

# INTRODUCTION ET PLAN DE L'OUVRAGE « APPROCHE ALGORITHMIQUE DE LA GRAMMAIRE DE L'ARABE »

André JACCARINI

## 1. Présentation générale

Nous exposons dans cet ouvrage les principes de construction d'une classe d'analyseurs de l'arabe écrit non voyellé, présentant la particularité de pouvoir fonctionner sans lexique. Cette approche ne signifie pas, naturellement, que l'on s'interdise par la suite, de manière systématique, toute interaction avec un lexique donné.

On proposera donc des analyseurs modulables dont l'intérêt sera de mettre en évidence la régularité du système morphologique de base de l'arabe. La morphologie sera pour nous avant tout un langage formel, c'est-à-dire un langage dont la proportion de règles terminales (qui correspondent au lexique) par rapport à celles qui ne le sont pas (règles structurales) demeure limitée.

L'explicitation de cette structure de base nous permettra de déduire des analyseurs efficaces pour ce qui est de la rapidité et a fortiori de l'espace.

Signalons également, bien que cela puisse de prime abord sembler paradoxal, que cette approche devrait aussi retenir l'attention du linguiste qui s'intéresse à la confection de lexiques informatiques, que ce soit en vue d'une analyse syntaxique plus fine, ou bien du traitement de la sémantique. Il importe en effet lorsqu'on se donne pour domaine d'application le traitement automatique, que le lexique soit très bien structuré, qu'il contienne des informations homogènes, cohérentes, non redondantes, que les catégories et les traits définis soient en accord avec la ou les grammaires pour lesquelles ce lexique a été conçu. La recherche du meilleur équilibre entre grammaire et lexique demeure un problème délicat au niveau de la modélisation linguistique. Nos programmes pourront alors être utilisés comme outils, en ce sens qu'ils permettront de bien faire apparaître les limites de la grammaire ; ils feront ressortir toutes les spécificités de chaque forme par rapport au système général des régularités. Cela suppose une très bonne gestion de l'ambiguïté et c'est ce à quoi nous nous sommes attachés en concevant nos propres analyseurs et outils informatiques.

Le souci de minimalité, tant au niveau descriptif qu'à celui de la reconnaissance, nous a d'abord amené à construire des modèles syntaxiques et morphologiques unifiés dans le cadre de la théorie des automates et des transducteurs ; cela a nécessité ensuite la mise au point du concept de variation de grammaire en vue de l'obtention de l'algorithme optimum, laquelle suppose une recherche, sur les plans expérimental et théorique, de méthodes d'évaluation des grammaires. Cet état de fait nous a d'ailleurs conduit à construire le noyau d'un environnement de génie linguistique : l'atelier de grammaires. Ce dernier est conçu avant tout comme un outil de recherche de stratégies d'analyse et de génération d'applications linguistiques. La généralisation du cadre formel devrait permettre, à un stade ultérieur, de fonder théoriquement le système .

Le travail de modélisation linguistique s'est ainsi accompagné d'une recherche en méthodologie, au niveau même de la conception des programmes. Pour des raisons de clarté nous avons toutefois préféré isoler l'exposé de cet aspect fondamental de nos recherches, du texte que nous présentons ici, bien qu'il en soit profondément solidaire. Nous espérons ainsi éviter tout risque d'interférences d'interprétation. Cette scission aura peut-être aussi l'avantage de satisfaire les critères discriminatoires - aussi ténus qu'ils puissent apparaître quelquefois - permettant de distinguer en tant que disciplines séparées la linguistique formalisée susceptible d'être mise en œuvre informatiquement de l'algorithmique propre aux problèmes linguistiques. Elle pourrait en revanche présenter l'inconvénient de masquer une unité profonde, à laquelle nous avons été particulièrement sensibles, grâce à notre méthode de travail, qui a essentiellement consisté en un va-et-vient continu entre le modèle théorique et sa mise en œuvre informatique.

