Chapitre I

La *minimalité*, méthode générale d'approche du traitement automatique de l'arabe.

L'architecture originale pour le traitement automatique de l'arabe proposée par Claude Audebert et André Jaccarini constitue la base de nos propres recherches au sein de l'équipe du DATAT, base à partir de laquelle nous tenterons dans cette étude tantôt un approfondissement des questions soulevées, tantôt une extension vers de nouvelles perspectives, mais aussi parfois des remises en question. C'est grâce à la réalisation d'un logiciel original, *Sarfiyya*, que nous avons pu développer notre point de vue, tenter de répondre à certaines questions et définir le contour de futures applications.

Les travaux sur le traitement automatique de l'arabe réalisés par le DATAT ont été publiés depuis 1986 sous formes d'articles. A. Jaccarini y a consacré une thèse, «Grammaires modulaires de l'arabe»[1] soutenue en 1997 et actuellement sous presse. Ces travaux ont fait également l'objet de plusieurs communications dans des congrès d'arabisants, de linguistes ou de spécialistes du traitement automatique des langues. Nous désignerons désormais par le terme de *minimalité* cette approche du traitement de l'arabe. Nous avons souhaité en formuler les objectifs principaux, et en préciser la démarche à la lumière de ces travaux, avec un souci de synthèse: cette présentation se concentre sur l'essentiel et n'abordera pas nombre détails pour lesquels nous renvoyons aux publications. À l'issue de cet exposé, nous montrerons ceux des points qui semblent, à notre sens, nécessiter un approfondissement, soit

1. [Jaccarini 97].

parce qu'ils sont les plus fragiles théoriquement et réclament une vérification statistique, soit parce qu'ils n'ont pas ou insuffisamment été mis en œuvre informatiquement. Ce bilan s'apparentera à l'annonce de nos propres objectifs et travaux.

I. Les objectifs de la minimalité

1. Aux sources de la minimalité: l'enseignement de l'arabe.

L'approche novatrice du traitement automatique de l'arabe amorcée par C. Audebert et A. Jaccarini débute par une réflexion sur l'apprentissage de l'arabe par les apprenants de cette langue, et propose de l'améliorer en la modélisant. L'article fondateur «À la recherche du *Ḫabar*, outils en vue de l'établissement d'un programme d'enseignement de l'arabe assisté par ordinateur»[2] consigne des observations issues d'une large expérience d'enseignement de l'arabe à des quasi-débutants. Le manque de vision globale de la phrase par les apprenants est expliqué en partie par la difficulté de reconnaissance des mots graphiques. Le doigt est donc mis sur une difficulté majeure, l'appréhension des flux d'information entre la syntaxe et la morphologie et leur «masquage» partiel par la non-voyellation généralisée. Le recours est le repérage prioritaire des mots-outils, dits *tokens*, et l'exploration de leur rôle de structurant dans la phrase: c'est une approche de *surface*. Ces constatations amèneront les auteurs à proposer, au moyen d'un modèle représenté par un programme fictif, une stratégie originale de décodage de la phrase préconisant le rôle directeur de la syntaxe. Cette stratégie s'appuie sur le rôle phare des tokens, s'abstient de recourir au lexique et, d'une manière générale, n'utilise qu'un minimum de règles. La vision de la phrase arabe vidée de ses occurrences lexicales et ne contenant que les tokens sert de support à l'élaboration d'hypothèses sur la structure de la phrase.

Ce point de départ souligné, nous ne poursuivrons pas ici une description de cette stratégie épousant le fil chronologique des publications, mais préférerons exposer une synthèse exprimée en termes d'*objectifs*, de *principes fondateurs* et de *méthodes*, permettant par un examen rapide puis détaillé d'embrasser d'un coup d'oeil

2. [Audebert, Jaccarini 86] p. 218.

l'essentiel de ses articulations.

2. Objectifs de la minimalité

O1. Objectif principal: traitement de l'arabe

Parvenir à une analyse morpho-syntaxique de la phrase arabe, sans toutefois en explorer la sémantique, par une exploitation optimale et une description algorithmique pertinente de toutes les régularités morphologiques et syntaxiques de la langue, sans recours au lexique ni à une liste exhaustive de règles de syntaxe.

O2. Objectif corollaire: connaissance linguistique de l'arabe

Connaissance des mécanismes de fonctionnement de l'arabe.

O3. Objectif corollaire: enseignement de l'arabe

Guider les processus d'appréhension de la phrase arabe par l'apprenant.

Ainsi exposée, la minimalité affiche un premier objectif ambitieux, préalable à tout traitement automatique des langues naturelles: une analyse syntaxique des composants de la phrase, que l'on représente communément sous forme d'arbre syntaxique. Cette analyse est très étroitement liée aux données morphologiques perçues dans leur caractéristique originale, la régularité. Mais elle va plus loin en privilégiant une règle, en proposant un postulat, dont l'ambition épistémologique est exprimée avec les objectifs O2 et O3.

Une modélisation théorique de la minimalité a été esquissée par A. Jaccarini dans sa thèse à propos de la morphologie; nous aurons l'occasion de l'évoquer plus amplement. Cependant une présentation axiomatique complète de la minimalité, qui s'appuierait donc sur la linguistique arabe, l'algorithmique et des modélisations informatiques, s'accommoderait mal, à notre sens, du sens profond de cette démarche. Il s'agit plus en effet d'un état d'esprit ou d'un principe général que d'une théorie, dont l'axiomatisation excessive serait nécessairement artificielle et incomplète. La minimalité est un pari dont les idées directrices doivent être comprises comme autant de pistes génératrices de méthodes. Elle s'affirme d'emblée comme un cadre d'échanges entre modélisation linguistique et optimisation de procédures de traitement automatique. Aussi, au risque de déplaire aux esprits les

plus formalistes, nous bornerons-nous à exposer l'essentiel de la problématique et des méthodes explorées en en formulant les principes constitutifs tels que nous les percevons et tels qu'ils apparaissent dans les travaux du DATAT.

3. Principes fondamentaux

Ces principes peuvent être regroupés en trois axes principaux: M, S et I désignant respectivement Morphologie, Syntaxe et Ingénierie linguistique ou traitement automatique. Ils peuvent se résumer ainsi:

– **M** Régularité de la morphologie permettant de s'affranchir du lexique;

– **S** Primauté de la syntaxe appuyée par le rôle central des mots-outils;

– **I** Ingénierie linguistique: reflet des principes M et S par le formalisme des AFND et des ATN[3].

Il n'existe pas de cloisonnement étanche entre ces axes. Une présentation linéaire de ces principes est le seul moyen d'exposer rapidement l'ensemble des idées, avant leur explicitation.

M – Régularité de la morphologie permettant de s'affranchir du lexique

M1 Régularité du système trilitère sain

Le système morphologique trilitère sain interne et externe est régulier.

M2 Extension de la régularité

Cette régularité s'étend à la morphologie trilitère dans son ensemble; or cette morphologie couvre, avec ses avatars bi-, quadri- et quinquilitère la grande majorité des mots du lexique.

M3 Légitimité de la catégorisation morpho-syntaxique

Le découpage du lexique arabe en schèmes est compatible avec la définition des catégories syntaxiques de la morphologie; il existe une borne inférieure pour cette catégorisation.

M4 Effacement du rôle du lexique

Des premiers principes découle la possibilité de construire des analyseurs morphologiques à large couverture, capables d'extraire un maximum d'information

---

3. AFND: Automate Fini Non Déterministe; ATN: Augmented Transition Network ou RTA, Réseau de Transitions Augmenté. Ces termes seront définis et explicités lors de l'exposé de I1.

morphologique à partir d'un minimum de recours au lexique.

M5 Existence d'un noyau d'invariants morphologiques

Il existe un ensemble de mots appelés *atomes*, au nombre très réduit d'éléments par rapport à l'abondance du lexique, et qui constituent les invariants du système morphologique: ni les règles de dérivation par schèmes ni les désinences casuelles ne s'y appliquent; en revanche, certains éléments obéissent à des règles de la morphologie externe.

S – Primauté de la syntaxe appuyée par le rôle central des mots-outils

S1 Les invariants morphologiques sont des tokens syntaxiques

Ces *atomes* au regard de la morphologie sont aussi des *tokens* au regard de la syntaxe: leur présence déclenche un certain nombre d'«attentes» syntaxiques, qui dispensent d'une analyse à caractère ascendant pour isoler les groupes syntaxiques. Ce rôle est visible lors de l'opération de *phrase vide*, où l'on supprime les occurrences lexicales pour ne garder que les tokens.

S2 Règles syntaxiques dominantes

Les «attentes» syntaxiques sont autant de fragments de syntaxe dont l'identification fait «chuter» l'indéterminisme des analyseurs syntaxiques; certaines d'entre elles offrent ainsi une vision globale de la phrase à partir d'un seul élément, laissant supposer l'existence d'une hiérarchie de tokens et de règles. La surabondance de l'information morpho-syntaxique est contournée par une approche *en surface*.

I – Ingénierie linguistique et traitement automatique: reflet des principes M et S par le formalisme des AFND et des ATN

I1 Morphologie et automates

Le principe de la régularité, réelle ou «étendue», rend naturel l'emploi des AFND simples et augmentés ATN non récursifs pour l'analyse morphologique. Il est cependant inévitable de rechercher un compromis dans cette modélisation par transduction, compromis entre complexité et couverture réelle des éléments du lexique.

I2 Syntaxe et automates

La description des attentes syntaxiques peut être menée grâce à l'emploi d'AFND et d'ATN, et fonde une classe d'analyseurs syntaxiques. Le groupe nominal GN/GND se

prête, après dérécursivation, à une description par automate.

I3 Moniteur morpho-syntaxique

Le pilotage de l'analyse morpho-syntaxique peut s'effectuer au moyen d'un programme cadre, le moniteur, dont le rôle est d'orchestrer en s'appuyant sur les tokens le dialogue entre les analyseurs syntaxiques et morphologiques.

I4 Variation des grammaires et réalisations informatiques

La mise au point des transducteurs évolue en fonction de l'objectif de traitement automatique que l'on s'est fixé, et doit être affinée par le programmeur-linguiste dans ce but. Il existe plusieurs manières de programmer les analyseurs d'automates, dont certaines reflètent la résolution d'équations algébriques «grammaticales». Disposer de systèmes ouverts afin de reconstituer les parcours d'automates présente un intérêt pédagogique.

I5 Mesure et évaluation des grammaires

Il est nécessaire de mener des analyses à échelle raisonable de textes variés, courts et préalablement étudiés pour mesurer précisément à l'aide d'outils statistiques le comportement des grammaires.

II Exposé des principes fondamentaux

Nous adopterons un ordre de présentation par thème de ces principes laissant entrevoir les déductions possibles, les passages et échanges d'informations entre les différents principes. Ces relations seront récapitulées à la fin de ce chapitre, en guise de conclusion.

1. Morphologie et automates : M1, I1 et M2

M1. Régularité du système trilitère sain

1. Définitions élémentaires

• Σ Ensemble des lettres de l'alphabet arabe.

Plusieurs sous-ensembles de Σ seront utilisés:

•$\Sigma_c = \Sigma - \{$و, ا, ى, ي, ة$\}$Sous-ensemble des lettres de l'alphabet arabe constitué des consonnes y compris la *hamza*.

•$\Sigma' = \Sigma - \{$ة$\}$

Sous-ensemble des lettres de l'alphabet arabe constitué des lettres susceptibles de

représenter la racine d'un mot dans sa réalisation graphique.

•$\Sigma_r = \Sigma - \{ة, ا, أ, إ, ؤ, ئ\} = \Sigma_c + \{و, ى\}$

Sous-ensemble des lettres de l'alphabet arabe constitué des lettres susceptibles de former la racine d'un mot par leur réunion en n-uplet, n variant de 2 à 5.

