MATHÉMATIQUES ET SCIENCES HUMAINES

MORRIS SALKOFF

# Analyse syntaxique automatique utilisant une grammaire en chaîne (« string grammar »)

*Mathématiques et sciences humaines*, tome 35 (1971), p. 19-30

<http://www.numdam.org/item?id=MSH_1971__35__19_0>

*Math. Sci. hum.* (9e année, n° 35, 1971, p. 19-30)

# ANALYSE SYNTAXIQUE AUTOMATIQUE UTILISANT UNE GRAMMAIRE EN CHAÎNE (STRING GRAMMAR)[1]

par

Morris SALKOFF [2]

*Lorsqu'on désire analyser une phrase à l'aide des règles explicites, il est alors nécessaire de substituer à l'analyse intuitive un programme d'analyse explicite, c'est-à-dire un algorithme dont une des tâches serait de donner toutes les analyses syntaxiques de la phrase analysée (si la phrase « hors contexte » est ambiguë, il y a plusieurs analyses possibles) et ensuite de choisir l'analyse qui soit compatible avec le contexte. Z. S. Harris a proposé le modèle des « grammaires en chaîne ». Une grammaire en chaîne est un exemple de programme d'analyse de toutes les phrases d'une langue donnée (ici le français) ; elle fournit une analyse exprimée dans une métalangue abstraite par rapport à la langue objet. C'est cependant une analyse qui reste très liée aux structures de surface.*

*J. P. Desclés*

## 1. INTRODUCTION

La plupart des grammaires utilisées jusqu'à maintenant dans des programmes d'analyse syntaxique sur ordinateur sont basées peu ou prou sur la grammaire en constituants immédiats (IC) développée aux États-Unis dans les années quarante par l'école de linguistique distributionnelle[3]. Les analyses produites par ces programmes laissent à désirer, en particulier en ce qui concerne la justesse des analyses. En effet, ce type de grammaire (IC) présente un inconvénient majeur : la difficulté de l'incorporation dans la grammaire des phénomènes grammaticaux discontinus, représentés, par exemple, par l'accord grammatical entre sujet et verbe :

(1.1) *a*) *Les hommes sont ici*
*b*) **Les hommes est ici*

ou l'accord entre le sujet et le prédicat d'*être* :

(1.2) *a*) *Cet homme est courageux*
*b*) **Cet homme est courageuse.*

Ici, ce sont les morphèmes singulier-pluriel (ou féminin-masculin) qui sont découpés en deux parties non contiguës : une moitié sur le sujet et l'autre sur le verbe (ou sur le prédicat).

---

1. Cet article a été réalisé en partie avec l'aide de la DGRST, contrat n° 69.01.591 ; l'article représente le développement d'une conférence donnée en août 1970 à l'École d'Été de Pise, dirigée par A. Zampolli.

2. CNRS, Laboratoire d'Automatique Documentaire et de Linguistique, 23, rue du Maroc, Paris, 19e.

3. Voir par exemple, le programme et la grammaire de S. Kuno. L'équivalence entre les diverses grammaires utilisées dans ce genre de travail est démontrée dans Gross [2].

On représente, dans les grammaires IC, les structures syntaxiques par des règles de ré-écriture imbriquées les unes dans les autres :

(1.3) *a*) $Ph \rightarrow GN_1\ GV$ où $Ph$ = phrase
*b*) $GV \rightarrow V\ GN_2$ $GN$ = groupe nominal
*c*) $GV \rightarrow V\ P\ GN_2$ $P$ = préposition
. . .
. . .
. . .

Un programme d'analyse, ou *analyseur syntaxique*, utilisant le modèle des grammaires IC a la structure d'un automate à mémoire en pile [1]. Dans un tel analyseur, les niveaux successifs de développement de la phrase (de la règle pour *Ph* dans (1.3)) sont stockés les uns sur les autres dans la mémoire centrale de l'ordinateur. Au moment donc où il faut vérifier un accord comme (1.1) ou (1.2), les deux éléments assujettis à cet accord — $GN_1$ dans (1.3) *a*) et $V$ dans (1.3.) *b*) — se trouvent séparés par au moins un niveau dans la mémoire en pile. En réalité, une grammaire IC élaborée avec plus de détails aurait davantage de règles intervenant entre (1.3) *a*) et (1.3) *b*) ; il serait par conséquent très difficile de « remonter » de $V$ à $GN_1$ afin de vérifier l'accord. En effet, il n'est guère commode de manier par programme les divers niveaux de la mémoire en pile.

Notons cependant, qu'il n'est pas impossible d'examiner les niveaux d'une grammaire IC, mais les difficultés de programmation qu'il faut surmonter sont telles qu'en pratique un maniement de ce genre n'a jamais pu être incorporé dans cet analyseur. Comme on le verra plus loin, le fait de ne pouvoir vérifier des accords du genre (1.1) et (1.2) amène l'analyseur à sortir de fausses analyses des phrases étudiées.

Afin d'éviter ces problèmes, Z. S. Harris a défini un nouveau type de grammaire, la grammaire en chaîne [2] (en anglais : *string grammar*). Dans cette grammaire, les éléments grammaticaux ayant un lien entre eux figurent dans la même structure syntaxique (chaîne) de la grammaire. On évite ainsi les déboires décrits à propos de la grammaire IC : $GN_1$ et $V$, dans l'exemple étudié plus haut, seront membres d'une même chaîne et se trouveront donc au même niveau. L'accord sera alors facile à vérifier, ainsi que tous les autres phénomènes grammaticaux entre éléments discontinus. En effet, ces derniers figurent toujours dans une même chaîne.

