



UNIL | Université de Lausanne

Unicentre

CH-1015 Lausanne

<http://serval.unil.ch>

Year : 2011

Un analyseur syntaxique pour le latin classique

Elisa Nury

Elisa Nury, 2011, Un analyseur ayntaxique pour le latin classique

Originally published at : Mémoire de maîtrise, Université de Lausanne

Posted at the University of Lausanne Open Archive.
<http://serval.unil.ch>

Droits d'auteur

L'Université de Lausanne attire expressément l'attention des utilisateurs sur le fait que tous les documents publiés dans l'Archive SERVAL sont protégés par le droit d'auteur, conformément à la loi fédérale sur le droit d'auteur et les droits voisins (LDA). A ce titre, il est indispensable d'obtenir le consentement préalable de l'auteur et/ou de l'éditeur avant toute utilisation d'une oeuvre ou d'une partie d'une oeuvre ne relevant pas d'une utilisation à des fins personnelles au sens de la LDA (art. 19, al. 1 lettre a). A défaut, tout contrevenant s'expose aux sanctions prévues par cette loi. Nous déclinons toute responsabilité en la matière.

Copyright

The University of Lausanne expressly draws the attention of users to the fact that all documents published in the SERVAL Archive are protected by copyright in accordance with federal law on copyright and similar rights (LDA). Accordingly it is indispensable to obtain prior consent from the author and/or publisher before any use of a work or part of a work for purposes other than personal use within the meaning of LDA (art. 19, para. 1 letter a). Failure to do so will expose offenders to the sanctions laid down by this law. We accept no liability in this respect.

Un analyseur syntaxique pour le latin classique
Prof. E. Wehrli
Prof. F. Bavaud

Maîtrise universitaire ès Lettres en IMM
Elisa Nury
Septembre 2011

Mémoire de Maîtrise universitaire ès Lettres en

Informatique et Méthodes Mathématiques

Faculté des Lettres
Université de Lausanne

Un analyseur syntaxique pour le latin classique

Par Elisa Nury

Sous la direction du Professeur Eric Wehrli (Genève)
co-directeur : Professeur François Bavaud

Session de septembre 2011

Table des matières

1 Analyse et analyseurs syntaxiques : introduction.....	4
1.1 Une définition par l'exemple.....	4
1.2 L'analyse automatique.....	5
1.2.1 La composante linguistique.....	7
1.2.2 La composante informatique.....	8
1.2.3 Les représentations du texte.....	9
2 Aspects techniques.....	11
2.1 Les grammaires formelles.....	11
2.2 Les algorithmes d'analyse.....	16
2.3 Analyse syntaxique du latin - Etat de la question.....	18
2.3.1 Grammatica Latina.....	19
3 L'analyseur syntaxique Fips.....	23
3.1 Présentation générale.....	23
3.2 Le lexique de Fips.....	24
3.2.1 La place de la morphologie.....	24
3.2.2 FipsMorpho.....	26
3.2.3 Le lexique latin.....	29
3.3 Les étapes de l'analyse : fonctionnement de Fips.....	32
3.3.1 Lecture.....	32
3.3.2 Analyse.....	32
4 Description des structures syntaxiques.....	34
4.1 Verbes.....	34
4.1.1 Position.....	34
4.1.2 Arguments.....	35
4.2 Noms et adjectifs.....	37
4.3 Autres.....	38
4.3.1 Les adverbes.....	38
4.3.2 Les particules enclitiques.....	38
4.3.3 Les compléments de la phrase.....	39
4.4 L'hyperbate.....	40
5 Examen d'un corpus de phrases avec Fips.....	41
5.1 Corpus.....	41
5.1.1 Noms.....	41
5.1.2 Adjectifs.....	41

5.1.3 Prépositions.....	42
5.1.4 Adverbes.....	42
5.1.5 Attributs du verbe esse.....	42
5.1.6 Compléments du verbe.....	43
5.1.7 Passif.....	44
5.1.8 Subordonnées.....	44
5.1.9 interrogation et négation.....	45
5.2 Test.....	46
5.2.1 Noms	46
5.2.2 Adjectifs.....	48
5.2.3 Prépositions.....	48
5.2.4 Adverbes.....	49
5.2.5 Attributs du verbe esse.....	50
5.2.6 Compléments du verbe.....	53
5.2.7 Passif.....	55
5.2.8 Subordonnées.....	56
5.2.9 interrogation et négation.....	58
6 Conclusion.....	60
7 Annexe : le fichier de morphologie latin.....	61
8 Bibliographie.....	62
8.1 Ouvrages.....	62
8.2 Articles.....	64
8.3 Sites internet.....	64

1 Analyse et analyseurs syntaxiques : introduction

Ce travail a pour but d'implémenter une version de l'analyseur syntaxique Fips destinée au latin classique, afin de pouvoir analyser un corpus de phrases simples et représentatives de la prose classique.

La création de ce programme nous permettra de nous interroger sur les particularités spécifiques au latin et les problèmes qui peuvent en découler. La richesse de la morphologie ainsi que la grande liberté dans l'ordre des mots en sont des exemples. La description des structures syntaxiques nous donnera un aperçu des points à traiter et fournira la plupart des phrases du corpus.

Après avoir présenté l'analyse automatique de façon très générale et plus détaillée, nous verrons comment l'analyseur Fips a été adapté au latin.

1.1 Une définition par l'exemple

Qu'est-ce que l'analyse syntaxique ? Ce concept peut paraître abstrait mais pourtant, lorsqu'il étudie un texte, tout latiniste accomplit une analyse syntaxique. Cette étape est en effet essentielle à la compréhension et la traduction du texte. Voici un exemple simple afin d'en donner une meilleure illustration :

*Huius consilium plerisque civitatibus displicebat*¹.

Comment s'y prendre afin de traduire cette phrase ? D'une façon générale, on peut distinguer deux étapes obligatoires.

Pour interpréter correctement la phrase, il est tout d'abord nécessaire de connaître non seulement la signification de chaque mot, mais aussi sa **morphologie** : le genre, le nombre et surtout le cas pour les noms, la conjugaison pour les verbes. Nous obtenons ainsi :

Huius : génitif sg. (masc., fém. ou neutre) du pronom démonstratif *hic*.

Consilium : nominatif - accusatif sg., neutre, de *consilium*.

Plerisque : datif - ablatif pl., masc. ou fém., de l'adjectif *plerique*.

Civitatibus : datif - ablatif pl., fém., de *civitas*.

Displicebat : imparfait actif de l'indicatif, 3e personne du singulier, du verbe *displicere* (déplaire à : + datif).

¹ NEP., *Them.* 3.1.

La seconde étape consiste à construire la **syntaxe** de la phrase. La première chose à faire est de rechercher le verbe principal :

Displicebat : le seul verbe conjugué de notre exemple est le verbe principal. Il demande un complément au datif.

Plersique civitatibus : *civitatibus* est donc le complément de *displicere* au datif. *Plersique* ne pouvant s'accorder avec *consilium*, il est par conséquent associé à *civitatibus*. Ensemble ils forment un syntagme nominal complément du verbe.

Huius consilium : nous déduisons que *huius* est le complément du nom *consilium*, pour des raisons de proximité et de contexte. Ce groupe constitue le sujet de la phrase, car *consilium* ne peut remplir une autre fonction (il ne peut être ni complément du verbe, ni complément du nom *civitatibus*).

Finalement, la construction syntaxique de la phrase se termine avec les éventuels compléments de lieu, de temps et les subordonnées.

Ce n'est qu'après être passés par ces deux étapes brièvement décrites, correspondant à l'analyse morphologique et l'analyse syntaxique, que nous pouvons traduire notre phrase avec un risque d'erreur minimal :

« Son plan déplaisait à la plupart des cités ».

Il reste encore à noter que certaines ambiguïtés ne peuvent se résoudre que grâce au contexte : c'est en effet le seul moyen de savoir que *huius* désigne Thémistocle, et qu'il est donc masculin...

1.2 L' analyse automatique

L'analyse syntaxique s'inscrit dans le contexte plus large du Traitement Automatique du Langage Naturel (TALN). Les domaines qui profitent de l'application pratique du TALN sont nombreux, et l'on peut citer entre autres :

- les correcteurs orthographiques
- la traduction automatique
- la recherche d'information
- la classification de documents
- la synthèse et la reconnaissance de la parole

etc.

La langue naturelle se définit comme la langue parlée et écrite par les êtres humains. Elle s'oppose donc aux langages artificiels tels que les langages de programmation informatique ou encore les langages formels utilisés en mathématiques et en logique.

Le traitement du langage naturel est une opération qui consiste à prendre un texte en langue naturelle et à en extraire une représentation². Cette étape correspond à l'analyse. La génération d'un texte à partir de cette représentation peut constituer une deuxième étape (ce serait le cas d'une traduction, par exemple).

Le TALN est donc une branche qui réunit les domaines de la linguistique et de l'informatique, et dont le but est de créer des programmes capables de traiter la langue naturelle.

Dans cet objectif, le TALN se heurte à certains défis, dont le principal est assurément l'ambiguïté des langues naturelles. En effet, bien que nous y soyons rarement attentifs, la plus banale des phrases peut receler une ambiguïté susceptible d'entraver le cours d'un programme informatique et qu'il sera nécessaire de lever. Voici un exemple typique d'ambiguïté, fréquemment cité dans les ouvrages théoriques :

« Jean regarde l'homme avec un télescope ».

Est-ce Jean qui observe avec un télescope ? Est-ce l'homme (que Jean regarde) qui possède un télescope ?

La recherche d'une couverture lexicale et grammaticale la plus large possible est un autre défi pour l'analyse. Il est en effet très difficile de définir un programme qui reconnaisse toutes les phrases possibles d'une langue, sans pour autant admettre des énoncés incorrects.

Enfin, notons encore que la liberté de l'ordre des mots est un problème spécifique lié à certaines langues telles que le latin classique.

En résumé, le but de l'analyse syntaxique est d'extraire une représentation d'un texte afin de rendre explicites les informations considérées comme pertinentes. Ainsi, les informations phonétiques nécessaires à la reconnaissance vocale ne seront d'aucune utilité dans le cas d'une traduction automatique.

L'analyse syntaxique nécessite deux types de connaissances : d'une part celles en rapport avec la langue, et d'autre part les connaissances informatiques, ce que l'on peut illustrer par le schéma suivant³ :

² cf. chapitre 1.2.3 Les représentations du texte..

³ cf. Wehrli (1997), p. 10 ; Bouillon *et al.* (1993), p. 28.

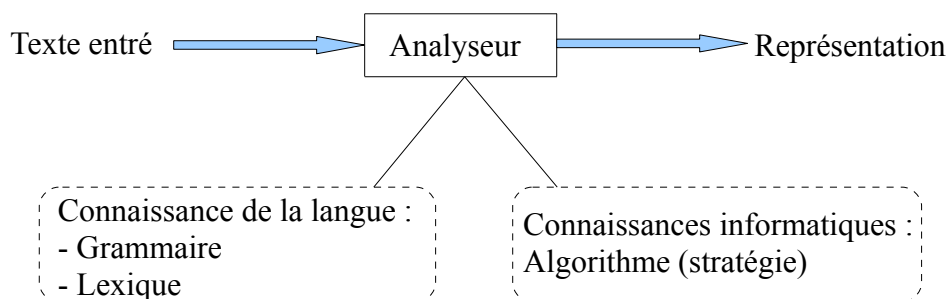


Illustration 1.1: Processus de l'analyse syntaxique

1.2.1 La composante linguistique

La description du langage se divise en différents niveaux qui correspondent aux domaines de la linguistique théorique. On peut distinguer sept de ces niveaux : la phonétique, la phonologie, la morphologie, la lexicologie, la syntaxe, la sémantique et la pragmatique. Chacun d'eux sera brièvement présenté dans les paragraphes suivants.

Phonétique / phonologie : la phonétique étudie les sons en eux-mêmes, leur formation, leur composition, tandis que la phonologie étudie la manière dont les sons sont agencés pour former des mots et des phrases.

Morphologie : la morphologie définit les règles qui régissent la construction des mots à partir de morphèmes (la plus petite unité de signification). Elle se subdivise en morphologie flexionnelle, dérivationnelle et compositionnelle. La morphologie flexionnelle a pour fonction de spécifier certains traits grammaticaux (genre, nombre, personne, cas, temps, ...) ; tandis que la morphologie dérivationnelle se rapporte à la formation de mots à partir d'un autre par ajout d'affixes. Enfin, la morphologie compositionnelle couvre la formation des noms composés.

Lexicologie : pour effectuer le traitement automatique, l'ensemble des mots d'un langage est stocké dans un lexique. Il ne s'agit pas uniquement d'une liste de mots, mais pour chaque entrée, un mot est associé à sa catégorie syntaxique, les formes fléchies sont répertoriées, de même que les constructions des verbes, etc.

Syntaxe : la syntaxe s'intéresse à la façon dont les mots se combinent entre eux pour former des phrases correctes. Les combinaisons valides peuvent être décrites par de multiples formalismes grammaticaux.

Sémantique : la sémantique se rapporte au sens des mots et de la phrase. Elle détermine les rapports de sens entre les mots (synonymes, antonymes etc.).

Pragmatique : cette branche concerne les éléments du langage dont la signification ne peut être comprise que grâce au contexte. Il est nécessaire dans certains cas de posséder des connaissances générales sur les concepts du monde, ou sur la situation de communication.

En résumé :

Sons et formation des mots -> phonétique, phonologie, morphologie.

Description de la structure -> lexicologie, syntaxe.

Signification -> sémantique, pragmatique.

Ces niveaux sont fortement liés entre eux, mais cette séparation ne correspond pas forcément à une réalité psychologique (ni à une réalité dans le traitement informatique). Cette décomposition pratique a toutefois été adoptée pour l'entier de la description linguistique.

Dans le cas de l'analyse syntaxique, seuls les points concernant la morphologie, le lexique et la syntaxe nous intéresseront. Les autres ne seront pas pris en considération lors de ce travail.

1.2.2 La composante informatique

Ce sont les algorithmes qui déterminent de quelle manière un programme informatique, appelé « analyseur », va interpréter les connaissances linguistiques.

Voici une définition de l'analyseur donnée par Bouillon et al. (1998) : l'analyseur est un « mécanisme informatique qui a la double fonction de déterminer, sur base de la description linguistique, si une phrase fait partie ou non du langage, et de lui associer une ou plusieurs représentations en fonction de l'ambiguïté de la phrase »⁴.

Alors que la description linguistique est une étape statique, l'analyseur correspond à une phase dynamique. Grâce à un ensemble d'instructions exécutables par une machine, et en se fondant sur la description d'un langage donné, le programme permet de décider si une phrase appartient bel et bien à ce langage.

Tout en s'assurant que la phrase est correcte, l'analyseur lui associe une représentation. Celle-ci sera unique si la phrase ne contient pas d'ambiguïté. Mais reprenons le cas de l'énoncé évoqué plus haut : « Jean regarde l'homme avec un télescope ».

Cette phrase recevra deux représentations : l'une où le groupe « avec un télescope » se rattachera au verbe « regarder », et l'autre où ce même groupe complètera « l'homme ».

⁴ Bouillon *et al.* (1998), p. 167.