C'est la raison pour laquelle cette introduction comporte une présentation générale (§1) suivie de deux autres exposés portant respectivement sur les objectifs initiaux et la logique de ce travail (§2) et l'aspect informatique (§3). Il nous a semblé en effet indispensable de présenter d'abord une vue d'ensemble de nos méthodes de travail, en soulignant notamment l'unité sous-jacente aux deux aspects mentionnés au paragraphe précédent. L'atelier de grammaire, qui relève normalement du second, ne fait donc pas l'objet d'un exposé dans cet ouvrage, mais dans celui intitulé : « Modélisation linguistique et théorie des automates. Méthode de la détermination de l'algorithme optimum par variation de la grammaire. Application à l'arabe » (ref.2). Il en est de même de la plupart des questions évoquées dans la troisième partie de cette introduction. En revanche nous ne pouvions en faire totalement l'impasse, dans la présentation générale : l'atelier de grammaire apparaît en effet comme une *entité synthétisée susceptible de conférer un sens à tout notre travail d'algorithmique*, lequel s'est caractérisé, nous l'avons déjà dit, par des rétroactions continues entre les aspects linguistique et informatique. Bien plus, c'est dans le prolongement de cette même idée, celle de *donner du sens* à une activité dont la part expérimentale est essentielle, que nous préconisons aujourd'hui un élargissement du système.

Il nous semble important enfin de souligner que notre approche du problème nous a conduit tout naturellement à considérer la conception des grammaires non pas comme celles de descriptions immédiates et immuables des régularités de la langue mais tout simplement comme des points de vue, ne pouvant être que partiels. L'exhaustivité d'une description linguistique ne pouvant être atteinte, il faut savoir se suffire de ces différents points de vue, portés sur la langue, tout en cherchant toutefois à les relier entre eux, idéalement à trouver des formules d'équivalence et de transformation qui nous permettraient de passer de l'un à l'autre et à postuler l'existence d'un invariant. Ainsi le langage, dans sa globalité, inaccessible par une description immédiate et effective pourrait être appréhendé de manière indirecte, en posant un principe d'invariance, qui reviendrait à construire un espace abstrait de grammaires où ces dernières auraient la possibilité de varier.

Il est tout aussi important de rappeler que cette perspective théorique n'a pas été posée *a priori* mais a émergé au fur et à mesure qu'avancait notre travail d'expérimentation en vue de dégager des critères linguistiques pertinents d'évaluation de nos grammaires. Nous en rendons partiellement compte dans cet ouvrage, tant au niveau morphologique (chapitres II, IV, V et VI) que syntaxique (chapitres VIII et IX). Signalons également que dans le cadre de ce travail un logiciel baptisé " *Sarfiyya* " a été développé par Christian Gaubert dont les fonctionnalités d'analyse morphologique (sans lexique) de textes non-voyellés reposent sur le concept de " variations des grammaires ". Il est maintenant possible de contrôler ces variations grâce à une ergonomie particulièrement soignée. Le premier objectif de ce logiciel - outre ses capacités d'analyse - est d'établir expérimentalement une hiérarchisation rigoureuse des bruits et des silences permettant entre autre d'isoler (tout en les hiérarchisant) ceux qui sont imputables à l'absence de lexique, ou encore à celle de listes de mots présentant certaines irrégularités morphologiques (le cas extrême étant celui des atomes qu'on ne peut en aucune manière ramener à un schème). Mais revenons à la stabilité du système morphologique car il s'agit là d'un point important.

