such metalanguages as very high level algorithmic languages offering complex data types and associated operators. Although this principle of separation has been criticized as imposing too much "rigidity" on the users, critics have failed to understand that this is only the case when the metalanguages are not adequate. A good comparison may be found in classical programming, where for example, the compiler and run-time package of PL/I is separated from programs written PL/I in exactly the same sense.

Semantics by features

The third classical principle touches *semantics*. In a second-generation MT systems, semantics may be only expressed by the use of *features* (concrete, abstract, countable,...), which are exactly like grammatical features. The theoretical framework is the one of a formal language, with a syntax describing the combination rules of the language units. There is no direct way, for instance, to relate two lexical units. In order for this to be possible, there should be a (formalized) domain, possibly represented as a thesaurus, and rules of interpretation. However, this limitation may be partially overcome in ARIANE-78's lexical transfer step. Remark also that semantic features may be extremely refined for some limited universe, and give surprisingly good results [TAUM-METEO, 1975].

2. Principles proper to GETA's system

We relate them to the three main principles exposed above.

Intermediate structures

In ARIANE-78, we split up each of the three main phases into two steps. This is essentially for algorithmic as well as for linguistic reasons. Morphological analysis, lexical transfer and morphological generation are undoubtedly very much simpler than the order steps, and it has seemed reasonable and linguistically motivated to keep them separate and to use simpler algorithmic models to realize them. However, this could not be the case in other environments, for example if the input would be very noisy (oral input).

ARIANE-78 uses a unique kind of data-structure to represent the unit of translation from morphological analysis to morphological generation, namely a *complex labeled tree structure* : each node of such a tree bears a value for each of the "grammatical variables" used in the current step.

GETA's system is *multilingual* by design : an analysis cannot explicitly use information from the target language, and generation is likewise independent of the source language. Moreover, in a given user space, ARIANE-78 ensures the coherence of the linguistic data written to construct a multilingual application.

Computer environment

The principle of separation of programs and linguistic data is strictly observed in our system. An additional feature is to propose *several algorithmic models* designed to be of maximal adequacy and generality as well as of minimal computational complexity.

Functions of an integrated MT system include preparation of the linguistic data, management of the corpora and execution of the linguistic data over texts. ARIANE-78 provides a *conversational environment* for these functions, hiding implementation chores to the user. It also includes a *specialized data-base management system* for the texts and the linguistic files.

Semantics

Semantic features may be declared as normal grammatical features in each step. At lexical transfer, the linguist may relate several source and target lexical units, these relations being elaborated in the succeeding structural transfer phase. This is however certainly not sufficient to call the system "third generation".

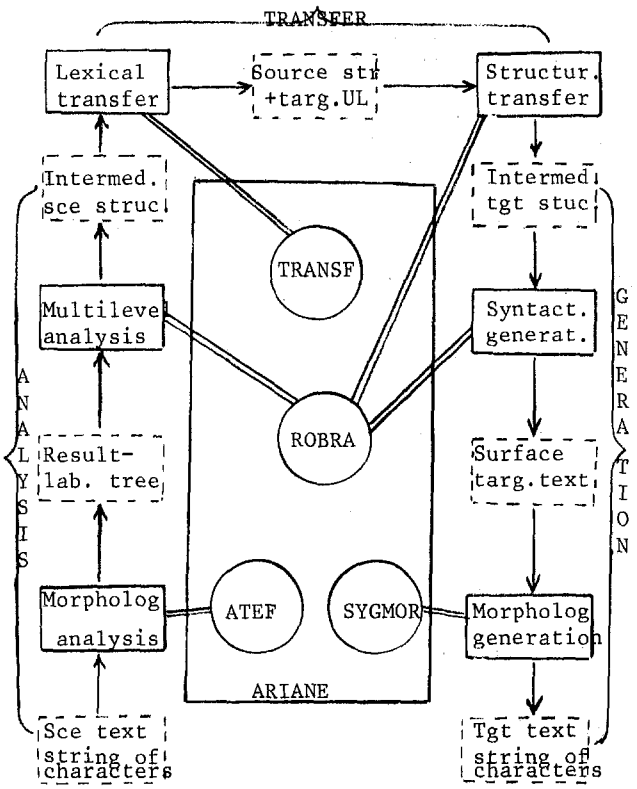
3. Organization of the translation process

Overall schema

The schema below shows the different steps of the translation process. The components of ARIANE-78 implementing the 4 different algorithmic models appear within circles, they are linked by double lines to rectangles corresponding to the linguistic data written in the associated metalanguage for the indicated step. Simple arrows indicate the flow of control.

Organization of a step

In each step, the linguistic data may be of four kinds : *grammatical variables* (like gender, number, semantic type), *classes*, describing useful combinations of values of variables, *dictionaries* and *grammars*, containing the rules and the strategy to use them.