L'ensemble de ces contraintes grammaticales constitue une partie importante d'une grammaire en chaîne [3] ; on verra dans 3, qu'un analyseur utilisant une grammaire en chaîne ainsi construite ne sortira aucune fausse analyse. Néanmoins, des problèmes difficiles subsistent ; l'analyseur n'est donc pas encore un outil de travail parfait (voir 5).

## 2. ESQUISSE D'UNE GRAMMAIRE EN CHAÎNE

L'hypothèse de base est la suivante : il est possible de décomposer chaque phrase correcte de la langue en une *chaîne centrale* accompagnée de zéro, un ou plusieurs *ajouts*. Une *chaîne centrale* est une séquence de catégories grammaticales telle que :

(*i*) Cette séquence, où l'on a remplacé chaque catégorie grammaticale par un mot appartenant à cette catégorie, est une phrase de la langue ;

1. Voir Hays, D.G.
2. Harris, Z. S. (voir bibliographie).
3. Un abrégé d'une grammaire en chaîne de l'anglais est donné dans le livre de Harris, cité dans la bibliographie. Pour une grammaire plus détaillée, l'attention du lecteur est attirée sur deux rapports du Linguistic String Project à New York, dans la bibliographie.

(*ii*) Aucune catégorie grammaticale ne peut être enlevée sans que la chaîne centrale, privée de cette catégorie, perde son statut de phrase de la langue.

Un *ajout* est une séquence de catégories grammaticales qui a sa structure propre et un statut défini de la façon suivante :

(*i*) L'ajout peut être inséré à gauche ou à droite d'un élément de la chaîne centrale, ou d'un ajout ;

(*ii*) L'ajout est facultatif dans la chaîne où il figure.

L'insertion d'un ajout dans une chaîne ne change pas le statut de cette chaîne.

Grosso modo, on peut considérer qu'une chaîne centrale est le squelette d'une phrase et les ajouts sont les modifieurs des catégories grammaticales de la chaîne centrale. Ainsi dans la phrase :

(2.1) *Le grand camion rouge roule très rapidement*

il y a une chaîne centrale *Le camion roule* — une occurrence du type $N\,tV$[1] — avec trois ajouts : *grand*, un adjectif ajout à gauche de $N$ ; *rouge*, un adjectif ajout à droite de $N$ ; et *rapidement*, un adverbe ajout à droite de $tV$. Ce dernier ajout contient lui-même un ajout à gauche, l'adverbe *très*. Notez que les ajouts sont facultatifs dans la chaîne où ils figurent, c'est-à-dire, leur présence n'est pas nécessaire pour que cette chaîne soit bien formée : la chaîne centrale *le camion roule*, sans aucun ajout, est aussi bien formée que (2.1). De la même façon, *très* est facultatif dans l'ajout où il figure.

Définissons maintenant quelques-uns des ajouts de la grammaire.

Soit :

(2.2) $Y = X_1\ X_2\ \ldots\ X_n$ une chaîne centrale de la grammaire.

J'utilise les symboles $g_X$ et $d_X$ pour noter un ajout à gauche ou à droite de la catégorie grammaticale $X$ :

(2.3) — $g_N$ Un ajout à gauche du nom $N$, ou un ajout inséré à gauche d'un autre $g_N$. Ainsi, dans la séquence *élégante, belle femme, belle* est un $g_N = A$, ajout à gauche de *femme*, et *élégante* un $g_N = A$, ajout inséré à gauche du premier ajout.

— $d_N$ Un ajout à droite du nom $N$, ou un ajout inséré à droite d'un autre $d_N$. Ainsi, dans la séquence *un vin fin de Bordeaux*, *fin* est un $d_N = A$, ajout de *vin* ; *de Bordeaux* est un $d_N = PN$, ajout de *vin*, inséré à droite du premier $d_N$.

— $g_A$ Un ajout à gauche de l'adjectif. Ainsi, dans *une très élégante femme*, *très* est un $g_A = D$ de *élégante*.

— $d_A$ Ajout à droite de l'adjectif. Dans *responsable de sa perte, de sa perte* est un $d_A = PN$ de *responsable*.

— $g_V$,$d_V$ Des ajouts à gauche (à droite) du verbe (et des différentes formes verbales). Dans *Il lit bien*, *bien* est un $d_V = D$ de $tV = lit$ ; dans *Il l'a bien lu*, *bien* est un $g_V = D$ de $Vé = lu$.

— $a_Y$ (ou *) Un ajout de la chaîne centrale (symbolisé aussi par l'astérisque). Il peut être inséré à gauche ou à droite de chacun des éléments d'une chaîne centrale. Ainsi, dans la chaîne centrale $Y = N\,tV$ = *Le camion roule*, l'ajout *aujourd'hui* peut être inséré à gauche de N *(Aujourd'hui le camion roule)* entre $N$ et $tV$ *(Le camion, aujourd'hui, roule)* ou bien à droite de $tV$ *(Le camion roule aujourd'hui)*. Autres ajouts $a_Y$ (ou *) : $D$ = *naturellement évidemment, ...* ; $PN$ = *de cette façon, à ce moment, ...*, etc.