De nombreux analyseurs ont été développés au cours des dernières décennies, et ils se distinguent les uns des autres selon plusieurs critères. Premièrement, il y a bien sûr le formalisme grammatical ainsi que l'algorithme d'analyse, qui seront discutés plus en détail dans le chapitre suivant. D'autre part, comme nous l'avons vu précédemment, le type de données linguistiques pertinentes varie selon l'application à laquelle l'analyseur est destiné. En outre, les informations associées à une analyse peuvent être plus ou moins étendues : un analyseur profond sera capable d'assigner une représentation complète de la phrase traitée avec la structure exacte des constituants, tandis qu'un analyseur superficiel se contentera d'une analyse syntaxique de surface, regroupant uniquement les constituants minimaux, tels les groupes nominaux ou les groupes prépositionnels, sans spécifier leur structure ou leur fonction.

1.2.3 Les représentations du texte

L'un des objectifs du traitement de la langue est de produire une représentation d'un texte, et cette tâche est dévolue à l'analyseur syntaxique. Il s'agit en effet pour le programme de rendre explicites les informations contenues dans le texte. Mais il convient maintenant de définir la forme que doit prendre cette représentation.

À chaque niveau linguistique correspond une certaine représentation. Nous avons déjà déterminé quels niveaux étaient pertinents dans le cadre de ce travail, c'est pourquoi nous allons nous pencher en particulier sur la morphologie et la syntaxe.

La représentation morphologique doit rendre compte des informations concernant les mots, leur forme de base (lemme), leur catégorie lexicale, genre, nombre etc. Il est donc nécessaire de considérer tout ce qui a trait aux morphèmes du mot, ces unités de sens qui le composent.

Si l'on prend par exemple le mot *civitatibus*, sa représentation morphologique devra donc contenir les indications suivantes :

- | | |
|--------------|-------------------|
| a. lemme | = civitas; |
| b. catégorie | = Nom (N); |
| c. genre | = féminin; |
| d. nombre | = pluriel; |
| e. cas | = datif, ablatif; |
| f. personne | = troisième. |

La représentation syntaxique concerne la structure hiérarchique des phrases en constituants (ou syntagmes). Les constituants identifiés peuvent s'imbriquer les uns dans les autres, et les arbres permettent de visualiser de façon intuitive cette structure hiérarchique. Exemple :

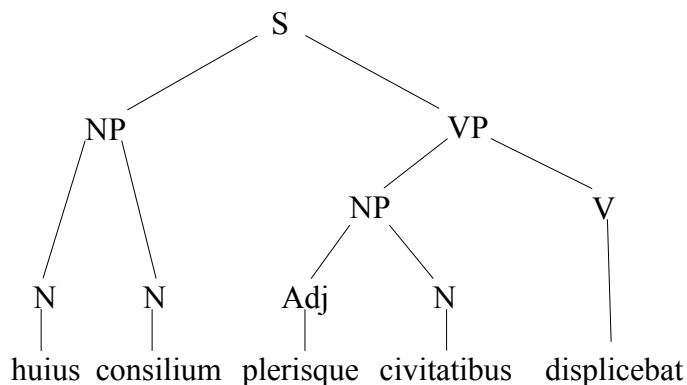


Illustration 1.2: Structure des constituants

Les parenthèses étiquetées sont un autre mode de représentation syntaxique. Chaque syntagme est encadré par une parenthèse avec, en indice, la catégorie à laquelle il appartient :

[S [NP huius consilium] [VP [NP plerisque civitatibus] displicebat]].

Afin de faciliter la compréhension de la suite du travail, voici la liste des abréviations utilisées pour les syntagmes et les catégories lexicales⁵ :

S	phrase	(sentence)
NP	syntagme nominal	(noun phrase)
VP	syntagme verbal	(verb phrase)
AP	syntagme adjectival	(adjective phrase)
AdvP	syntagme adverbial	(adverb phrase)
PP	syntagme prépositionnel	(prepositional phrase)

Det	déterminant
N	nom ⁶
V	verbe
Adj	adjectif
Adv	adverbe
Prep	préposition
Conj	conjonction

⁵ cf. Wehrli (1997), p. 36.

⁶ Le nom regroupe les noms communs, les noms propres ainsi que les pronoms.

2 Aspects techniques

Les aspects techniques concernent les formalismes grammaticaux ainsi que les algorithmes d'analyse. Pour replacer leur développement dans un contexte historique, il faut remonter aux années 50 : pendant la période de la Guerre froide qui suit la Seconde Guerre mondiale, la possibilité de traduire automatiquement les communications de l'ennemi constitue un important enjeu stratégique. Mais lorsque le monde scientifique entreprend de relever le défi de la traduction automatique, ce projet ambitieux se solde par un échec.

En effet, l'informatique n'en est encore qu'à ses débuts et ses moyens sont limités : la faible capacité en mémoire rend le stockage des données problématique, et la puissance de calcul est alors très peu élevée en comparaison des machines actuelles.

En outre, la linguistique n'est pas assez développée pour gérer les problèmes soulevés par la traduction automatique. Le courant structuraliste, qui tire ses origines de l'oeuvre de Saussure, ne s'occupe pas de la représentation du sens. Il en résulte qu'à défaut d'une réelle traduction, les programmes produisent principalement une transposition mot-à-mot du texte.

En 1966, le comité ALPAC⁷ chargé d'évaluer les avancées et la rentabilité de la traduction automatique publie un rapport particulièrement négatif qui va signer la fin du financement dans ce domaine. Les chercheurs doivent donc se tourner vers d'autres champs d'investigation, tels que la linguistique informatique, qui étudie le TALN, les grammaires formelles étudiées par Chomsky, ou encore l'intelligence artificielle.

Le but de ce chapitre n'est pas de donner une description détaillée des différents formalismes et algorithmes d'analyse, car ceux-ci sont abondamment décrits dans la littérature scientifique. Il s'agit plutôt de présenter un aperçu des concepts généraux, qui permettra au lecteur de connaître les principes de base et de mieux comprendre la suite du projet. Quelques pistes bibliographiques seront proposées pour ceux qui désirent approfondir le sujet.

2.1 Les grammaires formelles⁸

Noam Chomsky, linguiste et philosophe américain, développe durant la fin des années 50 la théorie des grammaires formelles qui va bouleverser l'approche de la linguistique. Dès 1957, il introduit la notion de grammaire générative transformationnelle. Il s'agit d'une description systématique de la génération du discours dans une langue, ou comment générer toutes les phrases possibles et uniquement celles-ci. La théorie de Chomsky postule qu'il existe deux niveaux de représentation syntaxique : un premier niveau abstrait (structure profonde) est lié par des règles de transformation

⁷*Automatic Language Processing Advisory Committee*. Il est mis en place par l'Académie des Sciences des Etats-Unis.

⁸Pour une première approche, cf. Wehrli (1997) chap. 2 et Bouillon *et al.* (1998) chap. 3.3.

à un second niveau correspondant à la phrase elle-même (structure de surface)⁹.

Cette théorie implique un grand progrès, car désormais le traitement de la langue est purement syntaxique et peut s'effectuer indépendamment du sens. En effet, la syntaxe et la signification sont autonomes puisqu'il est possible de créer des phrases grammaticalement correctes mais dépourvues de sens, et vice-versa. Chomsky s'est servi de la phrase suivante pour rendre cette idée plus explicite : « D'incolores idées vertes dorment furieusement ». Cet énoncé pourtant grammatical n'a strictement aucun sens. Au contraire, « Je parle très mieux français que toi, et je te merde ! » est parfaitement compréhensible, malgré une syntaxe boiteuse. C'est donc l'intuition des locuteurs natifs qui détermine quelles phrases sont acceptables dans le langage.

Afin de mieux comprendre le principe de grammaire formelle, prenons la définition de la grammaire telle que nous la connaissons, à savoir « l'ensemble des règles à suivre pour parler et écrire correctement une langue »¹⁰.

De même, les grammaires formelles permettent de construire les phrases correctes d'un langage grâce à un ensemble de règles. Appliquées de façon mécanique, ces règles transforment une chaîne d'entrée en une chaîne de sortie, c'est-à-dire une phrase correctement constituée par un procédé de transformations (ou réécritures) successives. Pour être plus précis, une grammaire formelle est composée de quatre parties :

- Un symbole initial S, qui constitue l'entrée à partir de laquelle toute phrase sera dérivée.
- Un ensemble de symboles non-terminaux, dont S fait partie, qui représentent les différents constituants de la phrase. $VN = \{NP, VP, AP, AdvP, PP\}$.
- Un ensemble de symboles terminaux qui représentent les catégories syntaxiques des mots. $VT = \{Det, N, V, Adj, Adv, Prep, Conj\}$.
- Un ensemble P de règles de production, aussi appelées règles de réécriture. Chaque règle comprend deux parties : une partie gauche avec un symbole non-terminal et une partie droite qui sert à réécrire la partie gauche. Par exemple, la règle :

$NP \rightarrow Det N$

signifie qu'un syntagme nominal (NP) se réécrit comme un déterminant (Det) suivi d'un nom (N). Les règles de réécritures sont appliquées à partir du symbole initial jusqu'à ce que la chaîne de sortie ne contienne plus que des symboles terminaux.

Une grammaire G est donc ainsi définie : $G = \{S, VN, VT, P\}$.

Voici un exemple de règles de réécriture pour une grammaire très simplifiée :

⁹ Exemple: le passif. « le chat mange la souris » possède la même structure profonde, ou presque, que « la souris est mangée par le chat » (même acteur de l'action) ; mais ces phrases diffèrent dans leur structure de surface (sujet syntaxique différent).

¹⁰ Le Grand Robert de la langue française, version électronique, 2e édition dirigée par A. Rey.

- a. S -> NP VP
- b. NP -> Det N
- c. NP -> N
- d. VP -> V NP
- e. VP -> V

En appliquant ces règles, on peut obtenir par dérivation les étapes suivantes :

- 1. S => a NP VP
- 2. NP VP => b Det N VP
- 3. Det N VP => c Det N V NP
- 4. Det N V NP => b Det N V Det N.

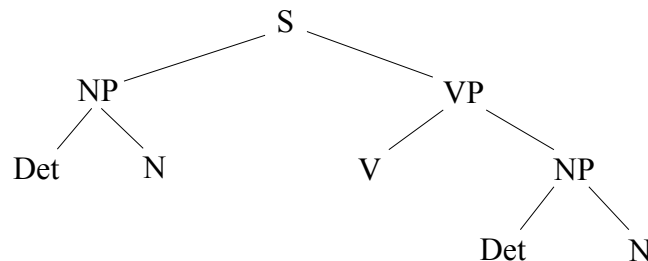


Illustration 2.1: Représentation sous forme d'arbre

Les grammaires utilisées pour décrire les langues naturelles sont dites indépendantes de contexte, ou « de type 2 » selon la hiérarchie établie par Chomsky. En effet, les règles de dérivation portent sur un symbole non-terminal, sans prendre en compte ce qui se trouve juste avant ou juste après. Le contexte n'a donc pas d'influence.

Il est aisé de se rendre compte intuitivement que notre modeste grammaire se limite à produire des phrases du type « le chat mange la souris ». Mais elle génère également des phrases comme « les chat mangeons un souris » etc. Pour éviter de tels énoncés agrammaticaux et traiter les phénomènes comme l'accord ou les dépendances non bornées¹¹, il devient nécessaire de multiplier considérablement les règles ainsi que les symboles qui les composent. Malgré tout, ces grammaires ne permettent pas de décrire de façon pleinement satisfaisante les langues naturelles. C'est pourquoi

¹¹ On utilise le terme de dépendance non bornée pour qualifier deux éléments d'une phrase éloignés l'un de l'autre, mais formant cependant un tout du point de vue du sens. C'est le cas dans les interrogatives :
« Quel livre Jean a-t-il lu ? »
Ici le syntagme interrogatif *quel livre* est la suite du verbe lire.

de nombreux autres formalismes seront élaborés afin de pallier à ces défauts, dont en particulier les grammaires d'unification.

La notion d'unification est commune à un ensemble de formalismes, développés en réaction aux théories de Chomsky¹². Les grammaires d'unification partagent deux points communs principaux :

- Les informations linguistiques concernant les mots, syntagmes, phrases etc. sont décrites par des structures de traits qui énumèrent leurs caractéristiques.
- L'unification des structures des traits est une opération fondamentale sur les structures permettant de comparer et de combiner les informations contenues dans ces dernières.

Les structures de traits se présentent comme une matrice contenant, par exemple pour le nom *consilium*, les traits de genre, nombre et cas :

[neutre]
[singulier]
[nominatif]

Dans le cas présent, les traits prennent une simple valeur unique, mais il est également possible d'avoir des valeurs complexes, c'est-à-dire des traits dont la valeur est à son tour une structure de trait. La structure ci-dessus pourrait ainsi devenir la valeur du trait « accord » :

[[genre : neutre]]
[accord :	[nombre :singulier]]
[[cas : nominatif]]

L'unification est la manipulation principale appliquée aux structures de traits. Cette opération consiste à combiner, sur le mode de l'union dans la théorie des ensembles, deux structures de traits afin d'obtenir une nouvelle structure compatible avec les deux premières, tout en conservant leurs informations respectives.

Soit deux nouvelles structures :

[[genre : neutre]]
[accord :	[nombre :singulier]]
[accord :	[nombre :singulier]]
[[cas : nominatif]]

¹² Cf. Abeillé (1993), ten Hacken (2007).

L'unification de ces structures correspondrait à la structure pour le trait « accord ». Il faut toutefois préciser que cela ne serait pas possible si l'une de ces deux structures avait la valeur pluriel pour le nombre, car cette contradiction empêcherait l'unification.

Grâce à l'unification, la grammaire construit pas à pas une représentation globale de la phrase. Même si en théorie les grammaires d'unification permettent de décrire avec précision les structures du langage naturel, la mise en place d'un système destiné à une application à grande échelle se heurte à certaines difficultés, telles que la multiplicité des résultats ou la fragilité du système. En outre, il existe une certaine redondance entre les informations contenues dans le lexique et les règles de réécriture : le verbe « manger » peut se construire de façon transitive ou intransitive¹³, et cela doit être indiqué dans le lexique. Mais ces deux constructions du verbe « manger » correspondent aux règles :

- d. VP -> V NP
- e. VP -> V

Un autre problème se pose avec les grammaires de règles : elles décrivent des structures de surface (au niveau de l'observation). Ces structures de surfaces peuvent être très différentes d'une langue à l'autre. L'exemple suivant est tiré de Serbat (p.137). Une traduction très littérale fait ressortir les contrastes entre le latin et le français, notamment la place du verbe en fin de proposition, les disjonctions entre le nom et l'adjectif épithète, entre le relatif et l'antécédent, etc.

Latin :	Mot à mot :	Traduction :
<i>Singulari militum nostrorum virtuti consilia cuiusque modi Gallorum occurebant,</i>	A l'exceptionnelle des soldats nôtres valeur des mesures de tout ordre des Gaulois s'opposaient,	A l'exceptionnelle valeur de nos soldats s'opposaient des mesures de tout ordre des Gaulois,
<i>ut est summae genus sollertiae atque ad omnia imitanda et efficienda quae ab quoque traduntur aptissimum.</i>	étant donné que c'est d'une extrême race ingéniosité et à toutes choses imiter et réaliser qui par l'un ou l'autre sont transmises éminemment apte.	étant donné que c'est une race d'une extrême ingéniosité et éminemment apte à imiter et réaliser toutes choses transmises par l'un ou l'autre.

Un modèle adapté au français sera donc très difficile à transposer pour la langue latine.

