UNIVERSITÉ PARIS-SORBONNE

ÉCOLE DOCTORALE X

Laboratoire de recherche X

T H È S E

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ PARIS-SORBONNE

Discipline : Linguistique, section Informatique

Présentée et soutenue par :

Dhaou GHOUL

Le : 07 décembre 2016

Classifications et grammaires des invariants lexicaux arabes en prévision d'un traitement informatique de cette langue.

«Construction d'un modèle théorique de l'arabe : la grammaire des invariants lexicaux temporels ».

Sous la direction de :

M. Amr Helmy Ibrahim – Professeur, Université Sorbonne Paris 4

Membres du jury :

M. Mounir Zrigui – Professeur, Université de Monastir, Tunisie– Rapporteur.
M. Mohamed Embarki – Professeur, Université de Franche-Comté, Besançon– Rapporteur.
M. André Jaccarini– Chercheur CNRS, USR3125.
M. Amr Helmy Ibrahim – Professeur, Université Sorbonne Paris 4.

REMERCIEMENTS

Ă l'issue de la rédaction de cette thèse, je suis totalement convaincu que la thèse est loin d'être un travail individuel. En effet, je n'aurais jamais pu réaliser ce travail sans le soutien et l'aide de quelques personnes.

Tout d'abord, je tiens à remercier sincèrement mon directeur de thèse, monsieur *Amr Helmy Ibrahim*, pour la confiance qu'il m'a accordée en acceptant de diriger ce travail de recherche, pour ses remarques pertinentes, pour sa disponibilité tout au long de la réalisation de ce travail. J'aimerais également lui dire que je vous dois en grande partie la finalisation et la réussite de ce travail doctoral et je vous souhaite de tout cœur une bonne santé.

J'adresse aussi mes remerciements à monsieur *André Jaccarini*, pour son accueil chaleureux à chaque fois que j'ai sollicité son aide ainsi que pour ses multiples encouragement. Je le remercie également d'avoir accepté de faire partie de mon jury et de m'avoir prodigué des remarques très pertinentes.

Je profite de cette occasion de remercier autant Madame *Claude Audebert* d'avoir accepté de me lire et corriger à plusieurs reprises ma thèse malgré ses multiples occupations.

Je souhaiterais exprimer ma gratitude à monsieur *Christian Gaubert* pour m'avoir donné l'occasion de passer un séjour d'un mois à l'IFAO (Institut Français d'Archéologie Orientale), pour son accueil chaleureux. Je le remercie aussi pour son aide durant ces années de doctorat.

Je remercie également *Samir Zardan*, directeur du service informatique de la MMSH et coresponsable au sein de cette institution du programme TALA de m'avoir invité de participer de manière active au cycle de séminaires « Invariances et calculabilité en langue arabe et en sémitique » (ICLAS) où j'ai pu à plusieurs reprises exposer l'état d'avancement de mes travaux et pour son franc soutien au sein de cette institution.

Mes remerciements anticipés s'adressent également à Monsieur *Mounir Zrigui*, professeur d'enseignement supérieur à l'université de Monastir en Tunisie et monsieur *Mohamed Embarki*, professeur à l'université de Franche-Comté, Besançon, qui m'ont fait l'honneur d'accepter la tâche d'être rapporteurs de ce travail.

Je remercie infiniment mes parents pour leur soutien durant toute ma vie.

Je remercie tous mes frères et mes sœurs ainsi que mes amis, qui m'ont toujours encouragé au cours de la réalisation de cette thèse. Merci à tous et à toutes.

Ghoul Dhaou

Position de thèse

L'approche algorithmique de la langue arabe et de sa grammaire est née bien avant que n'apparaissent les premiers systèmes informatiques pouvant traiter correctement les caractères et l'écriture arabe.

On peut en effet considérer que le traitement par algorithmes de la langue arabe a pour origine l'étude pionnière de David Cohen Vers un traitement automatique de l'arabe qui date de 1960. À la fin de cette étude, où les lexèmes arabes sont définis de manière standard comme le croisement d'un schème et d'une racine, figure la description d'un ensemble de procédures pouvant s'intégrer dans des analyses de mots graphiques. Dès 1961, André Lentin, professeur à la Faculté des Sciences de Paris, a démontré que cette description pouvait prendre une forme algorithmique.

Signalons aussi que vers 1978 un programme de tri et de reclassement de formes arabes en vue d'un prétraitement morphologique a été écrit en APL, qui est un langage fonctionnel d'opérateurs algébriques, au centre scientifique d'IBM-France. Ce programme attribuait à chaque codage normé de mot graphique arabe une valeur numérique représentant sa complexité a priori relativement à l'opération d'extraction de la racine ; les trois critères retenus étant la valeur radicale des caractères, leurs positions dans les mots graphiques qui les contiennent ainsi que la longueur de ces derniers. Ce prétraitement linguistico - algébrique est nécessaire pour se donner par la suite la possibilité de n'avoir à appliquer qu'une seule fois et en bloc des opérations constitutives à toutes les analyses possibles - ayant pour fin l'extraction de la racine - sur des classes entières de mots graphiques extraits du texte présentant des caractéristiques communes, afin d'éviter les procédures trop répétitives et se conformer ainsi à la logique d'APL .