•$\Sigma_r' = \Sigma r - \{ء, و, ى, ي\} = \Sigma_c - \{ء, أ, إ, ؤ, ئ\}$

Sous-ensemble des lettres de l'alphabet arabe constitué des lettres susceptibles de former la racine *saine* d'un mot par leur réunion en n-uplet, n variant de 2 à 5.

• **Lar**

Ensemble des mots arabes graphiques, c'est dire des mots compris entre deux blancs dans les textes arabes. Plus formellement, un sous ensemble strict d'éléments du monoïde libre Σ* muni de l'opération de concaténation.

• **Sous-chaîne**

Chaîne de caractères issue d'un élément de Lar par segmentation, cette dernière opération se définissant comme une succession d'effacements de lettres à droite ou à gauche du mot.[4]

• **Morphologie**

Ensemble des contraintes de liaison entre les symboles de la graphie arabe en vue de former un mot licite[5].

• **LEX**

Première définition: ensemble des lexèmes, éléments du lexique arabe.

L'étude de LEX, qui retiendra ici notre attention; se décompose en deux parties distinctes:

• **Morphologie externe**

Étude de l'appartenance à LEX des sous-chaînes des éléments de Lar. Par delà, étude de la définition de LEX.

• **Morphologie interne**

Étude de la structure interne des éléments de LEX.

Ces définitions résultent d'une approche «graphique» et donc surfacique et

4. C'est l'opération inverse de la concaténation: voir [Jaccarini 97] chapitre IV, I.3.

5. [Jaccarini 97] chapitre I, I.1.2.

directement appréhendable de la morphologie arabe. Il ne saurait en être autrement: les textes, «objets» soumis au traitement automatique, sont constitués de chaînes de caractères représentant, dans une graphie convenue et selon un code univoque, l'écriture arabe graphique.

2. Morphologie interne

2.1 Racine

La majorité des éléments de LEX sont issus d'une racine[6]. Cette racine est, dans la majorité des cas, une racine trilitère, c'est-à-dire un triplet appartenant à $\Sigma_r \times \Sigma_r \times \Sigma_r = \Sigma_r^3$. On note la racine trilitère $<$r1,r2,r3$>$.

Il existe également des racines bilitères, quadrilitères, quinquilitères, constituées respectivement de paires, quadruplets et quintuplets d'éléments de Σ_r, mais elles sont statistiquement minoritaires. À titre indicatif, l'ensemble de notre corpus référencé (voir chapitre III) comporte moins de 1% de racines bilitères ou quadrilitères.

Les racines trilitères sont dites «saines» lorsque:

pour tout i, ri $\in \Sigma_r$' et r2 $\neq$ r3 ; ce dernier cas exclut les racines dites «sourdes», pouvant produire des formes geminées.

• **RAC** désigne l'ensemble des racines saines.

Tous les n-uplets, n variant de 2 à 5, de Σ_r^n ne sont pas pour autant des racines attestées dans la langue. Nous rapportons ici les statistiques de nombre de racines collectées par *ʿAlī Ḥilmī Mūsā*[7] et Ayadi Chabir[8], qui portent sur trois dictionnaires

---

6. Nous ne tenterons pas une définition complète de la racine, cette question est laissée aux linguistes, théoriciens et historiens de la langue arabe. Ces points de vue, parfois très opposés à l'image de la controverse bilitère-trilitère, concernent avant tout l'étymologie. Nous nous plaçons ici du point de vue de l'effectivité: la notion classique de racine, utilisée par les principaux dictionnaires comme entrée lexicale, est celle qui rend compte le mieux de la grande majorité des phénomènes de la morphologie arabe. Une constatation s'impose en effet: la racine est un concept en partie imprécis, mais dont l'usage ne fait pas de mystère pour des millions d'arabophones et d'arabisants.
7. [Musa 71].
8. [Chabir 97] chapitre 4.

médiévaux:

Type de racine	**2**	**3**	**4**	**5**	**total**
Al-Zubaydī	489 (sic)	4269	820	42	5640
Al-Ṣiḥāḥ	21	4814	768	38	5641
Lisān al ʿarab	nd	6507	2742	nd	9249

Une majorité des triplets formant les racines trilitères répondent à des règles de compatibilité phonologique[9] concernant chacune des paires de consonnes, mais il existe de nombreuses exceptions à ces règles, imputables par exemple aux emprunt étrangers.

2.2 Schème

La notion de racine ne se conçoit pas sans celle de schème: un élément de LEX duquel on peut isoler une racine apparaît comme le «produit» de cette racine et d'un schème, selon le processus suivant:

– adjonction de lettres consonnes ou voyelles convenues, en positions éventuelles de préfixe, infixe et postfixe conservant l'ordre ri mais selon un agencement précis susceptible d'être appliqué à d'autres racines. L'ensemble des schèmes, fini, est noté **SC**.

– Les cas de conflits entre voyelles longues et voyelles brèves sont réglés par des règles complexes de permutation, d'élision et d'effacement; de même, certaines consonnes radicales influent sur les lettres du schème par assimilation phonologique. Le produit, brut, est donc *lissé* pour revêtir une forme prononçable sans heurts et répondre aux canons de la phonologie.

Opérateur ou structure commune, le schème apparaît comme un «paradigme de lexème».

2.3 Invariants

Deux catégories de mots échappent à ce système «radical»:

– les atomes ou mots outils, au nombre réduit: voir *infra*, le principe M6.

---

9. [Greenberg 50].

– Les emprunts aux mots étrangers, récents ou anciens, comme nombre de noms propres ne subissent pas ou partiellement de processus d'«arabisation»: ils sont souvent de simples transcriptions phonétiques.

3. Aspects formels de la morphologie interne

Un premier schéma de modélisation peut être entrepris en se restreignant à la morphologie interne saine. S'affranchir dans un premier temps des multiples règles de permutation et de «lissage phonologique» évoquées en plus haut conduit à une vision simplifiée, mais met en valeur les structures essentielles de LEX.

LEX désignera donc dans ce qui suit l'ensemble des lexèmes issus de racines saines.

Les principaux actes de cette formalisation se résument ainsi[10]:

3.1 le schème est un opérateur.

La production d'un élément de LEX à partir d'une racine et d'un schème peut être vue comme l'application d'un opérateur σ à un élément valide et sain de Σ_r^3. Le formalisme des lambda-expressions reflète la transformation associée aux schèmes:

$(\lambda\ r_1\ r_2\ r_3.\text{م}\ r_1\ r_2\ |\ r_3)\ <\text{ح، ت، ف}> = \text{مِفتاح}$,

où le premier membre de l'expression s'apparente à l'opérateur σ.

On peut alors définir l'opérateur inverse σ-1, tel que:

$\sigma^{-1}.\sigma\ <r_1,r_2,r_3> = <r_1,r_2,r_3>$

σ-1 correspond donc à l'opération mentale d'«extraction de la racine» saine d'un mot.

3.2 Partition de LEX en classes de racines.

Il est possible de définir une application Φ de LEX dans l'ensemble des racines saines, augmenté de 0:

pour tout m,si m atome, $\Phi(m) = 0$

sinon, $\Phi(m) = s^{-1}.m$

En effet, la règle dite de «spécialisation lexicale» peut être employée comme ultime recours pour trancher des cas pour lesquels au moins deux racines semblent plausibles, mais ce recours est statistiquement peu courant.

Cette application s'étend naturellement en relation d'équivalence ainsi définie:

quel que soit x, y ∈ LEX, x R y $<=>$ $\Phi(x) = \Phi(y)$

---

10. [Jaccarini 97] chapitre I, p. 29-34.

Les classes d'équivalences correspondantes ne sont pas homogènes : ce sont, par racine, les listes des schèmes effectivement attestés dans LEX.

3.3 Notion de clôture du lexique

Une homogénéisation de ces classes d'équivalence précédemment définies, obtenue par la complémentarisation des classes prises deux à deux (permutation de racine) étendue à l'ensemble, conduit à définir LEX ou la *clôture du lexique*:

• LEX est le plus petit ensemble contenant l'ensemble LEX et clos par opération de permutation de racine.

SC, l'ensemble des schèmes, peut être redéfini alors comme l'ensemble des plus petites classes non vides de LEX closes par l'opération de permutation de racine.

4. Morphologie externe

Les faits marquants de la morphologie externe arabe peuvent se résumer ainsi[11]:

1. existence d'une concaténation d'éléments avant ou après le mot élément de LEX.

2. Si l'on permute les racines des mots d'une phrase ou si l'on fait abstraction de ces racines pour leur donner un unique repésentant, tel <ف،ع،ل>, la phrase résultante est valide à condition d'accepter comme valides tous les éléments de LEX, et elle est de faible agrammaticalité.

Le tableau suivant rappelle les catégories possibles de ces éléments concaténables. Il est à souligner qu'une catégorie «possible» ne se réalise que dans un contexte syntaxique précis que nous ne mentionnons pas ici.

Catégories possibles des éléments pré et postfixés

	éléments pré-fixés	**éléments post-fixés**

11. [Jaccarini 97] chapitre I, p. 35-43.

N: **Noms &** **dérivés**	interrogatif أ coordonnants وَ, فَ corroborateur لَ prépositions بِ, كَ, لِ article ال article لْ après prépos. لِ	nisba يّ, ويّ féminin ة duel ين, ان pluriel m. ين, ون pluriel f. ات cas direct non déterminé m. اً pronoms personnels ى, كَ, كِ, ه, ها, هُما, نا, كُم, كُما, هُم, هُما, هُنّ
V: **Verbes**	interrogatif أَ coordonnants وَ, فَ corroborateur لَ particule de l'impératif ou subordonnant لِ subordonnant فَ	verbes transitifs: pronoms personnels نى, كَ, كِ, ه, ها, هُما, نا, كُم, كُما, هُم, هُما, هُنّ
T: **Atomes ou** **Tokens**	selon les catégories de tokens: interrogatif أَ coordonnants وَ, فَ prépositions بِ, كَ, لِ corroborateur لَ	pour certains tokens: pronoms personnels

5. Aspects formels de la morphologie externe

L'enjeu est ici d'étendre à la concaténation d'éléments externes les propriétés des classes d'équivalences définies pour LEX. De nouvelles notions et structures sont donc introduites.

Lar est maintenant considéré comme le monoïde libre Σ*, muni de l'opération associative et interne de concaténation et d'un élément neutre conventionnel, e :

quel que soit $x \in \Sigma^*$, $e.x = x.e = x$

La notion de *congruence syntaxique* $\sim L$ se définit ainsi:

soit L sous-ensemble de Σ^*,

$x, y \in \Sigma^*$, $x \sim L\ y$ si et seulement si pour tout $u, v \in \Sigma^*$:

$u\ x\ v \in L <=> u\ y\ v \in L$

On remarque le caractère distributionnel de cette notion, x et y pouvant commuter dans toute séquence où ils figurent sans que l'appartenance (ou la non

appartenance) de cette séquence à L ne soit remise en cause. Cette relation de congruence syntaxique définit à son tour une relation d'équivalence, dont les classes d'équivalences seront dénommées *classes de congruences syntaxiques*.
Un homomorphisme, application «transportant» la loi de concaténation entre deux monoïdes, peut donc être défini comme une projection qui fera correspondre à chaque mot ou suite Σ* sa classe de congruence syntaxique:

$\Pi : \Sigma^* \longrightarrow \Sigma^*/\sim_L$ tel que $\Pi(x.y) = \Pi(x).\Pi(y)$

Après l'élément *neutre* e, introduisons un élément *absorbant* dans la structure de monoïde, Ø: signifiant une «absence de séquence», cet élément est tel que:

quel que soit x, $x . \emptyset = \emptyset .x = \emptyset$

Or certaines séquences de lettres arabes n'appartenant pas à la langue arabe constituent des éléments absorbants de $\Sigma^*/\sim_{Lar}$, si l'on remarque que lorsqu'on les concatène à n'importe quel autre mot, le résultat n'est pas un mot arabe (ex.: ممم). En donnant pour image de ces séquences «absorbantes» par Π précédement défini l'élément Ø, on parvient à la construction d'un homomorphisme permettant de sélectionner les segmentations licites d'un mot graphique[12]. Ce dernier est en effet tel que:

Π(وَلِلْولد) = Π(وَ)Π(لِ)Π(لْ)Π(ولد) où

Π(وَ) = *coordonnant*, Π(لِ) = *préposition*, Π(لْ) = *article*, Π(ولد) = *nom*

mais aussi:

Π(المرصاد) = Π(ال)Π(مر)Π(صاد) = Ø car Π(مر) = Ø, alors que

Π(المرصاد) = Π(ال)Π(مرصاد) où Π(مرصاد) = *nom* est la bonne catégorisation.