Pour toutes ces raisons, une approche différente a été développée par Chomsky¹⁴ : la théorie du

¹³ Exemples : Paul mange une pomme. Paul mange.

¹⁴ La position de Chomsky a régulièrement évolué : *Syntactic Structures* (1957) et *Aspects of the Theory of Syntax* (1965) présentent la *grammaire générative et transformationnelle*, la théorie du *gouvernement et liage* est exposée dans *Lectures on Government and Binding* (1981), suivi de *The Minimalist Program* (1995). Sag et Wasow (1999) donnent un aperçu historique de la grammaire générative, pp.411-427. Smith (1999) analyse les travaux de

« Gouvernement et Liage », aussi connue sous le nom de « Principes et Paramètres ». Plutôt que de chercher à décrire la langue, il adopte une démarche explicative :

Chomsky émet l'hypothèse qu'il existe une grammaire universelle régie par des principes établis pour toutes les langues, tandis que les paramètres sont les options spécifiques servant à expliquer les différences syntaxiques qui existent entre chaque langue. Les règles de réécriture sont remplacées par la théorie X-barre¹⁵.

2.2 Les algorithmes d'analyse¹⁶

Un algorithme est un processus de résolution d'un problème, comprenant un ensemble fini d'instructions qui, appliquées mécaniquement, déterminent comment atteindre un résultat précis. Étant donné un langage défini par une grammaire, un algorithme d'analyse correspond à la procédure à suivre pour savoir quelles phrases appartiennent au langage et lesquelles ne sont pas en accord avec la grammaire.

Ces procédures se répartissent principalement en deux groupes : les stratégies descendantes et les stratégies ascendantes. Sans entrer dans les détails de l'algorithme, il faut savoir que les procédures descendantes cherchent à construire la représentation du texte depuis la racine (S) de l'arbre jusqu'aux éléments terminaux. Si la séquence de catégories syntaxiques obtenue concorde avec les mots de la phrase, alors l'analyse est réussie. Reprenons la grammaire présentée au chapitre précédent :

- a. S -> NP VP
- b. NP -> Det N
- c. NP -> N
- d. VP -> V NP
- e. VP -> V

La dérivation de la phrase « Sophie lit un livre » commencerait ainsi :

S => a NP VP
NP VP => b Det N VP

...

Mais elle s'arrêterait à ce moment-là, car la séquence « déterminant + nom » n'est pas équivalente à notre phrase. Lorsque l'analyse conduit à une telle impasse, l'algorithme est conçu pour remonter au dernier point où il a rencontré une ambiguïté, c'est-à-dire un choix entre plusieurs alternatives pour réécrire un symbole (dans notre cas, le choix se situe entre les règles b et c pour réécrire le

Chomsky et les controverses qu'ils ont suscitées (voir en particulier le chapitre 2 sur l'aspect linguistique), tandis que Notari (2009) propose une approche critique de la théorie de Chomsky.

¹⁵ cf. chapitre 3.3.2 Analyse.

¹⁶ cf. Wehrli (1997) chap. 3 et Bouillon et al (1998) chap. 4.

syntagme nominal). Si toutes les possibilités ont été épuisées sans succès, l'analyse échoue, car la phrase ne fait pas partie du langage.

Le second groupe d'algorithmes, basés sur la stratégie ascendante, fonctionne de manière inverse : les mots de la phrase constituent le point de départ de l'analyse. Ensuite, l'arbre syntaxique est établi petit à petit, en regroupant les constituants jusqu'à atteindre la racine S. Exemple :

Sophie lit un livre => N V Det N => N V NP => N VP => NP VP => S.

Dans cette analyse, nous n'avons pas tenu compte des alternatives, mais ce problème reste valable également pour les stratégies ascendantes. Il existe deux approches pour traiter le problème des alternatives. La première est séquentielle : les alternatives sont traitées une à la fois, « à la suite », et l'analyseur effectue un retour en arrière (*backtracking*) lorsqu'une alternative a échoué. D'autre part, l'approche peut être parallèle. Toutes les alternatives sont en effet développées en même temps, et celles qui sont incompatibles avec les données sont éliminées au fur et à mesure de l'analyse. À la fin de l'analyse, on obtient la liste de toutes les représentations possibles.

Cependant les deux stratégies, ascendante et descendante, ne sont pas sans inconvénient. Il faut bien réaliser que la complexité d'une grammaire pour la langue naturelle dépasse de beaucoup celle de la grammaire utilisée pour illustrer les exemples précédents. Les alternatives à traiter sont beaucoup plus nombreuses, et obligent l'analyseur à recommencer plusieurs fois l'analyse d'un même constituant¹⁷. Ce qui portait peu à conséquence avec notre petite grammaire peut rapidement devenir un obstacle important lors d'une application concrète au langage naturel. En effet, l'analyse nécessite alors plus de temps¹⁸ et d'espace pour le calcul.

Bien que ces procédures soient relativement simples à implémenter, elles ne sont toutefois pas suffisamment efficaces. C'est pourquoi les stratégies mixtes, qui réunissent à la fois l'aspect ascendant et descendant, sont préférables. Les analyseurs mixtes ont cet avantage de garder en mémoire dans une table les constituants bien formés, de telle sorte qu'ils ne doivent pas recommencer plus d'une fois la même analyse lorsqu'ils sont dans une impasse et reviennent en arrière. Ces analyseurs sont de type tabulaire¹⁹. D'autres algorithmes mixtes se servent de filtres afin d'éviter les règles incompatibles avec la phrase donnée²⁰.

¹⁷ Les procédures ascendantes n'ont pas de vision globale de la phrase (elles construisent des constituants incompatibles avec le contexte), tandis que les procédures descendantes ne prennent pas en compte les données.

¹⁸ De plus, le temps de traitement croît de façon exponentielle par rapport à la longueur de la phrase.

¹⁹ On peut citer dans cette catégorie l'algorithme d'Earley, celui de Cocke-Kasami-Younger ou le *chart-parser*.

²⁰ Exemple : l'algorithme du coin gauche, cf. Wehrli (1997), chap. 3.8.4.

Pour un résumé général de la situation concernant aussi bien les formalismes grammaticaux que les algorithmes, cf. Villemonte de la Clergerie & Rajman (2003).

2.3 Analyse syntaxique du latin - Etat de la question

Peu d'analyseurs syntaxiques pour le latin sont actuellement disponibles sur internet. La majorité des sites et programmes ne proposent pas une analyse syntaxique complète, bien qu'ils soient souvent qualifiés de *parser* (terme anglais pour l'analyse syntaxique). Ils s'arrêtent en effet à la description morphologique d'un ou plusieurs mots, parfois accompagnée d'une traduction.

En voici une liste non exhaustive :

- Collatinus, par Yves Ouvrard (<http://www.collatinus.org/collatinus>) ;
- Words, par William Whitacker (<http://www.archives.nd.edu/cgi-bin/words.exe>) ;
- Latin parser and translator, par Adam McLean (<http://www.alchemywebsite.com/latin/latintrans.html>) ;
- Latin parser, par Louis Ha (<http://arts.cuhk.edu.hk/~lha/latin/>).

Les deux premiers (Collatinus et Words) ont un lexique très riche et prennent en compte les particules telles que *-que*, *-ve*, sans forcément les reconnaître : l'analyse de *pellibusque sunt vestiti* par Collatinus montre que *pellibus* est extrait de la forme *pellibusque*, mais le *-que* n'est pas défini. Leurs analyses sont généralement correctes.

Les deux suivants sont nettement moins fiables dans leurs analyses et ne reconnaissent pas les particules. D'autre part, leur vocabulaire est assez limité. La particularité du programme d'A. McLean est qu'il prend une phrase entière et en donne la traduction de chaque mot dans le même ordre, ce qui produit une pseudo-traduction mot-à-mot (en anglais).

Le programme « Andromeda » se rapproche déjà un peu plus d'un analyseur syntaxique (<http://www.collatinus.org/andromeda/>). Il appartient à une série de programmes à but pédagogique créés par Yves Ouvrard, professeur de Lettres classiques à Angoulême.

La stratégie ne repose pas sur un formalisme grammatical, mais s'appuie sur le lexique de Collatinus. Sur la base des cas, le programme recherche :

- le verbe auquel est subordonné un accusatif, un ablatif ou un datif ;
- la préposition qui précède un accusatif ou un datif ;
- le nom avec lequel s'accorde un adjectif ou un participe.

Les mots subordonnants (*cum*, *ubi*, *ut*, etc.) sont également traités.

Les résultats peuvent être relativement précis. Voici l'analyse de la phrase *Cicero pater patriae vocatus est* :

est(3ème personne singulier indicatif présent actif) :
Cicero(nominatif masculin singulier):attribut du sujet ou sujet du V.
pater(nominatif masculin singulier) :attribut du sujet ou sujet du V.
patriae(génitif féminin singulier) : CDN au génitif
vocatus(nominatif masculin singulier participe parfait passif) :
participe, forme composée.

Cependant, comme le précise son auteur, Andromeda est encore « expérimentale et instable ». Quelques points en particulier restent à améliorer :

- Il n'y a pas de système de désambiguïsation, et le nombre d'analyses est de ce fait très élevé (pour la phrase précédente, l'analyse correcte se trouve parmi vingt autres propositions).
- La durée de l'analyse croît de façon exponentielle, en fonction du nombre de variantes morphologiques et des différentes combinaisons possibles pour lier les constituants entre eux. Dans certains cas, l'analyse se retrouve bloquée dans une boucle infinie, comme par exemple avec cette phrase : *ille cognita re caput Gorgonis eis ostendit omnesque ab humana specie sunt informati in saxum.*
- L'analyse des subordonnées relatives n'est pas possible (*video rosam quae pulchra est*) ;
- Andromeda accepte des constructions inusitées, ainsi que des incohérences : *Cicero pater patriam vocatus est* produit une analyse « réussie », bien que le cas de *patriam* soit bien entendu incorrect.

Il existe malgré tout au moins un véritable analyseur syntaxique pour le latin. Il s'agit de l'analyseur « grammatica latina », présenté en détail dans les paragraphes suivants.

2.3.1 Grammatica Latina

Site : <http://www.agfl.cs.ru.nl/lat/index.html>

Depuis 1991, un formalisme grammatical adapté à la description syntaxique des langages naturels est développé par le *research group on Compiler Construction* à l'Université Radboud de Nimègue aux Pays-Bas : l'AGFL, « Affix Grammars over a Finite Lattice ». Le but du projet AGFL est de développer des technologies et ressources pour le traitement du langage naturel qui soient disponibles dans le domaine public. Des analyseurs existent pour plusieurs langues différentes, en particulier l'anglais, le hollandais et le latin.

L'analyseur Latina pour le latin classique est disponible sur le site de l'AGFL Grammar Work Lab. L'interface permet d'entrer un énoncé ou de choisir au hasard une phrase de Saint Augustin. Le programme affiche ensuite en quelques secondes l'analyse qu'il considère comme la plus probable, sous la forme d'une « transduction ».

Exemple de transduction pour la phrase *huius consilium plerisque civitatibus displicebat* :

```
{{consilium <MODgen huius] <SUBJ[displicebat <MODdat [ciuitatibus <ATTR plerisque]]}}
```

Huius est le complément au génitif de *consilium*. Ce groupe forme le sujet du groupe verbal, où *plerisque* est attribut de *civitatibus* et les deux mots sont le complément au datif de *displicebat*.

Le système de l'AGFL permet la création de grammaires pour le langage naturel. Ces grammaires appartiennent à la famille des grammaires à deux niveaux²¹ : il s'agit d'une grammaire indépendante de contexte complétée par un second niveau de règles (*meta-rules*) qui définissent des affixes dont les valeurs sont choisies parmi un ensemble fini, afin d'exprimer l'accord entre les constituants. De ce fait, la grammaire est également apparentée aux grammaires attribuées.

Exemple d'affixes pour définir les traits de nombre, genre, personne et de cas :

```
NUM:: sg | pl.  
GENUS:: fem | masc | ntr.  
PERS:: prm | sec | trt.  
CASUS:: nom | voc | gen | dat | acc | abl | loc.
```

Les règles de syntaxes sont des procédures avec des paramètres, dans une notation proche de PROLOG. En cela la grammaire appartient aux grammaires de clauses définies.

Les règles décrivant la structure des constituants se présente ainsi pour les phrases contenant au plus sujet, objet, verbe et compléments :

```
[subject(PERS,NUM)] & v(MODUS,PERS,NUM,act) & [object] & [adject]
```

Le symbole & signale que l'ordre de ces constituants est libre.

Cette grammaire d'affixes n'est en soi pas plus puissante qu'un grammaire indépendante de contexte. Mais l'introduction des affixes, en la rendant plus concise, permet de traiter des grammaires plus complexes.

L'analyseur²² est généré automatiquement à partir de la grammaire du latin et d'un lexique regroupant plus d'un million d'entrées. Cela couvre environ 80% du vocabulaire de Saint Augustin et près de 75% de celui de César. Il n'est cependant pas forcément possible de juger avec exactitude l'efficacité du programme, car il ne renvoie qu'un seul résultat et ignore peut-être d'autres analyses qui pourraient s'avérer plus pertinentes.

²¹ Les grammaires à deux niveaux ont été introduites par A. Van Wijngaarden, avec le langage de programmation ALGOL 68 : le premier niveau, qui produit la langue cible, est une grammaire dont les règles de production sont engendrées par une autre grammaire au second niveau. Ainsi, il est possible d'avoir un nombre infini de règles qui dérivent le langage, et cela permet d'analyser certaines structures dépendantes de contexte. Les dépendances croisées, que l'on trouve notamment en néerlandais, en sont un exemple.
cf. Van Wijngaarden (1968).

²² L'algorithme est un *chart parser*. Sur la construction du lexique et de la grammaire latine, cf. Koster (2005).

On constate que l'ordre des mots est relativement libre, quoique certaines combinaisons rendent plus difficile la distinction entre adjectif attribut et épithète. Comparer : *bonus miles fortis est* et *fortis est bonus miles*. Cela peut poser problème également avec les formes composées du passif. Comparer : *Roma est a Romulo condita* et *Roma a Romulo condita est*.

Les phrases simples avec un sujet, un verbe et un complément sont correctement interprétées (*discipuli legunt, pater amat filium suum, filius amabatur a patre, vires mihi desunt*, etc.). Dès que la phrase contient une subordonnée, les analyses sont plus rarement justes (exemples : *scio vitam esse brevem ; video rosam quae pulchra est ; Galli timebant ne Caesar in Gallia veniat ; cum Caesar pontem faceret, Galli adgressi sunt*).

D'autres problèmes apparaissent avec la coordination : la phrase *ego sum nauta et tu es agricola* est bien interprétée, contrairement à *pater et mater sunt boni*. Quant aux particules, elles ne sont pas reconnues par le programme.

D'une manière générale, les phrases courtes restent plus faciles à analyser. Mais l'analyseur Latina n'a pas encore atteint sa version définitive, et il n'est pas exclu qu'une version future vienne combler les lacunes encore présentes dans le programme actuel²³.