Des algorithmes relevant du même domaine ont également été simulés dans le langage LISP au début des années 1980 (Centre Mondiale de l'Informatique, Paris), auxquels il est fait référence dans une étude datant de 1986 portant sur la possible optimisation et la désambiguïsation de toute analyse morphologique par un prétraitement syntaxique (Audebert et Jaccarini 1986).

L'intérêt de cette étude, qui a donné par la suite naissance à un programme de recherche en linguistique formelle arabe au CNRS, est double.

Premièrement, sur le plan strictement algorithmique : la nécessité de faire précéder toute procédure d'analyse morphologique par un programme de reconnaissance et d'isolement de toutes les formes arabes figées (celles en particulier qui ne peuvent s'analyser comme la combinaison d'un schème et d'une racine) afin d'éviter l'échec de l'analyseur, amène logiquement à essayer de tirer le meilleur bénéfice de cette première opération que l'on applique

sur l'ensemble du texte. Or, une fois cette reconnaissance effectuée, les informations que l'on peut extraire et exploiter en vue de la désambiguïsation et de la réduction des silences et des bruits (qui se produisent immanquablement même en recourant à des lexiques et des contrôles post-parsing) sont considérables. Ceci est dû au fait que ces éléments qui échappent au système de dérivation morphologique de l'arabe - c'est à dire auxquels on ne peut faire correspondre canoniquement un schème et une racine - recoupent pour une bonne part ceux que l'on désigne, faute de mieux, par mots outils ou particules , lesquels induisent sur leurs environnements des contraintes, notamment syntaxiques, plus fortes que celles de la plupart des lexèmes standard. C'est le principe de la segmentation entropique : ces éléments provoquent une chute du degré de liberté dans tout flux de caractères arabes respectant les contraintes grammaticales. Une fois ce fait constaté il apparaît alors normal de privilégier ces éléments dans toute description de régularité des flux grammaticaux. On trouvera ainsi dans ce travail un nombre important de graphes orientés récursifs étiquetés par des catégories minimales reflétant les contraintes, notamment syntaxiques, induites par ces éléments. Les contraintes induites, qu'elles soient globales ou locales, sont souvent suffisamment fortes pour que les éléments en question puissent être considérés comme des structurants de la phrase. L'argument algorithmique défendu est de nature top down : c'est une approche qui s'apparente à la méthode récursive. Les contraintes syntaxiques exprimées par les graphes peuvent en effet être facilement converties en automates. Quant aux étiquettes des arcs de leurs graphes de transition, elles peuvent être considérées comme des appels récursifs de sous-programmes morphologiques. Si l'on parvient ensuite à exprimer les contraintes morphologiques exclusivement sous forme d'automates fonctionnant lettre à lettre, on obtient alors une très grande unité dans la représentation. Cette unification des représentations syntaxiques et morphologiques, outre le fait qu'elle constitue un gage de consistance de la description, nous offre surtout la perspective de la construction d'un modèle syntaxique de l'arabe dont le vocabulaire terminal serait extrêmement réduit puisque constitué uniquement de l'alphabet de l'arabe (avec ou sans les signes diacritiques) et d'un petit ensemble de formes invariantes (150 environ). Cette possible économie descriptive constitue une spécificité forte de la langue arabe et a fait l'objet de plusieurs publications depuis 1988.

Mais pour la mise en œuvre d'une analyse de ce type, c'est à dire celle d'un processeur morphologique guidé par la syntaxe, il est nécessaire de concevoir un programme cadre qui orchestre les automates de différents niveaux, morphologiques et syntaxiques : « le moniteur syntaxique ». Sa fonction fondamentale sera celle d'un aiguilleur. Après avoir balayé le texte pour repérer les éléments lexicaux invariants et les attentes d'étiquettes qu'ils déclenchent au niveau de la phrase, il pourra diriger l'analyseur morphologique de l'occurrence qu'il traite vers les branches adéquates et éviter ainsi l'ambiguïté inutile et les explosions combinatoires. Dans les travaux d'A. Jaccarini (Jaccarini 1997), la gestion de l'indéterminisme des réseaux de

transition, ainsi que les questions liées à la suppression de la récursivité inhérente aux descriptions syntaxiques, sont traitées mathématiquement par résolution d'équations algébriques grâce à une formalisation - dans le cadre de la théorie du monoïde syntaxique et des corps non commutatifs - de l'algorithmique sous-jacente à l'analyse linguistique arabe. Il en de même de la question importante de synthèse de grammaire à partir de grammaires fragmentaires. Quant à l'ambiguïté linguistique au niveau morphologique elle est traitée de manière approfondie par (Christian Gaubert, 2001). Le comportement de cascades d'automates nominaux et verbaux est étudié en vue du classement des grammaires selon le niveau de l'ambiguïté produite. Pour ce faire, un logiciel de manipulation des automates linguistique a été conçu *Sarfiyya* ancêtre du système *Kawâkib* que nous utilisons dans notre travail (voir chapitre VII).

