Stratégie d'analyse et structures de données

Eric Wehrli
Département de linguistique - LATL
Université de Genève

Juillet 1991

1 Introduction

Ce rapport décrit sommairement les structures de données et l'algorithme d'analyse de l'analyseur syntaxique interactif en cours de développement au Département de linguistique de l'Université de Genève, dans le cadre d'un projet du Fonds national suisse de la recherche scientifique.

Il commence par un bref rappel des objectifs de cette recherche, avant de décrire, dans la section 2, les structures de données utilisées par l'analyseur, et dans la section 3, l'algorithme d'analyse.

1.1 Le projet IPS

Le projet `IPS` (*Interactive Parsing System*) vise à développer un analyseur syntaxique interactif pour l'anglais basé sur le modèle linguistique de la théorie chomskyenne dite du "gouvernement et liage" (*Government and Binding*, ou simplement *GB*)[1].

Les objectifs de ce projet sont à la fois d'ordre théorique et d'ordre pratique. Du point de vue théorique, il s'agit de mettre en évidence les avantages que présente pour l'analyse syntaxique automatique le choix d'une approche basée sur une théorie linguistique modulaire, basée sur des principes généraux, plutôt que sur des règles spécifiques comme les règles traditionnelles de réécriture. Du point de vue pratique, le projet vise à développer un analyseur syntaxique puissant, susceptible d'utilisations pratiques dans différents domaines du traitement automatique du langage, et en particulier dans celui de la traduction. Enfin, et c'est là la troisième dimension de ce projet, il s'agit également de montrer l'intérêt d'une approche interactive dans le domaine de l'analyse du langage, c'est-à-dire d'un système qui dialogue avec son utilisateur. Ainsi, l'analyseur `IPS` est interactif en ce sens qu'il peut requérir des compléments d'information de la part de l'utilisateur. L'interaction prend la forme de dialogues en temps réel entre le programme et son utilisateur.

[1] Voir Chomsky (1981, 1986) pour un exposé de la théorie "gouvernement et liage", Berwick (1987) et Wehrli (1988) pour une discussion des analyseurs basés sur cette théorie.

2 Spécifications de l'analyseur et structures de données

L'analyseur syntaxique développé pour le projet IPS repose sur l'analyse grammaticale de type GB développée dans les rapports de R. Clark, *cf.* Clark (1990a,b, 1991a-f).

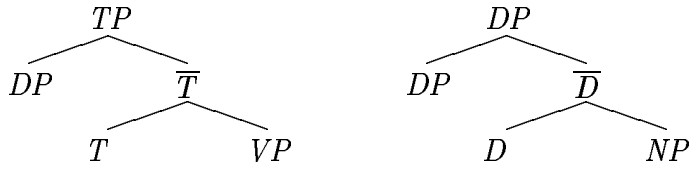
Le module de base de cette grammaire est le module $\overline{X}$, qui dicte la géométrie des structures syntaxiques. Le schéma $\overline{X}$ adopté pour cette recherche est donné en (1) :

(1) XP $\longrightarrow$ Spec X'
X' $\longrightarrow$ X Compl

où X est une catégorie lexicale ou fonctionnelle, et Spec et Compl correspondent à des listes (éventuellement vides) de projections maximales YP.

Comme l'indique (1), les projections maximales sont d'ordre 2, et les catégories lexicales d'ordre 0. L'ensemble des catégories lexicales comprend N, V, A et P, et celui des catégories fonctionnelles D(eterminer), T(emps) et C(omplémenteur). Nous adoptons l'hypothèse DP (*cf.* Abney, 1987, Clark, 1990a), avec comme conséquence le parallélisme entre les structures DP et TP, comme illustré en (2) :

(2)



Aussi bien les catégories lexicales que les catégories fonctionnelles peuvent sélectionner des projections lexicales ou fonctionnelles. Ainsi, par exemple, un déterminant peut sélectionner une projection de type DP ou NP, comme illustré en (3) et (4) respectivement, correspondant aux structures (5), (6) :

(3) [each, D, [+definite], [_[D,[numeral]]^{max}]

(4) [each, D, [+definite], [_[N,[singular]]^{max}]

(5)a. each five men.

b. [$_{DP}$ [$_{D}$, each [$_{DP}$ [$_{D}$, five[$_{NP}$ [$_{N}$, students]]]]]]

(6)a. each student.

b. [$_{DP}$ [$_{D}$, each [$_{NP}$ [$_{N}$, student]]]]

De même, les auxiliaires sélectionnent des projections de type VP, et la plupart des prépositions des projections de type DP. Quelques exemples de traits de sélection associés aux auxiliaires sont donnés en (7), et des exemples de structures correspondant à ces sélections en (8) et (9):

