ATEF

Mathieu Lafourcade

GETA, Institut IMAG
(UJF & CNRS)
BP53X, 38041 Grenoble Cedex, France
e-mail : lafourca@imag.fr

le 5/11/00

ATEF

Mathieu Lafourcade

GETA, Institut IMAG
(UJF & CNRS)
BP53X, 38041 Grenoble Cedex, France
e-mail : lafourca@imag.fr

avril 1991

Introduction

ATEF (Analyse de Textes en Etats Finis) est un langage permettant de spécifier des analyseurs morphologiques de langue naturelle. Conçu et implémenté entre 1971 et 1973 par messieurs J. Chauché, P. Guillaume et M. Quézel-Ambrunaz, ATEF a été utilisé depuis pour l'écriture d'analyseurs du français, du russe, du portugais, de l'anglais, du japonais et de l'allemand.

ATEF constitue un outil indispensable de la conception d'un système de traduction. L'analyse morphologique dans le processus de traduction devant être alors spécifié par le linguiste de manière souple et efficace.

Dans ce document, nous nous attacherons à faire une présentation générale du système ATEF. Cette présentation restera simple car il est impossible de voir en détail l'ensemble de ce système en un nombre restreint de pages. Donc, pour des précisions supplémentaires, on peut se référer à [1] et [3]. La première partie est un rapide parcours des différents éléments constitutifs d'ATEF. La seconde partie est consacrée à quelques aspects algorithmiques et linguistiques d'ATEF. Enfin, la troisième partie présente une (petite) application pratique.

Présentation générale

Cette partie constitue une rapide présentation des composantes du système ATEF. Tous d'abord nous replaçons ATEF dans le processus de traduction existant au GETA et présentons les principes directeurs de ce système. Une seconde partie contient les éléments linguistiques notables d'ATEF. Enfin, une dernière partie, s'attache à décrire le fonctionnement de l'automate qui constitue le moteur d'ATEF.

Contexte et principes

Contexte d'ATEF dans ARIANE

ATEF permet de spécifier la phase d'analyse morphologique dans un système de traduction. Dans le système ARIANE-G5 du GETA, cette phase est la première parmi les 6 (au moins) que comporte une traduction (voir figure 1). Cette phase permet la consultation des dictionnaires et le découpage des mots. La deuxième phase produit une structure intermédiaire hiérarchisée à partir de la structure fournie par l'analyseur morphologique (ces deux premières phases constituant l'*analyse)*. Le *transfert* (troisième et quatrième phases) fait alors correspondre cette structure en langue source à une structure équivalente en langue cible. La *génération* (cinquième et sixième phase) crée le texte cible.

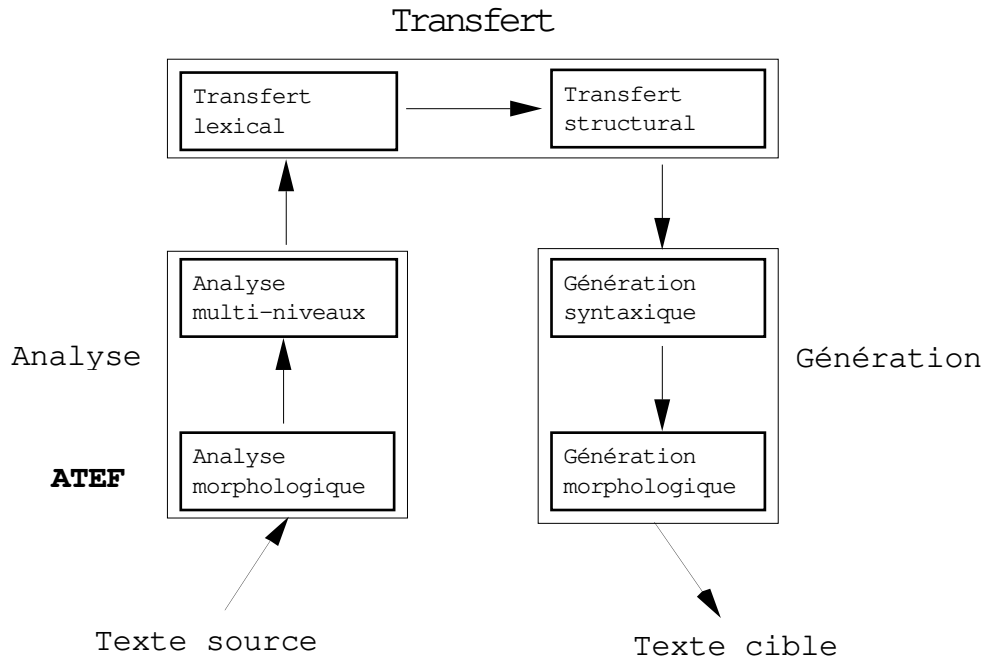

FIGURE 1 : Organisation logique du système de traduction ARIANE.
La phase d'analyse morphologique (AM) est la première d'une longue suite de traitements.

En fait, ATEF permet aussi de faire une *analyse pré-syntaxique*. Cependant cette analyse est peu profonde et ne semble pas simplifier le travail de l'analyse syntaxique. C'est pourquoi cette fonctionnalité n'est pas utilisée au GETA où ATEF est exploité uniquement pour l'analyse morphologique.

Il faut préciser qu'avant ARIANE-G5, avec ATEF était faite la seule et unique consultation de dictionnaires (en analyse). Maintenant la partie lexicale (EXPANS) est clairement séparée.