Enfin, pour ce qui est du choix du modèle de calcul, c'est à dire celui des machines finies (automates et transducteurs), il est à noter qu'une spécification de l'ensemble de l'étude pionnière et consistante de David Cohen - évoquée plus haut - par un simple automate fini, non déterministe, de complexité minimale et ne comportant que 6 états a été donnée par A. Jaccarini (Audebert et Jaccarini 1994). Cette concision dans la description linguistique est remarquable et justifie, à elle seule, le recours au formalisme des automates dans la mesure où l'on admet le principe selon lequel « toute économie d'écriture reflète un approfondissement conceptuel », qui est à la base de tout système axiomatique. La mise en œuvre est également intéressante grâce à l'analyseur Lisp pour automate non déterministe mis au point. En effet cet analyseur est transparent et offre ainsi la possibilité de contrôler l'évolution des états de la « pile à double entrée » (semblable à la structure DEQ décrite dans Sedgewick) : il est possible de contraindre l'évolution de cette pile en tenant compte de règles linguistiques, ce qui permet d'effectuer des élagages en cours d'analyse des hypothèses improductives correspondant à des tentatives de parcours inutiles (dans l'automate). De plus, cet analyseur (qui peut modulairement être enrichi pour traiter des grammaires augmentées de tests et d'actions) opère en un temps directement proportionnel à la longueur des chaînes analysées. Une spécification de cet analyseur par des équations algébriques, est donnée dans Jaccarini (Jaccarini 1997). Une deuxième grammaire « moins fruste » (30 états) est proposée dans la deuxième partie de l'article ; les comparaisons systématiques des « configurations » prises par la mise en œuvre des deux automates est faite en annexe ; cette étude est donc une introduction à la « méthode de la détermination de l'algorithme optimal par variation de grammaire » qui est l'un des fondements du programme TALA (Traitement par automates de la langue arabe) : les grammaires ne sont jamais figées et elles sont conçues pour être ajustées par feedback. Mais il ne faut pas perdre de vue que le couplage des processeurs morphologiques et syntaxiques peut s'avérer délicate en raison de risque de cercles vicieux (deadlock).

En conclusion de ce premier point concernant l'intérêt de cette approche sur le plan strictement algorithmique retenons surtout la possibilité, comme cas limite, de travailler sans lexique c'est à dire de concevoir des analyseurs modulables pouvant fonctionner sous différentes options allant du dictionnaire vide au lexique complet car il s'agit là d'une spécificité de la langue arabe (voir même du système sémitique dans son ensemble).

Deuxièmement, cette étude présente aussi un intérêt sur le plan cognitif en ce sens que, dès le départ, elle a été conçue comme une recherche de méthodes rigoureuses en vue de parvenir à modéliser, autant que cela est possible, le processus d'appréhension de l'arabe par un apprenant de cette langue. L'une de ses ambitions sous-jacentes est d'aboutir à la modélisation d'une hypothèse cognitive sur l'optimisation de la recherche dans le dictionnaire. En effet l'expérience de pensée qui en est à l'origine a consisté à supposer qu'il existait un processeur abstrait et optimal de décodage d'un texte arabe non vocalisé qui minimise le recours au lexique. L'originalité de cette approche réside dans l'adoption d'un point de vue s'inspirant du paradigme récursif, comme signalé plus haut. En effet, d'emblée le problème de l'analyse morphologique de la détermination de la racine que nécessite l'accès au dictionnaire est supposé résolu et l'on imagine à l'aide d'un programme fictif l'optimisation de l'ensemble des processus sous-jacents à cette analyse (méthodologie top-down), en se libérant de la contrainte de la progression mot à mot. L'idée principale est que la stratégie consistant à avancer dans le texte, en recourant le moins possible au dictionnaire, est la plus efficace à condition toutefois de savoir repérer les éléments les plus contraignants (segmentation entropique) et mettre en œuvre en en tirant judicieusement parti les attentes morphosyntaxiques qu'ils déclenchent.

L'expression de ces contraintes sous forme d'automates s'imbriquant les uns dans les autres découle donc de l'hypothèse, que suggère l'expérience pédagogique, de l'existence de ce processeur idéalisé. Or celui-ci, s'il existe, ne peut être appréhendé que par approximations successives et la définition de méthodes de construction des grammaires par agrégation de fragments en ayant recours à la théorie algébrique des automates. Par ailleurs la mise au point progressive des fragments de grammaire ne peut se faire qu'en procédant par rétroaction continue, d'où la nécessité de la mise au point d'un outil permettant d'assurer le feedback entre corpus et grammaires lequel permettra ainsi de boucler rétroactivement jusqu'à obtenir un niveau jugé satisfaisant. C'est dire que l'une des caractéristiques essentielles de ce programme est bien le souci d'une mise en parallèle des processus cognitifs et de l'algorithmique ainsi que la rétroaction continue - ou feedback - entre processus d'appréhension et algorithmes, ce que la transparence des analyseurs – que nous avons évoqué plus haut - permet d'envisager.

Alors, cette thèse porte sur la classification et le traitement des invariants lexicaux arabes qui expriment un aspect temporel identifiés manuellement au sein d'un large corpus d'arabe moderne contemporain afin de créer un modèle qui présente chaque invariant sous la forme

d'un schéma de grammaire (automates à états finis) compréhensible par une machine. Notre analyse est basée sur la méthode « *feedback* » à partir de ce corpus (contrôle de *feedback* entre grammaires et corpus). Dans ce travail nous avons limité notre traitement seulement pour 20 invariants lexicaux. Notre hypothèse part du principe que les invariants lexicaux sont situés au même niveau structural (formel) que les schèmes dans le langage quotient (squelette) de la langue arabe. Ils cachent beaucoup d'informations et entraînent des attentes syntaxiques qui permettent de prédire la structure de la phrase. Cette hypothèse nous amène à analyser, désambiguïser et comprendre le mode de fonctionnement ces invariants lexicaux ainsi que leur rôle dans la phrase en arabe.