(7)a. [have, V, [+aux], [_[V,[+ past participle]]^{max}]

b. [be, V, [+aux], [_[V,[past participle]]^{max}]

c. [be, V, [+aux], [_[V,[present participle]]^{max}]

(8)a. the men have arrived.

b. [$_{TP}$ [$_{DP}$ [$_{D}$, the [$_{NP}$ [$_{N}$, men]]]] [$_{T}$, have [$_{VP}$ [$_{V}$, arrived]]]]

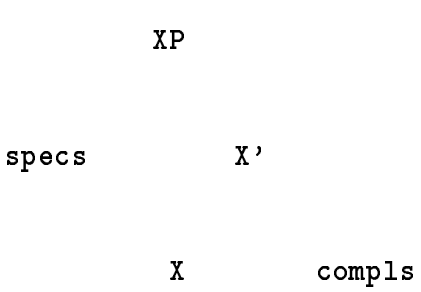
(9)a. the men must have been being cheated.

b. [$_{TP}$ [$_{DP}$ [$_{D}$, the [$_{NP}$ [$_{N}$, men]]]] [$_{T}$, must [$_{VP}$ [$_{V}$, have [$_{VP}$ [$_{V}$, been [$_{VP}$ [$_{V}$, being [$_{VP}$ [$_{V}$, cheated]]]]]]]]]]

On peut résumer comme suit les caractéristiques essentielles des structures de constituants :

1. Toute catégorie lexicale `X` entraîne une projection maximale de cette catégorie, `XP`.
2. Seuls les attachements de catégories `XP` sont légitimes.
3. Tous les constituants ont une structure de même architecture, comme illustré en (10), où `specs` et `compls` correspondent à des listes (éventuellement vides) de projections maximales :

(10)



La régularité absolue de ces principes permet l'adoption d'une structure de données qui fusionne en une structure unique les trois niveaux de la théorie $\overline{X}$. Cette structure unique, que nous appellerons `Projection`, comprend une tête lexicale, un ensemble de traits, une liste de spécificateurs et une liste de compléments, comme défini en (11). Il s'agit là d'une définition partielle et provisoire, qui sera complétée et modifiée par la suite, en particulier pour tenir compte des catégories vides et de la coordination.

(11) Structure d'une projection

```
TYPE Projection = RECORD
                    head :  ItemLexicalPtr;
                    feat :  FeaturesPtr;
                    spec :  List;
                    compl:  List;
                  END
```

Dans la structure définie en (11), le champ `head` correspond à la tête lexicale de la projection, et prend naturellement la forme d'un pointeur sur un item lexical. Le champ `feat` correspond à l'ensemble des traits associés à la projection. Pour des raisons d'implémentation, il prend la forme d'un pointeur sur une structure de traits. Enfin, les champs `spec` et `compl` correspondent respectivement aux spécificateurs et aux compléments de la projection. Comme nous l'avons déjà souligné, ces deux champs sont de type liste (éventuellement vide) de projections.

Les structures de traits associées aux projections contiennent toutes les propriétés spécifiques des projections en fonction de leur catégorie syntaxique. Ceux-ci comprennent par exemple, pour les syntagmes verbaux, les traits d'accord (`nombre, personne, temps`, *etc.*), les traits de sélection et les traits de sous-catégorisation; pour les syntagmes nominaux `DP`, les traits d'accord (`nombre, personne, genre, cas`), les traits inhérents (`humain, mass/count, etc.`) et les traits de sous-catégorisation. Les détails des structures de traits ne sont pas donnés ici en raison de leur complexité.

3 L'algorithme d'analyse

Dans le système `IPS`, l'analyse d'une phrase comporte deux niveaux distincts :

l'analyse lexicale, responsable de la segmentation d'une phrase d'entrée en unités lexicales (*tokenization*), et

l'analyse syntaxique, qui assigne à une phrase une ou plusieurs structures syntaxiques.

Ces deux niveaux sont décrits dans les sections suivantes.

3.1 L'analyse lexicale

Lorsqu'une phrase est soumise à l'analyseur, c'est tout d'abord le module lexical qui entre en jeu. Son rôle est de découper la phrase en unités lexicales et de constituer une structure de données appropriée pour l'analyseur syntaxique. Concrètement, une phrase est lue de gauche à droite, un mot après l'autre. Chaque mot est confronté à la base de données lexicale afin de déterminer s'il correspond à une unité lexicale, ou à une partie d'unité lexicale (dans le cas de mots composés)[2]. Toutes les combinaisons d'unités lexicales qui correspondent aux mots de la phrase d'entrée sont enregistrées dans une structure de données de type *chart*, c'est-à-dire un graphe orienté acyclique, qui sert d'entrée à l'analyse syntaxique proprement dite.