Pour fournir un fondement théorique solide à notre méthode de construction des grammaires et à la modélisation informatique qui lui est associée nous avons été conduit à exprimer cette stabilité en termes d'invariance. Le Chapitre I de cet ouvrage : Système morphologique et monoïde syntaxique, constitue un exposé de ce point de vue et la proposition par laquelle nous concluons ce chapitre n'en est que l'expression algébrique. Il nous semble en effet que l'une des caractéristiques fondamentales de l'arabe est que les "opérateurs" de projection et de catégorisation commutent : nous catégoriserons toujours de manière à ce que la construction de la grammaire ne soit pas affectée par l'opération qui consiste à réduire la langue à ses seuls paradigmes (les schèmes). Autrement dit notre grammaire doit respecter un principe d'invariance : il doit être indifférent de catégoriser d'abord et de projeter ensuite, ou vice-versa. La construction d'un programme informatique fonctionnant sans lexique n'est que l'exploitation de cette propriété. L'exposé rigoureux de ce point de vue suppose la notion de congruence syntaxique ainsi que celle de congruence associée aux schèmes, que nous introduisons au chapitre I. Mais s'il est en général plus élégant de travailler dans un cadre axiomatique, force est de constater que cela n'est pas toujours indispensable ; c'est la raison pour laquelle nous abordons directement, dès le chapitre II du présent ouvrage, " Les premières modélisations de la morphologie arabe dans le cadre de la théorie des automates ", et au chapitre III, les méthodes de constructions des automates de reconnaissance des atomes. Les chapitres IV, V et VI, où sont essentiellement traités les problèmes relevant de la morphologie nominale, vont nous permettre de pénétrer dans le vif du sujet. La distinction entre système morphologique de base et système général s'y trouve en effet explicitée, ainsi que ses principales conséquences méthodologiques et informatiques : le système de base y est décrit par un langage régulier tandis que le système général en est simplement déduit par une transduction finie et " minimale ". Nous y établissons également la nécessité de compromis en vue d'atteindre une description du système de base, d'un niveau de complexité raisonnable (il existe des centaines de paradigmes). Nous tentons ensuite de déterminer le coût que représentent ces compromis en termes d'ambiguïtés, de

bruits, de silences et enfin de temps d'exécution du programme, coût que l'on cherchera justement à optimiser par la méthode de variation de grammaire. Cette méthode, à laquelle nous aurons recours tout au long de ce travail, doit être étendue aux transducteurs par lesquels on décrit le système général (chapitre V) ainsi que le vocalisateur - ou voyelleur - automatique (sans lexique), que l'on désignera souvent sous le nom de transducteur morphologique (chapitre VI). Dans le chapitre VII, nous ne proposons qu'une description simplifiée des automates sous-jacents des formes verbales normalisées. Les principaux termes de la problématique linguistico-algorithmique étant essentiellement les mêmes pour les noms que pour les verbes, il nous a semblé en effet inutile de prolonger ce dernier, par des traductions systématiques, en termes d'automates, de raffinements descriptifs, lesquels sont par ailleurs assez faciles à trouver dans bon nombre d'ouvrages traitant de la grammaire de l'arabe.

Il serait en revanche très peu efficace de monter en parallèles les trois automates - et/ou transducteurs - représentant respectivement la grammaire des atomes (chapitre III), la morphologie nominale (chapitres IV, V et VI) et la morphologie verbale (chapitre VII), en vue d'obtenir un analyseur morphologique fonctionnant dans l'ignorance totale de son environnement. L'indéterminisme risquerait alors d'atteindre un seuil prohibitif. C'est la raison pour laquelle nous nous intéressons dans les chapitres suivants (VIII et IX) à la conception d'un moniteur syntaxique dont la fonction principale sera celle d'un aiguilleur. L'étude que constituent ces deux derniers chapitres nous permettra ainsi d'envisager la construction d'un premier programme de guidage de l'analyseur morphologique, mais nous signalons dans la conclusion que le raffinement de la fonction d'aiguillage ne pourra s'accomplir sans un travail préalable sur les méthodes d'évaluation des différents fragments de grammaire - issus d'un modèle de base, que nous n'avons d'ailleurs pas encore figé. Ainsi s'ouvre un champ de recherches complémentaires, aussi bien expérimentales que théoriques, qui nous permettra peut-être, non seulement de préciser les principes de l'évolution du moniteur mais d'en élucider également la théorie.