Notre travail s'articule en deux grandes parties. Dans la première partie de cette thèse, nous avons montré l'apport de l'analyse des invariants lexicaux dans la construction d'applications TAL performantes. En nous fondant sur des exemples concrets nous avons pu montrer les lacunes des quelques applications TAL à cause d'une analyse insuffisante ou l'écartement de ces invariants lexicaux (désambiguïsation syntaxique et sémantique) dans la phase d'analyse. De plus, nous avons abordé la notion « *invariant lexical* » en exposant les différents niveaux d'invariance. Il existe si l'on peut dire plusieurs niveaux d'invariants ceux par exemple que l'on peut définir comme les invariants du morphisme SC (ils sont leur propre projection et se situent donc au même niveau que les classes d'équivalence (congruence) définies par les schèmes, ceux dont la classe de congruence grammaticale « $\sim_g$ » est égale à l'unité et à l'inverse ceux qui sont permutables avec un nom, un verbe, des syntagmes nominaux ou verbaux, etc., ou ceux dont la sémantique est plus ou moins vide (voir ci-dessous) mais qui ne sont pas suppressibles, ceux qui le sont, ceux qui sont des « opérateurs » casuels et ceux qui ne le sont pas, etc. Ensuite, nous avons classé les invariants étudiés dans cette thèse selon plusieurs critères. Après cette classification, nous avons constaté que certains invariants lexicaux peuvent appartenir à plus d'une classe en même temps.

La deuxième partie de cette thèse fait l'objet de notre propre étude concernant les invariants lexicaux temporels. Ces derniers sont répartis en deux grandes classes : invariants simples et invariants complexes.

Comme nous l'avons noté plus haut, les invariants lexicaux sont identifiés manuellement au sein d'un large corpus d'arabe moderne contemporain. C'est pourquoi, au début de notre travail de recherche, nous avons décidé de construire notre propre corpus. Ensuite, nous avons annoté ce corpus syntaxiquement pour nos travaux à long terme

Dans cette partie, nous présentons notre méthode d'étude linguistique ainsi que la modélisation par schémas de grammaires des invariants lexicaux temporels étudiés. Ces schémas de grammaires représentent les résultats de notre recherche théorique ainsi que le fruit d'une expérimentation linguistique. Ensuite, nous abordons l'analyse proprement dite des

invariants lexicaux simples comme « *ḥattā, baʿda* » et complexes comme « *baʿdamā, baynamā* ». L'objectif de cette analyse est de dégager le maximum d'informations sous forme des règles linguistiques afin de les modéliser dans des grammaires régulières acceptables par des automates à états finis.

Nous finissons nos travaux de recherche par une présentation de l'application expérimentale « *Kawâkib* » qui nous avons utilisé pour détecter et identifier les invariants lexicaux en montrant leurs points forts ainsi que leurs lacunes. Avant la clôture de cette thèse, nous proposons une vision de la prochaine version de « *Kawâkib* » pour la rendre, pourquoi pas, une application pédagogique qui traite la langue arabe sans lexique.

Pour mener à bien notre recherche nous avons dû surmonter un certain nombre de difficultés. La première concerne la classification des invariants lexicaux, lorsqu'il s'agit notamment de séparer les critères purement syntaxiques de ceux qui sont de nature sémantique. La deuxième concerne l'ambiguïté graphique ainsi que grammaticale de certains invariants lexicaux qui peuvent être levées par le recours au contexte. Cette ambiguïté rend l'analyse de ces invariants plus difficiles. La troisième concerne l'intégration « le sens » dans la représentation formelle (schémas de grammaires) surtout lorsqu'il s'agit d'automates non augmentés.

L'une de nos contributions les plus importantes au programme TALA est celle qui a consisté à fournir des *schémas* d'automates attachés à des invariants lexicaux qui comptent parmi ceux dont « le poids sémantique » est le plus fort, à savoir les invariants lexicaux qui sont des marqueurs de temporalité et d'aspect. Les graphes de transition de ces schémas d'automates sont étiquetés par des labels de catégorie standard qui sont admises par la plupart des linguistes.

Notre travail nous a permis d'aboutir à différents constats. Parmi ces constats on cite :

- ✓ Les invariants lexicaux temporels jouent un rôle très important dans la construction de la phrase.
- ✓ Pour certains invariants, la forme graphique unique cache une ambiguïté grammaticale comme « *ḥattā* » qui peut jouer le rôle d'un adverbe, d'une conjonction de subordination ou d'une préposition.
- ✓ Certains invariants contiennent des marqueurs linguistiques : nombre, genre, mode, etc.
- ✓ Dans certains cas, la présence de l'invariant dans la phrase dépend de deux sous-phrases (noyaux) comma dans le cas de « *baynamā* ».
- ✓ La densité des invariants lexicaux permet de prédire la nature de texte : narratif, descriptif, …

- ✓ La suppression de la plupart des invariants lexicaux entraîne la perturbation du sens de la phrase. Cependant, ce n'est pas toujours le cas pour la structure. Dans certains cas, la suppression de l'invariant lexical ne perturbe pas la structure de la phrase.
- ✓ Certains invariants sont discontinus: ils attendent forcément un autre invariant plus loin comme « *baynamā….'iḏ…..* ».
- ✓ La commutation d'un invariant avec un autre en gardant exactement le même sens d'origine est très rare.
- ✓ Au niveau structural, les invariants lexicaux se situent au même niveau que les schèmes dans le langage squelette de la langue arabe.
- ✓ Construction d'une base de grammaires des invariants temporels.
- ✓ Possibilité d'analyser la langue arabe en nous fondant sur le langage squelette.