[2] Une description de la structure et du contenu de la base de données lexicale utilisée pour ce projet est donné dans Walther (1991), *cf.* annexe 8.

A ce niveau, on réduit le nombre d'hypothèses en associant toutes les lectures d'une unité lexicale de même catégorie à une seule arête du graphe, ou **item**. Autrement dit, un item correspond non à une unité lexicale, mais bien à l'ensemble des lectures de même catégorie de cette unité lexicale.

Ainsi, si le lexique contient les lectures données en (12a et b) pour les mots *notice* et *pet*, l'analyse lexicale créera deux items dans le premier cas (`N` et `V`), et trois items dans le second cas (`N`, `V` et `Adj`). Pour donner un exemple plus saisissant encore, dans le cas du mot *go*, l'analyse lexicale crée deux items (`N`, `V`), dont le deuxième correspond à la vingtaine de lectures verbales du verbe *to go*.

(12)a. **notice**

- N (warning)
- N (end of contract)
- N (announcement)
- N (review)
- V (perceive)
- V (review)

b. **pet**

- N (animal)
- N (favourite)
- Adj (favourite)
- V (indulge)

3.2 Synchronisation de l'analyse lexicale et de l'analyse syntaxique

L'analyseur syntaxique prend comme entrées les items produits par l'analyseur lexical selon les modalités que nous venons de discuter. La synchronisation de ces deux modules est essentielles, puisque le module syntaxique dépend des actions prises par le module lexical. Cette relation de dépendance exige que l'application du module lexical précède celle du module syntaxique. Il y a plusieurs façons de synchroniser ces deux modules en respectant cette contrainte de dépendance. Par exemple, on pourrait procéder à l'analyse lexicale complète d'une phrase avant de soumettre celle-ci à l'analyse syntaxique. Alternativement, on peut synchroniser les deux processus de façon plus serrée, en limitant l'application du module lexical à un mot à la fois et en déclenchant son application chaque fois que le module syntaxique a besoin d'accéder à l'item suivant. C'est cette seconde option que nous avons choisie, principalement pour des raisons de plausibilité psycholinguistique.

3.3 L'analyse syntaxique

La stratégie d'analyse syntaxique utilisée est de type gauche à droite parallèle, combinant une approche essentiellement ascendante avec un filtre descendant, comme nous allons le voir plus en détail.

L'idée de base est que l'analyseur doit travailler en fonction des mots rencontrés dans la phrase, c'est-à-dire de façon ascendante, plutôt que de construire de longues (et parfois nombreuses) séquences d'hypothèses comme le ferait un analyseur descendant. Cependant, une approche strictement ascendante n'est pas souhaitable non plus, car elle conduit d'une part à d'innombrables combinaisons de constituants, localement bien formées mais qui se révèlent incompatibles avec le contexte gauche, et d'autre part à de multiples répétitions d'actions déjà accomplies.

Les exemples suivants illustrent ces deux problèmes :

(13)a. Who could the children have invited ?

b. John must have given the students several of his books.

Dans la phrase (13a), lorsque l'analyseur lit le mot *have* il tente de le combiner avec les constituants dans son contexte gauche. Trouvant le constituant [$_{DP}$ the children] il va naturellement construire un constituant TP avec *the children* comme sujet et *have* comme verbe principal. Localement parfaitement bien formée, cette structure est bien sûr incompatible avec la présence du modal *could*.

La phrase (13b) illustre le deuxième problème posé par une approche strictement ascendante, à savoir la répétition d'actions déjà accomplies. Dans une structure de ce type, avec de nombreux branchements à droite (compléments), l'attachement d'un nouveau complément en bas à droite de la structure provoque une réaction en chaîne sur tout le contexte gauche. Ainsi, après avoir construit la séquence *must have given*, l'analyseur construit le constituant [$_{DP}$ the students] et l'attache au verbe *given*, ce qui détermine un nouveau constituant de catégorie VP, qui maintenant va s'attacher à l'auxiliaire *have*. Le consituant qui résulte de cette combinaison va lui-même se combiner avec un élément de son contexte gauche, *etc.* La même séquence d'actions se répètera au moins une fois, peut-être davantage lorsque l'objet directe de la phrase sera pris en considération. Pour une phrase d'une certaine longueur, il est tout-à-fait possible avec une telle stratégie que certains sous-constituants soient assemblés une dizaine de fois sinon plus.