Une définition formelle de LEX est désormais possible:

• LEX = {x, $x \in L_{ar}$ tel que pour tout u, v si x = uv alors Π(u)Π(v) = Ø}

Catégories syntaxiques

La notion de congruence syntaxique introduite avec le monoïde Σ* peut être étendue à L_{ar}* et à son sous-ensemble L, la langue arabe. Les étapes 2 et 3 peuvent

12. Cette construction importante est ici résumée dans ses grandes lignes: nous avons ainsi omi par souci d'allègement la notion de "concaténation pertinente" qui fixe des limites de cardinalité aux classes de congruence syntaxiques.

donc être reconduites et l'on obtient ainsi un homomorphisme Π' tel que:

{Catégories syntaxiques} = Π'(LEX)

LEX est donc partitionné en catégories syntaxiques.

La compatibilité entre schèmes et catégories linguistiques sera évoquée en M4.

I1 Morphologie et AFND-ATN

1. Il n'existe pas de modèle idéal

La régularité de la morphologie arabe a été soulignée en M1. Il est remarquable que cette morphologie possède en quelque sorte une double nature, obéissant d'une part à des règles de *concaténation*, avec un ordre strict de succession des éléments concaténés, et par ailleurs à des règles de *composition par schème* ou de transformation interne, pour la formation des lexèmes à base radicale.

Il n'existe pas de modèle idéal pour refléter cette double nature. En revanche, la concaténation trouve une modélisation simple et très efficace dans le formalisme des *automates*. C'est en examinant de près les possibilités offertes par cette théorie que l'on constatera l'adaptabilité des algorithmes au cas soulevé de la génération de LEX par le produit SC x RAC. Certaines difficultés nécessiteront alors une *algébrisation* des modèles, laquelle fournira un cadre théorique adéquat pour la construction de grammaires formelles. Cette construction reviendra à calculer l'homormorphisme Π évoqué en M1.

2. Rappel des notions de base liées aux automates

2.1 Automates

Donnons une définition formelle d'un *automate fini*:

• Un automate A est un quintuplet $\{Q, \Sigma, \delta, q_0, F\}$ où:

- Q ensemble fini d'états
- Σ alphabet, ensemble de symboles
- δ fonction de transition de Q x Σ —> Q
- q_0 état initial, appartenant à Q
- F ensemble des états finaux, sous-ensemble de Q

L'automate se trouve donc dans un nombre fini d'états, et les changements d'états enregistrés à la lecture d'un signal d'entrée composé d'éléments de Σ sont gouvernés

par la fonction δ. Lorsque le signal d'entrée est entièrement lu, si l'état courant est élément de F, le signal est accepté; dans le cas contraire, il est refusé.
L'ensemble des mots de Σ reconnus par un automate A est le *langage reconnu par A*.
La fonction de transition des automates finis est représentable graphiquement par un réseau d'états, reliés entre eux par des arcs sur lesquels les symboles permettant la transition sont consignés, et où les états remarquables (initial, finaux) sont démarqués.
En linguistique, les automates finis sont utilisés soit pour décoder des chaînes de caractères, et l'alphabet Σ est alors l'alphabet de la langue à analyser, soit pour décoder des phrases ou suites de mots, les symboles étant alors les mots entiers. C'est ici le premier cas qui retiendra notre attention.

2.2 Indétermisme, déterminisme

Dans le modèle *déterministe* des automates finis, la transition d'un état à un autre s'effectue de manière à ce que le choix soit limité à une seule possibilité par caractère lu et par état, ce qui implique notamment que chaque transition effectuée corresponde effectivement à la lecture d'un caractère dans la chaîne d'entrée.
Le modèle est *indéterministe* si: – plusieurs transitions sont possibles à partir d'un seul état;

– il est possible d'effectuer un «saut» d'un état à un autre sans lire pour autant de caractère. Ces transitions sont désignées par *epsilon-transitions*, nous les représenterons graphiquement par le symbole ε sur un arc reliant deux états.
On parle alors d'AFND, Automates Finis Non Déterministes.

Il existe une correspondance remarquable entre ces deux types d'automates: en effet,
∴ *tout langage reconnu par un AFND l'est également par un automate déterministe*.

Sans détailler ici la preuve de ce théorème[13], notons que les états de l'automate déterministe équivalent sont des regroupements judicieux d'états du premier automate. En revanche, le parcours dans l'AFND d'origine n'est pas reconstituable

13. Voir par exemple [Stern 90].

d'après le parcours dans l'automate original. Cette équivalence est aussi importante du point de vue de la programmation: si les automates déterministes sont aisés à simuler sous forme de programmes contenant des «sauts conditionnels», les automates indéterministes requièrent une analyse plus approfondie, nécessitant la programmation de «piles» ou des appels récursifs de fonctions. L'exécution de tels programmes est donc plus lente.

2.3 Langages réguliers, expressions régulières et grammaires non-contextuelles

• Si A est un langage d'alphabet Σ, la clôture de A , notée A*, est définie comme la réunion des langages A^n pour n entier, en convenant que A^0 est le singleton {e}.
On note de même m* le mot défini par $m = e + m + m.m + ... + m^n$.

• La classe des *langages réguliers* – ou *rationnels* – se définit sur un alphabet Σ de proche en proche:

– le langage vide est régulier;

– le langage réduit à chacun des symboles de Σ ainsi qu'au mot vide est régulier;

– si A et B sont des langages réguliers, il en est de même des produits de concaténation AB, l'union A | B, la clôture A*.

• La notation correspondant aux langages réguliers est celle des *expressions régulières*. Elles se définissent parallèlement aux langages rationnels:

– l'expression Ø est régulière et note le langage vide;

– les éléments du langage constituent des expressions régulières, auxquelles on adjoint l'expression désignant le mot vide $E_ø$;

– si E_1 et E_2 sont des expressions régulières, $E_1 + E_2$, $E_1.E_2$, E_1* le sont aussi, E* désignant l'expression somme $E^* = E_ø + E + E.E + ... + E_n + ...$

L'intérêt des langages et expressions réguliers apparaît avec la correspondance établie par le théorème de Kleene:

∴ *Tout langage régulier est reconnu par un automate fini, et tout langage reconnu par un automate fini est régulier*.

• Les *grammaires non-contextuelles*, ou *grammaires algébriques*, sont définies de la

façon suivante.
G grammaire algébrique comporte:

– Γ alphabet terminal, contenant les symboles terminaux;
– Σ alphabet auxiliaire, contenant les variables;
– un ensemble de règles de production ou de réécriture P

$A \longrightarrow \alpha$, où A variable et $\alpha \in (\Gamma \cup \Sigma)^*$;

– S variable initiale.

L'ensemble des mots de Γ^* obtenus par succession de dérivation partir de S est le langage engendré par la grammaire G.

La classe des langages algébriques se prête à son tour à l'inclusion suivante[14]:
$\therefore$ *Tout langage rationnel est algébrique.*

Sa réciproque est fausse. Cependant, on montre qu'une classe de grammaires non-contextuelles, les grammaires linéaires à gauche, ou à droite, peuvent s'exprimer sous forme d'une expression rationnelle, et donc engendrer un langage régulier:
• les grammaires *linéraire à gauche* (respectivement à droite) répondent à la définition précédente des grammaires algèbriques, avec la restriction:
chaque production est de la forme: $A \longrightarrow xB + a$ (respectivement $A \longrightarrow Bx + a$), où $A, B \in \Sigma$ et $a \in \Gamma$.

Le théorème de Kleene permet donc de caractériser les automates finis par les grammaires linéaires.
Quant aux langages algébriques, leur caractérisation est obtenue par la classe des langages reconnus par les automates à une pile. La définition de l'automate à pile s'inspire formellement de celle d'une machine de Turing déterministe à deux rubans, avec quelques aménagements[15].
Le tableau suivant résume les situations abordées , correspondant aux niveaux 1 et 2 de la hiérarchie de Chomsky:

---

14. [Stern 90] p. 271.
15. [Stern 90] p. 283.

Correspondance Grammaires-Automates

niveau	*grammaire*	*automate*
1	linéraires à droite (ou gauche)	AFNDs
2	non-contextuelles = algèbriques	Automates à une pile

Les niveaux 3, grammaires dépendant du contexte, et 4, grammaires transformationnelles, exigeant des analyseurs plus complexes, sont écartés de cette étude.

3 Modélisation: exemples progressifs

Ces quelques éléments de théorie des langages et des automates suffisent pour proposer des modèles simples d'accepteurs et d'analyseurs de formes morphologiques arabes, car ce formalisme reflète parfaitement la notion de concaténation.

3.1 Exemple du modèle des infixes[16]

Nous avons vu en **M1** que la morphologie nominale se distinguait par l'ajout possible de nombreuses particules concaténées avant le mot. L'ordre de ces concaténations est déterminé par des questions d'incompatibilité de catégories et par le respect d'un ordre syntaxique.

Il y a donc plusieurs approches possibles pour la modélisation de cette partie du mot.

• Un système de règles algébriques de réécriture:

Nous avons rappelé brièvement en **M1**-*Morphologie externe* les éléments concaténables au début du mot. Les catégories possibles obéissent aux règles de succession suivantes:

A : interrogatif / coordonnant / préposition / article أفبال...

B : coordonnant / corroboratif / préposition / article وللـ...

Dans le cas B, les premières règles de production du mot peut s'écrire:

16. [Jaccarini 97] chapitre II - sect. III.

Nom —> ExtraitNomInterrogatif

Nom —> ExtraitNomSansInterrogatif

où ExtraitNomInterrogatif désigne une chaîne extraite du nom contenant l'interrogatif, et ExtraitNomSansInterrogatif une chaîne ne le contenant pas. Puis l'on développe:

ExtraitNomInterrogatif —> أ + ExtraitNomCoordonnant

ExtraitNomInterrogatif —> ExtraitNomCoordonnant

ExtraitNomSansInterrogatif —> ExtraitNomCoordonnant1

.....

jusqu'à parvenir au mot privé de toute adjonction, $Base_{Nom}$.

Un développement identique pour B met en évidence, pour les deux dernières étapes, des équations identiques à la voie A: par opération de substitution, on est donc en mesure de présenter un système d'équations simplifiées de l'ensemble de cette grammaire des lettres concaténées du nom.