Malgré ses limites et comme tout travail de recherche, ce travail de thèse ouvre des perspectives pour de futures recherches dans le domaine du traitement de la langue arabe par automates (TALA). Parmi ces perspectives, nous citons :

- ✓ Tout d'abord, il nous semble très utile d'évaluer nos grammaires des invariants temporels sur une masse de données.
- ✓ Il nous semble aussi que les résultats de notre travail sur les invariants temporels motivent la continuation de ce travail sur les autres invariants lexicaux de l'arabe.
- ✓ Amélioration des schémas de grammaires en les rendent déterministes si possible afin de ne produire que des phrases correctes (valides).
- ✓ Factoriser les contextes gauche et droite des invariants lexicaux afin de développer les schémas de grammaires des invariants lexicaux.
- ✓ Vers une banque de grammaires des invariants lexicaux arabes : Modèle théorique original.
- ✓ évoluer nos grammaires par « *kawâkib* ».
- ✓ Intégration de « sens » dans les schémas de grammaires : vers automates à états finis augmentés.
- ✓ Réalisation d'une application efficace pour le traitement automatique de l'arabe : amélioration de « *kawâkib* ».

Nous devons rappeler que ce travail s'inscrit dans le cadre général du projet Mogador, brièvement exposé dans l'introduction de cette thèse[1].

Notre principale contribution à ce projet se situe au niveau de la constitution, dans une optique de linguistique de corpus, d'un stock consistant, quoique non exhaustif, d'informations concernant l'environnement morphosyntaxique des invariants lexicaux considérés. Ces invariants comptent parmi les plus importants puisque, concernant principalement les questions

[1] Voir en annexe G le diagramme représentant l'articulation des différents programmes qui le sous-tendent

de temporalité et d'aspect voire d'« argumentativité », ils constituent de forts marqueurs de la structure générale de la phrase.

Ces informations recueillies, à partir d'analyses plus ou moins approfondies mais non figées de corpus et de discussions sur leurs étiquetages, ont été ensuite systématiquement formalisées sous forme de diagrammes représentant des graphes orientés récursifs. Ces graphes constituent déjà des *schémas* d'automates finis (voir les diagrammes ci-dessous), dont les transitions pourront par la suite être éventuellement conditionnées par des tests et auxquels des actions minimales pourront aussi être associées[2].

La réduction à des machines finies est en effet toujours possible car il est à remarquer que la récursivité qui apparaît dans la plupart de ces schémas de grammaire est suppressible. Il ne s'agit pas d'une récursivité intrinsèque du point de vue algorithmique car les étiquettes représentent

- Soit des appels récursifs de sous automates morphologiques, lesquels sont toujours réductibles à des automates finis que l'on insère en remplacement de la transition récursive en question ;

- Soit des appels d'autres schémas de grammaire syntaxique, ne contenant pas non plus de récursivité irréductible ; car pouvant être transformées grâce au calcul en schémas *faiblement* équivalents ne comprenant pas de récursivité ; l'imbrication de structures en partie centrale du type $X \rightarrow a\ X\ b$, laquelle est irréductible (*pumping lemma*)[3], n'intervenant jamais après transformation adéquate.

Relativement à ce dernier point, remarquons aussi que la perte de pertinence linguistique immédiate, consécutive à cette transformation possible de grammaire, peut être avantageusement compensée, en cas de besoin d'un indicateur syntagmatique pertinent pour certains types d'opérations, par la création, contrôlée algébriquement, de registres d'automates minimalement augmentés (par une méthode de génération algébrique de transducteurs finis minimaux et optimisés).

En regard de tous les schémas de grammaire que nous avons présentés dans cette thèse nous avons pris soin à chaque fois de fournir les règles de production correspondant aux diagrammes de transition sous forme de règles de réécriture. Or ces règles peuvent être facilement traduites en systèmes d'équations dont les inconnues ne sont autres que les éléments

[2] Les actions ou les fragments de programme associés seront néanmoins suffisamment élémentaires (principe de minimalité) pour ne pas remettre en cause le principe des calculs algébriques effectuables sur ces schémas de grammaire en vue de les transformer, si nécessaire, en schémas « faiblement équivalents ». Ces tests et ces actions ne dépendent que des portions de parcours initiaux qui ont été déjà effectués dans l' « automate » au moment du traitement du symbole courant. Rappelons que deux grammaires sont dites « fortement équivalentes » si non seulement elles acceptent les mêmes chaînes de caractères mais si en plus elles fournissent les mêmes descriptions structurales.