L'algorithme utilisé s'appuie sur une stratégie dite du "coin droit", dont les principes fondamentaux sont les suivants :

(14) **Algorithme du "coin droit" :**

- dirigé par les données (*data driven*), *i.e.* on cherche à attacher à un contexte gauche un nouvel élément; cet attachement se fait au coin droit du contexte gauche;
- tous les attachements possibles sont considérés en parallèle;
- le contexte gauche spécifie un ensemble de noeuds actifs auquels le nouvel élément est susceptible de s'attacher;
- la combinaison résultant d'un attachement à un sous-constituant d'un contexte gauche n'entraîne pas d'autres opérations (pas de combinaisons itératives);

Il ressort de (14) que l'analyseur tente de combiner les nouveaux éléments non pas exclusivement à des constituants sur leur gauche, mais également à des sous-constituants des

constituants sur leur gauche. En contrepartie, la combinaison d'un nouvel élément avec son contexte gauche n'est pas itérative, sauf dans le cas d'attachement de spécificateurs, autrement dit d'attachement sur la gauche. L'effet attendu de la conjugaison ces deux propriétés est d'abord de permettre une approche ascendante qui évite de recalculer le contexte gauche à chaque nouvel attachement, ensuite, d'offrir une structure de données à partir de laquelle il sera aisé de donner priorité à des attachements en bas à droite. Au plan psycholinguistique, cette stratégie semble consistante avec les observations (banales) (i) que l'analyseur est dirigé par les données, mais (ii) que l'attachement d'un constituant sur la droite n'entraîne pas la recomposition itérative de toute la structure; autrement dit, l'attachement d'un élément (sur la droite) a lieu sur une structure aussi complète que possible et n'entraîne guère de modifications de cette dernière.

Une telle stratégie suppose un accès rapide et efficace aux sous-constituants d'un constituant qui correspondent aux sites d'attachement légitimes. Concrètement, on associe à un constituant une pile de noeuds actifs, un peu sur le modèle de la pile des noeuds actifs de Marcus, où un noeud actif correspond à un site d'attachement sur l'arête droite du constituant.

Un exemple détaillé permettra sans doute de mieux comprendre le fonctionnement de cette stratégie. Considérons la phrase (15) :

(15) John has bought some flowers.

Ignorant pour l'instant la façon dont l'analyseur construit la structure TP, examinons ce qui se passe lorsque le déterminant *some* est lu. A ce stade, le contexte gauche comprend entre autres la structure (16).

(16) [$_{TP}$ John has [$_{VP}$ bought]]

Le mot *some* entraîne la création d'un constituant DP, avec l'élément lexical *some* pour tête. Considérant les constituants sur sa gauche, l'analyseur trouve la structure TP (16), dont un des sites d'attachement est le verbe *bought*. Le constituant [$_{DP}$ some] se combine avec le constituant TP sur sa gauche et s'attache comme complément du verbe *bought*. Enfin, le syntagme DP est ajouté à la pile des noeuds actifs de la structure créée. L'analyseur lit maintenant le mot suivant, *flowers*. La lecture nominale de ce mot entraîne la création d'un syntagme [$_{NP}$ flowers]. Le contexte gauche de ce nouveau constituant comprend le constituant [$_{TP}$ John has bought some], dont un des sites d'attachement est le syntagme [$_{DP}$ some], auquel le syntagme [$_{NP}$ flowers] peut s'attacher comme complément.

Dans le cas d'attachement d'un spécificateur, les choses se passent un peu différemment. Si un nouveau constituant trouve dans son contexte gauche immédiat un constituant susceptible de fonctionner comme spécificateur, l'attachement se fait et détermine un nouveau constituant qui reste à attacher.

Ainsi, dans la phrase (17a), *flowers* se combine d'abord avec [$_{AP}$ nice], puis avec *bought*, un sous-constituant de la phrase, pour donner la structure complète (17b) :

(17)a. John has bought nice flowers.

b. [$_{CP}$ [$_{TP}$ [$_{DP}$ John [$_{T'}$ has [$_{VP}$ bought [$_{DP}$ [$_{NP}$ [$_{AP}$ nice [$_{N'}$ flowers]]]]]]]]]

3.4 Types d'actions

L'analyseur distingue les types suivants d'actions :

- Attachement à droite.
- Attachement à gauche.
- Projection.
- Substitution.
- Coordination.
- Création de traces.