They are expressed in a *metalanguage*. Their syntax and cohenrency is first checked by the corresponding compiler, which generates a compact intermediate code. At run-time, this code is interpreted by standard "execution programs". This approach separates the linguistic and algorithmic problems, and makes debugging and maintenance much easier.

The complete system is operational on IBM compatible machines under VM/CMS. ARIANE is the name of the interactive monitor interfacing with the user.

For more explanations about our terminology and our "intermediate structures", see [15, 22, 23].

II - An application to Russian-French translation

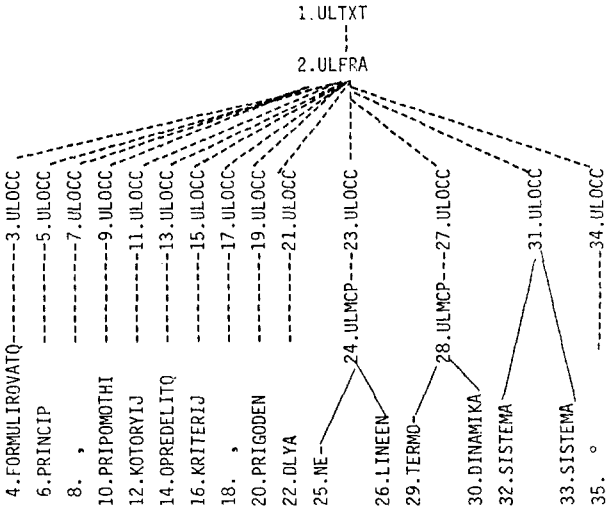
We will use a small size text as our standard example. Note that usual translation units are not sentences, but rather paragraphs. We use an unambiguous latin transcription.

Input text

*SFORMULIROVAN PRINCIP, S POMOTHQYU KOTOROGO OPREDELYAETSYA KRITERIJ, PRIGODNYIJ DLYA NELINEJNOJ TERMODINAMIKHESKOJ SISTEMYI.
(A principle has been defined, with whose help one defines a criterion useful for the non linear thermodynamic system).

1. Morphological analysis

The grammar, classes and dictionaries are written in the ATEF formalism [1, 8, 10, 19]. The strategy of the analyzer has been described in [16]. Its output is a "flat tree" with standard structure and with leaves labelled by the masks of variables computed by the analyzer.