[3] Ou « lemme de l'étoile » (Chomsky, Schützenberger, 1962).

du vocabulaire auxiliaire des schémas des grammaires, autrement dit les catégories, c'est-à-dire les étiquettes du graphe. Et il est possible de résoudre ces systèmes d'équations dans des structures algébriques adéquates - notamment dans des corps non commutatifs - en vue d'aboutir à des expressions régulières, éventuellement augmentées, codant des machines (automates ou transducteurs) finis ou d'effectuer plusieurs type de calcul pour vérifier par exemple des égalités d'expressions établissant l'équivalence faible de deux schémas de grammaire, les « dérécursiver » , transformer certains tronçons en sections déterministes, vérifier la cohérence de certaines jonctions, minimaliser la grammaire ou bien encore calculer le niveau d'indéterminisme ou le degré d'ambiguïté avec possibilité de les réduire ou les supprimer totalement, etc. Bien plus, les parseurs eux-mêmes - auxquels on est obligé de recourir en cas de non déterminisme, peuvent être, directement obtenus par simple calcul dans la structure algébrique en question avec leur preuve de validité et leurs optimalité mathématiquement établie. Ces calculs pouvant eux-mêmes être simulés par automates, ce qui ouvre une belle perspective d'unification de l'ensemble du cadre de travail.

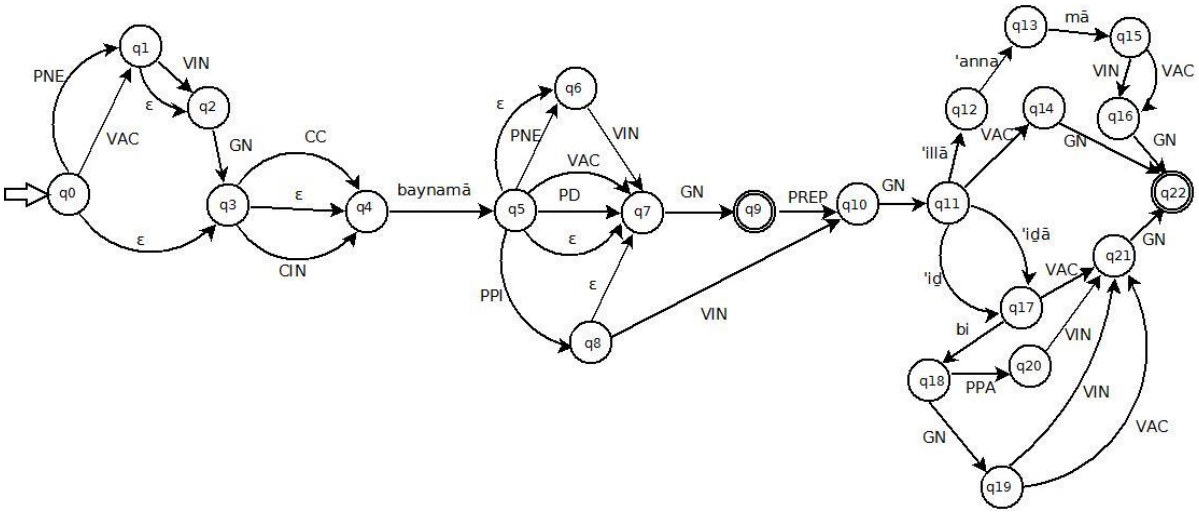

Figure 1: Schéma de grammaire de "baynamā".

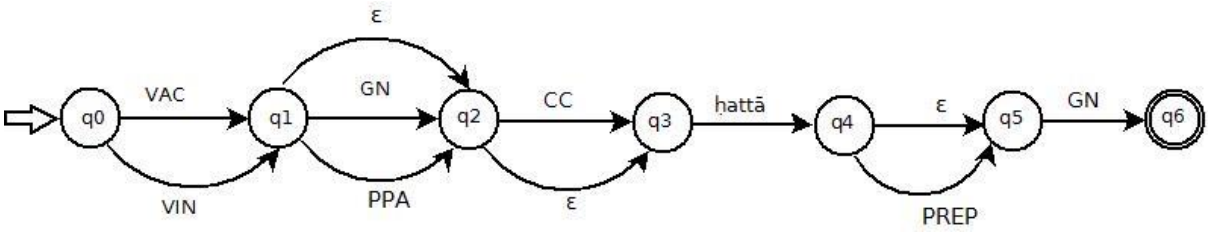

Figure 2: Schéma de grammaire de "ḥattā" dans le cas d'un adverbe

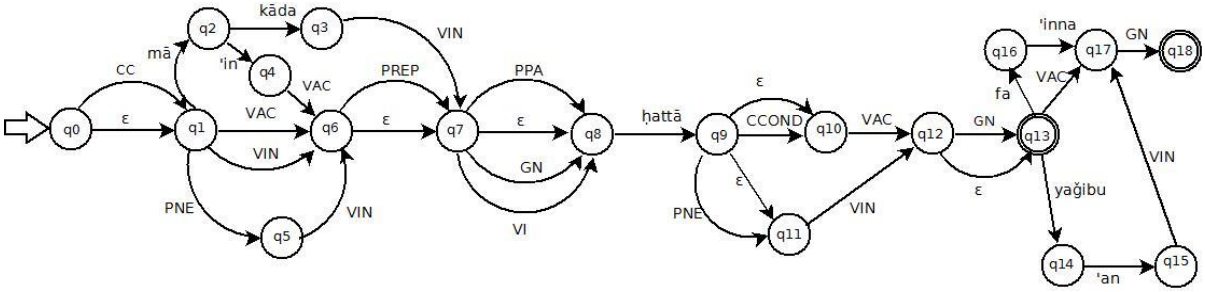

Figure 3: Schéma de grammaire de "ḥattā" dans le cas d'une conjonction de subordination.