3.5 Attachements à droite

On distingue deux types d'attachement de compléments, appelés respectivement attachement formel et non-formel. L'attachement formel est limité à des catégories adjacentes et est régi par des règles de sélection. Il concerne en particulier, les constituants verbaux (sélection d'auxiliaires, etc.) et les constituants nominaux (sélection de déterminants). L'attachement non-formel concerne les catégories `DP, CP, AP, AdvP, PP` comme compléments, les catégories $\overline{\mathrm{V}}$, $\overline{\mathrm{N}}$, $\overline{\mathrm{A}}$, $\overline{\mathrm{P}}$ comme sites d'attachement. La légitimation de ces attachements est du ressort des modules d'interprétation des compléments et des ajouts. Une précompilation des propriétés de l'anglais permet néanmoins de limiter quelque peu le foisonnement d'attachements au niveau purement syntaxique. Ainsi, il est possible de limiter les attachements de constituants `DP` aux catégories $\overline{\mathrm{V}}$ et $\overline{\mathrm{P}}$, excluant ainsi $\overline{\mathrm{A}}$ et $\overline{\mathrm{N}}$.[3]

3.6 Attachement à gauche

Dans des langues comme le français et l'anglais, l'attachement de spécificateurs correspond à un attachement à gauche de la tête. Cet attachement concerne principalement les constituants de type `DP` (spécificateur de `TP`), `Adv` (spécificateur de `VP` ou de `AP`), `Adj` (spécificateur de `NP`). De plus, tous les constituants qui portent le trait *wh* sont susceptibles d'un attachement comme spécificateurs de `CP`. Contrairement au cas d'attachement à droite, l'attachement à gauche peut être itératif en ce sens que le produit d'un tel attachement peut lui-même devenir un constituant à attacher. Considérons l'exemple suivant :

(18) John said Mary went home.

Dans une phrase comme (18), lorsque l'analyseur considère le mot *said*, un constituant [$_{\texttt{TP}}$ said] est projeté sur la base du trait [+tense] du verbe `said`. La présence dans le contexte gauche immédiat du constituant [$_{\texttt{DP}}$ John] permet une combinaison de ces deux syntagmes, avec attachement du `DP` comme spécificateur du syntagme `TP`, ce qui conduit à la création

[3] Du point de vue de la théorie linguistique, l'exclusion des attachements de `DP` aux projections $\overline{\mathrm{A}}$ et $\overline{\mathrm{N}}$ résulte de la théorie des cas, plus spécifiquement de l'absence d'assignation de cas aux compléments d'objets directs d'adjectifs et de noms.

du constituant [$_{TP}$ [$_{DP}$ John] [$_{T}$, said]]. Plus loin dans l'analyse de cette phrase, des circonstances similaires conduisent à la création d'un constituant [$_{TP}$ [$_{DP}$ Mary] [$_{T}$, went]]. Ce dernier constituant peut ensuite être attaché au verbe `said` pour former la structure (19):

(19) [$_{TP}$ [$_{DP}$ John] [$_{T}$, said [$_{TP}$ [$_{DP}$ Mary] [$_{T}$, went]]]]

3.7 Projection

L'opération de projection consiste à créer un constituant d'un certain type sur la base du constituant courant. Les cas typiques, en anglais, concernent la projection d'un constituant `DP` à partir d'un `NP` si celui-ci porte des traits tels que *plural*, *mass*, ou *proper name*, ou la projection d'un constituant `TP` sur la base d'un verbe non-conjugué.

Voici quelques exemples :

(20)a. [$_{DP}$ [$_{NP}$ old men]]

b. [$_{DP}$ [$_{NP}$ Paul]]

c. [$_{TP}$ [$_{VP}$ go]]

Dans l'exemple (20a), une structure `DP` est projetée sur la base d'un `NP` pluriel. Dans le cas de (20b), une structure `DP` est projetée sur la base d'un `NP` qui porte le trait `proper name`. Enfin, en (20c), on a une projection d'un `TP` sur la base d'un `VP` non-conjugué. Dans tous les cas de projection, la tête du constituant projeté est vide.

3.8 Substitution

L'opération de substitution consiste à remplacer un sous-constituant par un autre. Typiquement, cette opération a lieu lorsque l'analyseur se rend compte que le sous-constituant actif d'une analyse est en fait le spécificateur d'un constituanta été mal attaché, en ce sens qu'il il s'agit de remplacer le sous-constituant actif par un autre. Les exemples ci-dessous illustrent ce cas de figure :

(21)a. John met Mary's sister's lover.

b. John could not find his umbrella.

Dans l'exemple (21a), le constituant `Mary` est d'abord interprété comme l'objet direct du verbe principal (`met`). Poursuivant la lecture de la phrase jusqu'au mot *sister*, on se rend compte que cet attachement n'est pas correct. `Mary` n'est pas l'objet direct mais bien un possessif, c'est-à-dire un spécificateur de l'objet direct `sister`. La même erreur d'attachement a été répétée, puisqu'à la lecture du mot suivant, on se rend compte que le syntage [$_{DP}$ Mary's sister's] est un possessif associé à *lover*. La structure correcte associée à (21a) est donc celle donnée en (22a).

(22)a.