• Cet ensemble d'équations se traduit en une expression régulière:

$Nom = ((أ+\varepsilon)(ف+و+\varepsilon))+(ف+و+\varepsilon)(ل+\varepsilon))(ل+ك+ب+\varepsilon)(ا+\varepsilon)(ل+\varepsilon)\ Base_{Nom}$

• L'automate fini non déterministe correspondant peut être constitué (fig. 1):

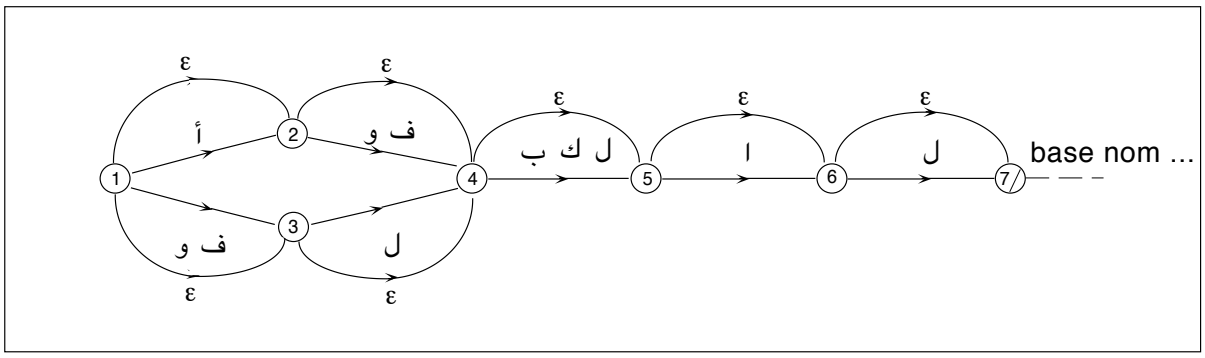


Fig. 1. *Automate des préfixes du nom.*

On remarque la correspondance état/catégorie de cette construction: une catégorie est associée à chaque état.

Un tel automate se décrit également sous forme de règles de réécriture:

Q_1 —> أ Q_2 $\quad\quad$ Q_1 —> ف Q_3

Q_1 —> ε Q_2 $\quad\quad$ Q_1 —> ε Q_3

Q_1 —> و Q_3 $\quad\quad$ etc.

Concevoir directement ces automates est la méthode générale adoptée ici. Le concepteur des grammaires devra, tant que faire se peut, tenter d'éviter des

embranchements trop larges en début d'automate: ces embranchements sont les plus coûteux en temps d'analyse. Des optima sont donc à rechercher, comme nous le verrons dans les principes suivants.

3.2 Exemple de la partie radicale[17]

Plus délicate est l'approche de la partie radicale du nom ou du verbe. Faisons maintenant abstraction des règles de compatibilité régissant l'enchaînement des lettres concaténées avant le mot, mais aussi après (les pronoms postfixes). On peut alors définir les ensembles suivants: – les lettres préfixées et les lettres des mots précédant l'occurrence de la première lettre radicale (ensemble Σ_1);

– les lettres postfixées et les lettres des mots suivant l'occurrence de la dernière lettre radicale (ensemble Σ_3).

Ces ensembles ne constituent pas à proprement parler des catégories, mais ils permettent une définition concise du mot arabe, que nous donnons directement sous forme d'expression régulière:

- $E_{MotArabe} = \Sigma_1^*.\ \text{PartieRadicale}.\Sigma_3^*$

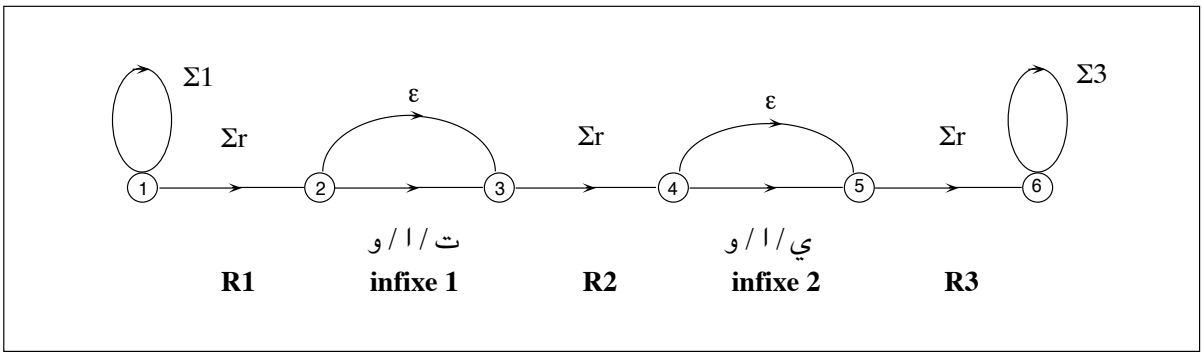
  $= \Sigma_1^*.\ R_1.\text{infixe}_1.R_2.\text{infixe}_2.R_3.\Sigma_3^*$

  où $R_i \in \Sigma_r$

Les *infixes* (voir exemple 3) sont représentés dans un premier temps par une seule lettre, ou une absence de lettre (epsilon-transition):

$\text{infixe}_1 \in \{ø, ت, و, ا\}$; $\text{infixe}_2 \in \{ø, ي, و, ا\}$

La forme AFND se construit aisément (fig. 2):



17. [Jaccarini 97] chapitre II - sect. II; [Audebert, Jaccarini 94] p. 80-81.

Fig. 2. *AFND de la partie radicale du nom.*

Cet automate présente donc la faculté d'accepter toutes les formes morphologiques constituées à partir d'une forme trilitère saine – ce qui ne signifie pas qu'il ne puisse accepter des formes non saines qui se prêteraient à ce gabarit (ex.: ال, ف); deux éventualités se présentent, pour son exploitation pratique:
– si l'on cherche à connaître les éléments radicaux, il convient d'en conserver la forme non déterministe et d'examiner les lettres permettant les transitions (1 à 2), (4 à 5) , (6 à 7): leur réunion reconstituera la racine;
– si l'on désire un simple accepteur de formes trilitères saines, un tel automate doit être transformé en son équivalent déterministe, avec la perte d'information catégorielle sur chaque caractère que cela implique.
Nous sommes dès ce modèle confronté à la réalité de l'ambiguïté, très importante dans l'arabe non-voyellé. C'est une qualité essentielle de la modélisation par automate non déterministe que de respecter l'ambiguïté naturelle, dès lors qu'il peut exister plusieurs chemins possibles dans le réseau de transitions pour accepter un mot. Nous reviendrons sur ce point.

3.3 Exemple d'automate nominal, illustration d'un compromis[18]

L'exemple précédent laisse déjà entrevoir un aspect capital de la modélisation morphologique: les schèmes n'apparaissent pas en tant que tels, comme des suites de transitions qui leur seraient réservées, mais comme des entités discontinues. C'est là un compromis entre efficacité d'analyse d'une part, les automates «larges» dès les premiers états conduisant à des analyses coûteuses en temps, et effectivité linguistique, car il est possible prévoir un grand nombre de schèmes courants à partir de quelques transitions.
Le modèle présenté ici n'a pas la prétention d'être définitif: il illustre simplement cette idée de compromis réalisable en restant dans le formalisme des AFND. Des *pseudo-catégories* sont introduites pour rendre compte de la discontinuité du schème: elles ne correspondent pas aux classes de congruence définies en M1. Ces pseudo-

18. [Jaccarini 97] chapitre VII - sect. I-II.

catégories sont :

– les préfixes de schème, éléments du schème avant R_1;

ex. مِ dans است, مِفتاح dans است,بال

– les infixes de schèmes, éléments du schème entre R_1 et R_2 ($infixe_1$) et R_2 et R_3 ($infixe_2$);

ex. ي dans ا, حكيم dans مناجم

– les suffixes de schèmes, éléments du schème après R_3.

Par ailleurs, les incompatibilités entre lettres concaténées après le mot ont été modélisées, prévoyant notamment la chute du ن des marques du duel et du pluriel dans les cas d'annexion.

Le résultat est l'automate de la fig. 3. Certaines transitions restent incompatibles les unes avec les autres, soit pour des motifs linguistiques, soit du fait de la méthode même de contruction fondée sur le compromis:

– l'article et le pronom personnel postfixe ne peuvent figurer dans un même mot, car ils marquent chacun la détermination qui ne saurait être multiple;

– les pseudo-catégories, lors de leur regroupement, doivent recréer un schème effectivement attesté, ce qui n'est pas le cas ici théoriquement;

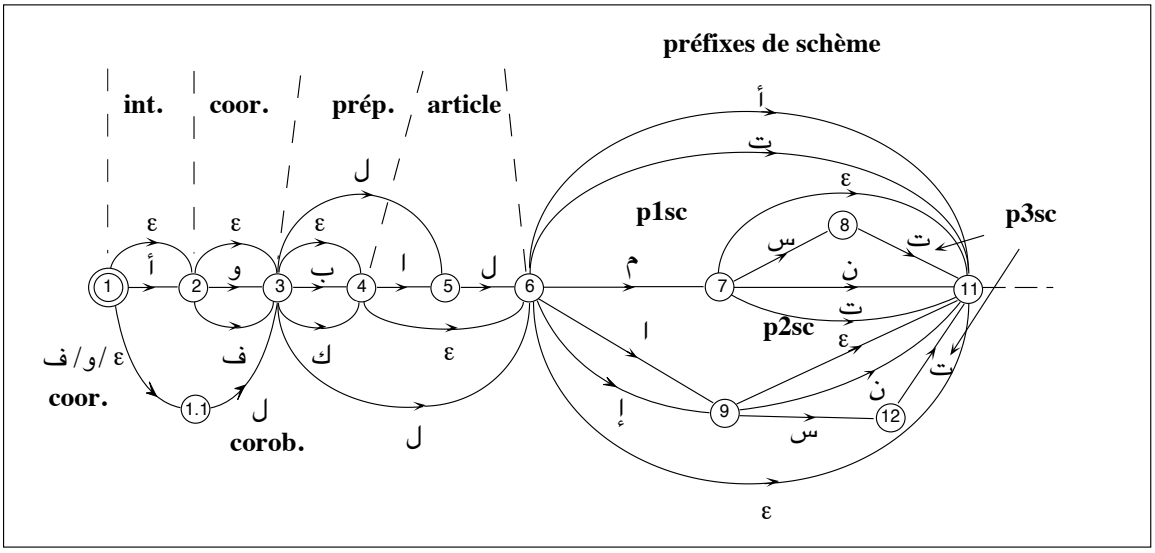
– les lettres radicales doivent aussi constituer un triplet valide, attesté dans la langue.

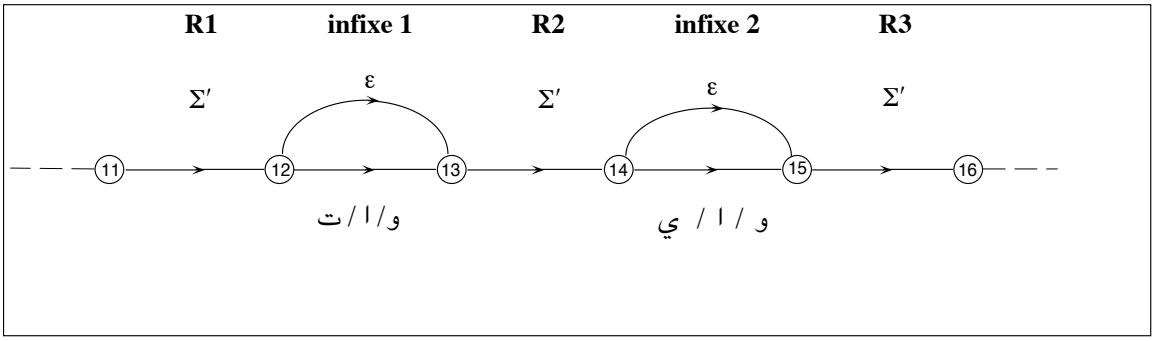
Les interprétations proposées par un tel automate comportent donc de nombreuses incohérences, des bruits lorsque l'interprétation produite est fausse, des silences lorsque la solution linguistiquement correcte n'apparaît pas. L'étude systématique de la réponse des analyseurs à des éléments tout venants de Lar doit faire l'objet d'une enquête approfondie, laquelle a pris la forme de premiers sondages et d'un début de classification, comme nous le verrons au principe **I4**. Un tel réseau ne se suffit pas à lui-même: il est nécessaire de l'augmenter, c'est-à-dire de disposer de contrôles supplémentaires à ceux pourvus par les seules transitions autorisées entre les états. Nous exposerons les principes d'augmentation en **M2**.