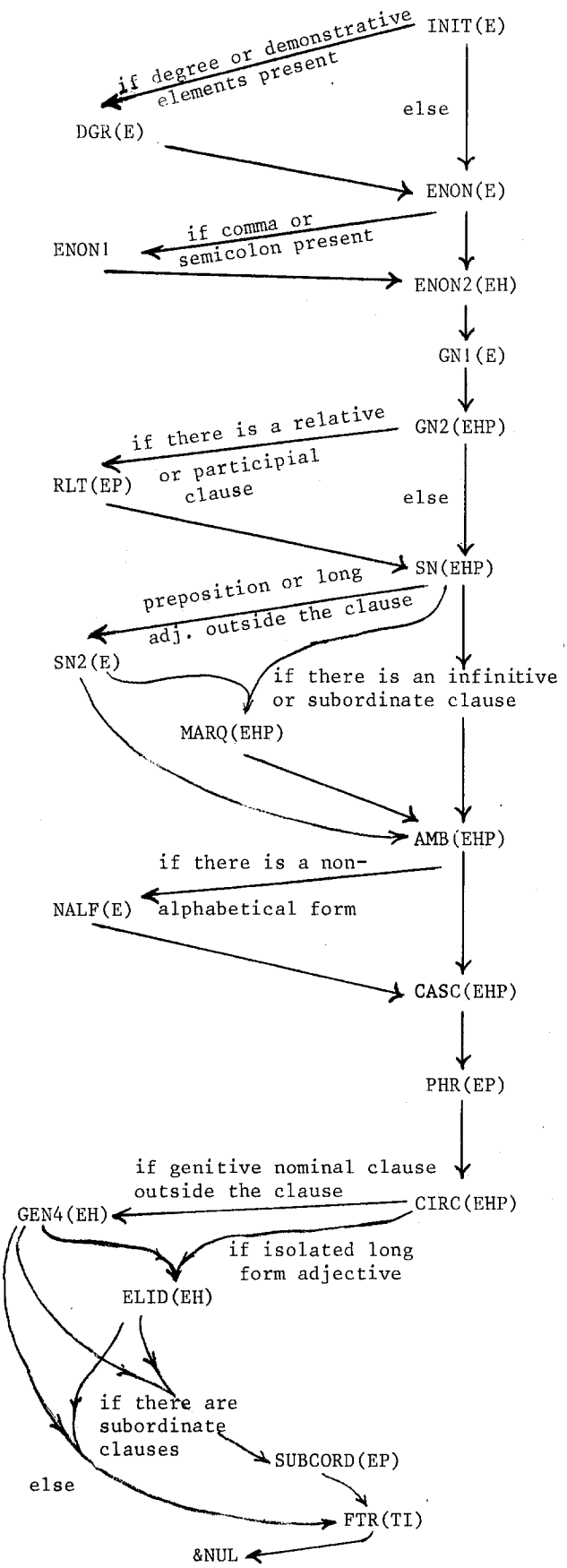


2. Multilevel analysis

This part is the most difficult. It is written in ROBRA [5, 6, 7, 8, 12], a general tree-transducer system. In order to build a whole *transformational system*, the linguist writes *transformational rules* (TR) and groups them in *transformational grammars* (TG). When a TG is applied to an object tree, all compatible occurrences of its TR are executed in parallel. The overall flow of control is described in the *control graph*. Using a built-in backtracking algorithm, ROBRA finds the first possible traversal of the control graph leading to an exit (&NUL symbol), thereby applying each traversed TG to the object tree.

Rules are grouped in grammars when they correspond to related linguistic phenomena, or when they express transformations used for a certain logical step of the linguistic process (here, multilevel analysis) or, more strategically, when they share the same execution modes (e.g., iterative rules will appear in "exhaustive" grammars, others in "unitary' grammars. This architecture makes it possible to limit the interaction between rules and avoid many combinatorial problems, to develop strategies and heuristics, and to test and modify TGs separately (different trace parameters may be associated to each TG).

Let us now give the control graph used in multilevel analysis of Russian, with some comments.



INIT is the first grammar, and is iterative (E). Its aim is to homogenize and to simplify the input tree.

DGR is used only when there is an analytic expression of degree, to represent it synthetically (NG variable).

ENON-ENON1-ENON2 : these 3 grammars break down the sentences into textually marked "utterances". Commas, unambiguous conjonctions and relative pronouns ... are used.

GN1 builds simple nominal groups like Adj + N or Prep + N or Num + N.

GN2 looks for further elements in the nominal groups, and solves certain ambiguities.

RLT looks for the nominal antecedents of relative and participial clauses constructed by ENON2.

SN searches for a personal verb as main element of the utterance, and for verbal modifiers, like negative and conditional particles or auxiliaries.

SN2 tries to solve the adverb - short form adjective ambiguity and builds embedded nominal groups.

MARQ builds all types of subordinate verbal and infinitive clauses. It further tries to solve the previous ambiguity.

AMB searches for the most important terms of the clause (subject, object, near dative), thereby resolving ambiguities between subject and object, adjectives and adverbs, etc.

NALF treats non-alphabetical forms as appositions or verbal complements.

CASC handles all genitive imbrications, by (provisionally) attaching dominated groups to non-ambiguous groups.

PHR marks all strongly governed groups subordinated to the utterance with logical relations as agent, patient, attribute... If possible, this is also done on dependent groups.

CIRC and GEN4 realize the distribution of prepositional and genitive nominal groups between their noun heads, according to several syntactic and semantic criteria.

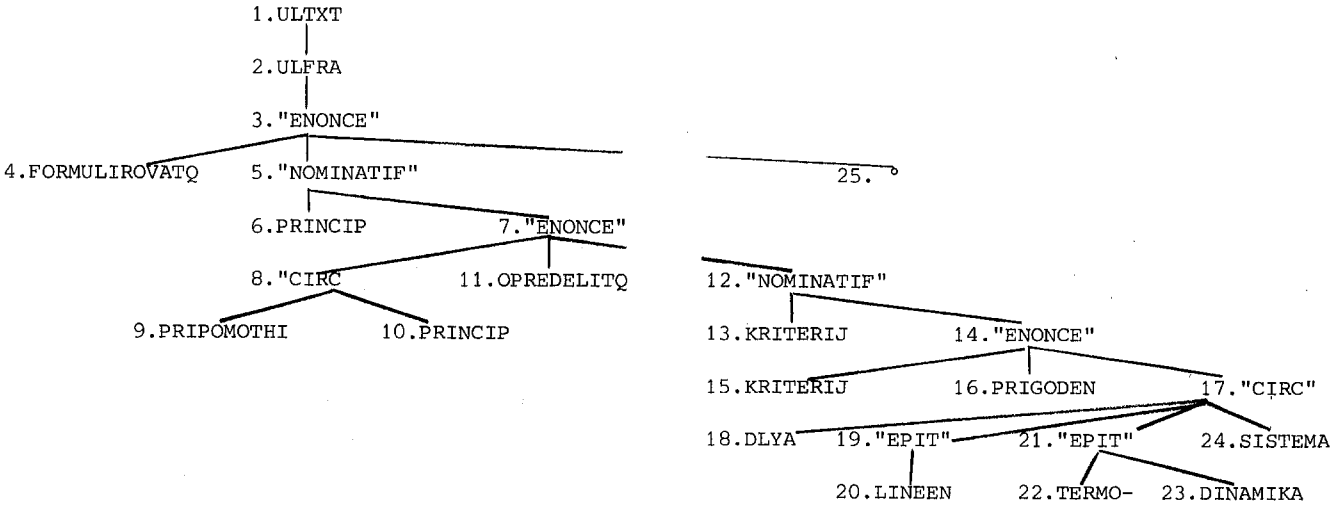
ELID searches for antecedents of pronominal expressions and isolated adjectives, and builds noun groups by copying the lexical unit of the antecedent. If the elliptic element is not a personal pronoun, it becomes qualifier or determiner according to its syntactic class. The syntactic and logical functions of the new group are computed.