Regardons maintenant de plus près les diverses formes possibles de la chaîne centrale. Notons par *C1* la chaîne centrale en laquelle peut être décomposée chaque phrase d'assertion de la langue (c'est-à-dire, une phrase terminée par un point).

---

1. J'utilise les symboles suivants : $N$ (nom) ; $tV$ (verbe fléchi) ; $V$ (verbe) ; *Vé* (participe passé) ; $A$ (adjectif) ; $D$ (adverbe) ; $P$ (préposition) ; $T$ (article) ; $R$ (pronom). L'article ne figure pas dans mes formules ( $NtV$ et non pas $TNtV$) bien que dans certains cas, il ne soit pas facultatif : * *camion roule*. Pour une discussion de ce problème, voir Harris, Z. S. et Sager, N.

Voici quelques-unes des formes possibles pour *C1* :

(2.4)

| Formule de *C1* | Exemples |
|---|---|
| *N tV* | *Pierre dort. Une solution existe* |
| *N tV N* | *Marie porte un chapeau* |
| *N tV PN* | *Paul dépend de l'administration* |
| *N tV N PN* | *Le professeur base sa théorie sur cette hypothèse* |
| . . | |
| . . | |
| . . | |

Il est à remarquer que chaque élément est indispensable à la chaîne. Ainsi, si l'on enlève le deuxième *N* dans la formule *N tV N*, nous n'obtenons plus une phrase : **Marie porte.*

Une certaine sous-classe de verbes comme *dire*, penser, ..., peut être suivie de *que* suivi lui-même de l'une quelconque des formules de (2.4) :

(2.5)

| Formule de *C1* | Exemples |
|---|---|
| *N dit que N tV* | *(que Pierre dort)* |
| *N dit que N tV N* | *(que Marie porte un chapeau)* |
| . . . . . | |
| . . . . . | |
| . . . . . | |
| *N dit que C1* | |

Nous voulons éviter de répéter après *dire*, dans (2.5), toutes les formules de (2.4) ; la séquence qui suit *dire* dans (2.5) est une autre chaîne, *que C1*, et non plus une suite de catégories grammaticales. En nous autorisant à avoir, à l'intérieur d'une chaîne, un élément qui soit lui-même aussi bien une chaîne qu'une catégorie grammaticale (ou suite de catégories), on tient compte dans la grammaire de la récursivité. Une phrase telle que :

*Marie dit que Jean a avoué que ... Pierre dort*

sera alors décrite comme une occurrence du type de (2.5) avec plusieurs chaînes centrales imbriquées les unes dans les autres après *dit*. A partir de (2.4) et (2.5) on peut définir une chaîne Ω (l'objet) qui prend une des valeurs :

(2.6) $\Omega$ = *N/PN/NPN/ ... / que C1/ ...* (et encore d'autres).

(Cette formule signifie que Ω prend soit la valeur *N*, soit la valeur *PN*, etc.) C'est la chaîne Ω qui figure maintenant après *tV* dans la formule générale pour *C1*.

De la même manière, on voit que *N* n'est pas seul à pouvoir précéder *tV* dans *C1*. Ainsi, entre autres, la chaîne *que C1* peut apparaître comme sujet, pour une sous-classe de verbes :

(2.7) *Que C1 tV* Ω *(Que Pierre dort n'est pas un problème).*

On définit donc la chaîne Σ (sujet) qui prend une des valeurs :

(2.8) Σ = *N/R/que C1/ ...* (et quelques autres).

La chaîne centrale prend maintenant la forme

(2.9) *C1* = Σ *tV* Ω

c'est-à-dire, une des chaînes de Σ (dans 2.8), suivie du verbe fléchi *tV*, puis une des chaînes de Ω (2.6) [1].

1. Il est clair que toute valeur de Σ n'est pas possible pour n'importe quelle valeur de *tV* et de Ω : * *Que Pierre dort porte un chapeau.* C'est précisément le rôle des restrictions, qui seront discutées un peu plus loin : empêcher qu'une séquence illicite ne figure dans l'analyse fournie par le programme.

Si l'on tient compte des ajouts définis dans (2.3), on peut développer *C1* jusqu'à obtenir la forme suivante, plus commode pour la discussion [1] :

(2.10) $C1 = {}^*\Sigma\ {}^*\ g_V\ tV\ d_V\ {}^*\ \Omega\ d_V\ {}^*$

L'ajout à droite du verbe, $d_V$, y figure deux fois parce qu'il peut apparaître tout de suite après le verbe : *Pierre lit rapidement tout ce qui l'intéresse*, ou bien après l'objet $\Omega$ : *Pierre a lu ce roman rapidement.*

## 3. LES RESTRICTIONS

On a déjà vu que toute valeur d'une variable dans la formule donnée pour *C1*, c'est-à-dire, toute valeur d'une catégorie grammaticale, n'est pas une valeur permise. Ainsi dans la chaîne centrale (2.5) :

*N pense que C1,*

le sujet *N* doit être « humain »:

(3.1) *Pierre pense que C1* ; * *La table pense que C1*

Ou encore, dans la chaîne centrale :

(3.2) *N est que C1*

*N* peut être *fait*, *problème*, etc., mais non pas *livre : *Le livre est que Pierre dort.*