Principes généraux d'ATEF

ATEF dispose d'un certain nombre de données linguistiques :

- des déclarations de variables,
- des déclarations de formats à partir des variables,

- des dictionnaires. Chaque article contient en entrée une chaîne de caractères (le "morphe") ainsi que diverses informations,
- une grammaire composée de règles. Une règle est composée d'une liste de "formats d'appel" (les formats acceptables pour appliquer cette règle), de conditions (à vérifier pour appliquer cette régle) et d'actions (exécutées si cette régle est applicable).

ATEF transforme le texte d'entrée en une arborescence (voir figure 2).

ATEF analyse suscessivement chaque forme du texte (les sous-chaînes de la chaîne d'entrée séparées par des espaces) en examinant toutes les solutions d'analyse possibles[1]. Chaque forme est alors découpée en plusieurs segments correspondant aux articles des dictionnaires.
Le système cherche ensuite à appliquer une des règles dont le format d'appel correspond au segment trouvé dans les dictionnaires.

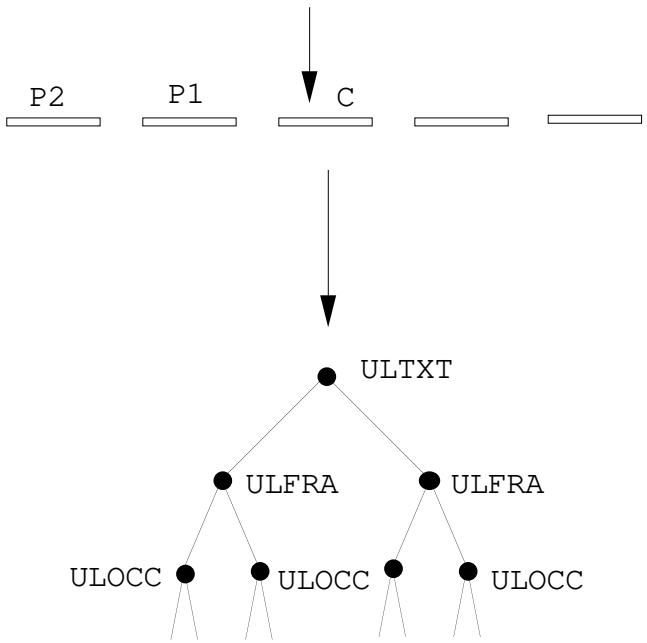

FIGURE 2 : ATEF reçoit en entrée une chaîne de caractères et fourni en sortie un arbre.

Les différentes analyses trouvées constituent l'arbre fourni en sortie.

La composante algorithmique d'ATEF n'est pas altérable. Celle-ci exploite la composante linguistique qui définit une *stratégie d'analyse* du texte d'entrée.

[1]- Ceci n'est vrai qu'en première approximation. En fait lors de l'écriture de ses règles, le linguiste peut avoir recours à des fonctions d'optimisation qui limitent le non-déterminisme. Dans ce cas le modèle n'est plus le modèle non-déterministe "pur". Cet aspect d'ATEF est développé dans la seconde partie de ce document "Quelques détails linguistiques et algorithmiques".

Eléments linguistiques

Dans cette partie on s'attache à une description des éléments linguistiques constituant ATEF.

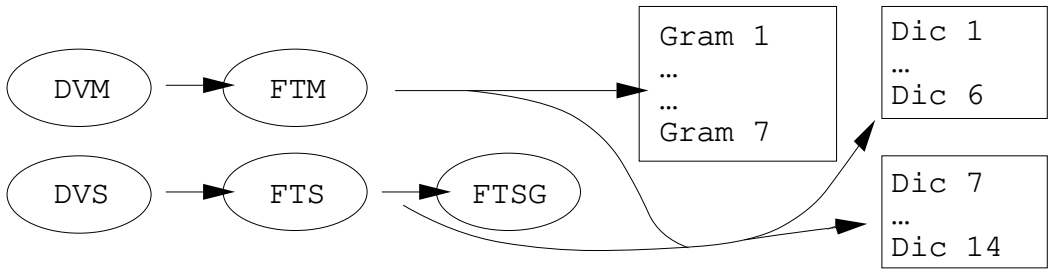

FIGURE 3 : Un analyseur morphologique construit avec ATEF est constitué de définitions de variables morphologiques (DVM), de définitions de variables syntaxiques (DVS), de formats morphologiques (FTM), de formats syntaxiques (FTS), de formats syntaxiques généraux (FTSG), de grammaires (jusqu'à 7) et de dictionnaires (de base et d'afixes).

Les Variables

Elles correspondent aux catégories linguistiques formalisées. Il existe deux types de variables : les variables morphologiques et les variables syntaxiques. En fait cette distinction est relativement artificielle, les linguistes pouvant utiliser les deux pour coder n'importe quel type d'informations.

Exemple de variable :

```
SUBN := (PR, PF, IP)

Cet exemple correspond à la morpholgie de l'allemand. PR se rapporte aux noms
propres, PF aux noms parfaits pouvant se passer d'article et IP aux noms
imparfaits nécessitant un article.
```

Une variable est donc un ensemble de valeurs élémentaires. Ces valeurs peuvent être *exclusives* (-EXC-) ou *non-exclusives* (-NEX-). Si les valeurs sont exclusives alors la variable ne peut prendre qu'une seule valeur à la fois parmi celles possibles. Une variable non-exclusive peut avoir comme valeurs un sous-ensemble des valeurs possibles.

Les variables non-exclusives, peuvent être manipulées à l'aide d'opérations ensemblistes (union, intersection, égalité, etc.). L'utilisation de ce type de variable permet de simplifier l'écriture des conditions et de diminuer sensiblement le nombre de règles de la grammaire (celle-ci gagne alors en clarté).

Il existe deux noms de variables réservés : UL et DICT.