Si nous signalons dans la deuxième partie de cette introduction l'intérêt de recourir à un moniteur morphosyntaxique, nous soulignons toutefois les difficultés que soulève une modélisation de la morphosyntaxe. Là encore, la méthode de "variations de grammaire" est la solution que nous proposons aux difficultés soulevées. La définition d'un cadre algébrique précis nous permet de mettre en oeuvre cette méthode. Toutefois des exemples d'application de certains calculs algébriques à des problèmes d'analyse de l'arabe, se trouvent exposés dès le chapitre IV du présent ouvrage.

Cependant, pour déterminer concrètement la grammaire optimale relativement à une application donnée la méthode expérimentale se révèle indispensable, mais pour pouvoir l'exploiter la conception et le développement d'outils informatiques particuliers sont nécessaires. Les raisons pour lesquelles nous avons cherché à réaliser nous-même ces outils se trouvent rapidement exposées dans la troisième partie de cette introduction et apparaîtront de toute manière de plus en plus clairement à la lecture de cet ouvrage.

Bien que les principaux aspects algébriques et informatiques de ce travail soient exposés dans « Modélisation Linguistique et Théorie des Automates » (ref. 2), nous avons choisi, afin de souligner les spécificités du traitement automatique de l'arabe, de ne pas séparer totalement l'exposé des problèmes linguistiques de celui de l'algorithmique qui leur est associée. Ainsi les programmes que nous avons écrits en LISP ou même les interfaces que nous avons réalisées se trouvent juste mentionnés ou très succinctement expliqués au fur et à mesure que sont exposés les problèmes linguistiques rencontrés. Nous mentionnerons par exemple au chapitre III, qui traite des atomes (les mots qui ne possèdent pas de paradigmes) et de leur reconnaissance, le programme de factorisation des expressions régulières ainsi que le programme qui construit l'automate déterministe minimal correspondant. L'ensemble de ces programmes permet à partir d'un schéma de grammaire, déduit d'une classification obéissant à des critères linguistiques précis de concaténation, de construire une véritable grammaire susceptible d'être mise en œuvre informatiquement (grâce à un analyseur). De la même manière, les versions d'analyseurs permettant de prendre en charge des grammaires "augmentées" sont mentionnées dans les chapitres traitant de "La morphologie nominale". Ces versions modifiées d'analyseurs vont nous permettre de traiter des formes où interviennent des racines non saines. Quant aux nouvelles versions de l'analyseur qui sont en train d'être implémentées (mode interprétatif ou semi interprétatif), elles ne seront évoquées que lorsque seront abordés les problèmes de "voyellation" automatique et de hiérarchisations des ambiguïtés. L'utilité de tels outils n'apparaîtra que plus clairement, si la difficulté qu'ils sont censés résoudre est exposée en même temps .

Nous tenons enfin à préciser que le concept d'atelier de grammaires que nous avons mentionné plus haut (voir partie III de cette introduction) ne s'est imposé à nous que lorsque nous avons cherché à regrouper toutes nos fonctions et à les hiérarchiser. C'est lorsque nous avons fait la synthèse de notre travail informatique qu'il nous est apparu clairement que l'ensemble de nos programmes pouvaient constituer le noyau d'un système de gestion et de traitement automatique des grammaires. C'est donc le regroupement de nos fonctions dans un ensemble organisé (ou système) qui est à l'origine de l'idée de faire correspondre à chaque solution apportée à un problème de modélisation linguistique une "navigation" dans ce système.

Cet atelier de grammaire qui nous permettra donc d'expérimenter et de manipuler des grammaires doit à son tour être compris comme le noyau d'un nouveau système - qui reste à réaliser - devant permettre de faire de "la programmation linguistique". En effet, à ce stade de développement, il est déjà possible d'envisager la création d'un langage de programmation permettant de définir dans un programme une séquence d'opérations sur les grammaires et autres objets linguistiques, alors qu'actuellement ces opérations ne peuvent être commandées et exécutées que séparément .