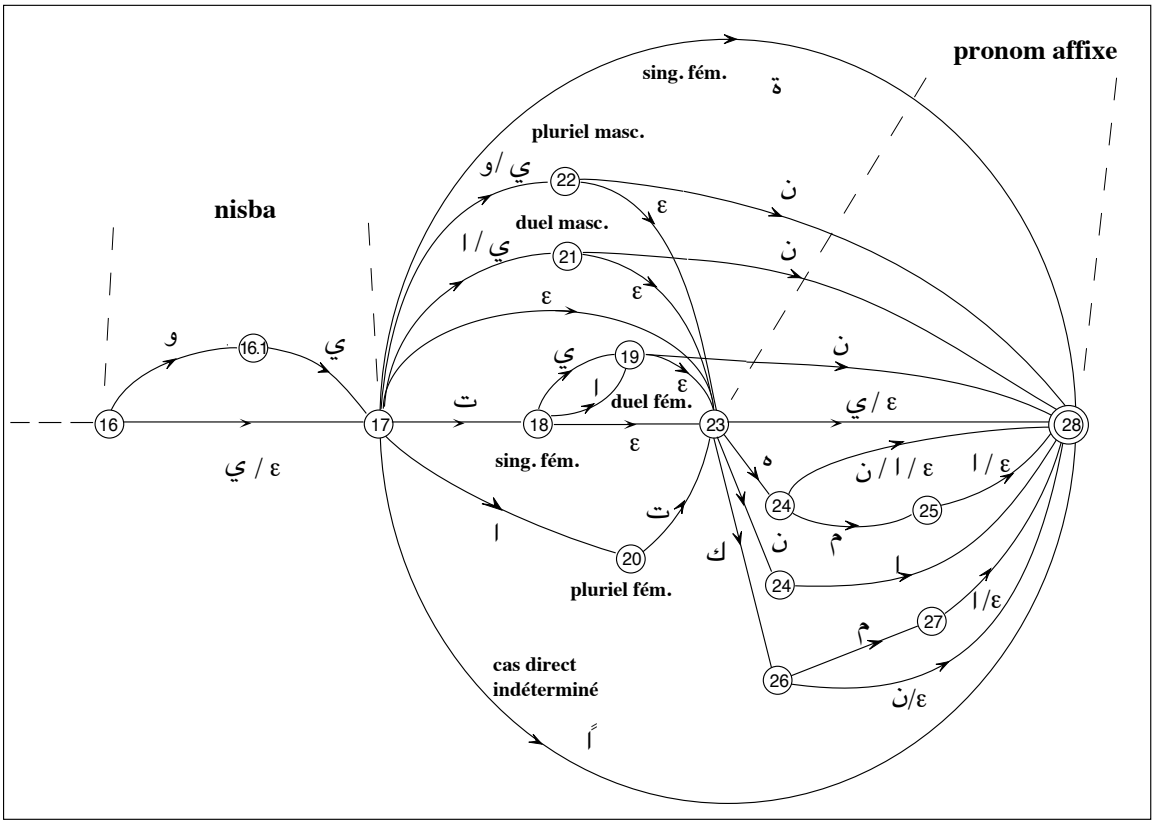
Pourtant, ce réseau fournit une base descriptive et effective remarquable de la morphologie nominale de l'arabe: il synthétise des connaissances morphologiques et les rend directement compréhensibles à la machine, par l'intermédiaire d'algorithmes d'analyse des AFND et ATN abordés en **I1**.

page suivante:

Fig. 3. *La grammaire F2 du nom trilitère sain*. Extrait de Jaccarini [97/02], chap. VII.II


préfixes de schème
int.
coor.
prép.
article
p1sc
p2sc
p3sc
coor.
corob.
ε
أ
و
ب
ا
ل
ف
ك
ت
س
ن
م
إ
1
1.1
2
3
4
5
6
7
8
9
11
12


R1
infixe 1
R2
infixe 2
R3
Σ′
ε
و / ا / ت
و / ا / ي
11
12
13
14
15
16


pronom affixe
nisba
sing. fém.
pluriel masc.
duel masc.
duel fém.
sing. fém.
pluriel fém.
cas direct
indéterminé
ة
ي /و
ي / ا
ي
ت
ا
ه
ن
م
ك
ل
ً
ε
و
16
16.1
17
18
19
20
21
22
23
24
25
26
27
28

M2 Extension de la régularité vers le non-sain

1. Racines non saines

Il existe plusieurs types de racines non saines, que les grammairiens ont pu classer par la position ou la présence simultanée des semi-consonnes ي, و , de la consonne *hamza* ء et l'égalité $R_2 = R_3$. Plusieurs types de phénomènes surviennent alors, que nous avons évoqués en **M1**. D'autres irrégularités peuvent se produire pour des racines régulièrement triconsonantiques. Sans donner de liste exhaustive de ces phénomènes, nous pouvons rappeler les plus courants:

- assimilation اصتدم —> اصطدم
- dissimilation قاول —> قائل
- gemination يمرر —> يمر
- suppression[19] ذئب —> ذيب
- orthographe de la *hamza* رئيس, بؤس, رأس, جزء
- «fusion» des *hamzas* أأكل —> آكل

Ces phénomènes peuvent être perçus comme autant de modifications, voire d'altérations, du réseau de base approché par les automates en **I1**. Sans entrer ici dans un débat morphologique, phonologique et historique, insistons sur le fait que la forme saine telle qu'elle devrait être sert toujours de référence aux linguistes dans leurs hypothèses d'explications des transformations dont les résultantes sont les formes irrégulières finales[20].

2. Transduction

Formellement, on considère qu'il existe un «noyau de base» décrit par une grammaire G', laquelle engendre la langue L(G'). La langue arabe peut alors être perçue comme la transduction finie L(G) d'une grammaire G: «nous chercherons à enrichir G' de la manière la plus simple en sorte que L(G) puisse être déduit par une correspondance homomorphique de L(G')[21]». Cette correspondance peut s'exprimer par un automate fini augmenté non récursif, autrement dit par l'enrichissement des réseaux de base. Ceci présente l'avantage de permettre la résolution d'équations

19. *ḥadf* de la *hamza*, cf. [Fleish 90], T. 2, p. 107.
20. H. Fleisch parle respectivement de «point de départ» et de «point d'arrivée» à propos des termes de cette évolution, dans le cas des "w et y intervocaliques". [Fleish 90] T. 1, p. 122.
21. [Jaccarini 97] chapitre VII, section I, p. 3.

algébriques définissant des langages réguliers, en restant dans une structure de corps non-commutatif (voir principes **I2**, syntaxe et automates et **I4**, variations de grammaires).

Nous exposons ici quelques cas possibles de transduction permettant de modéliser des phénomènes de morphologie non saine, en premier lieu desquels le cas d'assimilation des formes dérivées VIII des verbes à première radicale و. Rappelons la règle: le و radical, originellement situé entre le «préfixe de schème» ا et l'«infixe de schème» ت, se transforme en ت qui se gémine au ت infixe.

• assimilation اوتصل —> اتّصل

Cette transformation peut se transposer en une procédure d'augmentation en trois volets: 1. une transformation mineure du réseau, que l'on effectuera sur l'automate nominal présenté à la fin de **T1**, par l'ajout d'une ε-transition supplémentaire entre les états 11 et 12 (parallèlement à la transition dédiée à R_1), soit Q_{11} —> ε Q_{12};

2. le conditionnement de cette transition par le passage par le chemin Q_6(إ,ا)Q_9εQ_{11}, réalisé à l'aide d'un test;

3. la restitution de la lettre radicale و, elle aussi réalisée par le test de la présence de la transition suivante Q_{12} —> تQ_{13} et une «action» consécutive de reconstitution du و comme étant R1.

Ces tests et actions sont clairement exprimables dans le formalisme des lambda-expressions évoquées en **M1**:

(λxyz.(if(eq x 'و)(ا. تّ.y.z)(ا.x.ت.y.z))), où *if* est un opérateur définissable en λ-calcul.

Le langage Lisp, fondé sur le λ-calcul, se prête parfaitement à cette formalisation: nous y reviendrons en **I4**, en montrant diverses manières de programmer ces augmentations.

2. Morphologie et catégories syntaxiques : M4 et M5

M4 Légitimité de la catégorisation morpho-syntaxique

Pour parfaire la construction élaborée en **M1**, il convenait d'examiner la

compatibilité du «découpage» en schèmes du lexique et des catégories syntaxiques[22]. Plus précisément, le fait que «tous les mots qui présentent le même paradigme admettent ou excluent rigoureusement le même ensemble de caractères susceptibles de leur être concaténés» autorise à voir en la relation $\sim$SC ainsi définie une relation de congruence:

pour tout u,v L_{ar}*, u $\sim_{SC}$ v $=>$ x u y $\sim_{SC}$ x v y, quel que soit x, y Σ*

En lui associant l'homomorphisme SC tel que:

SC : $L_{ar} \rightarrow L/\sim_{SC}$

x: $\rightarrow$ x si x atome

x: $\rightarrow$ son schème sinon

On note L_{ar}/RAC = SC(L_{ar}) ce langage : c'est en quelque sorte la langue arabe réduite à une racine et aux mots outils; L_{ar}/RAC, inclus dans $\Sigma^*/\sim_{SC}$ est un langage «squelette» mais néanmoins grammatical (voir M1, *morphologie externe*, 2).

La compatibilité de la partition Lar/RAC avec la partition évoquée en **M1** qui définit les catégories syntaxiques mérite examen. Si certains éléments de réponse peuvent être apportés, en invoquant notamment le fait que «$\sim L_{ar}$ est la relation de congruence la plus grossière que l'on puisse définir sur Σ^* et qui sature L_{ar}», la compatibilité générale des deux partitions doit plutôt être considérée comme un postulat, un cadre général à respecter, et qui peut se reformuler ainsi:

«établir des classes syntaxiques que l'on sous-partitionne en schèmes, ou bien établir d'abord des schèmes que l'on regroupe en classes syntaxiques, revient au même».

Le travail direct sur la langue et les catégorisations nominales et verbales opérées par le linguiste équivaut à la construction d'une syntaxe de la langue squelette où les schèmes sont construits comme des classes d'équivalence du lexique. Ce principe est est aussi un programme de travail, selon lequel on ne va s'intéresser qu'aux règles de syntaxe qui le respectent, et mener le travail de catégorisation non pas dans la vraie langue, mais dans le langage quotient L/RAC.

M5 Effacement du rôle du lexique

Nous avons vu, à travers **M1**, **I1** et **M2**, qu'il était possible de construire des analyseurs morphologiques à large couverture, capables d'extraire avec un *minimum* de règles un *optimum* d'information morphologique. Autrement dit, le programme

22. [Jaccarini 97] chapitre. I, p. 43-45.

morphologique est construit indépendamment du lexique. Le dernier principe **M4** va plus loin en proposant, par l'étude de L/RAC, une abstraction du lexique comme point de vue extrême mais fécond, autrement dit le *dictionnaire vide*. C'est bien sûr de *minimum de recours au lexique* dont nous devons parler.

Anticipant sur ce principe latent mais pas encore exprimé, l'article *Ḫabar* supposait le problème morphologique résolu pour concentrer la réflexion sur les problèmes essentiels d'échanges morpho-syntaxiques et de prévisibilité, comme nous le verrons en **S1** et **S2**.