2.3.1.1 Liste des exemples :

- 1) {[miles <ATTR fortis] <ATTR bonus] <SUBJ[est]}
- 2) {[miles <ATTR bonus] <SUBJ[est <PRED fortis]}
- 3) {[Roma <ATTR [condita <MODa Romulo]] <SUBJ[est]}
- 4) { Roma <SUBJ[est <PRED condita <MODa Romulo]}
- 5) {discipuli <SUBJ[legunt]}
- 6) {pater <SUBJ[amat <OBJ [filium <ATTR suum]]}
- 7) {filius <SUBJ[amabatur <MODa patre]}
- 8) {vires <SUBJ[desunt <MODdat mihi]}
- 9) {[EGO <SUBJ[scio <OBJ[uitam <SUBJ[[esse <OBJ breuem]]]]]}
- 10) {[EGO <SUBJ[video <OBJ rosam]]
{[pulchra <ATTR quae] <SUBJ[est]}
- 11) { Galli <SUBJ[timebant]} {Caesar <SUBJ[ueniat <MODin Gallia]}
- 12) Caesar <SUBJ[faceret <OBJ pontem]}

²³ Lors de ce travail, la version 0.5 a été utilisée.

{[ILLI <SUBJ[[sunt <MODvoc , [adgressi <MODgen Galli]]]]}

13) {ego <SUBJ[sum <PRED nauta]} {tu <SUBJ[es <PRED agricola]}

14) {[pater]} { boni <SUBJ[[sunt <MODvoc mater]]]}

3 L'analyseur syntaxique Fips

La création d'un analyseur pour le latin classique ne va pas s'effectuer *ex nihilo*, mais il sera en réalité élaboré comme une version spécifique de l'analyseur Fips, développé par le Laboratoire d'Analyse et de Technologie du Langage (LATL) à l'Université de Genève.

3.1 Présentation générale

Lors de sa présentation en 1991, la première version de Fips²⁴ consistait en un système d'analyse syntaxique du français basé sur une grammaire chomskyenne dite de « Gouvernement et Liage ». Cependant en une vingtaine d'années, Fips a évolué et a connu de nombreux changements, notamment en ce qui concerne les principes linguistiques sur lesquels Fips est basé. La version actuelle de Fips²⁵ consiste en une adaptation du modèle minimaliste chomskyen, enrichi d'emprunts à d'autres modèles tels que LFG²⁶ ou Simple Syntax²⁷.

Avec le développement croissant d'internet et la globalisation des échanges, un système qui puisse traiter plusieurs langues à la fois est un enjeu important. Fips est ainsi devenu une plateforme multilingue. Des versions de Fips relativement complètes existent pour 5 langues : français, anglais, allemand, italien, espagnol. D'autres langues sont traitées, mais à un stade de développement moins avancé (grec moderne, roumain, russe, japonais, romanche).

Fips est un analyseur profond et robuste. Il doit être capable de faire preuve de souplesse lorsqu'il rencontre une construction qui n'est pas entièrement conforme à la grammaire, et d'accepter des phrases parfois incomplètes, car de grandes variations sont observées dans la langue écrite.

Fips est un analyseur universel : il ne se limite pas à une application en particulier, mais il est conçu pour toute application nécessitant une analyse linguistique (par exemple : la traduction automatique, l'analyse et synthèse de la parole, l'extraction terminologique, ...).

L'efficacité de ces applications dépend en grande partie de la qualité de l'analyse syntaxique ; or, l'analyse morphologique est un préalable nécessaire à l'analyse syntaxique, ainsi que l'analyse lexicale (séparation de la phrase en mots). On ne peut effectuer ni l'une ni l'autre de façon automatique sans un lexique. En effet, le lexique est l'un des éléments essentiels de la composante linguistique avec la grammaire²⁸. Le lexique est donc une base de données où sont stockées les

²⁴ FIPS est un acronyme pour *French Interactive Parsing System*.

²⁵ cf. Wehrli & Nerima (2009), p. 1.

²⁶ *Lexical Functional Grammar*, une grammaire d'unification (cf. Bouillon p. 100).

²⁷ Culicover & Jackendoff, 2005.

²⁸ cf. illustration 1.1, chapitre 1.2.

informations relatives aux mots telles que leur orthographe, leurs différentes formes flexionnelles, leur catégorie syntaxique, etc., sans lesquelles l'analyse syntaxique serait impossible.

3.2 Le lexique de Fips

3.2.1 La place de la morphologie

Afin de créer le lexique, il est nécessaire de décider en premier lieu de quelle façon intégrer la génération et l'analyse morphologique dans le processus de traitement de la langue. Le lexique doit en effet contenir les relations morphologiques qui lient les mots entre eux, et non uniquement une liste de mots. Il serait en effet peu utile que le lexique considère *amo* et *amare* comme deux mots différents sans aucune relation.

Il existe deux approches différentes à cette problématique. D'une part, l'approche dynamique permet d'analyser et dériver les formes fléchies à partir de la forme de base (ou lexème) à chaque emploi du mot, lorsqu'il est lu dans une phrase ou bien ajouté au lexique. Celui-ci contient tous les morphèmes ainsi que les règles qui déterminent comment les combiner entre eux.

Au contraire, le modèle relationnel comprend un lexique avec toutes les formes fléchies pour chaque mot. Les liens entre mots et lexèmes sont alors précisés sous la forme d'un ensemble statique de relations.

L'approche adoptée par Fips est un modèle relationnel. Pour être plus précis, les ressources lexicales de Fips sont constituées de plusieurs lexiques étroitement liés et représentant chacun un concept : les mots et leurs variantes orthographiques, les lexèmes et les collocations.

Le fait d'avoir toutes les formes fléchies dans le lexique facilite considérablement la recherche. C'est donc une méthode efficace et flexible, qui permet d'associer des informations soit au lexème soit aux formes fléchies. Cela peut se révéler particulièrement utile en phonétique pour indiquer la prononciation de chaque forme séparément, ou bien pour préciser des fréquences.

Concrètement, le lexique se présente à l'utilisateur sous la forme donnée par l'illustration 3.1, qui montre l'entrée du mot *civitatibus*.

Illustration 3.1 : Une entrée du lexique

Le mot *civitatibus* est accompagné de nombreuses informations réparties dans quatre zones différentes (*Paradigm*, *Word*, *Lexeme*, *Arguments*). En premier lieu viennent les informations concernant l'entier du paradigme : la catégorie syntaxique, la forme de base (*civitas*) et sa déclinaison. Les trois zones restantes se divisent en deux parties, selon qu'elles s'appliquent à une forme seulement ou bien à toutes.

À gauche se trouve la zone *Word* qui modifie en particulier la forme *civitatibus* : cas, genre, nombre, personne. C'est ici aussi que les temps, mode et voix sont précisés pour les verbes. Le lexique nous permet de donner la phonétique et de spécifier une variante orthographique. Ce dernier point sera utile pour les prépositions *ab* et *ex*, qui s'écrivent *a* et *e* devant consonne.

Du côté droit, les paramètres s'appliquent au lexème. Il s'agit, dans la partie supérieure *Lexeme*, de diverses caractéristiques ainsi que du type : pour les noms, les possibilités sont « nom commun », « nom propre » ou « pronom ». Les différentes options varient en fonction de la catégorie syntaxique.

Plus bas, la dernière zone *Arguments* sert à définir la structure argumentale, autrement dit la construction grammaticale. C'est dans cette zone que l'on indique si un verbe est intransitif, transitif, s'il peut être suivi d'une préposition, d'une proposition infinitive, etc. Le cas des compléments est aussi donné, et cela n'est pas uniquement valable pour les verbes. Les noms et adjectifs peuvent également avoir une construction particulière (« digne de » se traduit en latin par *dignus* + ablatif). Lorsqu'il existe plusieurs constructions pour un mot²⁹, chacune correspond à un lexème différent.

3.2.2 FipsMorpho

L'insertion de mots dans le lexique afin de l'enrichir est une tâche fastidieuse, mais si elle est accomplie soigneusement, elle permet d'atteindre un plus haut niveau d'efficacité. Lorsqu'un mot est entré dans le lexique de Fips, les règles morphologiques sont immédiatement appliquées afin de créer toutes les formes orthographiques liées à ce mot, et les liens entre les formes orthographiques et le lexème sont établis. Pour ce faire, un moteur morphologique baptisé FipsMorpho (ou FipsInflection) a été développé.

Le principe de FipsMorpho est simple : à partir d'une forme de base³⁰, l'ensemble du paradigme est généré en une fois, et toutes les formes obtenues sont directement insérées dans le lexique. Cela est plus avantageux que d'entrer les formes une à une, surtout en latin qui est une langue à la morphologie flexionnelle riche. Dans le cadre du latin, FipsMorpho concerne quatre catégories syntaxiques (les verbes, les noms, les adjectifs et les déterminants). Les autres classes de catégories syntaxiques ne présentent pas de flexion assez étendue ou sont trop irrégulières pour que cela soit intéressant de mécaniser le processus d'entrée dans le lexique.

²⁹ Par exemple le verbe *consulere* :
consulere (intr.) : « délibérer » ;
consulere + dat. : « veiller à / sur » ;
consulere + acc. : « consulter ».

³⁰ Dans certains cas, il peut y avoir plus d'une forme de base : les verbes sont entrés avec l'infinitif, le parfait et le participe passé pour obtenir les trois radicaux différents (ex : *agere, egi, actum*). Les déponents en ont seulement deux (ex : *mirari, miratus*). Les noms de la troisième déclinaison possèdent deux bases (ex : *homo, hominis*), de même que les adjectifs de deuxième classe (ex : *vetus, veteris*).

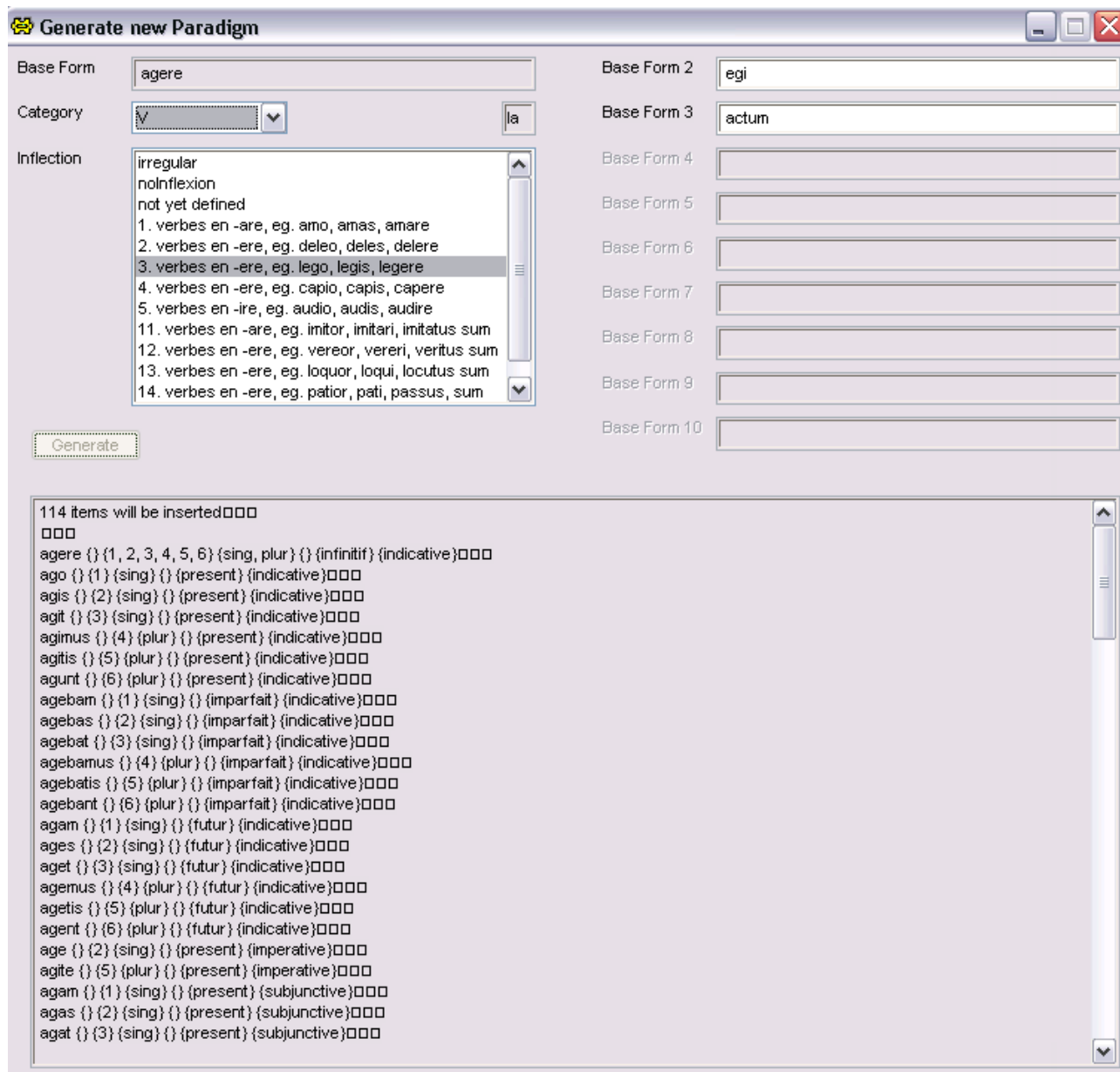


Illustration 3.2 : Entrée du verbe agere

À partir des formes de base, le programme déduit les radicaux essentiels à la construction de toutes les formes du verbe.

Un fichier contient les instructions nécessaires sous forme de règles de morphologie. Exemples :

- (1) INFL
`{-"e" > ""} (* lego *)`

`-"o" = (cat:V, inflClass:3, base:1, mode:indicatif, tense: présent,
voice:actif, pers:{1}, num:sing).`
- (2) INFL
`-"a" = (cat:N, inflClass:{1, 12}, base:1, gender:fem, pers:3, num:sing,
case:{nominatif, vocatif, ablatif})*;`

`-"am" = (cat:N, inflClass:{1, 12}, base:1, gender:fem, num:sing, pers:3,
case:accusatif);`
`...`
- (3) INFL (* superlatif des adj. de première classe *)

`-"issimus" = (cat:A, inflClass:1, base:1, degree:super, gender:masc,
num:sing, pers:{1, 2, 3}, case:nominatif);`
`...`

Les règles d'inflexion ont une structure qui se décompose en quatre points principaux :

- INFL : une règle commence toujours par ce mot-clé en majuscule, et se termine par un point.
- {...} : les accolades permettent d'indiquer une éventuelle modification à effectuer sur le radical. Dans notre fichier, *lege-* est le radical de *legere*. Pour obtenir la forme *lego*, il faut donc supprimer le *-e* final.
- `-" ... "` : le suffixe à ajouter au radical est donné.
- `= (...)` : à la suite du suffixe, les traits morphologiques associés au suffixe sont précisés. L'astérisque à la fin de la deuxième règle signifie que *ros-* (sans le *-a* du suffixe) est le radical du mot.

Ces règles ne sont pas toutes appliquées à chaque mot. Il serait peu utile de conjuguer le verbe *agere* sur le modèle d'*amare*. C'est pourquoi les mots sont répartis en classes d'inflexion que l'on choisit au moment de l'insertion dans le lexique (cf. Illustration 3.2). Seules les règles qui concernent la classe 3 (`inflClass:3`) dans la catégorie des verbes (`cat:V`) seront appliquées à *agere*.

Ce système permet de couvrir une grande partie de la morphologie. Le principal inconvénient réside dans le fait qu'une forme fléchie entrée dans le lexique ne peut être ni modifiée ni supprimée sans éliminer l'entier du paradigme. Si un mot compte une forme irrégulière (par exemple le datif-ablatif *bobus*, pluriel de *bos*), il faudra l'entrer entièrement à la main.