De la même manière qu'il existe, sur le plan informatique, une suite logique à nos activités qui est la réalisation d'un système qui nous permette, à partir d'une base de connaissances linguistiques, d'engendrer automatiquement des applications (voir ci-

dessous, partie III de cette introduction), il en existe également une sur le plan strictement linguistique. En effet, la nécessité de recourir à un moniteur morphosyntaxique nous amène à étudier le Groupe Nominal arabe qu'il soit déterminé ou non (chapitre X), ainsi que les "attentes" syntaxiques déclenchées par certains éléments (les atomes). Or l'étude de ces éléments qu'il nous arrivera aussi de désigner par le terme de "tokens" emprunté à la théorie de la compilation fait apparaître leur statut d'opérateur (au sens de Harris) lesquels s'appliquent à des phrases élémentaires ou noyaux. L'idée d'exploiter les treillis de décomposition de Harris dans une optique d'informatique documentaire n'est pas nouvelle, mais sa mise en application est loin d'être évidente. Il suffit de songer en effet aux difficultés de mettre au point des analyseurs pour grammaires transformationnelles. Toutefois, on pourrait tenter de définir un sous-ensemble de l'arabe qui nous permette d'une part de dialoguer avec une banque de données et qui d'autre part, soit tel que le processus d'analyse à mettre en œuvre pour obtenir la décomposition des phrases en phrases noyaux ou phrases élémentaires ne soit pas d'une complexité ingérable. Une fois obtenu des phrases élémentaires on pourrait alors leur associer des "formules sémantiques" combinables entre elles<sup>1</sup>.

Par ailleurs, le langage squelette de l'arabe, c'est-à-dire l'ensemble des associations licites de paradigmes (voir ci-dessous, partie II), constitue un objet linguistique qui, à notre connaissance n'a jamais été étudié en tant que tel, notamment sur le plan sémantique (pour ce qui est par exemple des invariances par rapport à la projection). Mais il s'agit là d'un champ d'étude qui dépasse le strict cadre de cet exposé. Nous ne voulions en fait qu'indiquer certaines perspectives, certains thèmes de recherches qui nous semblent être dans le prolongement direct de ce travail.

Cet ouvrage se divise en trois parties. Étant donné le rôle essentiel que nous conférons au principe d'invariance, à partir duquel découle la logique de notre exposé, il nous a semblé naturel d'isoler le premier chapitre sous le titre plus général d'Invariances, puisqu'il constitue, à lui seul, un exposé de celles qui nous semblent les plus caractéristiques du système de l'arabe. L'explicitation algébrique de ce principe nous fournirait par ailleurs une belle jonction théorique avec la notion d'automate largement utilisée dans ce travail. Les parties suivantes traitent principalement des objets de ces invariances, que ce soit au niveau de la morphologie (deuxième partie) ou de la syntaxe (troisième partie). Mais le plan que nous venons rapidement de présenter ne sera entièrement justifié qu'après un exposé plus rigoureux des structures d'invariance auxquelles nous avons fait allusion. Nous y reviendrons donc dans la conclusion du chapitre I.

---

<sup>1</sup> Il ne s'agit que d'exemples, parmi d'autres, d'extensions de cette étude vers des applications sémantiques.

## 2. Les objectifs initiaux et la logique de ce travail

### 2.1 Régularité du système morphologique

Le système morphologique ainsi que le système d'écriture de l'arabe, s'ils présentent dans un premier temps quelques difficultés pour l'analyse automatique, ne possèdent pas moins une stabilité remarquable qui nous permettra de les décrire entièrement au moyen d'un langage formel. Rappelons en effet, que les phénomènes de concaténation (agglutination) sont importants puisqu'il existe une dizaine de zones dans un mot graphique maximum. Cela signifie que les procédures de reconnaissance de type combinatoire (problèmes de segmentation) nécessiteront un calcul plus important que celui qui est généralement mis en œuvre dans le traitement d'autres langues. Par ailleurs, pour reconnaître la forme (le paradigme) d'un mot, il est nécessaire d'en isoler la racine qui ne constitue pas toujours à l'intérieur du mot graphique un ensemble connexe de graphèmes.