Pour définir rigoureusement les sous-classes, il est nécessaire de trouver des critères syntaxiques qui constitueraient des cadres dans lesquels pourraient s'insérer les membres de la catégorie grammaticale en question satisfaisant à la condition du critère, et seulement ceux-là. Afin que ces cadres soient syntaxiques et ne fassent appel au sens des mots [2], il ne doit y figurer que les chaînes de la grammaire, des constantes linguistiques (comme le verbe *être*, certaines prépositions comme *à* et *de*, etc.), et d'autres sous-classes, elles aussi déjà définies ainsi. La formule (3.2) est un tel cadre : elle ne contient que le verbe *être* et la chaîne *que C1*. Les noms qui peuvent figurer dans (3.2) forment une sous-classe $N_p$ de noms « opérateurs », c'est-à-dire, des noms qui peuvent servir de support à une chaîne du type *que C1*. La formule (3.1), par contre, n'est pas assez rigoureuse pour définir la sous-classe $N_h$ (les noms « humains »), puisque la sous-classe de verbes comme *penser*, *dire*, ..., est trop large pour que la classe de noms satisfaisant à (3.1) soit homogène pour le reste de la grammaire. Un meilleur cadre pour $N_h$ serait le suivant :

(3.3) $N_h$ *P qui*

Ainsi, *l'homme* fait partie de la sous-classe $N_h$ : *l'homme à qui (j'ai parlé)*, et non pas *table : * la table à qui (j'ai donné un coup de pied)* [3].

Les sous-classes ainsi définies pourraient être incorporées dans la grammaire de deux manières différentes. Dans la première méthode, on évite les séquences illicites, comme celles citées au-dessous de (3.1) et (3.2), en écrivant de nouvelles chaînes de la grammaire qui contiennent les sous-classes uniquement aux endroits où elles peuvent figurer, d'après le critère syntaxique utilisé dans leur définition.

---

1. Le développement complet de la chaîne *C1* est encore plus complexe que ne le laisse apparaître (2.10). Une grammaire en chaîne du français plus détaillée est en préparation.

2. Le problème de la définition rigoureuse de ces sous-classes n'est pas facile, et à l'heure actuelle n'est pas encore résolu. Il est parfois peu évident de savoir comment il conviendrait de déterminer telle ou telle sous-classe à l'aide des seuls critères syntaxiques. Pourtant, on hésite à les déterminer avec le recours de critères sémantiques en faisant intervenir le sens des mots ou encore l'avis d'un témoin de la langue sur le sens. Ces avis, en effet, varient d'une personne à une autre ce qui rend très difficile — sinon impossible — le travail lexicographique de la classification des mots de la langue selon leur appartenance ou non aux sous-classes ainsi définies.

3. La définition de la sous-classe $N_h$ est en réalité, beaucoup plus difficile et le cadre proposé n'est utile que pour les êtres humains. Le cadre (3.3) doit donc être considéré comme une première approximation.

La sous-classe $N_p$ (noms opérateurs), par exemple, définie par (3.2), peut être incorporée dans la grammaire en divisant la chaîne *C1* en deux :

$$C1_a = N\ tV\ \bar{\Omega} \qquad\qquad C1_b = N_p\ \ tV_e\ que\ C1$$

($V_{\hat{e}}$ est une sous-classe de verbes ayant un comportement semblable à *être : rester, demeurer...*) Une séquence illicite comme * *Le livre est que Pierre dort* est maintenant impossible, puisque seuls les noms $N_p$ figurent avec $tV_e$ et l'objet *que C1* ($\bar{\Omega}$ ne contient pas la chaîne *que C1*).

Le coût de cette première méthode est assez élevé : pour plusieurs dizaines de sous-classes de noms, verbes, adjectifs, etc., le nombre de nouvelles chaînes nécessaires à leur incorporation dans la grammaire gonflerait celle-ci de manière à ce qu'il devienne très difficile de la manier.

La deuxième méthode consiste à incorporer des sous-classes dans la grammaire à l'aide d'un ensemble de restrictions sur les chaînes. Rappelons que chaque chaîne de la grammaire s'écrit sous la forme d'une suite d'options, comme dans (2.6) et (2.8). Au cours de l'analyse d'une phrase, le programme choisit une des options dont un exemple peut être construit avec les mots de la phrase. Lorsqu'ont été associées les catégories grammaticales des mots à celles requises par l'option de la chaîne en question, une restriction attachée à cette option peut vérifier que la suite des sous-classes des catégories des mots ne constitue pas une séquence défendue, comme celles citées au-dessous de (3.1) et (3.2). Si une séquence illicite a effectivement été construite, le programme défait cette branche de l'arbre d'analyse.

Je m'efforcerai maintenant de démontrer que pour une grammaire en chaîne, il est aisé de tenir compte des phénomènes de discontinuité ; associée au jeu de restrictions du type décrit ci-dessus, la démonstration nous amènera à dire que le programme ne donne jamais une fausse analyse.

Grâce à la forme des chaînes et à la définition de l'adjonction, il est évident que deux éléments entre lesquels il existe une contrainte linguistique sont :

*a*) Soit deux éléments d'une même chaîne ;

*b*) Soit un élément et son ajout à gauche ou à droite.

Deux possibilités se présentent :

— Ou bien les éléments en question se trouvent séparés par un autre élément linguistique, mais sont membres de la même chaîne (cas *a*),

— Ou bien ils sont voisins puisqu'ils sont liés comme une catégorie et son ajout immédiatement à gauche ou à droite (cas *b*).