Le fichier complet contenant les règles pour la morphologie latine se trouve en annexe p. 61.

3.2.3 Le lexique latin

La construction du lexique soulève toutes sortes de problèmes. ceux-ci peuvent se manifester lors de la génération des formes fléchies ou, une fois le mot entré, au moment de l'attribution des différents paramètres.

L'interface (cf. Illustration 3.1) a dû être adaptée au fur et à mesure pour traiter certaines propriétés spécifiques au latin, notamment l'ajout du genre neutre, des degrés ainsi que des cas de l'adjectif.

Lorsque l'on insère un mot, la première décision à laquelle nous sommes confrontés est le choix de la catégorie grammaticale. Que faire par exemple du « pronom-adjectif possessif » *meus* ? En fait, bien que la catégorie du déterminant n'existe pas en latin, *meus* joue un rôle équivalent au déterminant français « mon » (*meus amicus*, « mon ami »), et au pronom « le mien ». C'est pourquoi *meus* doit être entré comme un déterminant et également comme un pronom³¹. Tous les mots qui appartiennent ainsi à plusieurs catégories grammaticales doivent être entrés deux fois. Les adjectifs ou les participes qui peuvent être nominalisés auront donc chacun deux paradigmes. Il faut souligner que cela engendre un grand nombre de formes et prend par conséquent beaucoup de place dans le lexique.

Pour ce travail, nous nous sommes concentrés en premier lieu sur le vocabulaire indispensable à l'analyse du corpus de phrases latines (chapitre 5, p. 41).

Une fois la catégorie définie, nous devons déterminer la classe d'inflexion. L'ensemble des classes est présenté au début du fichier de morphologie. Les verbes sont séparés en dix classes - cinq conjugaisons régulières et cinq conjugaisons déponentes. Tous les temps sont représentés à l'exception du supin, de l'impératif futur, du gérondif, de l'adjectif verbal et du participe futur. Les trois derniers sont déjà dans le fichier de morphologie et peuvent aisément s'ajouter au paradigme verbal. Dans le fichier actuel, il n'y a pas de classe pour les verbes réguliers sans participes passés (*timeo*, *careo*). Le participe est créé automatiquement par FipsMorpho et se compose uniquement des désinences sans le radical. Ils doivent être supprimés à la main.

Les classes sont encore plus nombreuses pour les noms, car il faut différencier non seulement les déclinaisons, mais aussi les genres ainsi que les noms propres qui ne se déclinent pas au pluriel.

En ce qui concerne les adjectifs, il en existe six classes. Les adjectifs au comparatif et/ou superlatif irrégulier ont leur propre classe. Ils sont entrés seulement au positif, puis leur classe d'inflexion doit être changée en « irrégulière »³² afin d'ajouter manuellement les formes du comparatif et du superlatif.

³¹ Il en va de même pour les pronoms démonstratifs (*hic-haec-hoc*, *is-ea-id*, etc).

³² L'option se situe dans la zone *Paradigm*, cf. Illustration 3.2, chapitre 3.2.1.

Une unique classe est présente pour les déterminants, et permet d'ajouter les pronoms possessifs formés sur le modèle des adjectifs de première classe (*meus, tuus, suus*). Les autres sont irréguliers.

Lorsque le mot est inséré dans le lexique, il reste encore certaines informations à préciser. Dans la zone *Lexeme*, le type et le sous-type doivent être complétés.

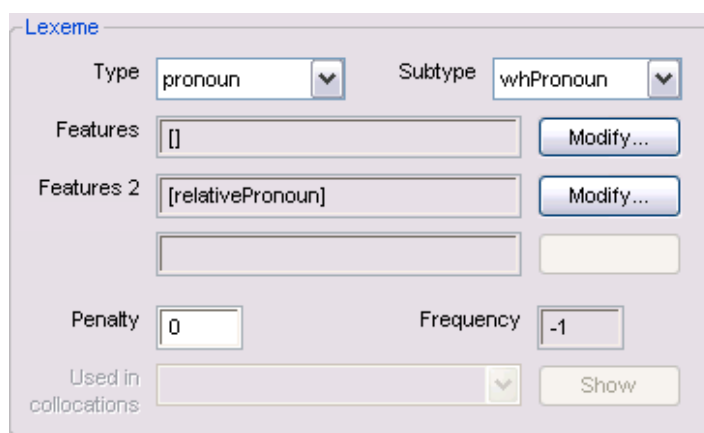


Illustration 3.3 : Lexème du pronom qui

Les noms peuvent être de type commun, propre ou pronom. Les pronoms interrogatif (*quis*) et relatif (*qui*) ont le sous-type « wh-pronouns ». Cette appellation provient de l'anglais, où les pronoms commençant par les lettres *wh-* forment un large groupe (*what, where, when, who, whom, whose, etc*). Les équivalents français et latins de ces pronoms sont souvent en *qu-* (*qui, quae, quod, quia, quando, quomodo* etc.). La distinction entre interrogatif et relatif s'opère grâce aux traits (*features*).

Les verbes entrés sont de type ordinaire, excepté *esse* qui est un auxiliaire.

Pour les adverbes, il est important de préciser le sous-type également quand il s'agit d'un adverbe d'intensité.

Les déterminants sont soit possessifs (*meus*), soit démonstratifs (*hic*), soit enfin numéraux (*unus*).

Pour affiner l'analyse, il est par ailleurs possible de compléter les traits dans d'autres cas, en précisant pour les noms s'il s'agit d'un nom comptable ; il existe aussi des traits pour les verbes (d'état, de mouvement...), les adjectifs, les préposition et, les conjonctions³³.

Enfin, la dernière partie du lexique nous permet d'ajouter des indications concernant la construction grammaticale des verbes, ainsi que des noms et des adjectifs.

Il est nécessaire ici de clarifier les notions d'argument, de rôle thématiques et de fonctions

³³ Il manque cependant l'option « subjonctif » pour les conjonctions suivies du subjonctif (*ut, ...*).

grammaticales.

La fonction grammaticale recouvre les termes de sujet, d'objet, de complément etc. Ce sont les rôles syntaxiques des constituants dans une phrase. Ils sont définis par leur cas.

Le rôle thématique, en revanche, caractérise la relation des arguments par rapport au verbe. Ils permettent de répondre aux questions « Qui fait quoi ? À qui ? Pour qui ? » etc. La situation décrite dans une phrase est en effet comparable au scénario d'un événement impliquant divers participant :

Nostris Caesar auxilium tulit.

Dans cette phrase, l'action est contenue dans le verbe *tulit*, « apporter ». Les participants de cette action sont César, l'aide et les soldats romains (*nostris*). Les rôles sont distribués en fonction de la scène : César est celui qui effectue l'action, il est l'agent ; *auxilium* est le thème, c'est-à-dire l'entité déplacée ; *nostris* est le bénéficiaire³⁴.

Les participants sont appelés les arguments du verbe. Les arguments ont donc un rôle syntaxique (sujet, objet, ...) et thématique (agent, thème, ...), mais il n'y a pas de stricte correspondance entre les deux : un agent ne sera pas toujours le sujet du verbe, comme on peut le constater dans les phrases passives.

Selon la construction du verbe, le nombre d'arguments³⁵ varie. Les verbes intransitifs possèdent un seul argument sujet (*verba volant*). Les verbes transitifs ont deux arguments (*audentes fortuna juvat*), tandis que les verbes ditransitifs, comme *ferro*, possèdent trois arguments.

Toutes ces données sont complétées dans la zone *Arguments* du lexique :

The screenshot shows a software interface for defining verb arguments. At the top, there are two dropdown menus: 'Sub Category' set to '[NP _ NP]' and 'Number of Arguments' set to '2'. Below this is a section titled 'Arguments' with a sub-header 'Argument Number' set to '1'. The main area contains several input fields: 'Grammatical Function' with the text 'subject', 'Thematic Function' with a dropdown menu set to 'agent', 'Features' with an empty list '[]', and 'Case' with the text '[nom]'. There are two 'Modify...' buttons to the right of the 'Features' and 'Case' fields.

Illustration 3.4 : Arguments d'un verbe transitif

³⁴ Il existe un grand nombre de rôles différents. Pour une liste des principaux rôles thématiques accompagnés d'exemples, cf. Devine et Stephens (2006), pp. 10-11.

³⁵ Le nombre d'arguments est appelé la valence du verbe.

3.3 Les étapes de l'analyse : fonctionnement de Fips

3.3.1 Lecture

L'analyseur commence par lire la phrase d'entrée. Cette première étape consiste à découper la chaîne d'entrée, considérée comme une suite de caractères par le programme, en unités lexicales appropriées pour l'analyse syntaxique. Ces unités lexicales ne sont pas forcément définies comme des mots séparés par des espaces. En français, les tirets et apostrophes sont une source d'ambiguïté, car ils font parfois office de séparateurs, mais pas dans tous les cas :

Sire, *rendez-vous* ce soir à la taverne.

J'ai pris un *rendez-vous* chez le dentiste.

Aujourd'hui.

Je *l'ai* fait.

En latin, ce problème ne se pose pas. On ne trouve ni séparateurs ni abréviations, contrairement au grec ancien qui réalise l'élision. Il faut toutefois tenir compte des particules enclitiques qui peuvent s'ajouter à la fin d'un mot, mais forment une unité lexicale distincte (*pellibusque sunt vestiti*). Ce phénomène est semblable à l'enclise des pronoms en italien (*lasciatemi cantare*).

On peut encore signaler le problème des variantes orthographiques pour les lettres i/j et u/v ainsi que certains préfixes comme *adf-/aff-*. L'orthographe adoptée pour ce travail est celle donnée par le Gaffiot.

3.3.2 Analyse

Lorsqu'il lit un mot d'une catégorie lexicale X, Fips crée une structure appelée projection qui correspond au syntagme XP³⁶. En outre, deux catégories fonctionnelles donnent lieu à une projection : un verbe conjugué ou à l'infinitif produit une projection TP (*Tense Phrase*) ; les structures fonctionnelles FP sont générées pour les propositions réduites³⁷.

³⁶ cf. p. 10 : liste des catégories et des syntagmes correspondants.

³⁷ Une proposition réduite est une structure prédicative, c'est-à-dire une structure qui comporte des arguments sans verbe conjugué. Ex : « Je vois *Sophie assise sur le canapé* ».

Le mot *lu* occupe la tête X du schéma suivant³⁸ :

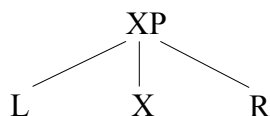


Illustration 3.5: Schéma X-barre

$[_{XP} L X R]$

Les positions L et R de la tête sont complétées par d'autres projections facultatives qui viennent s'attacher à gauche et à droite de la tête. De cette façon, les projections se combinent entre elles en suivant des règles d'attachement.

Prenons l'exemple du groupe nominal *Caesaris adventus*. À la lecture du premier mot *Caesaris*, la projection $[_{NP} Caesaris]$ est créée. *adventus* génère également une projection $[_{NP} adventus]$. Ces deux projections se combinent en suivant les règles d'attachement qui se trouvent dans un fichier spécial. La règle suivie dans cette situation est la suivante :

N+N

a.HasCase(genitive)

Cette règle indique qu'un nom peut s'attacher à gauche d'un autre nom s'il est au génitif (complément du nom). Une seconde règle similaire permet d'attacher le complément à droite (*adventus Caesaris*).

Les règles nous permettent donc d'indiquer que le complément du nom en latin peut se placer indifféremment à droite ou à gauche du nom, contrairement au français :

l'arrivée de César
Caesaris adventus
adventus Caesaris

Les règles précisent également que l'auxiliaire gouverne le participe passé, ou que le déterminant gouverne le nom : c'est le groupe nominal qui va s'attacher à droite du déterminant.

Elles décrivent donc les constituants dans leur structure générale et les attachements à gauche ou à droite. La structure des arguments (savoir ce qui est sujet, objet, etc.) est quant à elle encodée dans un autre fichier³⁹ contenant des procédures.

³⁸ Ceci est une version simplifiée du schéma proposé par Chomsky dans la théorie X-barre.

³⁹ Ces deux aspects de Fips, règles et procédures, ont été peu traités dans ce travail car les fichiers étaient déjà préexistants.

4 Description des structures syntaxiques

Dans ce chapitre, nous allons étudier la structure des constituants latins : la façon dont ils sont agencés dans la phrase ainsi que les phénomènes d'accord. Nous nous limiterons à examiner des exemples tirés de la prose classique. En effet, notre but n'est pas de couvrir la syntaxe latine de manière complète, de même que le but du lexique n'était pas une couverture exhaustive du vocabulaire latin.

L'ordre des mots en latin n'est pas aussi rigide qu'en français, car il ne joue pas un rôle primordial dans la compréhension de la phrase. La fonction de chaque mot ne dépend pas en effet de sa place, mais elle est connue grâce à la flexion nominale. Il en résulte que l'ordre des mots n'a jamais pris de signification syntaxique. Cependant certaines tendances et préférences se sont affirmées⁴⁰. Il est ainsi possible de classer le latin parmi les langues SOV⁴¹. Mais ces préférences n'ont rien de règles strictement observées, et l'usage courant est souvent modifié. L'ordre des mots peut être sujet à la thématization ou au *scrambling*⁴². Ces variations se produisent pour des raisons sémantiques, afin de mettre en évidence un mot, une expression. Car, pour citer Marouzeau, « L'ordre des mots en latin est libre, il n'est pas indifférent. Libre, en ce sens que, sauf exception, il n'y a pas pour chaque terme de la phrase une place attitrée, obligatoire. Mais non pas indifférent, parce qu'en général deux ordres possibles ne sont pas synonymes »⁴³.

4.1 Verbes

4.1.1 Position⁴⁴

La position usuelle du verbe est à la fin de la proposition, mais il n'y a pas besoin de chercher longtemps pour trouver une phrase où le verbe n'est pas situé en position finale.

Plusieurs facteurs favorisent par exemple la première place, comme illustré ci-dessous : une insistance particulière sur l'action, l'impératif, les verbes qui indiquent l'existence ou l'apparition, le passif, les verbes d'état mental :

conatus est Caesar reficere pontes

cedant arma togae

erat vallis inter duas acies

fit magna caedes

⁴⁰ cf. Ernout & Thomas pp. 161-163.

⁴¹ Devine & Stephens (2006) déterminent un ordre des mots neutre (par défaut) précis, cf. p. 79.

⁴² Modification apportée à l'ordre neutre afin de placer le thème (élément connu) avant le rhème (élément nouveau).

⁴³ Marouzeau (1922), p. 1.

⁴⁴ cf. Devine & Stephens (2006), chapitre 2.

datur negotium militibus
moverat plebem oratio consulis

Le verbe peut également se trouver en avant-dernière position, lorsqu'il se produit une inversion entre verbe et argument afin de mettre en évidence ce dernier :

opus quidem incipit nemo
cum in Italiam proficisceretur Caesar

Les temps verbaux formés sur le radical du parfait ont un passif « périphrastique », c'est-à-dire composé d'un participe ainsi que de l'auxiliaire conjugué *esse*. Le participe s'accorde alors de la même façon qu'un adjectif attribut (*puer et puella vocati sunt*).

L'ordre naturel place l'auxiliaire à la suite du participe, en fin de phrase :

in Galliam profectus est.