Une telle démarche est centrale dans le projet de la minimalité, et ses «conséquences» peuvent se mesurer dans les domaines suivants:

• la recherche d'un meilleur équilibre entre grammaire et lexique;

• la gestion en amont des ambiguïtés lexicales;

• l'optimisation des programmes d'analyse morphologiques;

• la structuration du ou des lexiques (information homogène, cohérente et non redondante...)

Enfin, un niveau supplémentaire d'abstraction correspond à celui d'une sémantique quotient, qui attriburait à chaque schème une valeur sémantique à croiser avec celle représentée par la racine.

Il va de soi que la minimalité n'interdit pas le recours ultérieur au lexique, et cette perspective est maintes fois annoncée.

3. Tokens: M6 et S1

M6. Existence d'un noyau d'invariants morphologiques

1. Définition

Selon la terminologie développée aux principes **Mi** précédents, nous sommes en mesure de définir formellement un ensemble d'invariants[23]; il s'agit de l'ensemble AT des *atomes*:

$$AT = LEX \cap \{x;\ SC(x) = \{x\}\ \}$$

Cette définition peut s'illustrer ainsi: SC(فإنهم) = ف SC (إن) فإنهم = هم

Les atomes sont donc à la fois dans le vocabulaire terminal de L et de L/RAC: ce sont des mots arabes dépourvus de schème. A. Jaccarini recense environ 150

23. [Jaccarini 97] chapitre. VI, p. 1.

atomes; nous exposerons chapitre V une version exhaustive de cette liste.

2. Catégorisation morphologique des tokens et détection

Le tableau **M1** morphologie externe 4 donnait les catégories possibles des éléments concaténables aux tokens. Une catégorisation morphologique fine est souhaitable pour programmer l'analyse et donc la détection de ces éléments dont nous verrons l'importance en **S1**.

Dans le diagramme suivant (fig. 4), issu de [Audebert Jaccarini 88], un certain nombre de possibilités de concaténations des tokens sont explicitées.

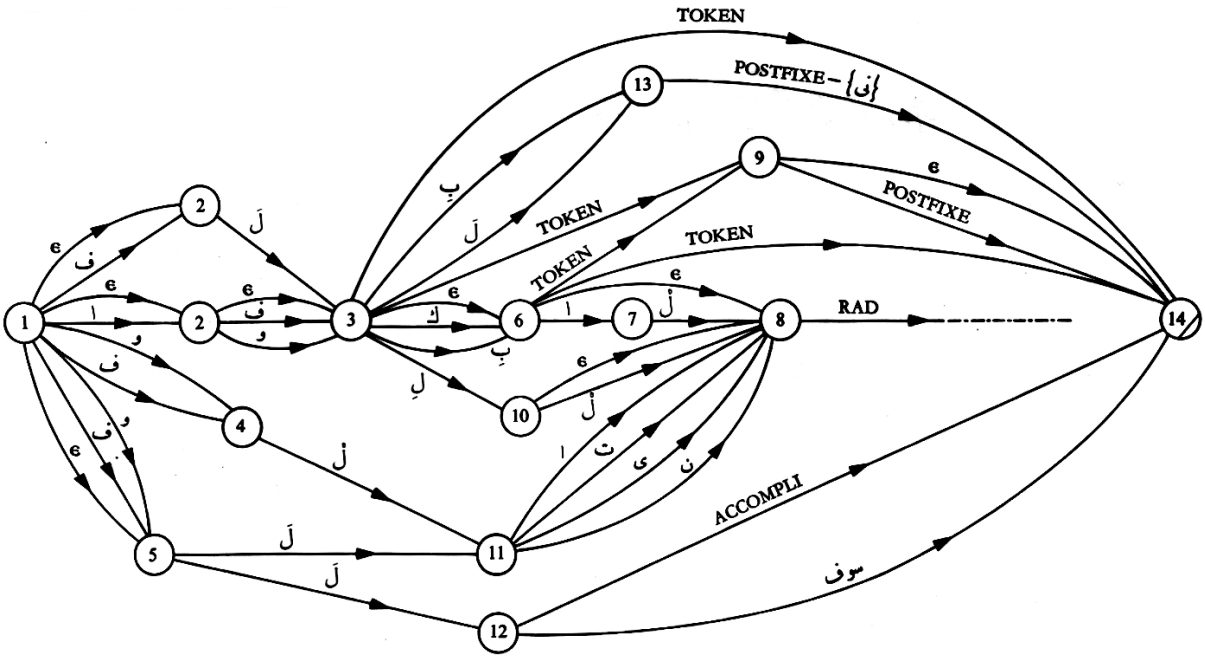


Fig. 4. *Diagramme décrivant certaines des concaténations possibles des tokens.*

Il convient cependant de bien séparer les problèmes et de ne pas aborder en même temps morphologie des atomes et implications syntaxiques. A. Jaccarini propose une démarche en deux étapes:

1. établissement d'un schéma de grammaire morphologique des atomes[24], où l'on s'intéresse d'abord aux possibilités de concaténation – morphologie externe – et aux regroupements d'atomes en famille au regard de ces possibilités. Sont ainsi isolées

24. [Jaccarini 97] chapitre. VI-2 et 3.

10 familles, représentant chacune une combinaison attestée de concaténations; la grammaire non déterministe décrivant ces combinaisons comporte alors 10 états, et autant de terminaux que de familles et de catégories réunies, soit 14. Le jeu des compatibilités entre les particules concaténées fait apparaître 27 règles de production ou arcs de l'automate équivalent, et qui permettent l'«aiguillage» d'un mot comportant un atome vers sa catégorie morphologique.

Ainsi, l'analyse de la séquence «أفببعضهم» fournira-t-elle la décomposition: int.coor.prep.tok_{15}.post, où tok_{15} note la famille permettant toutes les concaténations possibles; de même, «بما» s'analysera comme prep.tok_2.

2. Affinement de ce schéma par un travail d'optimisation «lettre à lettre», obtenu par la recherche de facteurs communs *à gauche* à l'intérieur de chaque famille[25]. En effet, le choix – ou le compromis – existe entre un automate «ouvert» et lisible mais lent car fortement indéterministe, comportant autant de bifurcations que d'atomes dans chaque famille morphologique, et un automate compact et rapide mais difficile à mettre au point, si ce n'est en utilisant des procédures de parenthésage automatique.

Un premier programme de détection permet d'obtenir un taux de 85% de détection, résultat certes insuffisant, mais largement perfectible par un travail à vaste échelle.

3. Ambiguïté graphique

L'ambiguïté graphique, si fréquente à l'arabe en général, n'épargne pas les atomes. Plusieurs formes graphiques ambiguës sont traitées en profondeur dans l'article *Ḫabar*[26]:

– ان, qui est soit إنّ, أنّ (*aḫawāt inna*), soit subordonnant (أنْ), soit conditionnel (إنْ);

– الا, qui est exclamatif (ألا), interro-négatif (أ لا), subordonnant (أن لا), ou exception (إلا);

– من, qui est préposition (مِنْ), relatif, interrogatif ou conditionnel (مَنْ).

Chaque cas est étudié en fonction de sa place relative dans la phrase: l'atome est-il précédé ou non d'un groupe nominal ou verbal, quelle est sa position absolue, etc. Ces critères, l'analyse morphologique nominale et verbale supposée résolue,

25. [Jaccarini 97] chapitre. VI-4.

26. [Audebert, Jaccarini 86] p. 223-228.

conduisent à la désambiguïsation théorique des formes atomes ambiguës.

S1. Les invariants morphologiques sont des tokens syntaxiques

1. Idée générale et définition

L'ensemble des invariants morphologiques, les atomes décrits en **M6** reçoivent l'appellation de *token*s pour les raisons suivantes[27]:

– «c'est ainsi que l'on désigne, en théorie de la compilation, des éléments que l'on a intérêt à faire figurer dans le lexique plutôt que dans l'analyseur»;

– «un jeton [token] est une entité visible qui déclenche un mécanisme».

Comme nous l'avons entrevu au principe **M6.3** (ambiguïté), les atomes morphologiques ont un rôle éminemment syntaxique : «il s'agit le plus souvent de mots-outils ou de mots vides»[28]. Un aperçu, non exhaustif, des classes de mots-outils de l'arabe aide à prendre la mesure des implications syntaxiques de ces invariants morphologiques:

– إنّ et ses sœurs (*aḫawāt inna*), coordonnants, relatifs, conditionnels, subordonnants, interrogatifs, prépositions, pronoms personnels et démonstratifs, etc.

2. Les clés d'une stratégie de décodage

L'objectif **O3** d'enseignement de l'arabe souligne le soucis d'un parallèlisme entre traitement automatique et décodage de l'arabe. Ainsi, C. Audebert remarquait que «la difficulté du décodage ne vient pas tant d'une ignorance que d'un excès d'information dont le texte est hérissé»[29]. L'auteur poursuit en proposant un cadre d'élaboration de stratégies de décodage: «le premier procédé d'élimination de ce trop plein d'information est fourni par la focalisation sur les tokens qui permettent de baliser la phrase sans se lancer dans une analyse morphologique mot à mot. Car deviner la structure d'une phrase c'est choisir la structure la plus probable et la vérifier en des points à la fois minimum et nécessaires». Le repérage préalable de ces tokens est donc une étape prioritaire, et «cette opération coïncide avec l'expérience d'enseignement qui consiste à conduire l'étudiant à rechercher d'emblée les points

27. [Audebert, Jaccarini 88] p. 269.

28. [Audebert, Jaccarini 88] p. 269.

29. [Audebert 86] p. 1.

de repère»[30].

Nombre des tokens, les plus importants d'entre eux, déclenchent une ou plusieurs *attentes syntaxiques*: l'apparition d'un groupe verbal (GV), nominal (GN), d'un thème ou d'un prédicat. Nous aurons l'occasion de préciser certaines de ces attentes en **S2**.

Les tokens, en tant que «révélateurs de structures», sont qualifiés par A. Jaccarini d'«opérateurs syntaxiques». Un paradoxe de la détection est mis en évidence: si le comportement des principaux opérateurs est décrit de manière trop détaillée, l'algorithmique et par conséquent l'analyse risque d'être plus complexe que l'objet linguistique à décoder[31]. Il faut donc s'en tenir à des esquisses de description, mais aussi «n'exploiter que des contraintes de très haut niveau», la responsabilité du choix de telle ou telle version des grammaires incombant au moniteur syntaxique (voir principe **I3**).

3 Critères et hiérarchie des tokens

Plusieurs critères ou traits sont indispensables à la conduite de stratégies de décodage. La première tentative de dégagement de ces critères est réalisée dans l'article *Ḫabar*[32].

Un tableau y représente des critères de positionnement du token, du niveau de globalité qu'il engage et de son environnement.

Les cas de أنْ, أنّ, إنّ, إنْ, الذى, مَنْ, مِنْ, هو, او, الا sont ainsi inventoriés dans leurs divers emplois.

Le «critère de globalité» précise le nombre de noyaux engagés. Le terme de noyau signifie ici les deux membres d'une phrase nominale (thème -prédicat) ou verbale (verbe - sujet et compléments). Les dépendances de ces structures fondamentales sont qualifiées d'ajouts (voir principe **S2**). C'est ce critère de globalité qui est retenu pour instaurer une hiérarchie des token. En effet, dans une optique de décodage, c'est bien les termes les plus structurants, «qui réduisent d'un coup le *niveau entropique*», qu'il convient de repérer en priorité. Voici quelques exemples de cette hiérarchie:

---

30. [Audebert, Jaccarini 86] p. 221.
31. [Jaccarini 97] chapitre IX, p. 6.
32. [Audebert, Jaccarini 86] p. 222.