Dans les deux cas, *les deux éléments sont contigus par rapport à la chaîne qui les contient.* Le problème des éléments discontinus qu'on avait dans les grammaires IC (1.3), n'existe plus pour une grammaire en chaîne.

Examinons maintenant quelques contraintes entre deux éléments liés comme dans *a*) ou *b*). Nous verrons que l'addition à la grammaire d'une restriction qui tienne compte de la contrainte est toujours faisable (et facile à faire), et aussi, le programme qui utilise la grammaire en chaîne ne sort pas de fausses analyses basées sur une infraction à la contrainte en question.

### *a*1. *L'accord grammatical entre sujet et verbe*

La phrase anglaise suivante :

(3.4)*a* *We can use the original data and these results too*
*(Nous pouvons nous servir des données du départ et (de) ces résultats aussi)*

ne peut être analysée comme un exemple de :

(3.4)*b* *C1 and C1'* où *C1* = Σ *(We) tV (can use)* Ω *(N = the original data)*
et *C1'* = Σ *(these) tV (results)* * *(too)*

puisqu'il n'y a pas accord entre *these* et le verbe *results* : *these result*, mais **These results* (où *results* est un verbe fléchi).

Cette fausse analyse peut être évitée en ajoutant une restriction à la chaîne *C1* (2.10) qui impose la règle : si le verbe est singulier (pluriel), alors le sujet n'est pas pluriel (singulier) [1]. La même contrainte se voit bien dans la phrase française qui commence par :

*Les deux avions des forces armées...*

La restriction qui vient d'être citée empêchera le programme de s'embarquer sur la piste de la fausse analyse dans laquelle Σ = *Les deux*, *tV* = *avions*, et Ω = *N* = *des forces armées...* L'erreur ici provient du manque d'accord en personne entre Σ et *tV* : *Les deux avaient*, **Les deux avions* (où *avions* est ici le temps passé d'*avoir*). Cela évitera ou bien une fausse analyse ou bien un long détour par ce faux chemin et peut représenter une économie considérable du temps de calcul.

*a2. Compatibilité des sous-classes*

L'exemple (3.1) peut sembler inutile pour les besoins pratiques d'un programme d'analyse automatique, puisqu'on peut s'imaginer qu'une séquence telle que *la table pense que...* ne surviendra jamais dans une phrase correcte de la langue. Pourtant, le programme peut découper une phrase correcte en une séquence de chaînes incorrectes (en construisant une fausse analyse) dans laquelle une contrainte du type (3.1) est enfreinte.

Il y a une sous-classe de verbes — appelons la *V1* — qui donne un passif en *de* [2] :

(3.5)*a* *Toute la classe admire le professeur*
*b* *Le professeur est admiré de toute la classe*

On retrouve cette dernière phrase aussi dans une séquence telle que :

*c* *... le professeur admiré de toute la classe...*

Considérez maintenant une séquence :

(3.5)*d* *... (qui était) l'incarnation admirée et haïe de la révolte de la jeunesse*

L'analyse correcte montrera *de la révolte* comme $d_N = PN$, ajout de *l'incarnation (l'incarnation... de la révolte)*, avec *admirée et haïe* comme $d_N = A$, aussi sur *l'incarnation* [3].

Mais, puisque *admirer* et *haïr* appartiennent à la sous-classe *V1*, le programme pourrait également trouver une fausse analyse dans laquelle (3.5)*d* est découpée comme (3.5)*c*. En comparant ces deux, on voit *l'incarnation*, puis *admirée et haïe* et finalement *de la révolte* dans les positions de *le professeur*,

---

1. Les entrées lexicales pour chaque mot de la phrase doivent alors faire figurer non seulement les catégories grammaticales auxquelles appartient le mot, mais aussi toutes les sous-classes de chacune de ces catégories dont ce mot est un membre. Ainsi, l'entrée pour *thèse* contiendrait *R* (pronom), sous-classe pluriel, tandis que celle de *results* serait *N*, sous-classe pluriel (et peut-être encore d'autres) et *V*, sous-classe singulier (et encore d'autres). En effectuant la vérification nécessaire, la restriction tenant compte de l'accord grammatical se référerait à ces sous-classes.

2. La sous-classe *V1* comprend *admirer*, *haïr*, *choisir*, ... et beaucoup d'autres. Exemple d'un verbe qui n'est pas membre de *V1* : *trouver*. Ainsi, *Paul trouve son argent* ne donne pas : * *Son argent est trouvé de Paul.* Voir Gross [1], § 2.1.1. (p. 100).

3. Cette analyse montre donc que la séquence dans (3.5)d provient d'une phrase *(Quelqu'un) admire et hait l'incarnation de la révolte.*

*admiré*, et *de toute la classe*, respectivement. Dans cette analyse, la séquence (3.5)*d* serait la transformée d'une phrase

(3.6.) * *La révolte admire et haït l'incarnation*

qui est incorrecte, puisque *admirer* et *haïr* ne prennent que $N_h$ en position sujet. *La révolte* est donc un sujet défendu pour ces verbes.

Comme pour (3.4), la façon d'écarter cette fausse analyse consiste à ajouter une restriction à la chaîne

(3.7) $d'_N$ = *Vé de* $N_1$

qui vérifie que, si *Vé* appartient à la sous-classe *V1*, alors $N_1$ n'est pas dans une sous-classe défendue comme sujet de $V$ [1]. Notez que le nom et le verbe liés comme $\Sigma$ et $V$ se trouvent tous deux dans la même chaîne ($d'_N$), comme $\Sigma$ et $tV$ dans *C1* (2.10).