En revanche, le système morphologique possède une grande régularité car :

- les racines sont presque toujours trilitères,
- les schèmes (très nombreux) peuvent être décrits de manière formelle,
- les concaténations d'éléments obéissent à des règles strictes de précedence et de compatibilité,
- enfin, les règles de permutation des semi-consonnes *alif/A*, *wāw/U* et *yā'/I*, quoique complexes, sont formalisables, tout comme les règles purement phonologiques (exemple: *izdihār*).

Le mot graphique possède donc une grammaire. La fonction du programme morphologique sera justement de construire l'arbre interne correspondant à cette grammaire. Il s'agit donc essentiellement d'un problème de parsing.

En d'autres termes, la morphologie arabe constitue un langage formel dont le vocabulaire de base est constitué des graphèmes principaux de l'alphabet arabe et, selon que l'on considère des textes vocalisés ou non, un ensemble de symboles représentant les signes diacritiques.

Ce fait a une conséquence extrêmement importante; le programme morphologique est construit indépendamment du lexique.

Il s'agit là d'une caractéristique majeure de ce projet : nous avons été guidés tout au long de ce travail par le principe du dictionnaire vide, ce qui revient à raisonner sur des phrases squelettes, obtenues à partir de phrases réelles en réduisant toutes les racines à un seul représentant R1 R2 R3. Cela signifie que l'on considère la transformation  $L \rightarrow L/RAC$  où L représente la langue naturelle, et L/RAC, l'ensemble obtenu en "quotientant" l'ensemble des séquences de L par l'ensemble des racines.

À notre sens, la caractéristique majeure de l'arabe - voire des langues sémitiques - est que l'ensemble quotient L/RAC obtenu, constitue à son tour un langage.

Dans cette transformation, les algorithmes de reconnaissance morphologique (ainsi que ceux qui traitent une partie non négligeable de la syntaxe demeurent invariants autrement dit, ils ne sont pas affectés par les permutations de racines (qui on le rappelle, constituent les entrées lexicales principales dans un dictionnaire). La morphologie arabe pourra être entièrement décrite par des automates finis sans boucle dont les arcs de transition sont étiquetés par des éléments de S ou bien encore, par une seule expression régulière sans étoile de Kleene (il n'existe pas de mots infinis).

## 2.2 La nécessité d'un moniteur syntaxique

La grande difficulté à laquelle on se heurte, lorsqu'on traite automatiquement l'arabe, provient d'une règle d'écriture qui veut que les mots ne soient presque jamais vocalisés ; ainsi, le mot "avenir", *mustaqbal*, se note *mstqbl*.

Nous désignons par décodeur morphologique, l'analyseur morphologique du texte arabe non vocalisé. Toutefois, un programme de décodage morphologique conçu pour fonctionner mot à mot (sans tenir compte du contexte) laisse subsister un grand nombre d'ambiguïtés - qui, pour une part non négligeable sont dues à la devocalisation (dé voyellation). Un grand nombre de ces ambiguïtés peuvent être levées par le recours au contexte, d'où la notion de moniteur syntaxique qui guidera le programme morphologique. Ses principales fonctions seront donc:

1. de lever les ambiguïtés susceptibles d'affecter les entrées dans un dictionnaire (ambiguïtés de la racine);
2. de lever les ambiguïtés des schèmes (à l'intérieur du dictionnaire);
3. d'optimiser le programme morphologique; et
4. de déterminer une stratégie de décodage.