• UL (Unité Lexicale) indique la référence lexicale d'une forme. Par exemple, les formes conjuguées du verbe "être" renvoient tous à la même référence lexicale (qui est le lemme "être"). UL est une variable exclusive.

• DICT permet de déclarer les dictionnaires utilisés. Il n'est possible d'avoir au maximum que 6 valeurs (6 dictionnaires au maximum). Cette variable est obligatoirement

de type morphologique (elle appartient à VARNM, c'est à dire les variables non-exclusives morphologiques).

La déclarations des variables est répartie sur deux fichiers, l'un contenant les variables morphologiques et l'autre les variables syntaxiques. À l'intérieur de ces fichiers, les variables exclusives et non-excluives sont regroupées et précédées respectivement par les mots-clés "-EXC-" ou "-NEX-".

Le linguiste peut aussi créer des variables dites "statégiques". Elles ne contiennent pas d'information linguistique mais permettent de contrôler ou de guider l'application des règles.

Les Formats

Un format constitue une classe de mots définie par le linguiste. Un format appelle une ou plusieurs règles de la grammaire. À chacune des classes qu'il a définies, le linguiste fait correspondre une combinaison de valeurs. Cet ensemble de valeurs constitue un format. Chaque format a un nom.

Exemple de formats :

```
SUBCOP := (COOR, CJS, PRP)
```

Cet exemple correspond à la morpholgie de l'allemand. SUBCOP est une sous catégorisation des lexèmes d'hypotaxe ou de parataxe (sic)[2]. COOR est une conjonction de coordination, CJS une conjonction de subordination et PRP une préposition.

Comme pour les variables il existe deux types de formats : les *formats morphologiques* (FTM) et les *formats syntaxiques* (FTS). L'information peut donc être (grosso modo) d'ordre morphologique ou syntaxique. Ce système impose certaines contraintes, par exemple l'impossibilité d'utiliser une variable syntaxique (VARS) dans un format morphologique (FTM).

Un format peut être vide. Dans ce cas son rôle consiste à activer une ou plusieurs règles.

Les Dictionnaires

Les dictionnaires contiennent l'ensemble des morphes d'une langue, c'est à dire toutes les chaînes élémentaires intervenant lors du découpage des occurrences du texte par l'automate.

Il y a des dictionnaires de bases avec UL qui contiennent des formes complètes fléchies, mots invariables, tournures figées, etc. Il y également des dictionnaires de désinences, sans UL, contenant les affixes.

À chaque article de dictionnaire est associé les deux formats (FTS et FTM) qui définissent les informations linguistiques associées à cet article.

Exemple d'entrée de dictionnaire :

```
UND == FMINV (FSCOOR, UND)
```

[2] - L'hypotaxe correspond à la subordination. La parataxe correspond à la coordination.

Cet exemple correspond à la morpholgie de l'allemand. Le premier UND est l'article du dictionnaire. FMINV est le nom de format FTM. FSCOOR est le nom de format FTS. Le deuxième UND est la référence lexicale (UL).

Il ne peut y avoir découpage des mots qu'après consultations des dictionnaires. L'accès aux dictionnaires est contrôlé dans la grammaire au moins avec la règle obligatoire RDICT (les dictionnaires disponibles au début du traitement)[3]. Il existe aussi une règle automatique d'ouverture des dictionnaires, elle correspond aussi à l'affectation automatique de la valeur d'UL.

La grammaire et les règles

La grammaire est une suite de règles. Celle-ci doit comporte au moins deux règles obligatoires RDICT (règle d'ouverture des dictionnaires) et MOTINC (règle du mot inconnu).

Une règle se compose d'une *partie gauche* (à gauche de "==") et d'une *partie droite* (à droite de "==").
En partie gauche, on trouve le nom de la règle et une liste (non vide) de noms de formats FTM[4]. Les règles sont appliquées sur les morphes dont la classe correspond à l'un des formats de la liste de formats de la partie gauche.

En partie droite, on trouve des *instructions* et des *conditions*.

Les instructions peuvent être des affectations de valeurs aux variables ou des transformations de chaînes (TChaînes).

Les conditions sont des conditions d'application de la règle et des sous-règles qui constituent alors des conditions complémentaires. Au moins une des sous-règles doit s'appliquer pour que la condition soit vérifiée.

Exemple de règle :

```
TC01A : V01A == VARS(C) := VARS(A), VAREM(C) := VAREM(A),
                VARNM(C) := VARNM(C) -U- VARNM(A) /
                / TCHAINE(0,'','01').
```

Cet exemple correspond à la morpholgie de l'allemand. Le nom de la règle est TC01A. Cette règle est appliquée pour le segment courant correspond au format V01A. Dans ce cas les valeurs des variables VARS (variables syntaxiques) du segment trouvé dans le dictionnaires (A) sont affectée aux variables correspondantes du segment courant (C). Il en est de même pour les VAREM (variables exclusive morphologiques). Pour les variables VARNM (variables non-exclusives morphologiques), on demande l'union entre les valeurs courantes et celles issues du dictionnaire. D'autre par on applique une TChaine (on modifie la chaine courante en insérant à la position courante[5] la chaine "01").

L'application d'une régle consiste à exécuter les instructions de calcul qu'elle contient ; elle dépend de la satisfaisabilité des conditions de cette règle.

La règle du mot inconnu (MOTINC) est une "sortie de secours" au cas ou on a trouvé aucun article dans les dictionnaire pouvant constituer un morphe acceptable (il n'y a pas de sous-chaine constituant un préfixe de la chaine courante). On est pas arrivé dans un état final de l'automate, il y a échec !