SUBCORD is purely tactical (modification of the hierarchy of certain subordinate clauses.

FTR copies certain information from non-terminals onto terminal "head" nodes, to prepare for lexical transfer.

We give now the result of the multi-level analysis of our standard example. Note that node 5 (noun group with head node 6 "PRINCIP") has correctly been given syntactic function subject and logical relation patient (A2). Syntactic functions of non-terminals appear as auxiliary lexical units (UL).

Remark the anaphoric resolution on node 13 ("whose"), on which the UL of the antecedent (PRINCIP) has been copied. Node 13 has been generated in place of the absent noun. The nodes with "ULO" are strategical delimiters of utterances generated at the beginning of the analysis.




```
Node 3: K(AQ),MD(PRT),K1(PH),A(P),T(PAS),FM(FOC),      G(M),N(S),P(3),RF(PF),ABS(A2,SJ),CP1(ACC)
     4:  LX(GOV)
     5: K(NM),K1(GN),AG(A2),G(M),N(S),P(3),            MRQ(RELAT)
     6:  LX(GOV)
     7: K(VB),MD(VRB),K1(PH),A(I),T(PRE),AG(A6),       G(M),N(S),P(3),RF(R),ABS(A2,SJ),CP1(ACC)
     8: K(NM),K1(GP),G(M),N(S),ANF(RLT)
     9: K(PP),FT(PP)
    10:  LX(GOV)
    11:  LX(GOV)
    12: K(NM),K1(GN),AG(A2),G(M),N(S),P(3),            MRQ(RELAT)
    13:  LX(GOV)
    14: K(AQ),K1(MD),AG(A6),FM(FOL),G(M),N(S)
    15: K(NM),ANF(RLT),FT(DEB),G(M),N(S)
    16:  LX(GOV)
    17: K(NM),K1(GN),G(F),N(S)
    18: K(PP),FT(PP)
    19: K(AQ),MD(ADJ),K1(GA),FM(FOL),NG(NE),           G(F),N(S)
    20:  LX(GOV)
    21: K(AQ),MD(ADJ),K1(GA),FM(FOL),                  G(F),N(S)
    22: K(AV),LX(PX)
    23:  LX(GOV)
    24:  LX(GOV)
    25: K(VG),FT(FIN)
```

3. Lexical transfer

Lexical transfer is written in TRANSF. It essentially includes a bilingual multichoice dictionary of "transfer rules" accessed by the UL. Each rule is a sequence of 3-uples (condition, image subtree, assignments), the last condition being empty (true).

The automaton traverses the input in preorder, creating the object tree as follows. The UL of the current node is used to access the dictionary. The first triplet of the item whose condition is verified is chosen. The image subtree (generally consisting of only one node) is added to the output, with values of variables computed by the assignment part.

Hence, the output tree is very similar to the input tree. The possibility to transform one input node into an output subtree may be used to create compound words or to create auxiliary nodes used in the following step (structural transfer) to treat idioms.