3.9 Coordination

La coordination est traitée séparément ici, bien qu'en fait elle corresponde à un cas particulier d'attachement. Cependant, comme nous allons le voir, les propriétés particulières du phénomène de coordination nécessitent un traitement ainsi que des structures de données spécifiques. Nous nous bornerons ici à un rapide survol des données et de l'algorithme d'analyse tel qu'il a été implanté à ce jour. Une discussion plus détaillée est offerte dans Clark (1991d), d'où provient d'ailleurs l'essentiel de cette section.

Les cas de coordination pris en considération sont ceux impliquants les conjonctions *and* et *or*. Les énumérations ainsi que les autres cas de coordination du type *but, as well as, both...and, either...or, etc.* ne sont pas pris en compte pour l'instant. Enfin, l'algorithme décrit dans cette section est limité à la coordination de mêmes catégories, excluant ainsi les cas (très rares, il est vrai) de coordination de deux constituants de catégories différentes.

Les propriétés fondamentales des constructions coordonnées notées par Clark (1991d) peuvent se résumer comme suit :

- La coordination est possible entre deux constituants de n'importe quelle même catégorie `X` et `XP`, mais pas de catégorie `X'`[4].
- La structure d'un constituant coordonné est d'un type particulier, qui ne respecte pas le schéma $\overline{\mathrm{X}}$. La structure assignée à un constituant coordonné est donnée en (23), où `n` prend ses valeurs dans l'ensemble de valeurs {0,2}, soit `X` et `XP`.

  (23) $[_{\mathtt{X}^n}\ \mathrm{X}^n\ \textit{conj}\ \mathrm{X}^n]$

- Du point de vue de l'analyseur, le problème fondamental que pose la reconnaissance d'une structure coordonnée est le non-déterminisme, ou l'absence de critères permettant de prédire la nature catégorielle de la structure au moment où la conjonction est lue.

Les exemples (23) donnés par Clark (1991d) attestent de la variété catégorielle du phénomène de coordination :

(24)a. John is late and Bill is sick. (clauses)

b. It is obvious [that John is late and that Bill is sick]. (`CP`s)

c. [the old men and the fat women] left. (`DP`s)

d. [each and every] republican is an alcoholic. (`D`s)

e. The [old men and fat women] left. (`NP`s)

f. This box is a combination [refrigerator and war-head] (`N`s)

g. John is [very tall and extremely arrogant] (`AP`s)

h. John is [both proud and ashamed] of himself. (`A`s)

i. John ran [down the stairs and out the door] (`PP`s)

j. John ran [in and out] the door. (`P`s)

[4] A l'exception de la catégorie `T'`, comme dans la phrase *John ate bread and drank beer.*

k. John [killed the pig and ate it] (`T's`)

l. John has [killed the pig and eaten it] (`VPs`)

m. John [washed and dried] the dishes. (`Vs`)

n. John answered the question [slowly and stupidly] (Adverbs)

o. You [can and will] write a paper about conjunction. (Modals).

Etant donné la nature particulière des structures coordonnées, une modification de la structure de données utilisée pour représenter les projections s'avère nécessaire, comme annoncé au début de ce papier. En effet, si une projection est de type coordination on veut spécifier le type de coordination (`and, or, etc.`) et la liste des conjoints. Par contre, la notion de tête, dans ce cas, n'a guère de sens. La structure modifiée pour prendre en compte cette disjonction est définie en (25). Dans cette structure les champs `coordination, feat, spec et compl` sont définis pour toutes les projections. Par contre, les champs `coordinationType` et `conjuncts` ne sont définis qu'au cas où `coordination` prend la valeur `TRUE`, alors que le champ `head` n'est défini que si `coordination` prend la valeur `FALSE`.

(25) Structure d'une projection

```
TYPE Projection = RECORD
                    CASE coordination :  BOOLEAN OF
                      TRUE :     coordinationType  :  Value;
                                 conjuncts         :  List;
                      FALSE:     head              :  ItemLexicalPtr;
                    END;
                    feat   :  FeaturesPtr;
                    spec   :  List;
                    compl  :  List;
                  END
```

L'algorithme de traitement de la coordination consiste en deux parties. D'une part il s'agit de faire l'hypothèse d'un constituant coordonné, d'autre part de compléter ce constituant. La première opération est déclenchée par l'apparition d'une conjonction de coordination. Ainsi pour une phrase telle que (26),

(26) John believes Bill and Mary.

c'est au moment où l'analyseur lit la conjonction de coordination *and* qu'il déclenche le processus de coordination. La présence de la conjonction permet de faire l'hypothèse d'un constituant coordonné. Le pas suivant consiste à déterminer les conjoints de ce constituant, en l'occurence les constituants de même catégorie à gauche et à droite de la conjonction. Dans notre exemple, il s'agit des constituants `DP` *Bill* et *Mary*, ce qui nous donne la structure (27) :