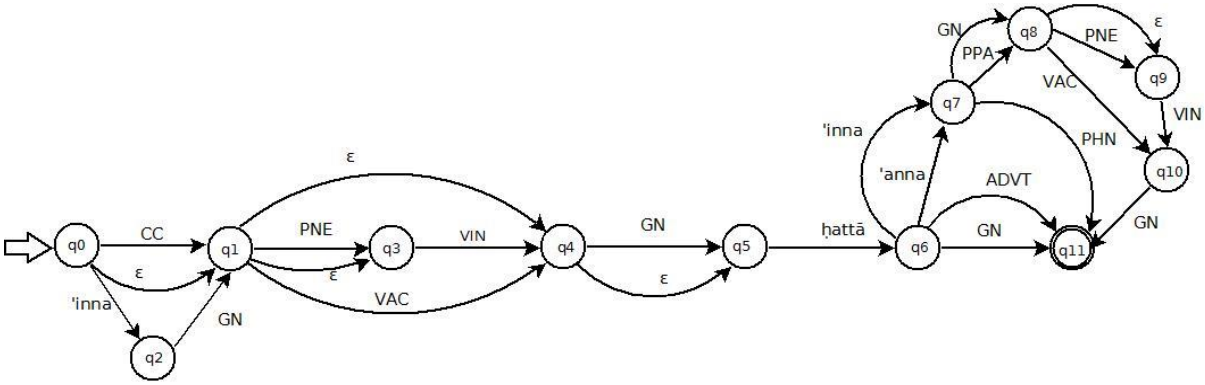

Figure 4: Schéma de grammaire de "ḥattā" dans le cas d'une préposition.

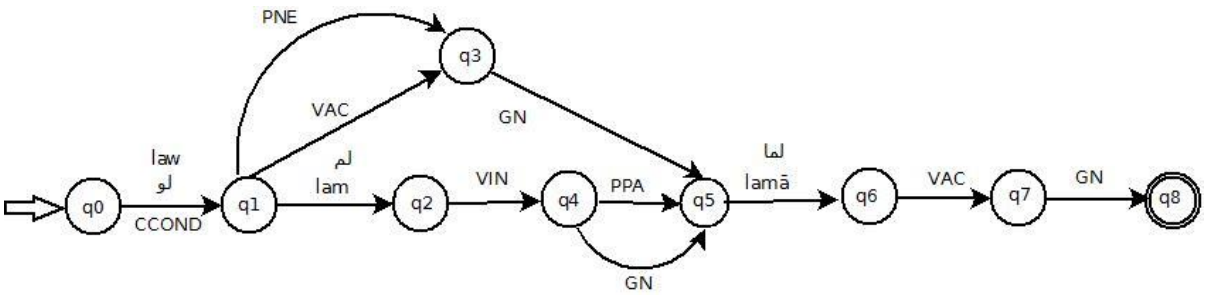

Figure 5: Schéma de grammaire de "lamā" dans le cas d'une particule de négation.

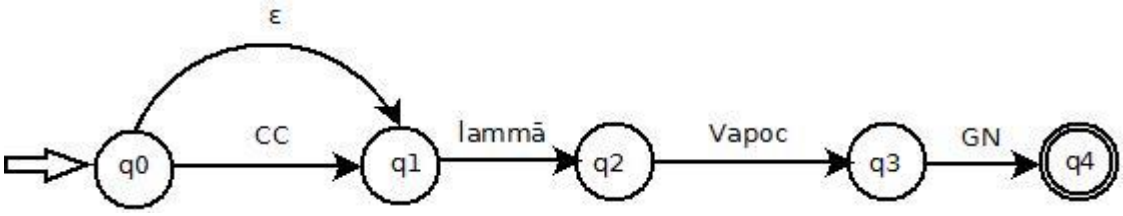

Figure 6: Schéma de grammaire de "lammā" dans le cas d'une particule de négation.

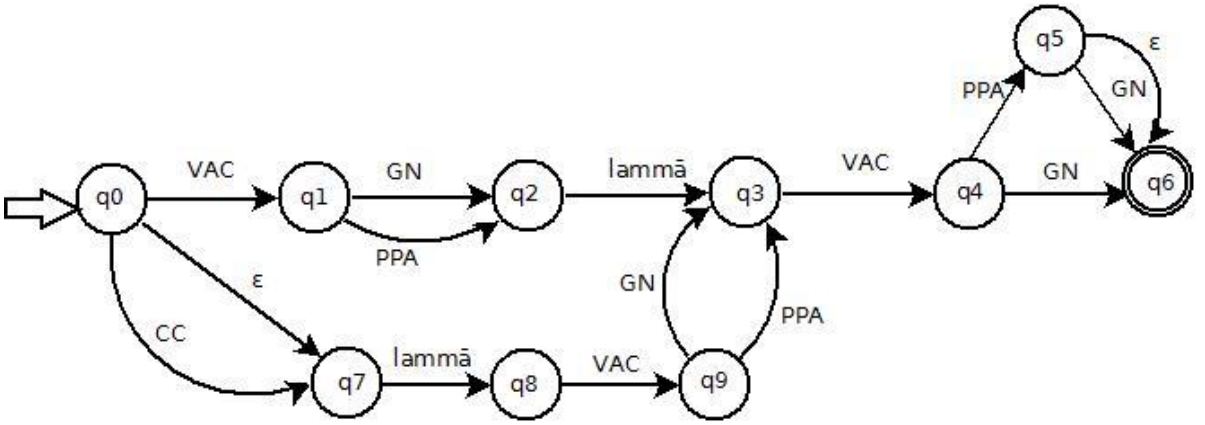

Figure 7: Schéma de grammaire de "lammā" dans le cas d'une conjonction de subordination.