---

3 - Il peut aussi y avoir un contrôle dans les autres règles, mais ce n'est pas obligatoire.
4 - Ces formats FTM constituent le vocabulaire d'entrée de l'automate.
5 - C'est l'origine 0, c'est-à dire la fin du morphe courant.

Avec cette règle on peut définir des “stratégies de rechange”.
On peut éventuellement se mettre en mode intéractif et demander à l’utilisateur de corriger une éventuelle faute d’orthographe. On peut aussi essayer de déterminer la base du mot inconnu et l’indexer dans les dictionnaires.

Le problème avec MOTINC est qu’elle affecte l’UL avec la chaîne non analysée. On voudrait peut-être transformer l’UL. Actuellement il n’y a pas de manipulation de chaînes sur la valeur d’UL.

Fonctionnement de l’automate

L’automate est défini par la grammaire. Cet automate segmente les formes de texte d’entrée en morphes compatibles entre eux. Cette compatibilité est assurée à l’aide du traitement linguistique.

Processus d’analyse

Le parcours de l’arbre des solutions possibles se fait en profondeur d’abord. Toutes les solutions possibles sont exhibées à moins que le linguiste n’utilise des fonctions heuristiques (dans les règles) permettant de contrôler le non-déterminisme (voir plus bas).

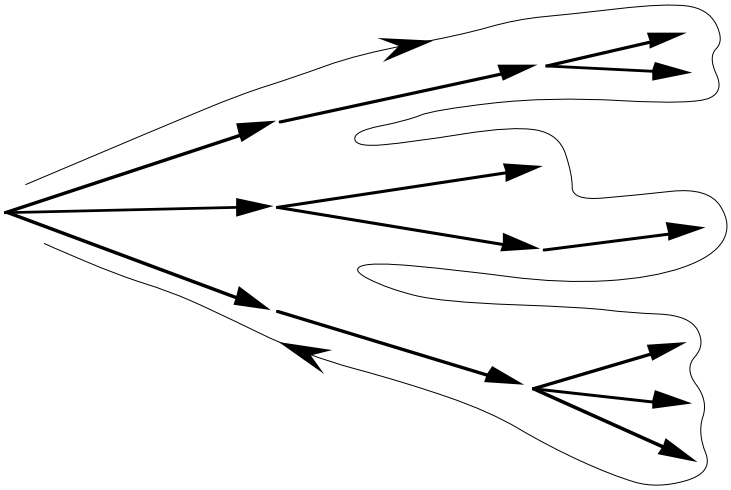

FIGURE 4 : L’automate produit toutes les solutions possibles.

L’état courant de l’automate est déterminée par :

- la position du pointeur sur la chaîne de caractères courantes,

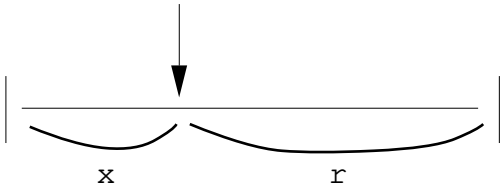

FIGURE 5 : Le pointeur dans le chaîne de caractères courante.
x est la sous-chaîne déjà analysée et r est la sous-chaine restant à analyser.
On a ici une analyse de la gauche vers la droite.

- l'ensemble des informations linguistiques du moment.

Les informations linguistiques sont contenues dans un "masque de variables" C (comme *état Courant)*. C contient l'ensemble des valeurs des variables VRAM, VARS et éventuellement la variable d'Unité Lexicale (UL). Au départ C est vide.

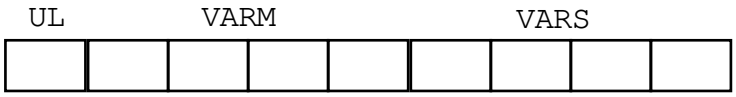

FIGURE 6 : l'Etat courant contient l'ensemble des variables déclarées. Il indique quelle valeur a l'état courant pour chaque variable.

Un morphe est *candidat* dans l'état courant s'il appartient à un dictionnaire ouvert et s'il coïncide (comme chaîne) à une sous-chaîne de ce qui reste à analyser (la sous-chaîne r).

À chaque morphe candidat sont associées ses informations linguistiques[6]. On associe également une ou plusieurs règles de la grammaire. Une règle (potentiellement applicable) est appellée si et seulement si le format FTM du morphe est inclu dans la liste de formats FTM de la règle.

La règle appelée vérifie si ses conditions sont vérifiées. Ces conditions portent sur C et/ou A(m). Si les conditions sont vérifiées alors la partie action de la règle peut-être exécutée.

Si une des règles associée au morphe candidat est applicable alors le morphe est dit *acceptable.* Dans ce cas C est modifié selon la partie action de la règle et le repère est déplacé :

6 - Cette information linguistique est un masque A(m) de même structure que C. A(m) est construit directement à partir des formats FTS et FTM associés pour ce morphe dans les dictionnaires.

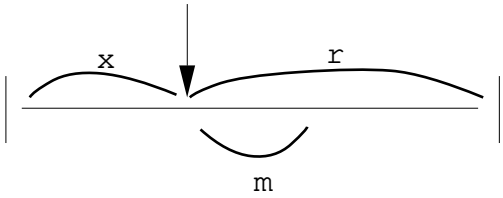

FIGURE 7 : Déplacement du repère

Si le repère se trouve en bout de chaîne et si UL(C) est différent de "UL0" (valeur indiquant qu'il n'y a pas d'unité lexicale à analyser) alors il y a succès du découpage. C constitue alors un résultat de l'analyse de la forme.