Cependant, lorsque l'information communiquée porte spécialement sur un élément, l'auxiliaire peut monter jusqu'à cet élément ; mais ce mouvement est bloqué par la négation, qui empêche le verbe de se déplacer plus haut dans la phrase :

pons paene erat refectus
ex duabus legionibus quae sunt traditae a Caesare
perfectum opus non erat

4.1.2 Arguments

Par défaut, c'est le sujet qui précède tous les autres arguments. Il n'est pas forcément à la première place absolue de la phrase, car celle-ci peut être occupée par un complément circonstanciel ou un adverbe.

Lorsqu'il est expressément mentionné, le sujet est au nominatif. Il s'accorde en nombre et en personne avec le verbe (*discipulus legit, discipuli legunt, ego et tu legimus*).

Si un verbe compte plusieurs sujets, il peut s'accorder parfois avec le plus proche (*tempus necessitasque postulat*).

Le rôle du sujet peut être rempli par un verbe à l'infinitif : *errare humanum est, nefas est deserere patriam*.

Passons maintenant en revue les différents arguments autres que le sujet.

L'objet direct du verbe est le plus souvent à l'accusatif (*amo patrem, patrem imitor*). Selon les verbes, il peut cependant être au génitif (*injuriarum obliviscor*), au datif (*vires mihi desunt*), ou à l'ablatif (*nulla re caret*).

Les compléments du verbe indirects recouvrent les différents circonstanciels de lieu, temps, d'attribution etc. Ils s'expriment par un groupe prépositionnel (*venit in hortum, auxilium a Caesare petere coeperunt*) ou un cas. La manière, par exemple, est à l'ablatif (*magna voce clamat*) ; le complément d'attribution se réalise, quant à lui, avec le datif (*poeta Elissae rosam dedit, sol omnibus lucet, mihi est liber*).

L'agent du passif est introduit par la préposition *a / ab* + ablatif : *amor a patre. a fortuna deserebantur. a militibus pugnatur*. Mais le complément du passif se met à l'ablatif seul pour exprimer la cause ou le moyen : *maerore conficior*.

Le complément du verbe peut être une proposition réduite (FP) : *pecunia homines beatos non reddit*.

Enfin, le verbe peut être complété par un infinitif seul, une proposition infinitive ou une subordonnée comprenant un verbe conjugué. La distinction entre ces trois arguments de type « sentence » (S) s'effectue dans le lexique grâce aux options : *infinitivalS* (infinitif seul), *infinitivalACI* (proposition infinitive) et *tensedS*.

Dans le cas de l'infinitif seul, le sujet du verbe principal contrôle également le verbe à l'infinitif (*incipit ludere, volo esse clemens*).

Les subordonnées circonstancielles jouent le rôle d'un complément circonstanciel (temporel, de but, de conséquence, de comparaison, de condition, de cause, de concession) pour un verbe. Elles sont introduites par un subordonnant suivi d'un verbe conjugué.

Ex : *dum haec a Caesare geruntur, Galli impetum parant*.

Les subordonnées complétives jouent un rôle de complément d'objet.

La proposition infinitive est un type de subordonnée complétive sans subordonnant, dont le verbe est à l'infinitif et le sujet à l'accusatif. Elle suit les verbes de sentiment, d'opinion (*dico, credo, scio, sentio, ...*) et les verbes de volonté (*volo, iubeo, patior, veto, ...*).

Exemples de complétive : *suadeo tibi ut legas, scio vitam esse brevem*.

L'usage du passif personnel est fréquent dans la langue classique : *Galli dicuntur in Italiam transisse* plutôt que *Gallos dicitur in Italiam transisse*. Dans ce cas, l'attribut du sujet est au nominatif, exemple : *Homerus dicitur caecus fuisse*. (il n'existe par contre pas d'exemple avec un participe nominatif). Ce cas se rapproche alors d'un infinitif seul.

4.2 Noms et adjectifs

L'adjectif s'accorde en genre, nombre et cas avec le nom qu'il qualifie.

Exemples : *maximus ardor, bona mater, vir bonus, nihil utile, pater est bonus, pecunia homines beatos non reddit.*

L'adjectif épithète qui porte sur plusieurs noms ne s'accorde qu'avec un seul d'entre eux, d'ordinaire le plus proche : *ardor gaudiumque maximum / maximus ardor et gaudium.*

L'adjectif attribut s'accorde le plus souvent⁴⁵ avec l'ensemble des noms. Si les noms désignent des personnes, l'adjectif s'accorde au masculin ; il s'accorde au neutre si les noms représentent des choses. Exemples : *pater et mater sunt boni, pulchra sunt domus et hortus.*

Les relatives ont une fonction équivalente à celle d'un adjectif. Le subordonnant relatif s'accorde en genre et nombre avec son antécédent. Dans le cas où il possède plusieurs antécédents, il se comporte comme un adjectif attribut. Le cas du relatif est déterminé par sa fonction dans la proposition qu'il introduit.

Exemples : *timeo virum qui venit. Caesar misit legiones quae in Gallia erant.*

Le complément du nom est en général au génitif (*liber Petri, adventus Caesaris, metus hostium, pars civium, ...*). Il est possible de le trouver à l'ablatif, si c'est un adjectif qui exprime une nuance de qualité (*puer egregiae indolis / puer egregia indole*). Il peut également être formé d'un groupe prépositionnel (*vas ex auro, pugna ad Cannas*). Tout comme les adjectifs, les compléments du nom se placent soit à gauche, soit à droite du nom.

Selon les adjectifs, le complément peut être à l'accusatif (mesure), au génitif (possession, savoir, désir), au datif (utilité, aptitude, voisinage, amitié), à l'ablatif (privation, abondance, ...) et avec *ad* + accusatif (inclination).

Exemples : *murus decem pedes altus, avidus laudis, utilis civitati, dignus laude, paratus ad bellum*

Le complément du comparatif est plus souvent à l'ablatif, ou bien exprimé grâce à une subordonnée elliptique introduite par *quam*, et le deuxième terme de la comparaison s'accorde alors avec le premier (*Paulus est doctior quam Petrus* ou *Paulus Petro doctior est*). Quant au complément du superlatif, il est soit au génitif partitif, soit introduit par la préposition *e / ex* + ablatif ou *inter* + accusatif (*altissima arborum* ou *altissima ex arboribus, felicissimus inter principes*).

Sans complément, le comparatif exprime une intensité (*doctior*, « assez, particulièrement, savant »)

⁴⁵ Cf. Ernout-Thomas p. 129, § 150.

et le superlatif est absolu : *doctissimus* « très savant ».

Pour quelques adjectifs, le comparatif et le superlatif se forment à l'aide des adverbes *magis* et *maxime* : *magis dubius*, *maxime idoneus*.

4.3 Autres

4.3.1 Les adverbes

Les adverbes sont répartis dans quatre types différents : adverbes d'interrogation, de négation, adverbes de phrase et adverbes standards.

Les adverbes d'interrogation se placent en début de phrase (*ubi*, *quando*, *quomodo*, *num*, *nonne*...) excepté la particule enclitique *-ne*.

Ex : *vidistine Romam ? Nonne amicus meus es ?*

Parmi les adverbes de négation, on trouve : *non*, *haud*, *ne* (ne pas confondre avec la particule *-ne*), *neque* (*nec*), *neve*, *nihil*, *ne quidem*, *non jam* etc.

Ex : *non legit, ludit neque legit, ne legat !, ne in Gallia quidem, jam non veniet.*

Deux négations s'annulent et valent une affirmation : *non nescio*, « je n'ignore pas, je sais ».

Cependant, selon que *non* suit ou précède une autre négation, cela change le sens de certaines expressions : *nemo non venit*, « tout le monde est venu », *non nemo venit*, « Quelques-uns sont venu ».

Les adverbes de phrase jouent un rôle de connecteur. Ils se rencontrent en général au début de la phrase.

Ex : *etiam, scilicet, nunc, tum, interea, nam, enim, autem, itaque, igitur ...*

Le reste des adverbes se classe dans le type standard, comme c'est le cas des adverbes de quantité. Ceux-ci portent sur le verbe, mais aussi sur un adjectif, un autre adverbe ou encore un nom au génitif.

Ex : *minus laboras, minus doctus, minus docte, multum vini.*

4.3.2 Les particules enclitiques

Les quatre particules *-que*, *-ve*, *-ne* et *-ce* sont enclitiques : elles s'attachent à la fin du mot qu'elles modifient.

Les particules *-que* et *-ve* sont équivalentes aux conjonctions de coordination *et* et *vel*. La première exprime la conjonction (« et »), la seconde la disjonction (« ou »).

Exemples : *inter Euboeam continentemque terram, lacte et carne vivunt pellibusque sunt vestiti, Myrmidonum, Dolopumve aut duri miles Ulixi.*

La coordination se fait entre deux groupes de mots qui remplissent la même fonction syntaxique, qu'il s'agisse de noms, d'adjectifs ou de propositions entières. Lorsque le second groupe comprend plusieurs mots, la particule se lie au premier mot. Cependant, s'il s'agit d'un groupe prépositionnel, la particule *-que* a tendance à se lier avec le mot suivant la préposition : *ex omnique genere*. Cela peut se produire aussi avec un adverbe comme *tam* : *propter tot, tantos, tam praecipitesque casus.*

L'interrogation peut être marquée par *-ne*. Dans ce cas, la particule s'attache au mot sur lequel porte l'interrogation. Il s'agit le plus souvent du premier mot de la phrase.

Exemple : *Venisne ?*

-ne peut également se trouver en corrélation avec *an* dans une interrogative indirecte, pour former une interrogation double.

Exemple : *Qaero verumne an falsum sit.*

La dernière particule épithétique *-ce* est beaucoup plus rare. Elle renforce les pronoms démonstratifs (exemple : *huiusce*). Elle est assez peu utilisée pour qu'il soit aisé d'entrer les occurrences directement dans le lexique.

4.3.3 Les compléments de la phrase

Certains compléments portent sur la phrase entière. Il s'agit des circonstancielles introduites par un subordonnant, ainsi que les ablatifs absolus :

*cum in Italiam proficisceretur Caesar, legiones in Galliam misit.
partibus factis, verba fecit leo.*

Comme la proposition infinitive, l'ablatif absolu est une proposition sans mot subordonnant. Le sujet est exprimé à l'ablatif et le verbe à l'ablatif du participe.

Lorsque le participe est absent, il faut alors sous-entendre un participe du verbe « être » : *Cicerone consule*, « Cicéron étant consul ». Le verbe *esse*, en effet, ne possède pas de forme participiale.

4.4 L'hyperbate⁴⁶

Cette figure de style, extrêmement courante en latin, entraîne la disjonction de plusieurs éléments appartenant à un même constituant.

Le nom et l'adjectif sont souvent séparés par une préposition, le complément du nom, un verbe, ou plusieurs éléments à la fois. L'hyperbate peut également dissocier un nom et son complément, ainsi que des constituants coordonnés.

magna cum cura

Caesar duas legiones in citeriore Gallia novas conscripsit

quem ad finem sese effrenata jactabit audacia

summam inter eos habet auctoritatem

duas a te accepi epistulas

hae permanserunt aquae dies complures

apud Helvetios longe nobilissimus fuit et ditissimus Orgetorix

⁴⁶ cf. Devine & Stephens (2006), chapitre 6.

5 Examen d'un corpus de phrases avec Fips

La matière du chapitre précédent nous a fourni un grand nombre d'énoncés illustrant divers points de syntaxe. À partir de ces exemples, nous avons constitué un corpus afin de tester l'analyseur Fips.

5.1 Corpus

5.1.1 Noms

maximus ardor
ardor et gaudium
ardor gaudiumque
honore praemiove primo
bona mater
mater bona
nihil utile
adventus Caesaris
Caesaris adventus
pars civium
hominum memoria
magna corporis parte
magna parte
magna parte corporis
homo egregiae indolis
puer egregia indole
vas ex auro
pugna ad Cannas
meus amicus
hanc urbem

5.1.2 Adjectifs

avidus laudis
dignus laude
utilis civitati
Paulus est doctior quam Petrus

altissima arborum
altissima ex arboribus

5.1.3 Prépositions

apud patrem
apud bonum patrem
cum magno viro
magna cum cura
in hortum
ab amicis
post mortem
ex auro
de Caesaris adventu
de deorum potestate
ex omnique genere
in conspectu meo

5.1.4 Adverbes

minus laboras
fortiter pugnat
miles fortiter pugnat
pugnat fortiter
miles pugnat fortiter
minus doctus
minus docte
tam bene
multum vini

5.1.5 Attributs du verbe *esse*

ego sum nauta
ego nauta sum
tu es agricola
ego sum nauta, tu es agricola
ego sum nauta et tu es agricola

rosa est pulchra
domus pulchra est
pater bonus est
pater est bonus
bonus est pater
pater et mater sunt boni
pulchra sunt domus et hortus
Elissa regina erat
Elissa est regina
est Elissa regina
Elissa regina non erat
Elissa non erat regina
Cicero est Romae consul
Cicero est consul Romae
Cicero consul Romae est
Cicero Romae consul est
multae sunt provinciae romanae
Cicero et Caesar magni homines fuerunt
Cicero et Caesar sunt magni homines
Caesar fuit calvus
agricolae sumus
non nautae sumus
nautae non sumus
puellae bonae et pulchrae estis
errare humanum est
nefas est deserere patriam

5.1.6 Compléments du verbe

discipulus legit
discipuli legunt
puer et puella legunt
ego et tu legimus
amo patrem
patrem amo
pater amat filium
pater filium amat

filium amat pater
amat pater filium
magnam habet auctoritatem
Romulus urbem condidit
pecunia homines beatos reddit
pecunia reddit homines beatos
pecunia homines beatos non reddit
injuriarum obliviscor
vires mihi desunt
mihi est liber
Caesar pepercit barbaris
villa abundat lacte
nulla re caret
incipit ludere
poeta rosam Elissae dedit
poeta Elissae rosam dedit

5.1.7 Passif

amor ab amico
amatur ab amico
ab amico amatur
maerore conficior
a fortuna deserebantur
relicti estis ab amicis
ab amicis relictis estis
ab hostibus captus eram
tribunus plebis interfectus erat
Galli dicuntur in Italiam transisse
Gallos dicitur in Italiam transisse
Homerus dicitur caecus fuisse

5.1.8 Subordonnées

scio vitam esse brevem
pater dicit vitam esse brevem
scio me clementem esse

volo esse clemens
scio patrem venire
regina amat hominem qui venit
Caesar misit legiones quae in Gallia erant
suadeo tibi ut legas
dum haec a Caesare geruntur, Galli impetum parant
cum in Italiam proficisceretur Caesar, legiones in Galliam misit
partibus factis, verba fecit leo

5.1.9 interrogation et négation

nonne amicus meus es ?
vidistine Romam ?
non legit
ne legat
ludit neque legit
jam non veniet
nemo venit