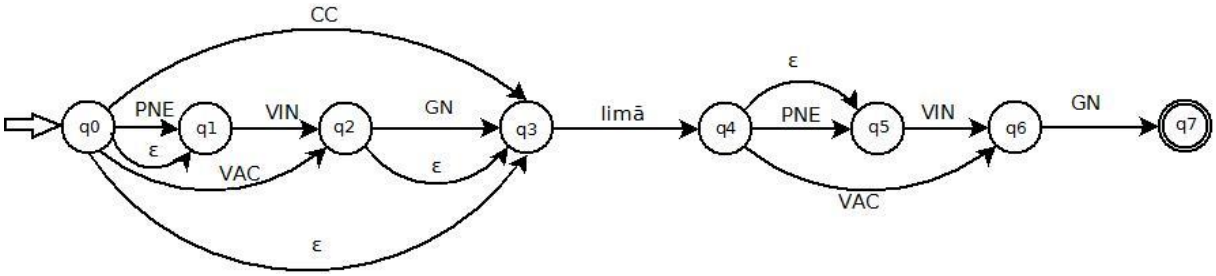

Figure 8: Schéma de grammaire de "limā" dans le cas d'une conjonction d'interrogation.

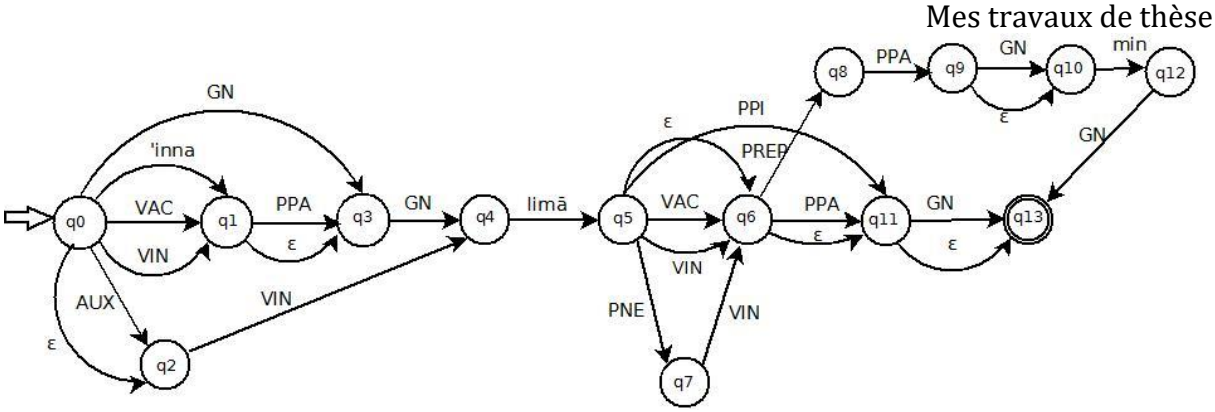

Figure 9: Schéma de grammaire de "limā" dans le cas d'une conjonction de subordination.

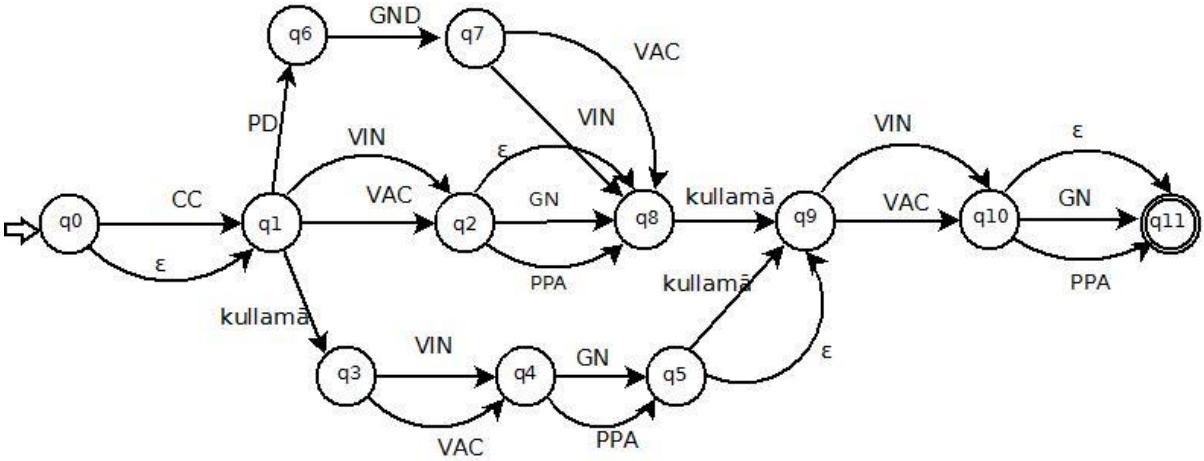

Figure 10: Schéma de grammaire de "kullamā".