L'automate affectue alors un retour-arrière (processus de *backtracking*) à l'état imédiatement précédent. Il essait alors les autres solutions possibles de découpages.
Les alternatives sont soit :

- la règle suivante applicable pour le même morphe, soit
- un autre morphe candidat.

Si le repère n'est pas en bout de chaîne et qu'aucun morphe n'est acceptable, il y a *échec* de l'analyse en cours. Il y a alors un retour-arrière et une autre solution est essayée.

Pour un "pas" de segmentation effectué, trois opérations élémentaires sont réalisées :

- choix d'un dictionnaire,
- choix d'un segment,
- application d'une régle.

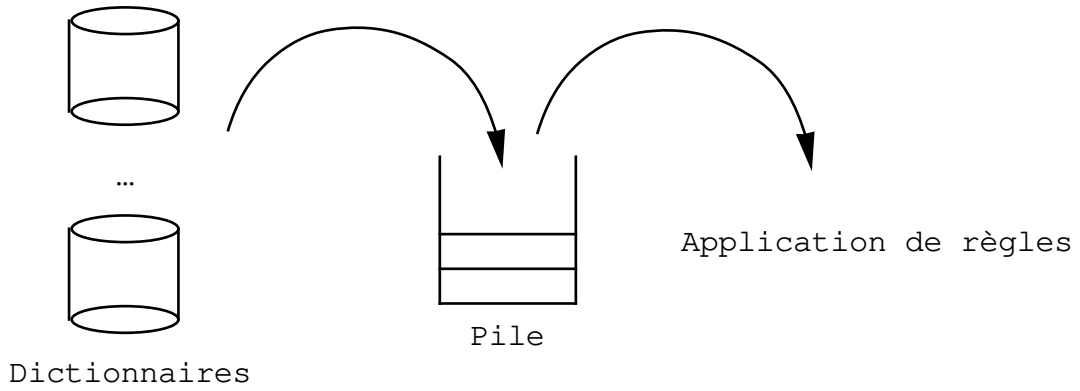

FIGURE 8 : Les trois phases d'une segmentation.
On choisit les dictionnaires (à partir de la régle obligatoire RDICT ou à partir d'actions de la règle précédente). À partir des dictionnaires, on sélectionne et empile tous les segments candidats. On applique ensuite la régle courante aux candidats.

La connaissance du fonctionnement de l'automate permet (à l'aide de fonctions de contrôle) de contourner le non-déterminisme.

Formes de sortie

Le produit de l'analyse morphologique est donc bien constitué par un ou plusieurs résultats.

Les résultats de l'analyse peuvent être présentés sous deux formes différentes : avec ou sans homophrase.

• Sans homophrase, si pour une occurrence on a plusieurs résultats d'analyse ces résultats sont factorisés sur un seul arbre.

• Avec homophrases, les différents résultats d'analyses sont distribués sur autant d'arbres que nécessaires.

La solution avec homophrases prend beaucoup de place mémoire (beaucoup de duplications) c'est pourquoi elle n'est pas utilisée au GETA. Cependant pour le reste de l'analyse (ROBRA) une telle solution simplifierait le travail.

Quelques détails algorithmiques et linguistiques

Dans cette partie, on présente quelques aspects particuliers d'ATEF. On a estimé qu'il était particulièrement intéressant de parler du *Contrôle sur l'Arborescence*, de l'utilisation des *TChaines* ainsi que de l'*Analyse de Contexte*.

Contrôle du non-déterminisme

Le linguiste à la possibilité de contrôler l'automate de façon à ce qu'il n'ait pas un comportement non-déterministe "pur". C'est ce qu'on appelle le *Contrôle de l'arborescence des choix.*

Dans ce cas l'automate de parcours pas toutes les solutions possibles.

Ce contrôle s'effectue grâce à des fonctions spéciales incluses dans les règles.

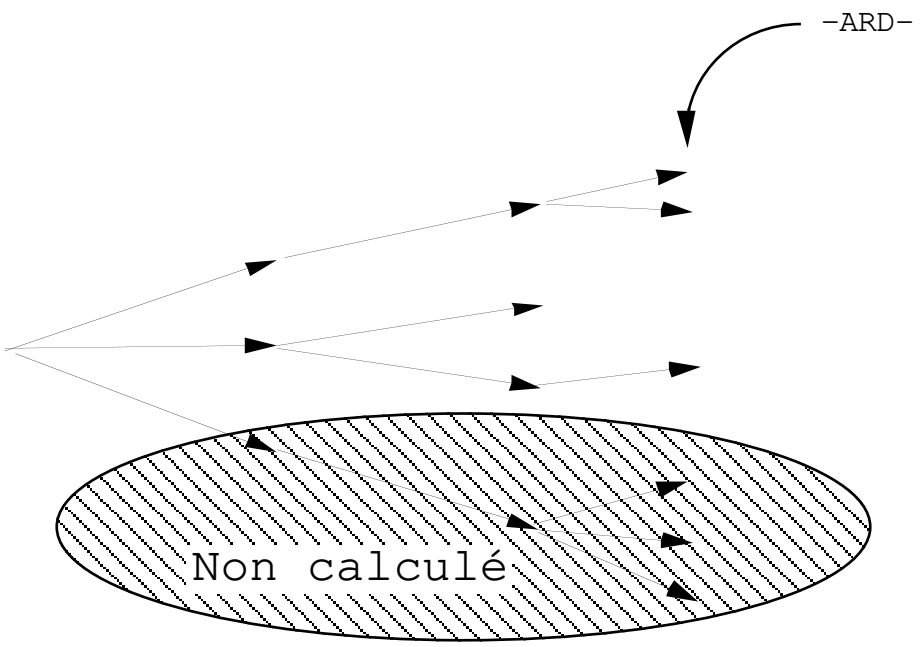

FIGURE 9 : Utilisation de la fonction -ARD-.
Cette fonction a deux effets : 1) elle empêche l'essai d'application d'autres règles candidates au traitement du segment courant ; 2) elle élimine les morphes en attente qui proviennent du même dictionnaire et qui sont de longueur inférieure.