(27) [$_{\texttt{TP}}$ John believes [$_{\texttt{DP}}$ [$_{\texttt{DP}}$ Bill] and [$_{\texttt{DP}}$ Mary]]]

Ce qui paraît très simple vu du point de vue d'une machine non-déterministe l'est beaucoup moins lorsqu'il s'agit de spécifier un processus déterministe. En effet, lorsque l'analyseur lit la conjonction, rien ne lui permet d'affirmer la forme précise du constituant coordonné. Etant donné la grande généralité de la coordination, n'importe quel constituant de type `X` ou `XP` dans le contexte gauche immédiat de la coordination est un candidat potentiel à considérer. Pour illustrer ce problème, reprenons l'exemple (26), au moment où l'analyseur lit la conjonction. Dans cette configuration, l'analyseur a lu la portion de phrase représentée en (28a), et rien ne lui permet de décider si le constituant coordonné est de catégorie `DP`, comme en (28b), de catégorie `TP`, comme en (28c). De plus, dans le cas d'une coordination de `DP`, ce constituant peut s'attacher soit comme complément d'objet du verbe *believe*, soit comme sujet de la phrase infinitive, comme en (28d).

(28)a. John believes Bill and

b. John believes Bill and Mary.

c. John believes Bill and Mary is to blame.

d. John believes Bill and Mary are to blame.

Le non-déterminisme inhérent à la coordination conduit souvent à des ambiguïtés globales, comme illustré en (29), où l'absence de marque d'accord sur le modal ne permet pas de distinguer entre les lectures (b) et (c) :

(29)a. John believes Bill and Mary will be to blame.

b. John believes [$_{\texttt{TP}}$ [$_{\texttt{DP}}$ Bill and Mary] will be to blame].

c. [$_{\texttt{TP}}$ John believes Bill] and [$_{\texttt{TP}}$ Mary will be to blame].

L'algorithme développé pour traiter la coordination tire parti de la stratégie du coin droit. En effet, parmi les constituants immédiatement à gauche d'une conjonction de coordination, on privilégie ceux qui font partie d'une structure active. Concrètement, l'algorithme fonctionne comme suit : lorsqu'une conjonction de coordination a été repérée, on parcourt de haut en bas la structure correspondant au contexte gauche immédiat de la conjonction, en suivant l'arête droite de cette structure, et on construit une liste de tous les sommets de type `XP`, $\overline{\texttt{T}}$ et `X` situés sur cette arête. Cette liste donne toutes les possibilités théoriques de coordination, y compris celle de $\overline{\texttt{T}}$ qui constitue un cas particulier, traité comme tel dans notre implémentation. Sur la base de cette liste, on construit donc tous les constituants potentiels (et partiels). Autrement dit, pour chaque constituant `W` dans cette liste, on construit un constituant `W'` qui est de la même catégorie que `W` mais de type coordination que l'on subsitue au constituant `W` dans toutes les structures où ce dernier apparaît. Le constituant `W'`, dont le premier conjoint est `W`, et le deuxième la conjonction de coordination, ne sera validé que lorsqu'un autre constituant de même catégorie que `W` sera trouvé dans le contexte droit immédiat de la conjonction.

Considérons l'exemple suivant :

(30) the old men and women.

Lorsque l'analyseur parvient au mot *and*, il a dans son contexte gauche (entre autres) le constituant (31) :

(31) [$_{DP}$ the[$_{NP}$ [$_{A}$ old] [$_{N}$, men]]]

L'établissement de la liste des noeuds situés sur l'arête droite de ce constituant donne (DP, NP, N), correspondant aux chaînes (*the old men, old men, men*). On construit donc trois constituants de type coordonné, respectivement DP, NP, N dont le premier conjoint est le constituant DP, NP,N à gauche de la conjonction et le deuxième constituant la conjonction elle-même.

La deuxième partie de ce traitement consiste à trouver un constituant, cette fois dans le contexte droit de la conjonction, capable de compléter le constituant coordonné. Ce constituant doit être de même catégorie que le constituant coordonné, puisque nous avons fait l'hypothèse (raisonable) de ne considérer que les cas de coordination de consituants de même catégorie.