5.2 Test

5.2.1 Noms

maximus ardor

```
[TP [VP[NP[AP maximus ] ardor ] ]]      55      0.0  
[NP[AP maximus ] ardor ]      60      0.0  
[TP [VP[FP[DP ] [AP maximus ]][NP ardor ] ]] 130      0.0
```

ardor et gaudium

```
[TP [VP[NP [ConjP[NP ardor ] et [NP gaudium ]]] ]] 155      0.0  
[NP [ConjP[NP ardor ] et [NP gaudium ]]]      160      0.0
```

ardor gaudiumque

```
[TP [VP[NP ardor [DP gaudiumque ] ] ] ] 141      0.0  
[TP [VP[NP ardor ] [DP gaudiumque ] ] ] 141      0.0  
[NP ardor [DP gaudiumque ] ]      146      0.0  
[TP [VP[NP ardor [AP gaudiumque ] ] ] ] 171      0.0  
[NP ardor [AP gaudiumque ] ]      176      0.0  
[TP [VP[NP ardor ] [FP[DP ] [AP gaudiumque ] ] ] ] 231      0.0
```

honore praemiove primo

```
[TP [VP[NP honore [DP praemiove ]][FP[DP ] [AP primo ] ] ] ] 246      0.0
```

bona mater

```
[TP [VP[NP[AP bona ] mater ] ] ]      55      0.0  
[NP[AP bona ] mater ]      60      0.0  
[TP [VP[FP[DP ] [AP bona ]][NP mater ] ] ] 130      0.0
```

mater bona

```
[TP [VP[NP mater [AP bona ] ] ] ] 70      0.0  
[NP mater [AP bona ] ]      75      0.0  
[TP [VP[NP mater ] [FP[DP ] [AP bona ] ] ] ] 130      0.0
```

nihil utile

```
[TP [VP[NP nihil [AP utile ] ] ] ] 70      0.0  
[NP nihil [AP utile ] ]      75      0.0  
[TP [VP[NP nihil ] [FP[DP ] [AP utile ] ] ] ] 130      0.0
```

adventus Caesaris

```
[TP [VP[NP adventus ] [DP Caesaris ] ] ] 40      0.0  
[NP adventus [DP Caesaris ] ]      45      0.0
```

Caesaris adventus

[TP [VP[DP Caesaris] [NP adventus]]] 40 0.0
[DP Caesaris [NP adventus]] 45 0.0

pars civium

[TP [VP[NP pars [NP civium]]]] 70 0.0
[TP [VP[NP pars] [NP civium]]] 70 0.0
[NP pars [NP civium]] 75 0.0

hominum memoria

[TP [VP[NP hominum] [NP memoria]]] 70 0.0
[NP hominum [NP memoria]] 75 0.0

magna corporis parte

[TP [VP[FP[DP] [AP magna]] [DP corporis] [NP parte]]] 246 0.0

magna parte

[TP [VP[NP[AP magna] parte]]]

magna parte corporis

[TP [VP[NP[AP magna] parte [DP corporis]]]]

homo egregiae indolis

[TP [VP[NP homo [NP[AP egregiae] indolis]]]] 100 0.0
[TP [VP[NP homo] [NP[AP egregiae] indolis]]] 100 0.0
[NP homo [NP[AP egregiae] indolis]] 105 0.0
[TP [VP[NP homo] [FP[DP] [AP egregiae]] [NP indolis]]] 175 0.0

puer egregia indole

[TP [VP[NP puer] [NP[AP egregia] indole]]] 100 0.0
[TP [VP[NP puer] [FP[DP] [AP egregia]] [NP indole]]] 175 0.0

vas ex auro

[TP [VP[NP vas] [PP ex [NP auro]]]] 95 0.0

pugna ad Cannas

[TP [VP[NP pugna] [PP ad [NP Cannas]]]] 95 0.0

meus amicus

[TP [VP[DP meus [NP amicus]]]] 40 0.0
[DP meus [NP amicus]] 45 0.0

hanc urbem

```
[TP [VP[DP hanc ] [NP urbem ] ]] 40 0.0  
[DP hanc [NP urbem ] ] 45 0.0
```

Remarques : problème de coordination avec les particules enclitiques *-que* et *-ve* (valable dans tout le corpus). Le groupe nom-adjectif est correctement interprété, hormis le groupe *magna corporis parte* où le génitif complément de *parte* est enclavé. Le complément du nom à l'ablatif et le complément avec préposition ne sont pas reconnus. Les déterminants ne posent ici aucune difficulté.

5.2.2 Adjectifs

avidus laudis

```
[TP [VP[FP[DP ] [AP avidus ]][NP laudis ] ]] 130 0.0
```

utilis civitati

```
[TP [VP[FP[DP ] [AP utilis ]][NP civitate ] ]] 130 0.0
```

dignus laude

```
[TP [VP[FP[DP ] [AP dignus ]][NP laude ] ]] 130 0.0
```

Paulus est doctior quam Petrus

```
*** no analysis [TP [VP[DP Paulus ]i est [FP[DP ] [AP doctior ]][AdvP quam ]]] 60 0.0  
[DP Petrus ] 0 0.0
```

altissima arborum

```
[TP [VP[FP[DP ] [AP altissima ]][NP arborum ] ]] 130 0.0
```

altissima ex arboribus

```
[TP [VP[FP[DP ] [AP altissima ]][PP ex [NP arboribus ] ]] 155 0.0
```

Remarques : problèmes avec les compléments de l'adjectif, au positif, comparatif et superlatif.

5.2.3 Prépositions

apud patrem

```
[PP apud [NP patrem ] ] 45 0.0  
[TP [VP[PP apud [NP patrem ] ]] ] 50 0.0
```

apud bonum patrem

[PP apud [NP[AP bonum] patrem]] 75 0.0
[TP [VP[PP apud [NP[AP bonum] patrem]]]] 80 0.0

cum magno viro

[PP cum [NP[AP magno] viro]] 75 0.0
[TP [VP[PP cum [NP[AP magno] viro]]]] 80 0.0

magna cum cura

[TP [VP[FP[DP] [AP magna]][PP cum [NP cura]]]] 155 0.0

in hortum

[PP in [NP hortum]] 45 0.0
[TP [VP[PP in [NP hortum]]]] 50 0.0

ab amicis

[PP ab [NP amicis]] 45 0.0

post mortem

[PP post [NP mortem]] 45 0.0
[TP [VP[PP post [NP mortem]]]] 50 0.0

ex auro

[PP ex [NP auro]] 45 0.0
[TP [VP[PP ex [NP auro]]]] 50 0.0

de Caesaris adventu

*** no analysis [PP de] 0 0.0
[DP Caesaris] 0 0.0
[NP adventu] 30 0.0

de deorum potestate

*** no analysis [PP de] 0 0.0
[NP deorum] 30 0.0
[NP potestate] 30 0.0

ex omnique genere

*** no analysis [PP ex] 0 0.0
[VP omnique [NP genere]] 76 0.0

in conspectu meo

[TP [VP[PP in [NP conspectu]][DP meo]] 666 0.0

Remarques : problème lorsque l'adjectif est mis en évidence avant la préposition (*magna cum cura*), ainsi qu'avec un complément du nom inséré entre la préposition et son régime (*de Caesaris adventu*). Pour le reste, le groupe prépositionnel « normal » est bien identifié. Le déterminant postposé n'est pas rattaché au nom qu'il gouverne (*in conspectu meo*).

5.2.4 Adverbes

minus laboras

```
*** no analysis [AdvP minus ]    30    0.0
[VP laboras ]30    0.0
```

fortiter pugnat

```
[TP [VP[AdvP fortiter ][DP ] pugnat ]] 85    0.0
[VP[AdvP fortiter ][DP ] pugnat ]      100   0.0
```

miles fortiter pugnat

```
[TP [VP[NP miles ][AdvP fortiter ][DP ] pugnat ]] 50    0.0
```

pugnat fortiter

```
[TP [VP pugnat [AdvP fortiter ]]]      260   0.0
```

miles pugnat fortiter

```
[TP [VP[NP miles ] pugnat [AdvP fortiter ]]] 25    0.0
```

minus doctus

```
[AP[AdvP minus ] doctus ] 60    0.0
```

minus docte

```
[TP [VP [AdvP[AdvP minus ] docte ]]] 55    0.0
[AP[AdvP minus ] docte ] 60    0.0
[AdvP[AdvP minus ] docte ]60    0.0
```

tam bene

```
[TP [VP [AdvP[AdvP tam ] bene ]]]      55    0.0
[AdvP[AdvP tam ] bene ] 60    0.0
```

multum vini

```
*** no analysis [AdvP multum ]    30    0.0
[NP vini ] 30    0.0
```

Remarques : tous les groupes adverbiaux sont correctement analysés excepté *minus laboras* et *multum vini*.

5.2.5 Attributs du verbe *esse*

ego sum nauta

[TP [VP[DP ego] sum [NP nauta]]] -65 0.0

ego nauta sum

[TP [VP[DP ego [NP nauta]] sum]] 235 0.0

tu es agricola

[TP [VP[DP tu] es [NP agricola]]] -65 0.0

ego sum nauta, tu es agricola

*** no analysis [TP [VP[DP ego] sum [NP nauta]]] -45 0.0
 [DP tu] 0 0.0
 [VP[NP agricola] es] 75 0.25

ego sum nauta et tu es agricola

*** no analysis [TP [VP[DP ego] sum [NP [ConjP[NP nauta] et]]]] 5 0.0
 [DP tu] 0 0.0
 [VP[NP agricola] es] 75 0.25

rosa est pulchra

[TP [VP[NP rosa]i est [FP[DP] [AP pulchra]]]] -15 0.0

domus pulchra est

[TP [VP[NP domus]][FP[DP] [AP pulchra]] est]] 45 0.0
 [TP [VP[NP domus]][FP[DP] [AP pulchra]] est]] 85 0.0

pater bonus est

[TP [VP[NP pater [AP bonus]] est]] 265 0.0

pater est bonus

[TP [VP[NP pater]i est [FP[DP] [AP bonus]]]] -15 0.0

bonus est pater

TP [VP[FP[DP] [AP bonus]] est [NP pater]]] 25 0.0

pater et mater sunt boni

[TP [VP[NP [ConjP[NP pater]i et [NP mater]]] sunt [FP[DP] [AP boni]]]]

pulchra sunt domus et hortus

[TP [VP[FP[DP] [AP pulchra]][DP] sunt [NP [ConjP[NP domus] et [NP hortum]]]]]] 145
0.0

Elissa regina erat

TP [VP[DP Elissa [NP regina]] erat]] 195 0.0
[TP [VP[DP Elissa [NP regina]] erat]] 235 0.0

Elissa est regina

[TP [VP[DP Elissa] est [NP regina]]] -105 0.0

est Elissa regina

[TP [VP est [DP Elissa] [NP regina]]] -125 0.0
[TP [VP est [DP Elissa [NP regina]]]] 175 0.0

Elissa regina non erat

[TP [VP[DP Elissa] [NP regina] [AdvP non] [DP] erat]] -55 0.0
[TP [VP[DP Elissa] [NP regina] [AdvP non] [DP] erat]] -15 0.0

Elissa non erat regina

[TP [VP[DP Elissa] [AdvP non] [DP] erat [NP regina]]] -35 0.0

Cicero est Romae consul

*** no analysis [TP [VP[DP Cicero] est [NP Romae]]] -85 0.0
[NP consul] 30 0.0

Cicero est consul Romae

[TP [VP[DP Cicero] est [NP consul [NP Romae]]]] -60 0.0

Cicero consul Romae est

[TP [VP[DP Cicero [NP consul [NP Romae]]] est]] 280 0.0

Cicero Romae consul est

[TP [VP[DP Cicero] [NP Romae] [NP consul] [DP] est]] 160 0.1

multae sunt provinciae romanae

[TP [VP[FP[DP] [AP multae]] sunt [NP provinciae [AP romanae]]]] 70 0.0

Cicero et Caesar magni homines fuerunt

[TP [VP[DP [ConjP[DP Cicero] et [DP Caesar]]]][NP[AP magni] homines] fuerunt]] 35
0.0

Cicero et Caesar sunt magni homines

[TP [VP[DP [ConjP[DP Cicero] et [DP Caesar]]] sunt [NP[AP magni] homines]]] 15
0.0

Caesar fuit calvus

[TP [VP[DP Caesar]i fuit [FP[DP] [AP calvus]]]] -45 0.0

agricolae sumus

[TP [VP[NP agricolae] sumus]] 180 0.0

non nautae sumus

*** no analysis [AdvP non] 30 0.0
[DP nautae] 30 0.0
[VP sumus] 30 0.5

nautae non sumus

[TP [VP[NP nautae] [AdvP non] [DP] sumus]] 250 0.0
[TP [VP[NP nautae] [AdvP non] [DP] sumus]] 290 0.0

puellae bonae et pulchrae estis

[TP [VP[NP puellae [AP [ConjP[AP bonae] et [AP pulchrae]]]] estis]] 305 0.0

errare humanum est

[TP [VP[TP[DP] [VP errare]][FP[DP] [AP humanum]][DP] est]] 225 0.125

nefas est deserere patriam

*** no analysis [TP [VP[NP nefas]]] 45 0.0
[VP[DP] [TP[DP] [VP deserere [NP patriam]]] est] 50 0.16666666666666667

Remarques : problème lorsque l'attribut est placé avant le verbe. Les phrases *domus pulchra est* et *pater bonus est* ne produisent pas le même résultat. La coordination entre deux sujets est reconnue (*Cicero et Caesar magni homines fuerunt*), contrairement à la coordination entre deux phrases (*ego sum nauta et tu es agricola*). Le groupe nominal *Romae consul* pose problème, tandis que *consul Romae* est convenablement analysé. La négation qui est séparée du verbe n'est pas identifiée (*non nautae sumus*). La dernière phrase avec un infinitif pour sujet n'est pas reconnue.

5.2.6 Compléments du verbe

discipulus legit

[TP [VP[NP discipulus] legit]] 120 0.0

discipuli legunt

[TP [VP[NP discipuli] legunt]] 80 0.0

puer et puella legunt

[TP [VP[NP [ConjP[NP puer] et [NP puella]]][DP] legunt]] 390 0.0

ego et tu legimus

*** no analysis [DP [ConjP[DP ego] et [DP tu]]] 50 0.0
[VP legimus]30 0.0

amo patrem

[TP [VP amo [NP patrem]]]160 0.0

patrem amo

[TP [VP[NP patrem] amo]]220 0.0

pater amat filium

[TP [VP[NP pater] amat [NP filium]]] -75 0.0

pater filium amat

[TP [VP[NP pater] [NP filium] amat]] -15 0.0

filium amat pater

[TP [VP[NP filium] amat [NP pater]]] -35 0.0

amat pater filium

TP [VP amat [NP pater] [NP filium]]] -95 0.0

magnam habet auctoritatem

[TP [VP[NP[AP magnam] [AP habet] auctoritatem]]] 186 0.0
[NP[AP magnam] [AP habet] auctoritatem] 191 0.0
[TP [VP[FP[DP] [AP magnam]][DP habet] [NP auctoritatem]]] 246 0.0
[TP [VP[FP[DP] [AP magnam]][NP[AP habet] auctoritatem]]] 261 0.0

Romulus urbem condidit

[TP [VP[NP Romulus] [NP urbem] condidit]] 25 0.0

pecunia homines beatos reddit

[TP [VP[NP pecunia]][NP homines [AP beatos]][DP] reddit]] 30 0.0

pecunia reddit homines beatos

*** no analysis [TP [VP[NP pecunia]][DP] reddit]] 0 0.0
[NP homines [AP beatos]] 75 0.0

pecunia homines beatos non reddit

[TP [VP[NP pecunia]][NP homines [AP beatos]][AdvP non]][DP] reddit]] 60 0.0

injuriarum obliviscor

[TP [VP[NP injuriarum]][DP] obliviscor]] 260 0.0

vires mihi desunt

[TP [VP[NP vires]][DP mihi]][DP] desunt]] -5 0.0

mihi est liber

[TP [VP[DP mihi]][DP] est [NP liber]]] -75 0.0

Caesar pepercit barbaris

[TP [VP[DP Caesar]][DP] pepercit [NP barbaris]]] -65 0.0

villa abundat lacte

[TP [VP[NP villa]][DP] abundat [NP lacte]]] -75 0.0

nulla re caret

[TP [VP[NP[AP nulla] re]][DP] caret]] 290 0.0

incipit ludere

[TP [VP[DP] incipit [TP[DP] [VP ludere]]]] 40 0.0

poeta rosam Elissae dedit

[TP [VP[NP poeta]][NP rosam [DP Elissae]][DP] dedit]] 0 0.0

poeta Elissae rosam dedit

[TP [VP[NP poeta [DP Elissae]][NP rosam]][DP] dedit]] 0 0.0

Remarques : l'accord sujet-verbe est en ordre, hormis pour *ego et tu legimus* où les deux sujets coordonnés ne sont pas à la même personne. Les compléments d'objet direct sont bien analysés,

ainsi que *incipit ludere*. Cependant l'analyseur ne reconnaît pas l'hyperbate et *magna habet auctoritatem* conduit donc à une analyse erronée. La construction du verbe ditransitif *dare* semble également problématique.