L'intérêt d'un tel contrôle est tel qu'il est utilisé systématiquement dans les analyseurs morphologiques conçus au GETA. Il permet par exemple, d'accélérer la recherche des solutions acceptables et d'éviter de sortir des résultats parasites.

Par exemple, supposons que l'on ait la forme "prévisible" à analyser. Cette forme n'est pas ambigüe. Pourtant "pré" constitue un article du dictionnaires des préfixes. Il faudra donc s'arranger pour ne considérer que la solution "prévisible" et ignorer la solution "pré-visible"[7].

Les fonctions les plus importantes sont :

`-FINAL-, -STOP-, ARRET-, -ARD-,` et `-ARF-`

Elles permettent d'éliminer certains choix (déjà fait ou à venir). Pour le détail de ces fonctions, on peut se reporter à [3].

La démarche habituelle est de travailler avec des règles générales et d'affiner l'analyse avec de telles fonctions de contrôle. Elles permettent alors d'écarter les solutions parasites. Leur utilisation est de toutes façon conditionnée par des raisons linguistiques.

[7] - L'automate cherche les sous-chaînes par taille croissante et les empile. Donc c'est la sous chaîne la plus longue qui sera d'abord dépilée et dont on appliquera la règle. Avec le contrôle du non-déterminisme, c'est cette règle qui empêchera de considérer les autres solutions possibles (la sous-chaîne "pré" dans notre exemple).

TChaînes

Les TChaînes permettent de modifier la chaîne d'entrée. Cette possibilité permet d'accélérer notablement la recherche de solutions acceptables et de faciliter la mise au point des règles de la grammaire.

Pour un exemple de TChaîne voir l'exemple de règle.

L'utilisation des TChaînes peut avoir deux motivations :

- Rendre le traitement plus efficace

Dans le processus de traduction, le temps passé dans l'analyse morphologique avec ATEF est négligeable par rapport à celui passé dans l'analyse syntaxique. C'est pourquoi des impératifs d'optimisation ne fournissent pas une très bonne justification.

- S'en servir comme stratégie pour arriver à la solution

L'idée est d'avoir une organisation des données de telle façon que le premier résultat trouvé soit nécessairement le bon (c'est un peu le même objectif qu'avec les fonctions de contrôle du non-déterminisme).
Par exemple, dans le cas de l'allemand, les mots composés sont indexés tels quels. Dès qu'une solution a été trouvée on n'en cherche pas d'autres.

Les TChaînes sont d'un emploi pratique. C'est un ajout très utile à ATEF qui offre cependant la possibilité de s'en passer. Il existe toujours des solutions évitant les TChaînes. Ces alternatives disposent d'une écriture des règles souvent plus compacte qu'avec les TChaînes mais dont la mise au point est moins facile. Il semblerait que l'utilisation des Tchaînes rallonge légèrement le temps de traitement[8].

L'analyseur morphologique de l'allemand implémenté et utilisé au GETA fait un emploi systématique des TChaînes.

Analyse de contexte

Avec ATEF, il est possible de faire une analyse de contexte portant sur la forme courante ou sur les formes voisines.

Une partie non modifiable du contexte cible peut être testé. Il est constitué de :

- valeurs des solutions déjà trouvées pour l'occurrence courante.
- valeurs des solutions précedentes

Une partie modifiable du contexte cible peut également être testée :

- Condition de sous-chaînes (SCHAINE) tansformable par les TChaînes. Cette condition porte sur la chaîne courante.

- Une valeur de décoration S associée à la prochaine occurrence.

[8] - Si on considère la rapidité relative de l'analyse morphologique, cette dégradation des performances n'est rien vis à vis du gain en clarté d'écriture que peut fournir l'utilisation des Tchaînes.

Applications pratiques et problèmes

L'application (très modeste) présentée ici permet de préciser la méthodologie utilisée pour écrire un analyseur morphologique en ATEF. Elle permet également d'éclairer sur plusieurs solutions possibles quand à la spécification des composants de l'analyseur.

Présentation

L'analyseur envisagé doit pourvoir être capable d'analyser quelques verbes conjugués au présent, futur et imparfait de l'indicatif. Ces verbes sont uniquement ceux du premier groupe.

On considère deux représentants des verbes du premier groupe : les verber en -ER (comme AIMER) et les verbes en -GER (comme MANGER).

On voudra aussi être capable de reconnaître des verbes aux participes (passé et présent) éventuellement accordés en genre et en nombre.

On veut aussi reconnaître directement (sans décomposition de la forme) quelques pronoms, articles, noms (communs et propres) et signes de ponctuation. Ces formes sont indexées directement dans les dictionnaires. L'ajout de ces formes permet de fournir en entrée de petites phrases (ce qui est plus sympatique que des mots isolés).

Solution

L'analyse se fera de la gauche vers la droite. Ce sens d'analyse correspond au sens intuitif : on trouve une base puis on attend une désinence. Ce choix est toutefois relativement arbitraire, on pourrait toujours faire autrement.
Cependant on peut remarquer que d'une manière générale, les désinences sont plus ambiguës que les bases et l'on a toujours intérêt à analyser ce qui est le moins ambigu d'abord.