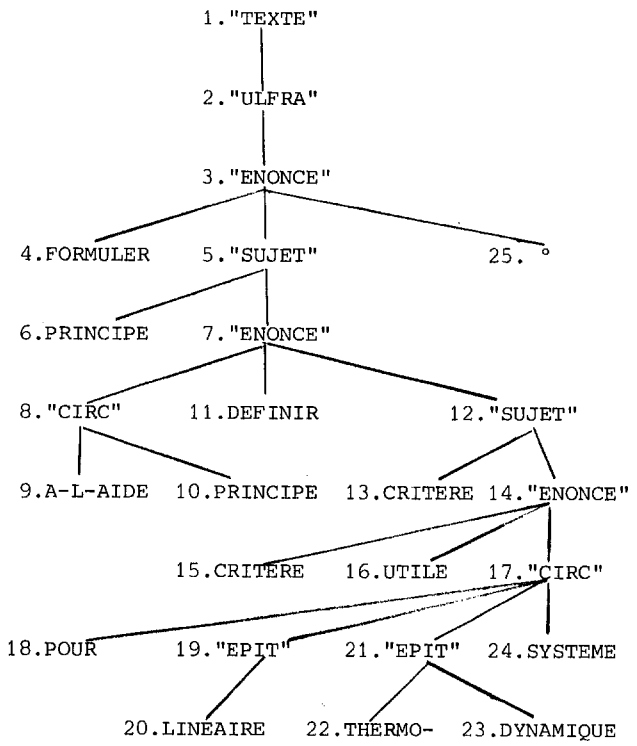
As this model is algorithmically very simple, it is the only one where no trace is provided. The example below gives an idea of the metalanguage of the dictionary.

```
'FORMULIROVATQ' == /      /'FORMULER'    ,+VBF1,
                                          $RFPF.
'PRINCIP'       == /      /'PRINCIPE'    ,$NMAS.
'PRIPOMOTHI'    == /      /'A-L-AIDE'    ,+MPCD.

'NAPRIMER'      == /O(1,2)/O:'*LOCF'     ,+VIDE ;
                           1:'PAR'       ,*PP ;
                           2:'EXEMPLE'   ,*NMMS.
```

"O(1,2)" describes the image subtree for "NAPRIMER". The other ones are reduced to one node (default). "+VBF1" says that the non-null values of variables in format VBF1 will be copied into the target node. RFPF is an assignment procedure. "*PP" says that all variables of format PP (except the UL) will be copied onto node 1.

The following structure is the result of this step on our standard example.

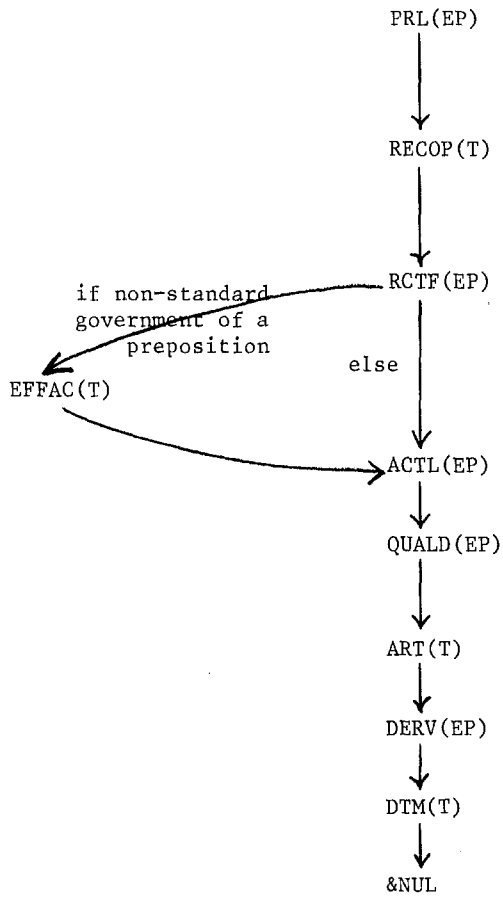



```
Node 4: KF(VB),SXF(ION),RFL(RF3)
     6: KF(NM)
     9: MPC(DE)
    10: KF(NM)
    11: KF(VB),SXF(ION)
    13: KF(NM)
    15: KF(NM)
    16: KF(AQ),SXF(ITE),PRG(AJQ),NGF(IN)
    20: KF(AQ),SXF(ITE),PRG(AJQ)
    23: KF(AQ),PRG(AJQ)
    24: KF(NM),G(M)
```

4. Structural transfer

The algorithmic component used in this step is again ROBRA, which has been very briefly presented in 2. The aim of this step is to realize all transformations of contrastive nature, so as to produce the desired intermediate target structure as output.

The following gives the control graph of the TS written for this step in the current version of our translation system.



PRL handles idioms, predicted in lexical transfer by generating auxiliary subtrees. It checks whether predicted idioms are present and takes appropriate action.

RECOP copies certain information (required mode, type of adjective, postponed preposition inversion of arguments) from terminal "head" nodes onto their fathers.

RCTF handles non-standard government, particular uses of "DE", erases some prepositions, takes care of passive-active transformations, etc.

EFFAC erases remaining auxiliary nodes generated in TL (idioms, non standard prepositions).

ACTL handles particular idiom translations, like "ESLI + Inf" → "SI ON + Present", etc.

QUALD handles actualization and qualifiers (modes, tenses, determination...), and generates the correct order in nominal groups.