– مَنْ situé en tête de phrase est de priorité maximale, car il peut gouverner deux noyaux verbaux (conditionnel), ou un noyau verbal ou nominal (interrogatif);
– هو qui peut représenter à lui seul le prédicat est alors aussi prioritaire (et aussi هذا, هناك, etc.);
– إنّ gouverne une phrase nominale, en dominante GND (thème) puis GN (prédicat), il est donc moins prioritaire;
– مِنْ ne gouverne qu'un GN ou GND.
Les contraintes de positionnement sont réparties entre contraintes fixes (souvent la première occurrence après le token) et lâches.
Cet angle de vue n'est pas celui des grammaires classiques où chaque token, pris éventuellement par classe, peut être étudié individuellement dans son comportement syntaxique, mais sans recul global au niveau de la phrase.

4. Un passage à la limite: le raisonnement sur la phrase vide

Le concept de phrase vide est évoqué dès 1986 dans l'article *Ḫabar*. Il s'agit, nous le rappelons, de ne garder dans le texte que les occurrences tokens et de masquer noms et verbes. C'est donc un passage à la limite, plus radical encore que celui qui consisterait à ne garder qu'une racine témoin فعل et conserver le schème. Des variantes peuvent être envisagées, où l'on décide de faire apparaître les articles, les coordonnants, les prépositions ل, ك, ب et les pronoms postfixes car ces informations ne sont pas lexicales. La phrase vide devient alors un support de réflexion pour l'analyse syntaxique. Quels éléments privilégier? Quelle structures syntaxiques faut-il tenter de repérer d'emblée? Le maître mot est ici le dépassement des tentatives combinatoires[33], et la conduite d'une véritable stratégie de décodage «top-down». La surabondance de l'information morpho-syntaxique est contournée par une approche *en surface*: «Tokens are tree indicators»[34].
Les tokens repérés, les autres occurrences sont moins capitales: elles sont N ou V, et ce sont les V qui renseignent le plus souvent sur le noyau.
ex.:

---

33. «This search for strategy implies the definition of an heuristic integrating elements of predictability in the parser in order to avoid exclusively combinatory procedures». [Audebert 89], p. 303.

34. [Audebert 89] p. 354.

_ هذه _ التي _ ان _ التي _ _ لا _ لها _ _ في _ _ _

19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

وV هذه الـ التي V ان الـ التي V الـ لا الـة لها بالـ الـة في الـ الـة الـة

Recherche sans recours aux articles ni aux marques extérieures:

1. Points de certitude: 3 = N, 5 = V, 7 = N, 9 = N, 17 = N
2. Recherche du noyau. Deux possibilités se présentent:
   – H1, N en 1 :
      – soit 10 n'est pas régi par le V 9 et c'est le prédicat de 1;
      – soit 10 est régi par le V 9 et le prédicat commence en 11 (لا);
   – H2, V en 1:
      – nous sommes en présence d'une phrase verbale.

L'analyse de plusieurs phrases vides révèle qu'en général la connaissance de quelques points clés suffit pour dévoiler la structure de la phrase.

Plusieurs remarques sont émises dès ce stade:

– ce décodage s'appuie sur des critères ou traits de connaissance des tokens (voir infra);

– le repérage de la ponctuation est capital et d'autant plus délicat que les و ou les ف se concatènent aux mots;

– une phrase sans ou avec peu de tokens présente une plus grande complexité d'analyse.

4. Syntaxe globale : S2, I2 et I3

S2. Règles syntaxiques dominantes

Une règle dominante est une règle qui a la plus forte probabilité de se produire parmi d'autres éventualités. Ainsi, dans une phrase nominale, la dominante est que le thème *mubtada*ʾ soit déterminé, et constitue la première hypothèse de travail bien qu'il existe 14 autres cas dont la fréquente phrase locative (فى الحديقة ولد). Un autre exemple de règle dominante est la prépondérence, mesurée par un petit dépouillement de phrases de presse[35], du verbe en première position dans les

35. [Audebert 86] note 2.

discours énonciatifs (V = 47, N = 23, T = 11). Au contraire, la rubrique religieuse présente une domination des tokens (T = 24, N = 6, V = 4).
La hiérarchie de tokens évoquée ci-dessus (principe **S1.3**) reflète ces règles qui lui préexistent, et c'est la conjonction des deux permet qui d'établir de fortes présomptions lors du décodage.
Les notions de noyaux (termes des phrases nominales et verbales), d'ajouts (ce qui en dépend) et de coordination sont en effet fondamentales, et doivent permettre d'aller à l'essentiel dans la description de la phrase. La belle part revient donc au comportement global des tokens, qui domine sur leur comportement local. La vocalisation désinentielle joue un rôle ici, mais mineur à cause de la non-voyellation généralisée.
Les règles dominantes sont bien sûr celles à enseigner en priorité (objectif O3), elles seront d'autant mieux assimilées qu'elles seront présentée comme telles: «Les indications sur la direction de la phrase, les attentes après certaines occurrences auront amené l'apprenant à formuler des hypothèses. Que celles-ci soient contradictoires voire erronées est moins grave que n'en point formuler du tout, ce qui est fréquemment le cas dans la procédure du mot à mot. L'essentiel est qu'elles soient formulées sur l'ensemble de la phrase. Les connaissances morphologiques rigoureuses viendront alors dans un deuxième temps lever les doutes.»[36]

I2. Syntaxe et automates

Les premières descriptions syntaxiques développées sous forme de réseaux de transitions de type AFND sont publiées dans l'article *Ḫabar*. Il s'agit d'un réseau de 11 automates, détaillés à des degrés variés, et qui modélisent les fragments de syntaxe et de situations suivants:

– I fragment du groupe nominal (GN);
– II groupe nominal déterminé (GND);
– III phrase verbale;
– IV structure commençant par من + GN;
– V structure commençant par أنْ + GV;
– VI structure commençant par GV + أنْ ou على + GN + أنْ.
– VII variante de la précédante incluant أنّ;
– VIII structure commençant par إنّ;
– IX structure commençant par إنْ;
– X groupe adjectival (GAD);
– XI variantes du groupe adjectival;

36. [Audebert, Jaccarini 86] p. 219.

La figure 5 rappelle le diagramme II du GND.

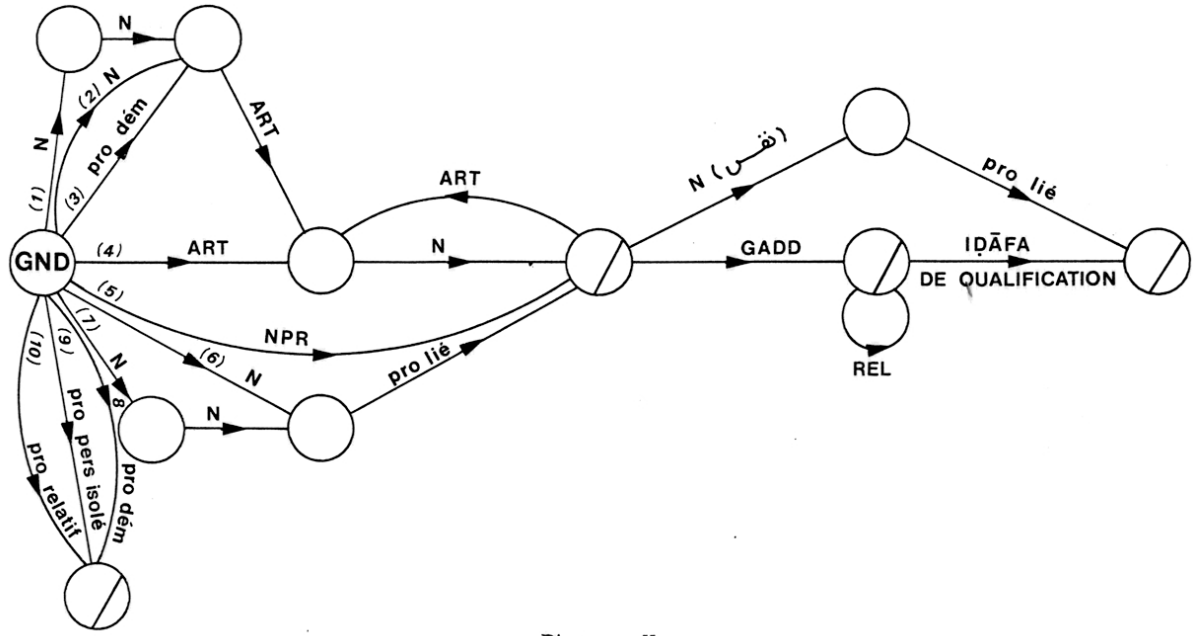


Diagramme II.

Fig. 5. *Diagramme II, GND.* Audebert Jaccarini [86].

Ces automates sont décrits comme autant de «successions d'attentes». Leur fonctionnement suppose l'existence d'un mécanisme de lecture (un seul symbole à la fois) et d'un mécanisme de contrôle (lire et tester l'entrée et faire passer la machine d'un état à un autre). La mémoire de cette machine est interrogeable et on peut y effectuer des tests et conduire des actions.

Ces diagrammes reflètent les règles dominantes dont il a été question plus haut (principe S2). Ils sont même la manière la plus simple de donner priorité à ces tokens, de leur attribuer le rôle de guide dans le décodage syntaxique: c'est la traduction du principe **S2** de rôle syntaxique des tokens.

À chaque token peut ainsi être associé un sous-langage régulier $L(tok_i)$ intégrable au moniteur morpho-syntaxique (voir principe **I3**), représenté idéalement par un AFND ou un ATN: «La cardinalité de leur [tokens] classe de congruence syntaxique est très réduite (souvent égale à l'unité); on en déduit qu'ils entraînent sur leurs environnements des contraintes plus importantes que celles de la plupart des autres lexèmes. C'est pourquoi il nous sera possible de les considérer comme des opérateurs syntaxiques entraînant des attentes et de leur associer des sous-langages»[37].

Le choix des AFND ou des ATN se justifie par la possibilité offerte par ce formalisme de transformer, réduire et éventuellement dérécursiver partie ou totalité d'un sous-langage. C'est cette possibilité qui fonde le principe de *variation des grammaires* (voir principe **I4**).

En effet, les AFND se situent au plus bas niveau dans la hiérarchie des grammaires formelles, mais les possibilités d'augmentations (ATN de Woods), offertes par exemple par les tests de registres suivis d'action, ne remettent pas en cause les calculs de transformation des grammaires. Ce modèle est équivalent à celui des grammaires linéaires à droite. Les pertes engendrées par ce choix (moins de capacité générative et pertes de finesse syntagmatique) sont compensées par la souplesse offerte par les tests/actions[38].

dérécursivation

---

37. [Jaccarini 2000] p. 181. Voir aussi [Jaccarini 97], chapitre IX, p. 6.

38. [Jaccarini 97] chapitre III. section I, p. 13.

Grâce aux langages réguliers, certaines grammaires de niveau 2 (c'est-à-dire modélisables par automate à pile, ou par des règles de réécriture autorisant la récursivité) sont réductibles au niveau 1, celui des automates finis modélisables par des règles de réécriture simples de type $q_i \rightarrow aq_j + q_j$. C'est la résolution d'équations rendue possible par ces langages qui permet de supprimer les situations récursives à droite, telle celle de la phrase conditionnelle arabe et du fragment COND associé[39] (إن فعلت ذلك وإن فعلت ذلك فافعل). Le prix en est l'obtention d'un accepteur (équivalence faible). Ces calculs se conduisent toutefois dans un corps non commutatif ($ab \neq ba$).