5.2.7 Passif

amor ab amico

[TP [VP[NP amor] [PP ab [NP amico]]]] 95 0.0

amatur ab amico

*** no analysis [VP[DP] amatur] 70 0.0
[PP ab [NP amico]] 45 0.0

ab amico amatur

[TP [VP[PP ab [NP amico]] [DP] amatur]] 375 0.0

maerore conficior

[TP [VP[NP maerore] [DP] conficior]] 260 0.0

a fortuna deserebantur

[TP [VP[PP ab [NP fortuna]] [DP] deserebantur]] 375 0.0

relicti estis ab amicis

*** no analysis [TP [VP[FP[DP] [VP relicti]] [DP] estis]] 155 0.0
[PP ab [NP amicis]] 45 0.0

ab amicis relicti estis

[TP [VP[PP ab [NP amicis]] [FP[DP] [VP relicti]] [DP] estis]] 455 0.0

ab hostibus captus eram

[TP [VP[PP ab [NP hostibus]] [FP[DP] [VP captus]] [DP] eram]] 455 0.0

tribunus plebis interfectus erat

[TP [VP[NP tribunus [NP plebis]] [FP[DP] [VP interfectus]] [DP] erat]] 185 0.0

Galli dicuntur in Italiam transisse

*** no analysis [TP [VP[NP Galli] [DP] dicuntur]] 80 0.0
[PP in [DP Italiam]] 15 0.0
[VP transisse] 30 0.0

Gallos dicitur in Italiam transisse

```
*** no analysis [TP [VP[NP Gallos ]][DP ] dicitur ]] 80 0.0  
[PP in [DP Italiam ]] 15 0.0  
[VP transisse ] 30 0.0
```

Homerus dicitur caecus fuisse

```
*** no analysis [TP [VP[DP Homerus ]][DP ] dicitur ]] 50 0.0  
[AP caecus ] 30 0.0  
[VP fuisse ] 30 0.5
```

Remarques : ne pas oublier d'ajouter la variante *a* de la préposition *ab*. *Amor* n'est pas considéré comme un verbe (passif de *amo*). Problème lorsque l'agent du passif est situé après le verbe (*amatur ab amico*, *relictis estis ab amicis*). Le passif personnel, ou suivi d'un infinitif, n'est pas analysé.

5.2.8 Subordonnées

scio vitam esse brevem

```
*** no analysis [VP[DP ] scio ] 70 0.0  
[NP vitam ] 30 0.0  
VP[FP[DP ] [AP brevem ]] esse ] 135 0.25
```

pater dicit vitam esse brevem

```
*** no analysis [TP [VP[NP pater ]][DP ] dicit [NP vitam ]]]-15 0.0  
[VP[FP[DP ] [AP brevem ]] esse ] 135 0.25
```

scio me clementem esse

```
*** no analysis [VP[DP ] scio ] 70 0.0  
[DP me ] 0 0.0  
[AP clementem ] 30 0.0  
[VP esse ] 30 0.5
```

volo esse clemens

```
[TP [VP[DP ] volo [TP[DP ] [VP[FP[DP ] [AP clemens ]] esse ]]]] 145 0.125
```

scio patrem venire

```
*** no analysis [VP[DP ] scio ] 70 0.0  
[NP patrem ] 30 0.0  
[VP venire ] 30 0.0
```

regina amat hominem qui venit

```
*** no analysis [TP [VP[NP regina ]][DP ] amat [NP hominem ]]] -55 0.0  
[DP qui ] 0 0.0
```

[VP[DP] venit] 70 0.0

Caesar misit legiones quae in Gallia erant

*** no analysis [TP [VP[DP Caesar]][DP] misit]] 10 0.0
[NP legiones] 30 0.0
[DP quae] 0 0.0
[PP in [DP Gallia]] 15 0.0
[VP[DP] erant] 70 0.5

suadeo tibi ut legas

*** no analysis [TP [VP[DP] suadeo [DP tibi]]] -10 0.0
[CP ut] 30 0.0
[VP[DP] legas] 70 0.0

dum haec a Caesare geruntur, Galli impetum parant

*** no analysis [CP dum] 30 0.0
[DP haec] 0 0.0
[VP a [DP Caesare]] -55 0.0
[VP[DP] faciuntur [NP Galli [DP impetum]]] 81 0.0
[DP parant] 101 0.0

cum in Italiam proficisceretur Caesar, legiones in Galliam misit

*** no analysis [PP cum] 0 0.0
[PP in [DP Italiam]] 15 0.0
[VP[DP] proficisceretur [DP Caesar]] -65 0.0
[NP legiones] 30 0.0
[PP in [DP Galliam]] 15 0.0
[VP[DP] misit] 70 0.0

partibus factis, verba fecit leo

*** no analysis [TP [VP[NP partibus]][FP[DP] [VP factis]]] 135 0.0
[NP verba] 30 0.0
[VP[DP] fecit [NP leo]] -35 0.0

Remarques : il reste encore des progrès à faire pour pouvoir analyser les proposition subordonnées.

5.2.9 interrogation et négation

nonne amicus meus es ?

*** no analysis [AdvP nonne] 30 0.0
[NP amicus] 30 0.0
[DP meus] 0 0.0
[VP es] 30 0.5

Vidistine Romam ?

```
[TP [VP[DP vidistine ] [NP Romam ] ]] 201 0.0  
[TP [VP[NP[AP vidistine ] Romam ] ]] 216 0.0  
[NP[AP vidistine ] Romam ] 221 0.0
```

non legit

```
[TP [VP[AdvP non ] [DP ] legit ]] 185 0.0  
[VP[AdvP non ] [DP ] legit ] 200 0.0
```

ne legat

```
*** no analysis [AdvP ne ] 30 0.0  
[VP legat ] 30 0.0
```

ludit neque legit

```
[TP [VP [ConjP[VP[DP ] ludit ] neque [VP[DP ] legit ]]]]
```

jam non veniet

```
*** no analysis [AdvP jam ] 30 0.0  
[AdvP non ] 30 0.0  
[VP veniet ] 30 0.0
```

nemo venit

```
[TP [VP[NP nemo ] [DP ] venit ]] 60 0.0
```

6 Conclusion

Au cours de ce travail, nous nous sommes penchés sur l'analyse syntaxique et les problèmes qu'elle soulève en ce qui concerne la langue latine. Certains défis ne sont pas encore parfaitement maîtrisés, comme la figure de l'hyperbate, pourtant extrêmement courante dans la prose classique. D'autre part, les difficultés liées au lexique et à la richesse morphologique du latin ont été résolues de façon à couvrir le vocabulaire le plus complètement possible, bien que l'on ne puisse pas toujours le faire mécaniquement avec le programme FipsMorpho.

Afin de développer une version de Fips pour traiter la langue latine, nous avons constitué un lexique et adapté la grammaire déjà existante.

Puisque la précision des informations entrées dans le lexique détermine l'efficacité de l'analyse, une grande attention a donc été portée au lexique et à la morphologie. Ainsi, à l'exception de quelques temps verbaux, toutes les formes morphologiques de chaque catégorie syntaxique peuvent être représentées.

L'examen des tests pratiqués dans Fips au chapitre précédent nous montre que l'analyse est plutôt efficace en ce qui concerne les phrases très simples comprenant sujet, verbe et attribut ou complément d'objet direct. Cependant, dès que l'on passe à un niveau de complexité légèrement supérieur, les problèmes se font plus fréquents.

Enfin, un point reste à mentionner : comment mettre à profit le logiciel ? L'aide à la traduction serait une réponse à cette question. Les textes classiques bénéficient déjà de multiples traductions et commentaires. Il pourrait alors être intéressant de se tourner également vers les nombreuses sources historiques médiévales dont aucune traduction n'a jamais été publiée faute de moyens.

7 Annexe : le fichier de morphologie latin

Cf. le document joint.

8 Bibliographie

8.1 Ouvrages

ABEILLÉ A., *Les nouvelles syntaxes : grammaires d'unification et analyse du français*, Paris : A. Colin, 1993.

AHO A., ULLMAN J. D., *The Theory of Parsing, Translation and Compiling*, Englewood Cliffs N.J. : Prentice-Hall, 1972-197, 2 vol.

BALDI P., CUZZOLIN P. (édd.), *New Perspectives on Historical Latin Syntax*, Berlin : Mouton de Gruyter, 2009- 2010, 3 vol.

BOUILLON P. et alii., *Traitement automatique des langues naturelles*, Bruxelles : Duculot, 1998.

CART A. et alii., *Grammaire latine*, Paris : Nathan, 1955.

CHOMSKY N., *Syntactic Structures*, The Hague [etc.] : Mouton, 1966⁶ (1957¹).

CHOMSKY N., *Aspects of the Theory of Syntax*, Cambridge Mass. : M.I.T. Press, 1965.

CHOMSKY N., *Lectures on Government and Binding*, Dordrecht ; Cinnaminson : Foris Publ., 1981.

CHOMSKY N., *The Minimalist Program*, Cambridge Mass. ; London : MIT Press, 1996² (1995¹).

CULICOVER P., JACKENDOFF R., *Simpler Syntax*, Oxford ; New York : Oxford University Press, 2005.

DEVINE A. M., STEPHENS L. D., *Latin Word Order. Structured Meaning and Information*, New York : Oxford University Press, 2006.

ERNOUT A., THOMAS F., *Syntaxe latine*, [Paris] : Klincksieck, 2002 (1951¹).

HAEGEMAN L., *Introduction to Government and Binding Theory*, Oxford ; Cambridge Mass. : B. Blackwell, 1991.

LAEZLINGER C., *Initiation à la Syntaxe formelle du français : le modèle Principes et Paramètres de la*

Grammaire Générative Transformationnelle, Berne : P. Lang, 2003.

MAROUZEAU J., *L'ordre des mots dans la phrase latine*, Paris : Champion : Les Belles Lettres, 1922-1949, 3 vol.

MATTHEWS C., *An Introduction to Natural Language Processing through Prolog*, London [etc.] : Longman, 1998.

NOTARI C., *Chomsky et l'ordinateur : approche critique d'une théorie linguistique*, Toulouse : Presses universitaires du Mirail, 2009.

OUHALLA J., *Introducing Transformational Grammar. From Rules to Principles and Parameters*, London ; New York [etc.] : E. Arnold, 1994.

PINKSTER H., *Latin Syntax and Semantics*, London ; New York : Routledge, 1990.

ROBERTS I., *Comparative Syntax*, London ; New York [etc.] : Arnold, 1997.

SAG I., WASOW T., *Syntactic Theory : A Formal Introduction*, Stanford, Calif. : Center for the Study of Language and Information, 1999.

SERBAT G., *Les structures du latin : avec un choix de textes traduits et annotés de Plaute aux Serments de Strasbourg*, Paris : Picard, 1994⁴ (1975¹).

SMITH N., *Chomsky : Ideas and Ideals*, Cambridge [etc.] : Cambridge Univ. Press, 1999.

TEN HACKEN P., *Chomskyan Linguistics and its Competitors*, London : Equinox, 2007.

TOURATIER C., *Grammaire latine : introduction linguistique à la langue latine*, Paris : Ed. Sedes, 2008.

VAN WIJNGAARDEN A. (éd.), *Draft Report on the Algorithmic Language. Algol 68*, (Amsterdam), Mathematisch Centrum, 1968.

WEBELHUTH G. (éd.), *Government and Binding Theory and the Minimalist Program : Principles and Parameters in Syntactic Theory*, Oxford ; Cambridge Mass. : Blackwell, 1995.

WEHRLI E., *L'analyse syntaxique des langues naturelles : problèmes et méthodes*, Paris [etc.] : Masson, 1997.

8.2 Articles

CARROLL J., « Parsing », in R. MITKOV (éd.), *The Oxford Handbook of Computational Linguistics*, Oxford [etc.] : Oxford Univ. Press, 2003, pp. 233-248.

KOSTER C.H.A., « Constructing a Parser for Latin », *Lecture Notes in Computer Science* 3406 (2005), pp. 48-59.

RAJMAN M., VILLEMONTÉ DE LA CLERGERIE E., « Petit panorama des approches en analyse syntaxique », *TAL* 44.3 (2003), pp. 7-14.

SAINT-DIZIER P., SZPAKOWICZ S., « Logic programming, logic grammars, language processing », in P. SAINT-DIZIER, S. SZPAKOWICZ (édd.), *Logic and Logic Grammars for Language Processing*, New York [etc.] : Ellis Horwood, 1990, pp. 9-27.

WEHRLI E., « Un modèle multilingue d'analyse syntaxique », in AUCLIN A., FILLIETTAZ L. (édd.), *Structures et discours : mélanges offerts à Eddy Roulet*, Québec : Ed. Nota bene, 2004, pp.

WEHRLI E., « Fips, a Deep Linguistic Multilingual Parser », in *Proceedings of the 5th Workshop on Important Unresolved Matters*, Ann Arbor, 2005, pp. 120-127.

8.3 Sites internet⁴⁷

BONAMI O., « Argument », in D. Godard, L. Roussarie et F. Corblin (édd.), *Sémanticlopédie : dictionnaire de sémantique*, GDR Sémantique & Modélisation, CNRS, 2006, <http://www.semantique-gdr.net/dico/>.

SAINT-DIZIER P., « Rôles thématiques », in D. Godard, L. Roussarie et F. Corblin (édd.), *Sémanticlopédie : dictionnaire de sémantique*, GDR Sémantique & Modélisation, CNRS, 2006, <http://www.semantique-gdr.net/dico/>.

WEHRLI E., NERIMA L., « L'analyseur syntaxique Fips », Journée thématique ATALA, 11th International Conference on Parsing Technologies, 2009,

⁴⁷ Consultés le 30.06.2011.

Un analyseur syntaxique pour le latin classique
Prof. E. Wehrli
Prof. F. Bavaud

Maîtrise universitaire ès Lettres en IMM
Elisa Nury
Septembre 2011

<http://alpage.inria.fr/iwpt09/atala.fr.html>

Andromeda : <http://www.collatinus.org/andromeda/>.

Grammatica Latina : <http://www.agfl.cs.ru.nl/lat/index.html>.

Latin parser : <http://arts.cuhk.edu.hk/~lha/latin/>.

Words : (<http://www.archives.nd.edu/cgi-bin/words.exe>).