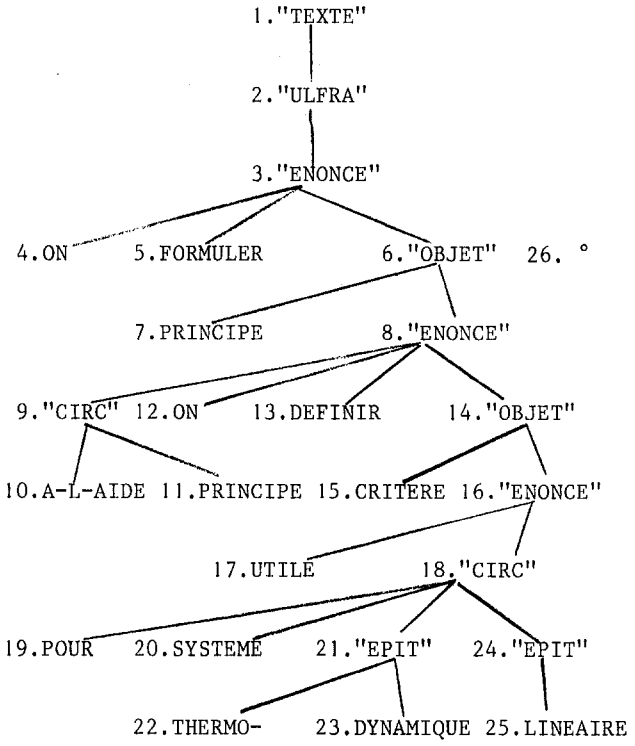
ART uses the remaining designators to compute the determination of nominal groups.

DERV handles derivations (-ANT, -EUR, -ITE, etc.), negation (NON, PEU, IN...), prefixes and others.

DTM makes the final computation of determination of noun groups.

As we see, structural transfer is relatively simple in this version. However, many improvments are planned in our future version.

The result of this step is given below. Note the modification of order in the last nominal group, as well as the generation of the impersonal "ON".



Nodes 4,12: NBR(SIN),TPN(SJA),G(M),P(3)

5: TF(PRE),MF(IND),NBR(SIN),AF(1),RF(N)

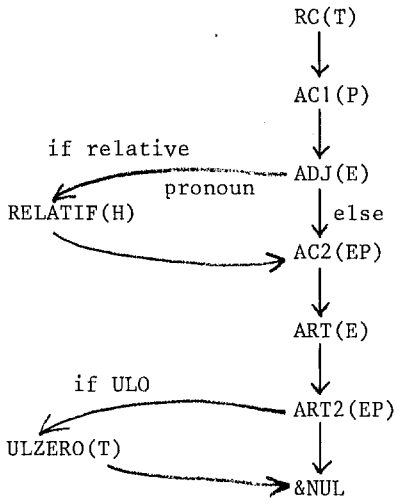
6,14,18: ART(DEF)

13: TF(PRE),MF(IND),NBR(SIN),RF(N)

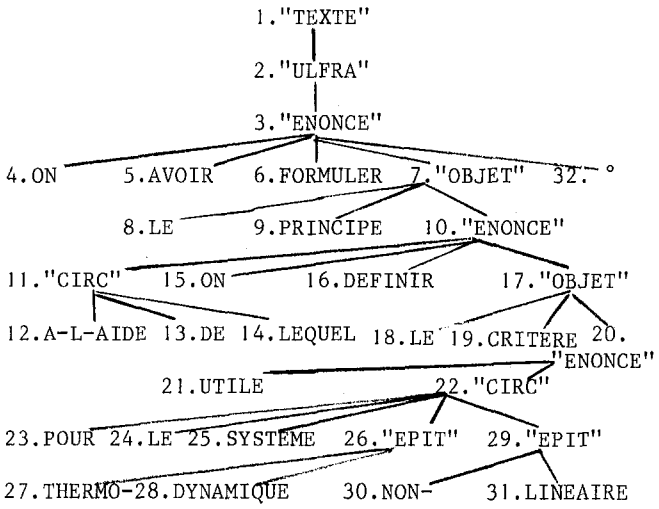
25: NGF(NON)

5. Syntactic generation

ROBRA is also used in this step as algorithmic component. The aim of this step is to produce a tree structure where the terminal nodes contain all the information necessary for generating the output text, and to give the final surface order of the words. This is a constraint imposed by the nature of the algorithmic component SYGMOR, used in the last step.