Les variables

On commence par définir les variables. Tous les variables sont non-exclusives[9].

```
CAT == (V, PRON, ART, NOM, PONC)

CAT est la CATégorie de la forme. V correspond à un Verbe, PRON à un PRONom,
ART à un ARTicle, NOM à un NOM et PONC à un signe de ponctuation.


SUBV == (VC, PPA, PPR)

SUBV est la sous catègorie d'un verbe. VC correspond à un Verbe Conjugué, PPA à
un Participe PAssé et PPR à un Participe PRésent.


SUBN == (NP, NC)
```

[9] - Il existe certaines contraintes sur les formats qui justifient ce choix. Les FTS sont définis à partir de variables syntaxiques uniquement et les formats FTM ne contiennent que des variables morphologiques non-exclusives. Il faut garder à l'esprit que les variables exclusives sont scalaires et les variables non-exclusives ensemblistes (d'un emploi plus souple mais plus gourmant en mémoire).

```
SUBN est la sous catègorie d'un nom. NP correspond à un Nom Propre et NC à un
Nom Commun.


MT == (PRES, IPF, FUT)

MT est le Mde Temps du verbe. PRES correspond au PREsent et IPF à l'ImParFait
et FUT au FUTur.


PERS == (1, 2, 3)

PERS est la personne. 1, 2  et 3 correspondent aux première, seconde et
troisième personnes.


NBR == (SING, PLUR)

NBR est le nombre. SING correspond au SINGulier, PLUR au PLURiel.


GNR == (MAS, FEM)

GNR est le genre. MAS correspond au MASculin, FEM au FEMinin.
```

On va définir dans ce qui suit une variable d'état qui ne représente pas d'informations linguistique mais servent uniquement à des fins stratégiques.

Cette variable d'état contient à tout moment l'état dans lequel se trouve l'automate :

```
ETAT == (1, 2, 3, 4, 5, 6, F)

L'automate dispose de 6 états plus un état final.
```

Les formats

Les formats morphologiques (FTM) servent dans notre exemple, exclusivement à l'appel des règles. Il ne contiennent pas d'information linguistique. Les formats syntaxiques (FTS) contiennent l'information linguistique[10].

L'idée est d'avoir peu de règles et beaucoup d'informations différentes.

On définit d'abord des formats pour les bases :

```
BDV == CAT -E- V.

BDV correspond à une Base De Verbe.


PRONON == CAT -E- PRON.

PRONON correspond à un PRONON.


ARTICLE == CAT -E- ART, CAT -E- PRON.

ARTICLE correspond à un ARTICLE ou à PRONON.


NOM == CAT -E- NOM.
```

[10] - Il s'agit des informations attachées aux affixes (ici les désinences) et aux bases.

```
NOM correspond à NOM.

PONCTUATION == CAT -E- PONC

PONCTUATION correspond à signe de PONCTUATION.
```

Il faut maintenant définir des formats spécifiques correspondant aux différentes désinences :

Formats syntaxiques	*Désinences*	*Instanciation des variables*
	Présent	
FSE	e	MT -E- PRES, SUBV -E- VC, PERS -E- 1, PERS -E- 3, NBR -E- SING
FSES	es	MT -E- PRES, SUBV -E- VC, PERS -E- 2, NBR -E- SING
FSONS	ons	MT -E- PRES, SUBV -E- VC, PERS -E- 1, NBR -E- PLUR
FSEZ	ez	MT -E- PRES, SUBV -E- VC, PERS -E- 2, NBR -E- PLUR
FSENT	ent	MT -E- PRES, SUBV -E- VC, PERS -E- 3, NBR -E- PLUR
	Imparfait	
FSAIS	ais	MT -E- IPF, SUBV -E- VC, PERS -E- 1, PERS -E- 3, NBR -E- SING
FSAIT	ait	MT -E- IPF, SUBV -E- VC, PERS -E- 2, NBR -E- SING
FSIONS	ions	MT -E- IPF, SUBV -E- VC, PERS -E- 1, NBR -E- PLUR
FSIEZ	iez	MT -E- IPF, SUBV -E- VC, PERS -E- 2, NBR -E- PLUR
FSAIENT	aient	MT -E- IPF, SUBV -E- VC, PERS -E- 3, NBR -E- PLUR
	Futur	
FSERAI	erai	MT -E- IPF, SUBV -E- VC, PERS -E- 1, NBR -E- SING
FSERAS	eras	MT -E- IPF, SUBV -E- VC, PERS -E- 2, NBR -E- SING
FSERA	era	MT -E- IPF, SUBV -E- VC, PERS -E- 3, NBR -E- SING
FSERONS	erons	MT -E- IPF, SUBV -E- VC, PERS -E- 1, NBR -E- PLUR
FSEREZ	erez	MT -E- IPF, SUBV -E- VC, PERS -E- 2, NBR -E- PLUR
FSERONT	eront	MT -E- IPF, SUBV -E- VC, PERS -E- 3, NBR -E- PLUR
	Participe Présent	
FSPPR	ant	SUBV -E- PPR, GNR -E- MAS, NBR -E- SING
	Participe Passé	
FSPPA	é	SUBV -E- PPA, GNR -E- MAS, NBR -E- SING

Il y a de plus les cas du genre et du nombre

	Féminin	
FSFEM	e	GNR -E- FEM, NBR -E- SING
	Pluriel	
FSPLUR	s	NBR -E- PLUR

On a un problème d'ambiguïté pour la désinence "e". En effet, celle-ci peut correspondre aussi bien à la désinence dont le format est FSE qu'à celle dont le format est FSFEM.

On a aussi un problème avec le "e" de la forme "mangeons".

Dans l'écriture des règles nous avons deux solutions pour résoudre ces problèmes.