Dans le cas étudié, la restriction dans $d_N$ trouverait que $N_1$ *(révolte)* est dans la sous-classe non-$N_h$ (non-humain), qui est une des sous-classes de sujet défendues pour *Vé (admiré)*. L'échec de cette restriction obligera le programme à défaire la fausse analyse.

*a*3. *Éléments discontinus*

Voici un cas de deux éléments entre lesquels il existe une contrainte de dépendance, mais qui se trouvent très éloignés l'un de l'autre dans la chaîne où ils figurent. Une phrase comme

(3.8)*a* *Plus de gens ont commandé du fromage qu'on ne s'y attendait*

peut être analysée comme une occurrence de *C1* dans laquelle

(3.8)*b* $\Sigma$ = *plus de gens ;* $tV$ = *ont commandé ;* $\Omega$ = *du fromage ;* * = *qu'on ne s'y attendait.*

La proposition *qu'on ne s'y attendait* est casée dans l'ajout à la phrase (noté *) qui paraît après l'objet $\Omega$. Cette proposition ne peut apparaître dans *C1* à cet endroit que si l'un des éléments $\Sigma$, $tV$ ou $\Omega$ contient un « marqueur » du degré comparatif comme *plus* ou *moins*:

(3.8)*c* *Les gens ont plus souvent commandé du fromage qu'on ne s'y attendait*
*Les gens ont commandé plus du fromage qu'on ne s'y attendait*
* *Les gens ont commandé du fromage qu'on ne s'y attendait*

Pour tester toutes ces possibilités, et vérifier que la proposition du type *qu'on ne s'y attendait* est effectivement précédé par *plus (moins)* dans une des positions indiquées, il suffit (afin d'éviter une fausse analyse basée sur l'infraction illustrée dans la dernière phrase de (3.8) *c*) d'ajouter une restriction à *C1* allant dans ce sens. Le couple d'éléments liés — l'ajout * plus l'un des $\Sigma$, $tV$ ou $\Omega$ — bien qu'ils se trouvent à une certaine distance l'un de l'autre, sont dans la même structure de la grammaire, la chaîne centrale *C1*.

*b Entre N et* $d_N$

Nous avons déjà eu un exemple dans la contrainte entre $d_N$ et $N$ (voir (3.7) et la note 3, page 25). Voici un autre exemple :

L'adjectif $A$ dans une séquence comme

(3.9) $N_1$ *de* $N_2$ $A$

1. Utilisant la même méthode que celle décrite dans la note 1 (page 25), chaque verbe contiendra dans son entrée lexicale, une liste de sous-classes de noms qui ne peuvent figurer comme son sujet. De la même manière, le $N$ à la droite duquel le $d'_N$ de (3.7) a été inséré ($N$ $d'_N$ = $N$ *Vé de* $N_1$, qui est la séquence (3.5.)d) ne doit pas appartenir à l'une des sous-classes défendues comme objet du verbe.

peut être rattaché soit à $N_1$, soit à $N_2$ [1] ; les analyses de ces deux cas auront les formes :

$$(3.10)\begin{matrix} a \\ b \end{matrix} \quad \begin{matrix} N_1\,[\text{de } N_2]\,[A] = N_1\,d_{N1}\,d'_{N1} & \text{avec} & d_{N1} = \textit{de}\ N_2 \quad \text{et} \quad d'_{N1} = A \\ N_1\,[\text{de } (N_2\ A)] = N_1\,d_{N1} & \text{avec} & d_{N_1} = \textit{de}\ N_2\,d_{N2} \quad \text{et} \quad d_{N2} = A \end{matrix}$$

Dans la première analyse, $d'_{N1}$ — l'adjectif — est un deuxième ajout au nom $N_1$, inséré à droite du premier ajout. Les deux ajouts $d_{N1}$ et $d'_{N1}$ se rapportent tous deux à $N_1$. Dans la deuxième analyse, il n'y a qu'un seul ajout sur $N_1$ ; l'adjectif est un ajout sur $N_2$ et se trouve à sa droite.

Le point important : dans les deux analyses, l'adjectif se trouve toujours à droite du nom qu'il modifie, et donc est dans une même chaîne avec le nom, la chaîne $Nd_N$. Dans (3.10)*b*, $A$ est tout de suite à droite de $N_2$, auquel il se rapporte ; dans (3.10)*a*, $A$ est à droite de $N_1$, qu'il modifie, puisque $d_N$ (qui est à droite de $N_1$) est constitué de la suite $d_{N1}\ d'_{N1}$. Les deux éléments $N$ et $A$ étant donc toujours contigus (dans une même chaîne), il est facile de vérifier l'accord de l'adjectif et le nom afin d'éviter des fausses analyses.