## 2.3 Le principe de construction du moniteur syntaxique:

Dans la transformation  $L \rightarrow L/RAC$  mentionnée plus haut, certains éléments demeurent invariants; ce sont justement les éléments que l'on ne peut rattacher au système général de dérivation, et que l'on ne peut, par conséquent, ramener automatiquement à une racine  $R_1 R_2 R_3$ . Ces éléments (au nombre de 150 environ) qui échappent aux règles de formation du mot, c'est à dire à la grammaire, devront donc nécessairement figurer dans un lexique.

Cette classe d'éléments recoupe une catégorie d'éléments que l'on désigne en général par "mots outils". Dans une optique d'informatique documentaire, on les désigne en général par le terme "mots vides" car il s'agit d'une catégorie de mots qui n'apportent pas (ou peu) d'informations (sémantiques), et que l'on regroupe justement dans un anti-dictionnaire. Or, la "vacuité" de ces éléments, est en général en relation avec les contraintes qu'ils induisent sur leur environnement. En fait nous verrons que

cet état de chose peut être immédiatement déduit de l'invariance que nous venons de mentionner.

Ces atomes ont donc une répercussion syntaxique importante, soit qu'ils engagent la globalité de la phrase, soit qu'ils aient une incidence sur l'environnement immédiat, ce qui pose le problème de leur hiérarchisation.

Le fait que nous ayons focalisé notre attention sur ces éléments pourrait aussi se justifier par la remarque suivante : lorsqu'on décode un message, il faut d'abord repérer l'élément de ce message qui engage le plus grand environnement; il faut dégager en premier lieu, les plus forts structurants qui réduisent d'un coup le niveau entropique (c'est à dire, le degré d'indétermination du flux informationnel), donc ceux qui introduisent les contraintes les plus importantes. Il semble ainsi naturel d'associer à chacun de ces éléments un automate (en général récursif), décrivant les attentes qui lui sont attachées ; ce qui revient à construire le noyau d'une grammaire indépendante du contexte à partir de ces éléments .

La grammaire ainsi définie, sur laquelle s'appuiera le moniteur syntaxique, reflètera alors un sous-ensemble des contraintes syntaxiques de L/RAC. On voit ici que la définition de L/RAC est essentielle dans la constitution du noyau de la grammaire.

## **2.4 Difficultés et solutions proposées**

Les automates récursifs, associés à cette grammaire, auront tous leurs arcs terminaux, correspondant normalement à des recherches lexicales, remplacés par des sous-programmes morphologiques. Or ces derniers doivent justement faire appel au moniteur syntaxique dans de nombreux cas pour arriver à un niveau satisfaisant de résolution. L'étude de l'interaction entre ces deux niveaux est donc cruciale mais elle n'est pas des plus aisée. Il faut en effet éviter les cercles vicieux, c'est-à-dire que ne se produisent des situations telles que les deux processeurs morphologiques et syntaxiques soient en attente l'un de l'autre ou bien encore des situations absurdes, par exemple que le procédé d'optimisation soit plus complexe que le processus à optimiser.

C'est pour se donner les moyens de résoudre ces difficultés, tout en conservant une marge suffisante de liberté au niveau de la modélisation linguistique, que l'on a choisi de travailler dans des structures algébriques précises qui nous permettent d'effectuer des calculs sur les grammaires. Mais pour bien cerner les passages d'information entre la syntaxe et la morphologie, il était nécessaire avant tout de donner de cette dernière une description cohérente, qui soit compatible avec les représentations syntaxiques. Pour cela nous nous sommes toujours efforcé de rester au niveau le moins élevé dans la hiérarchie des grammaires (automates et/ou transducteurs), dont les propriétés mathématiques sont bien connues. L'aspect algébrique inhérent à ce travail ainsi que l'algorithmique sous-jacente est exposée de manière plus explicite dans « Modélisation linguistique et théorie des automates » (ref.2).