• Utilisation des TChaînes pour le "e" de "mangeons". On supprime dans la chaîne "mangeons" de "e" à l'aide de TChaine. Cela se fait après analyse de la base. On ramène bien le cas particulier du verbe "manger" au cas général "aimer".

• Pour le "e" flexionnel on dispose de deux solutions

Une fois que l'on a trouvé la base, on transforme ce qui suit avec des TChaînes. On insère par exemple les chaînes d'indexage "01" ou "02". Ensuite on a plus qu'à analyser normallement ce qui suit. Dans les dictionnaires les deux "e" différents ont comme entrées "01e" et "02e".

Si on a pas de Tchaînes alors on va rechercher les 2 "e" dans les dictionnaires. On rajoute alors une variable tactique exprimant la flexion possible à ce moment là. On demande que l'intersection entre cette variable tactique portée par la désinence et celle qui a été trouvée dans les dictionnaires ne soit pas nulle.
Cette variable peut être la variable d'état de l'automate que l'on a élaboré.

Ces deux solutions sont relativement équivalentes[11].

Les dictionnaires

On dispose d'un dictionaire de désinences contenant toutes les désinences rescencées (et uniquement elles).

On dispose également d'un dictionnaire contenant les bases (bases des verbe, articles, noms communs et propres, signes de ponctuation, etc.).

La grammaire

Plutôt que d'écrire la grammaire sous forme de règle, il semble plus explicite de la présenter sous forme d'automate.

[11] - Une troisième solution serait en fait possible en utilisant plusieurs dictionnaires pour les désinences. Sur la règle indique dans quel dictionnaire aller chercher la désinence attendue. Bien qu'utilisable ici, cette technique n'est pas très pratique en grandeur réelle à cause du nombre limité de dictionnaires.

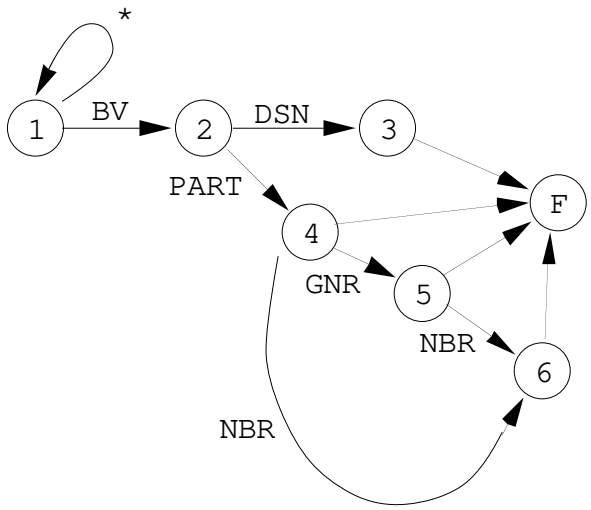

FIGURE 10 : L'automate définissant notre analyseur morphologique.
L'état F est l'état final. Sur les arcs on a indiqué le nom de la règle à appliquer.
Les deux règles obligatoires (RDICT et MOTINC) ne figurent pas dans l'automate mais existent bien.

Détails des transitions :

- * correspond à une majuscule ;
- BV correspond à la reconnaissance d'une base verbale ;
- DSN correspond à la reconnaissance d'une désinence verbale ;
- PART correspond à la reconnaissance d'un participe ;
- GNR correspond à la reconnaissance d'une désinence de genre ;
- BV correspond à la reconnaissance d'une désinence de nombre ;

Les transitions vers l'état final ne font aucune action.

Commentaire

On a vu grâce à cette application comment il était possible de spécifier un petit analyseur morphologique.

On a vu également deux solutions pour résoudre des problèmes d'ambiguïtés. Souvent, on conjugue à la fois TChaînes et variables tactiques. D'une certaine manière les TChaînes permettent le marquage de l'état de l'automate dans la chaîne (c'est uniquement vrai pour cette utilisation des TChaînes).

Conclusion

ATEF est un outil très efficace, abondamment utilisé au GETA (dans la chaîne ARIANE). Ses performances sont tout à fait satisfaisantes surtout si on compare les temps d'analyse morphologique par rapport aux temps d'analyse syntaxique. On ne s'apperçoit pas du temps d'exécution d'ATEF dans le processus global de traduction.

ATEF offre au linguiste la possibilité de structurer de multiples façons ses données. La liberté d'action est très large (peut-être trop) lors de la définition des variables, des formats et de l'automate. Chaque utilisateur peut développer ses petites "astuces" de manipulation de chaînes ou de variables tactiques.

ATEF mériterait de fonctionner sur d'autres systèmes que celui existant au GETA. Une éventuelle réécriture en vue d'une portabilité sous d'autres environnements semble souhaitable.

Remerciements

Je tiens tout particulièrement à remercier J.P. Guibaud et P. Guillaume pour leurs remarques et conseils, sans qui ce document n'aurait pas vu le jour.

Bibliographie

[1] CHAUCHE J. (1974) Transducteur et arborescences (Etudes et réalisation de systèmes appliquées aux grammaires transformationnelles). Thèse d'Etat, USMG, décembre 1974.

[2] GETA (1982), C. BOITET, rédacteur *Le point sur Ariane - 78. début 1982 (DSE - 1).* Volume I, Partie I : Le logiciel. GETA - CHAMPOLLION, CAP SOGETI FRANCE. avril 1982, 252 p.

[3] GUILBAUD J. P. (1980) Analyse morphologique de l'allemand en vue de la traduction par ordinateur de textes techiques spécialisés. Thèse de 3ème cycle, juin 1980 .