Ainsi, considérons les séquences :

(3.11)*a* *une zone d'orages étendue*
(3.11)*b* *un régime de vents faibles.*

La première ne sera pas analysée comme une occurrence de (3.10)*b*, puisque $A$ ne s'accorde avec $N_2$ ni en nombre, ni en genre ; de la même manière, la seconde ne sera pas analysée comme une occurrence de (3.10)*a* puisque $A$ ne s'accorde pas avec $N_1$ en nombre. Et parfois, même si l'adjectif s'accorde avec $N_1$ et $N_2$ à la fois, il peut y avoir une autre contrainte de compatibilité de sous-classes entre l'un des noms et l'adjectif, contrainte qui oblige à écarter l'une des analyses. Ainsi, la séquence :

(3.11)*c* *chef de groupe statistique*

ne peut être analysée comme une occurrence de (3.10)*a*, dans laquelle *A (statistique)* se rapporte à $N_1$ *(chef)*, car la phrase *(le) chef est statistique* n'existe pas. En effet, l'adjectif *statistique* appartient à une sous-classe d'adjectifs qui ne se rapporte pas à un $N_h$ (nom d'être humain), et justement, *chef* appartient à la sous-classe $N_h$. Une restriction est ajoutée à la chaîne $Nd_N$ pour empêcher l'analyse erronée de (3.11)*c*.

## 4. L'ANALYSEUR SYNTAXIQUE

Le processus d'analyse peut maintenant être esquissé, sans entrer toutefois dans les détails. La grammaire en chaîne est un ensemble de chaînes, chacune ayant une ou plusieurs options (ou valeurs possibles). Les restrictions décrites ci-dessus accompagnent les diverses options ; chacune d'elles empêche le programme de choisir telle ou telle option (d'une chaîne) dans une analyse en cours, lorsque les conditions particulières à cette restriction ne sont pas satisfaites par les mots de la phrase. La grammaire se présente par conséquent comme un ensemble :

1. Ou encore successivement aux deux, lorsque l'adjectif peut s'accorder soit à $N_1$, soit à $N_2$. Ceci constitue un cas d'ambiguïté (voir plus loin § 5).

(4.1)

chaîne centrale [1] = $R$, $C1$ / $R$, $C2$ / ...
$C1 = R, * \Sigma * g_V\ tV\ d_V * \Omega\ d_V *$
$\Sigma = R, N / R,$ *que* $C1 / R, V\ \Omega$ / ...
$N = R, g_N\ N\ d_N$ / ...
. .
. .
. ., etc.

On trouve chaque mot de la phrase à analyser dans le lexique, où il figure avec toutes ses appartenances aux diverses catégories de la grammaire ; les verbes, en particulier, sont accompagnés d'une liste de sous-classes de noms qui ne peuvent occuper la position sujet (ou objet) de ce verbe [2].

L'analyseur syntaxique lit la phrase à analyser, cherche ensuite les entrées lexicales pour chaque mot de la phrase, puis transfère la grammaire en chaîne dans la mémoire centrale. Partant de la première chaîne de la grammaire (chaîne centrale dans le schéma de (4.1)) qui en est l'axiome [3], le programme essaie d'associer les catégories grammaticales qui figurent dans les diverses chaînes avec la séquence de mots de la phrase (par un balayage de gauche à droite). Chaque fois qu'une occurrence de telle ou telle chaîne est construite, toutes les restrictions sur cette chaîne (s'il y en a) sont vérifiées. L'échec de l'une d'entre elles oblige le programme à revenir en arrière et à tenter une autre analyse.

Si le jeu intégral des restrictions définies dans la grammaire recouvre tous les phénomènes linguistiques de la langue (tels qu'ils ont été esquissés dans le paragraphe 3), le programme ne pourra que fournir des analyses justes. La définition des sous-classes et la formulation des restrictions les utilisant apparaît donc comme une des tâches principales de la confection d'une grammaire en chaîne.

## 5. PROBLÈMES RÉSIDUELS

Bien que le programme ne fournisse que des analyses correctes, ceci ne signifie pas que le problème de l'analyse syntaxique automatique ait été résolu une fois pour toutes — tant s'en faut. Une fois écartées les analyses erronées, le problème de l'ambiguïté de la langue reste entier. Notons tout d'abord qu'une grammaire en chaîne, par sa structure, n'apporte pas de solutions rapides à ce problème ; elle nous donne plutôt un cadre dans lequel il est facile d'incorporer une solution. A l'heure actuelle, il me semble que la solution du problème de l'ambiguïté consisterait à trouver des sous-classes de plus en plus fines et mieux définies ; les restrictions, qui incorporeraient ces sous-classes plus fines, permettraient de trancher entre les structures ambiguës proposées par le programme, limité par le manque de finesse de la grammaire actuelle.

Évidemment, pour les phrases qui ont une ambiguïté inhérente, le programme doit toujours donner plusieurs analyses. Reprenons l'exemple (3.10) ; une séquence telle que :

(5.1) *zone de pollution croissante*

---

1. La barre diagonale sépare les options. $C1$ est la chaîne d'assertion décrite dans 2. $C2$, $C3$, etc., sont d'autres chaînes centrales pour les phrases interrogatives, impératives, ... qui ne peuvent être décrites ici, faute de place. Toutes les $R$ devraient être indicées pour les distinguer.

2. Il y a beaucoup d'autres renseignements encore qui doivent figurer dans chaque entrée lexicale. L'établissement de ces entrées lexicales pour les mots de la langue, dans le cadre d'une analyse automatique basée sur une grammaire en chaîne, n'est donc pas une tâche aisée.

3. C'est-à-dire, chaque phrase bien formée de la langue doit pouvoir être analysée comme contenant une occurrence de C1, ou de $C^2$ etc.