I3. Moniteur morpho-syntaxique

Le modèle de moniteur syntaxique est esquissé dans l'article *Ḥabar*, et affiné dans les travaux postérieurs. «La fonction principale [du moniteur syntaxique est] d'améliorer le niveau de résolution morphologique du texte et d'optimiser les analyses. Ce moniteur est essentiellement conçu comme un aiguilleur, fondé sur des fragments de grammaires syntaxiques, permettant d'éviter les ambiguïtés, de court-circuiter les tentatives inutiles au niveau de l'analyse morphologique et de réduire les silences. Il s'agit donc d'une procédure dont le rôle le plus important est de prévenir les analyses parasites plutôt que de les élaguer après une analyse morphologique hors-contexte et d'éviter ainsi les explosions combinatoires d'ambiguïtés de différents niveaux»[40].

Le fonctionnement de ce moniteur se résume en quelques caractéristiques essentielles:

– le moniteur avance mot à mot.

– Il comporte plusieurs piles enregistrant simultanément en mémoire:

1. la position dans la phrase,

2.1 le n° de l'automate ou diagramme courant,

2.2 l'état courant dans cet automate.

2.3 2.1 et 2.2 sont mémorisés pour autant de sous-traitements effectués: lorsqu'un diagramme est parcouru avec succès, la successsion d'état est mémorisée

39. [Jaccarini 97] chapitre IV. section I.

40. [Jaccarini 2000] p. 180.

et une étiquette globale affectée aux groupes d'occurrences identifiés (GN, GACC, Thème, etc.). On désempile alors pour retrouver l'automate qui a conduit ici.

– L'observation de la règle dominante commande de donner priorité au choix proposé par l'automate de hiérarchie la plus haute, le premier emprunté.

– Les séquences identifiées avec certitude, résultats locaux non ambigus, ne sont pas remises en question.

Ce fonctionnement suppose, au niveau morphologique,

– une détection complète de tous les tokens, l'éventuelle désambiguïsation s'effectuant grâce aux informations consignées dans un tableau et l'examen morphologique du voisinage immédiat;

– la détection des noms et verbes avec un niveau de détail suffisant pour pouvoir les distinguer et appliquer certaines règles d'accord, mais sans plus.

5. Variations et mesures de grammaires : I4 et I5

I4. Variation des grammaires et réalisations informatiques

Le principe de la variation des grammaires exprime l'idée selon laquelle les grammaires ne sont pas figées, mais reflètent un point de vue et réalisent un optimum entre un but de traitement et des données linguistiques. Le travail de modélisation est aussi celui de la vérification de la *cohérence* de ces modèles. L'augmentation des grammaires correspond à la volonté délibérée de «ne plus considérer les analyseurs comme des boîtes noires, mais analyser ce qui se passe à l'intérieur»[41]. Il s'agit de pouvoir extraire des fragments de grammaires robustes afin de les réemployer, les combiner pour en former de nouveaux et approcher l'algorithme optimum, utilisant ainsi les propriétés algébriques des grammaires.

Dans ce but a été créé un ensemble de programmes informatiques LISP modulaires baptisé «atelier de grammaires», reposant sur un analyseur de complexité linéaire à la structure transparente déduite d'un calcul dans une structure algébrique et enrichi de manière à gérer les tests et actions[42]. Un éditeur structurel de grammaires permet l'édition des grammaires à partir de leur règles de transitions. Cet éditeur est également un interpréteur, qui vérifie la structure syntaxique des objets qui lui sont

41. [Jaccarini 97] chapitre VII, section II.

42. [Jaccarini 97] chapitre IV, section II.

soumis et engendre le code LISP de l'automate correspondant, éventuellement dans sa version calculée déterministe[43].

Traduction ou interprétation

Il existe une alternative importante, en morphologie comme en syntaxe, pour la restitution de l'information fournie par l'analyse d'AFND et son traitement en vue d'augmentations:

– le mécanisme de traduction, qui consiste à traiter la sortie de l'analyseur d'AFND par un second programme qui saura par exemple attribuer une catégorie morphologique ou syntaxique à chaque transition lue, à partir de données codées dans une fonction de correspondance;

– le mécanisme d'interprétation, qui consiste en l'augmentation par procédures du noyau de base, au moyen de tests et d'actions effectués pendant le traitement même et codables en LISP par des lambda-expressions.

A. Jaccarini souligne[44] que les deux voies peuvent être explorées, en note toutefois qu'il faut se garder avec la traduction de s'éloigner du modèle de base en multipliant de coûteuses epsilon-transitions.

I5 Mesures et évaluation des grammaires

Un première typologie des bruits est esquissée[45] à partir d'une liste de 25 mots analysée par une grammaire nominale (celle du principe **I1**, 3.3). La solution est le plus souvent trouvée avec un coefficient d'ambiguïté égal à 2. Cette typologie fait apparaître les problèmes d'extraction de la racine; certains bruits sont irréductibles même avec un recours au lexique, d'autres sont réductibles par le recours à la syntaxe.

L'idée déjà exprimée (principes **M1** et **I1**) selon laquelle la morphologie découle d'abord d'un noyau fondamental sain et que les grammaires doivent être complexifiée à partir de ce noyau stable est aussi à la base de notre première

43. [Jaccarini 97] chapitres XI, XII et [Jaccarini 2001].

44. [Jaccarini 97] chapitre VII, p. 10.

45. [Jaccarini 97] chapitre VII, sect. II.

contribution[46] à ces travaux: un prototype de logiciel, *Sarfiyya*, constitué autour d'un analyseur d'AFND écrit en langage C (voir chapitre II) permet d'analyser des textes tout venant et de mesurer les réponses. Un texte littéraire est analysé qui autorise une première classification des réponses à grande échelle.
Plus généralement, un pendant indispensable à la variation des grammaires sera le contrôle sur corpus du comportement des grammaires: étude des niveaux de bruit et de silence, de l'ambiguïté, respect des grandes catégories morphologiques et syntaxiques, performance d'extraction de la racine, etc. Les aspects de complexité et de temps d'exécution d'une part et d'encombrement mémoire d'autre part sont autant de paramètres dont il faut tenir compte dans le choix d'une grammaire pour résoudre un problème donné. Ceci peut déboucher sur une métrologie[47] des grammaires ou une méthode d'évaluation, leur classification en fonction de critères rigoureux.

III Bilan

Après ces rappels des multiples aspects de la minimalité, nous nous devons de signaler certains aspects délaissés ou encore insuffisamment explorés. Convaincus que le sujet en vaut l'étude et qu'il est riche de développements potentiels, nous émettons cependant une série d'interrogations sur des aspects qu'il nous semble indispensable d'étudier de manière plus approfondie. Ces remarques ne constituent pas en elles-mêmes une critique de fond de la théorie, et nous sommes conscient du fait qu'il ait fallu, au gré du développement des idées, privilégier telle piste et abandonner temporairement telle autre.

Morphologie

Les efforts ayant porté dans cette phase sur la théorie des automates et sur son adéquation aux propriétés de la morphologie arabe en tant que paradigme de calcul, un bilan prospectif de la poursuite de cet ouvrage se traduit par les constatations suivantes.

- La méthode d'augmentation par tests/actions présente l'intérêt de respecter

46. [Gaubert 95], [97] et [2000].

47. [Jaccarini 97] conclusion.

strictement le noyau régulier de la morphologie, puis de le complexifier par transduction. Cependant, cette méthode peut être complexe à mettre en œuvre pour la morphologie non saine qui peut comporter des centaines de cas. Ce n'est pourtant pas la seule méthode possible: une autre piste à explorer est la méthode de modification locale du réseau de base, dans des limites raisonnables de complexité, pour la prise en charge d'une partie de ces cas qui devront ensuite être filtrés dans un post-contrôle des interprétations.

• En morphologie nominale comme verbale, le contrôle de la validité de la racine extraite est nécessaire, celui du schème également: se passer de ces contrôles conduit à un niveau intolérable d'ambiguïté.

Sur le plan des réalisations, le modèle est incomplet:

• la morphologie nominale est insuffisamment décrite, notamment pour la morphologie non saine qui n'a été qu'effleurée alors qu'elle représente presque 50 % des cas[48].

• La morphologie verbale est aussi incomplète (pas de non sain) et les grammaires verbales n'ont pas été testées et évaluées à grande échelle.

• La détection des tokens est insuffisante, de nombreux cas (tokens en deux parties, suites de tokens, et cas particulier de concaténations) ne sont pas pris en compte.

Il reste donc à mener une étude statistique globale, même rapide, de l'impact des grammaires sans lexique et du phénomène des ambiguïtés croisées entre N, V et T à l'échelle d'un corpus; cette étude conduirait à dégager des critères d'évaluation des grammaires.

Syntaxe

L'étude de la syntaxe doit également être poursuivie dans plusieurs directions.

• Les classes de tokens sont insuffisamment décrites; seuls quelques cas – il est vrai essentiels – le sont. Il reste à modéliser une micro-syntaxe qui permettrait d'affermir la détection des tokens et de leur conférer un rôle de désambiguïsateur local.

• Les règles dominantes restent insuffisamment explorées, une étude sur corpus ciblé de la syntaxe permettrait d'isoler ces règles.

---

48. Voir [Gaubert 95] et chapitre III pour des statistiques à ce sujet.

• L'étude de la syntaxe minimale pourrait s'appuyer sur l'approche par «phrase vide» dont l'intérêt est patent, non seulement pour le traitement automatique, mais aussi pour les objectifs O2 et O3 (connaissance et didactique) exprimés au début de ce chapitre.
• La théorie de la minimalité fait peu de cas de la sémantique, alors qu'elle peut aussi jouer un rôle important dans la désambiguïsation syntaxique; une «sémantique quotient» reposant en partie sur les schèmes reste à développer.
• Il reste à réaliser une simulation informatique de la «phrase vide», ainsi que l'implémentation des désambiguïsations permises par les tokens. Une possibilité simple de variation des grammaires devra être développée, au moyen d'une interface graphique.
Le moniteur syntaxique reste le but ultime qui devra s'appuyer sur ces éléments.

Positionnement cette étude
La confrontation des méthodes développées avec un corpus de textes nous semble être une première étape indispensable à partir de laquelle découle non seulement la validation mais aussi la poursuite du volet syntaxique et éventuellement les remises en question. C'est l'axe principal de cet ouvrage. Notre but sera donc de contribuer à une théorie de la validation des analyseurs morphologiques, à travers l'exposé de la complexification de grammaires. Ce travail se propose ainsi de poursuivre et d'illustrer les principes I précédemment décrits, et particulièrement I4, I5 et I3.
Nous prendrons le parti de réaliser un nouvel analyseur dans un langage déclaratif classique, le C, auquel seront joints plusieurs niveaux de contrôle des interprétations, et notamment un niveau d'étiquetage, indispensable pour l'analyse contrôlée de corpus. Faire varier l'automate en introduisant des catégories intermédiaires tant que cela sera possible nous permettra de réserver à la partie AFND le maximum de traitements, tout en évitant de trop coûteuses epsilon-transitions.
Dans un second mouvement, nous tenterons d'améliorer la détection des tokens en prenant en compte leur rôle syntaxique dans leur catégorisation, ce qui n'a pas été le cas jusqu'à présent. Cette détection devra donc s'opérer en collaboration avec les analyseurs morphologiques mis au point dans la phase précédente, de comportement désormais connu.

C’est sur la base de ces détections que nous pourrons alors envisager un traitement syntaxique, qui demeure à nos yeux la finalité de cette approche.

Soulignons enfin la possibilité de renversement de la définition de la minimalité. Celle ci devient alors un programme de recherche, qui peut être ainsi formulée:

– en n’utilisant que des méthodes minimales, à quel résultat aboutit-on?

– Parvient-on à une description syntaxique complète, ou laconique et à quel degré?

– Existe-t-il d’autres retombées, moins fortes *a priori* que la description syntaxique, mais qui peuvent avoir leur importance dans des application diverses de l’informatique linguistique, de l’informatique documentaire à l’enseignement de l’arabe par l’ordinateur?