RC copies variables from head nodes onto their fathers, and checks for number and gender correctness. AC1 handles noun coordination, place of subject, and generation of preposition before infinitive, or of periphrases. ADJ handles agreement in gender and number between nouns, adjectives and articles. RELATIF chooses the relative pronoun (DONT, QUI, LEQUEL). AC2 handles homographs and noun ellipses. ART generates the correct article (UN, LE), and ART2 reflexive pronouns, auxiliary verbs, negations (NE...PAS, NON, IN-) and special punctuation marks to present alternate translations in case of doubt. ULZERO is strategical.



6. Morphological generation

This is the last step of the translation process. Words of the output text are generated. Some facilities must be provided by the algorithmic component, SYGMOR to handle elisions and contractions.

SYGMOR realizes the composition of two transducers : the first, "tree-to-string", produces the frontier of the object tree ; the second transforms this string (of masks of variables) into a string of characters, under the control of the linguistic data. These data are made of declaration of variables, formats and condition procedures, dictionaries (with direct addressing by the values of certain declared variables, whereby the first dictionary must be referenced by the UL, and a grammar.

Each item in a dictionary gives a list of <condition / assignment / string> triplets, the last one having an empty (true) condition.

```
A-L-AIDE   ==        / VID / 'A L'AIDE

AVOIR      ==  TP1A / VID / 'AI,
           ==  PSSPT/ V3H / 'EU,
           ==  TP3A / V3A / 'A.

LEQUEL     ==  NIB  / VID / 'LAQUELLE,
           ==  NID  / VID / 'LESQUELLES,
           ==  PLU  / VID / 'LESQUELS,
           ==       / VID / 'LEQUEL.
```

TP1A, PSSPT, TP3A are names of condition procedures. VID, V3H, V3A are names of formats. The apostrophs ('AI) are used in the grammar to make contractions.

It should be noted that, unlike ATEF, SYGMOR realizes a finite-state deterministic automaton, thus reflecting the lesser complexity of the synthesis process. To process a mask, SYGMOR looks for the first applicable rule (at least one must have an empty condition), applies it and follows the transitions indicated, unless it finds an inapplicable obligatory rule. In this case, the system executes the special rule ERREUR or a default action if this rule has not been declared. It is thus possible to generate an arbitrary error string at that point. For instance, non translated source lexical units will be printed between special markers.

The output of SYGMOR on our standard example is the following text, which is then transformed by ARIANE in a script file and formatted, thereby adding documentary information.

Output text

ON A FORMULE LE PRINCIPE A L'AIDE DUQUEL ON DEFINIT LE CRITERE UTILE POUR LE SYSTEME THERMODYNAMIQUE NON LINEAIRE.

RUSSE RAPPORT

LANGUES DE TRAITEMENT: RUS-FRA

TEXTE D'ENTREE:

SIMPOZIUM POSVYATHEN YADERNOJ SPEKTROSKOPII I STRUKTURE ATOMNOGO YADRA . VO VSTUPITELQNOM SLOVE PODKHERKIVAETSYA VAZHNAYA ROLQ , KOTORUYU SIMPOZIUM SYIGRAL V RAZVITII YADERNOJ FIZIKI SLABYIX YENERGIJ V SOVETSKOM SOYUZE . V XODE SIMPOZIUMA OBSUZHDEN RYAD VAZHNYIX ISSLEDOVANIJ , OSUTHESTVLENNYIX SOVETSKIMI UKHENYIMI . V KHASTNOSTI , IZUKHENO NESOXRANENIE KHETNOSTI V YADERNYIX PROCESSAX , SOZDANIE MODELI NEAKSIALQNOGO YADRA , SPONTANNOE DELENIE IZOTOPOV SVERXTYAZHELYIX YELEMENTOV I OBNARUZHENIE YEFFEKTA TENEJ PRI RASSEYANII KHASTIC .· SOBRANYI UBEDITELQNYIE STATISTIKHESKIE DANNYIE , OTRAZHAYUTHIE ROST KHISLA PREDLOZHENNYIX DOKLADOV . OTMEKHAETSYA PRISUTSTVIE SREDI UKHASTNIKOV SPECIALISTOV IZ ZARUBLZHNYIX STRAN .

TEXTE DE SORTIE:

----- (TRADUCTION DU--1 MARS 1980 11H 12MN 37S) -----
VERSIONS : (A :-29/01/80 ; T :-29/01/80 ; G :-21/09/79)