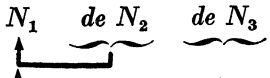
doit donner lieu aux deux analyses de (3.10), puisque l'adjectif *croissante* peut se référer soit à *zone*, soit à *pollution*.

Cette remarque est vraie lorsqu'un programme d'analyse travaille sur des phrases isolées de leur contexte. Il est fort possible que les ambiguïtés inhérentes d'une phrase isolée puissent être écartées si l'on tient compte des rapports entre la phrase étudiée et les autres phrases de son voisinage : on arriverait ainsi à ne trouver qu'une seule analyse qui correspondrait à l'interprétation voulue par son auteur. Mais les travaux sur l'analyse du discours sont encore trop peu avancés : les difficultés linguistiques sont énormes.

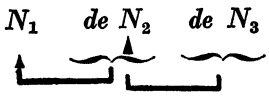
Puisque les ambiguïtés inhérentes dans des séquences comme (5.1) sont connues par leur propre structure, il n'est pas utile de présenter ces multiples analyses à un utilisateur éventuel du programme d'analyse. Regardons le cas d'une séquence ambiguë qui est tout à fait semblable à (3.9) :

(5.2)

*a*




*b*




Dans la première analyse — qui est semblable à (3.10)*a* — les deux groupes prépositionnels se rapportent à $N_1$, comme dans la séquence *lycée de garçons d'État*; dans la deuxième — semblable à (3.10)*b* — le deuxième groupe *(de $N_3$)* se rapporte au premier groupe *(de $N_2$)*, lui-même se rapportant à $N_1$, comme dans la séquence – *lampe à vapeur de mercure.*

Appelons le premier type d'analyse *répétition d'ajout*, et le deuxième *imbrication d'ajout*. Il est clair que toute séquence de la forme (3.10) ou (5.2) sans restrictions qui empêchent l'une ou l'autre analyse (restrictions dont on a donné un exemple en (3.11)) fournira deux analyses. Les inconvénients de cette double analyse apparaissent dans une séquence comme :

(5.3) ... *voir l'homme dans un parc avec un télescope...*

où les deux groupes prépositionnels peuvent être rattachés soit à *homme* (comme dans 5.2), soit à *voir* (comme $d_V$ : *voir dans un parc* ou *voir avec un télescope*) ; nous avons ainsi 4 analyses possibles. Si le nombre de ces groupes prépositionnels dans la phrase augmente, le nombre d'analyses possibles pour les diverses combinaisons de cas de répétition et d'imbrication croît alors exponentiellement.

Ces multiples analyses alourdissent énormément la présentation des résultats sans pour cela apporter de renseignements supplémentaires [1]. Aussi, le programme d'analyse ne fournit-il que les analyses où les ajouts sont imbriqués les uns dans les autres, comme dans (5.2)*b* [2]. En attendant que des recherches linguistiques apportent une solution au problème du choix entre les analyses des structures ambiguës du type (5.2), il est très commode de pouvoir supprimer sélectivement toutes les analyses qui n'apportent pas une information que l'on jugera intéressante.

---

1. Puisque l'existence et la structure de l'une forme des analyses sont connues à partir de l'autre analyse. Ainsi, on peut déduire la forme d'analyse due au placement des ajouts par imbrication si l'on connaît déjà l'analyse sous forme de répétition d'ajouts.

2. A moins qu'il n'y ait une restriction qui empêche l'analyse sous forme d'imbrication, comme dans (3.11) a et (3.11) c.

## 6. CONCLUSIONS

Un programme qui utilise une grammaire en chaîne se révèle être un outil puissant pour l'analyse des phrases d'une langue naturelle. En raison de la structure de cette grammaire, l'insertion des contraintes de natures diverses (contraintes de compatibilité entre classes de mots, contraintes d'accord entre éléments non-contigus, etc.) entre éléments des chaînes est non seulement commode mais aussi pratique.

L'existence d'un analyseur qui donne les analyses correctes d'une phrase et seulement celles-là (dans les limites discutées en 5) ouvre la voie aux recherches sur le traitement de l'information à partir de l'analyse syntaxique du contenu. En particulier, les recherches dans le domaine de la documentation automatique, où les rapports entre mots porteurs de renseignements significatifs jouent un rôle capital dans l'analyse de tout document, profiteraient de cet analyseur en l'incorporant dans un système de recherche documentaire.

## BIBLIOGRAPHIE


[1] GROSS, M., *Grammaire transformationnelle du français : Le verbe*, Paris, Larousse, 1968.

[2] GROSS, M., "On the equivalence of models of language used in the fields of machine translation and information retrieval", *Information storage and retrieval*, Vol. 2, pp. 43-57 (1964).

HARRIS, Z. S., *String analysis of sentence structure*, La Haye, Mouton, 1961.

HAYS, D. G., *Introduction to computentional linguistics*, Chapitres 6 et 7, New York, American Elsevier, 1967.

KUNO, S., "The predictive analyzer and a path elimination technique", *A.C.M.*, 8 (7), July 1965, pp. 453-462.
(Disponible aussi dans *Readings in automatic languages processing* in: D.G. Hays (ed.), New York, American Elsevier, 1966.)

SALKOFF, M., SAGER, N., *Restrictions in a string grammar of English*, String Program Reports n° 5, Linguistic String Project, 2 Washington Square Village, New York, 1969.

SAGER, N., *A computer string grammar of English*, String Program Reports, n° 4, *ibid.*