LE SYMPOSIUM EST CONSACRE A LA SPECTROSCOPIE NUCLEAIRE ET A LA STRUCTURE DU NOYAU ATOMIQUE. DANS LE MOT D'ENTREE ON SOULIGNE LE ROLE IMPORTANT QUE LE SYMPOSIUM A JOUE DANS LE DEVELOPPEMENT DE LA PHYSIQUE NUCLEAIRE DES FAIBLES ENERGIES EN UNION SOVIETIQUE. PENDANT LE SYMPOSIUM ON A EXAMINE LA SERIE DES ETUDES IMPORTANTES REALISEES PAR LES SAVANTS SOVIETIQUES. EN PARTICULIER, ON A ETUDIE LA NON-CONSERVATION DE LA PARITE DANS LES PROCESSUS? PROCEDES? NUCLEAIRES, DIVISION SPONTANEE DES ISOTOPES DES ELEMENTS SUPERLOURDS ET DECOUVERTE DE L'EFFET DES OMBRES PENDANT LA DISPERSION DES PARTICULES. ON A REUNI LES DONNEES STATISTIQUES CONVAINCANTE QUI REFLETENT LA CROISSANCE DU NOMBRE DES RAPPORTS PROPOSES. ON REMARQUE LA PRESENCE PARMI LES PARTICIPANTS DES SPECIALISTES DES PAYS ETRANGERS.

Bibliography

[1] Axtmeyer M. - Gaudey J. - Torre L. "Indexage morpho-syntaxique du russe". Juin 1979.

[2] Boitet Ch. "Un essai de réponse à quelques questions théoriques et pratiques liées à la traduction automatique". Définition d'un système prototype. (Thèse Docteur Es-Sciences Mathématiques - Avril 1976).

[3] Boitet Ch. "Problèmes actuels en traduction automatique : un essai de réponse". (COLING 76 - OTTAWA 1976).

[4] Boitet Ch. "Mechanical Translation and the Problem of Understanding Natural Languages". (Table-Ronde IFIP - TORONTO - Août 1977 et Colloque Franco-Soviétique MOSCOU 1977).

[5] Boitet Ch. "Extension de ROBRA et utilisation de l'environnement de contrôle interactif à l'exécution". Mars 1979.

[6] Boitet Ch. "Automatic Production of CF and CS Analyzers using a General Tree-Transducer". (2. Internationales Kolloquium über Maschinelle Uebersetzung, Lexikographie und Analyse, - SAARBRUCKEN - 16-17/11/1979). November 1979.

[7] Boitet Ch. - Guillaume P.- Quézel-Ambrunaz M. "Manipulation d'arborescences et parallélisme : le système ROBRA". (COLING 78 - BERGEN 1978).

[8] Chauché J. "Transducteurs et arborescences". Etudes et réalisations de systèmes appliquées aux grammaires transformationnelles". (Thèse Docteur Es-Sciences Mathématiques - Décembre 1974).

[9] Chauché J. "Présentation du système C.E.T.A." (G.3100.A - Janvier 1975).

[10] Chauché J.-Guillaume P.-Quézel-Ambrunaz M. "Le système A.T.E.F." (Analyse des Textes en Etats-Finis). (G.2600.A - Octobre 1972).

[11] Communication groupée. "Où en est le GETA début 1977". (also available in English and German). (Colloque : Franchir la barrière linguistique : LUXEMBOURG - Mai 1977).

[12] Guillaume P. "Le modèle de transformations d'arbres : ROBRA". (Présentation des caractéristiques externes de la grammaire complétant celles du modèle C.E.T.A.). (G.3200.A - Mars 1978).

[13] Jaeger D. "SYGMOR". (SYstème de Génération MORphologique). (G.3300.A - Mars 1978).

[14] Jaeger D. "SYGMOR". Nouvelle version. Exécuteur en Assembleur 360. Avril 1979.

[15] Nédobejkine N. "Niveaux d'interprétation dans une traduction multilingue : application à l'analyse du russe". (COLING 76 - OTTAWA 1976).

[16] Nédobejkine N. "Application du système A.T.E.F. à l'analyse morphologique de textes russes". (International Conference on Computational Linguistics - PISA 1973).

[17] Quézel-Ambrunaz M. "ARIANE. Système interactif pour la traduction automatique multilingue". (Version II). (G.3400.A - Mars 1978).

[18] Quézel-Ambrunaz M. "Transfert en ARIANE-78. Le modèle TRANSF". Novembre 1979.

[19] Quézel-Ambrunaz M. - GUILLAUME P. "Analyse automatique de textes par un système d'états-finis". (International Conference on Computational Linguistics - PISA 1973).

[20] Thouin B. "Système informatique pour la génération morphologique de langues naturelles en états-finis". (COLING 76 - OTTAWA 1976).

[21] Vauquois B. "La traduction automatique à Grenoble". Dunod - 1975 - Documents de Linguistique Quantitative n° 24. (en librairie).

[22] Vauquois B. "Automatic Translation. A survey of Different Approaches". (COLING 76 - OTTAWA 1976).

[23] Vauquois B. "L'évolution des logiciels et des modèles linguistiques pour la traduction automatique". (Colloque Franco-Soviétique - MOSCOU - Décembre 1977).