Dans l'exemple (30), le contexte droit de la conjonction de coordination est constitué par le mot *women*. Ce mot est tout à la fois un substantif N, un syntagme nominal NP et, comme il porte le trait pluriel, un DP. Cela permet de compléter les trois syntagmes coordonnés potentiels et donne par conséquent les trois structures suivantes pour cette phrase :

(32)a. [$_{DP}$ [$_{DP}$ the[$_{NP}$ [$_{A}$ old] [$_{N}$, men]]] [$_{ConjP}$ and] [$_{DP}$ [$_{NP}$ women]]]

b. [$_{DP}$ the[$_{NP}$ [$_{A}$ old] [$_{N}$, [$_{N}$, men and women]]]]

c. [$_{DP}$ [$_{NP}$ the[$_{NP}$ [$_{A}$ old] [$_{N}$, men]]] [$_{ConjP}$ and] [$_{NP}$ women]]

3.10 Le traitement des chaînes $\overline{\text{A}}$

Une des difficultés bien connues du traitement de la syntaxe des langues naturelles est le traitement des syntagmes extraposés. Ce qualificatif s'applique à une large classes de phénomènes syntaxiques, correspondant en gros, à toutes les transformations de mouvement de la théorie transformationnelle classique. Les raffinements successifs de la théorie ont fait apparaître des distinctions fondamentales à l'intérieur de cette classe, que ce soit en termes des propriétés de cas, de propriétés thématiques ou encore de contraintes de localités. Dans la théorie "gouvernement et liage" on distingue ainsi les mouvements $\overline{\text{A}}$, qui concernent l'antéposition de compléments ou d'ajouts vers une position non-argument (typiquement une position de spécificateur), et les mouvements A, qui concernent le déplacement d'un constituant d'une position d'argument vers une autre position d'argument. Le premier type de déplacement peut se faire de façon itérative sur une distance non limitée et n'implique ni le module des cas ni le module thématique. Au contraire, le mouvement d'éléments A est soumis à des règles de localité beaucoup plus strictes et interagit directement avec le module de cas puisqu'il affecte des syntagmes nominaux soumis au filtre des cas mais engendrés dans des positions dépourvues de cas.

Nous adoptons dans ce papier un point de vue représentationnel plutôt que dérivationnel de l'extraposition. Autrement dit, la relation entre un syntagme extraposé et sa trace est vue non pas comme un déplacement mais simplement sous forme d'une chaîne[5].

Pour l'instant, notre traitement des chaînes se limite aux chaînes $\overline{\text{A}}$, c'est-à-dire des chaînes dont la tête est dans une position $\overline{\text{A}}$, en général dans la position de spécificateur

[5]Voir Rizzi (1986) pour une discussion des points de vue dérivationnels et représentationnels, ainsi que pour la notion de chaîne.

de C. L'exemple typique, en anglais, est celui des interrogatives ouvertes et des relatives, mais les chaînes $\overline{\text{A}}$ se rencontrent également dans les constructions d'extraposition de l'objet ("tough-movement") et de topicalisation, bref ce que l'on appelle les constructions *wh*, et dont quelques exemples sont donnés en (33)-(36) :

(33)a. [$_{\text{DP}}$ who]$_i$ t_i saw John?

b. [$_{\text{DP}}$ what]$_i$ did John see t_i

c. [$_{\text{DP}}$ who]$_i$ did John give the book to t_i

d. [$_{\text{DP}}$ who]$_i$ did John say he gave the book to t_i

e. [$_{\text{PP}}$ to whom]$_i$ did John give the book t_i

f. [$_{\text{DP/AdvP}}$ when]$_i$ did John give the book to Bill t_i

g. [$_{\text{AdvP}}$ why]$_i$ did John give the book to Bill t_i

h. [$_{\text{DP}}$ which day]$_i$ did John give the book to Bill t_i

i. [$_{\text{AdjP}}$ how raw]$_i$ did John eat the meat t_i

(34)a. the man who$_i$ t_i saw John?

b. the book that John saw t_i

c. the man who$_i$ John gave the book to t_i

d. the man to whom$_i$, John gave the book t_i

e. the day John gave the book to Bill t_i

f. the reason why$_i$ John gave the book to Bill t_i

g. the way John gave the book to Bill t_i

h. the degree of rawness that John ate the meat t_i

(35) This violin is easy to play this sonata on t_i

(36)a. [$_{\text{DP}}$ this book]$_i$, John saw t_i

b. [$_{\text{DP}}$ this man]$_i$, John gave the book to t_i

c. [$_{\text{PP}}$ to this man]$_i$, John gave the book t_i

d. John promised he would run away and [$_{\text{VP}}$ run away]$_i$ he will t_i

D'une façon générale, la création de chaînes $\overline{\text{A}}$ et l'insertion de traces sont fonction des éléments suivants :

1. La présence d'un élément $\overline{\text{A}}$ en positon spécificateur de C.

2. La catégorie grammaticale de l'élément $\overline{A}$ qui peut être
   DP, *e.g. who*
   PP, *e.g. for whom*
   AdvP *e.g. when*
   AP *how beautiful*

3. La nature du gouverneur local, c'est-à-dire

N	- spec, T	sujet
	- compl, T/V	objet
	- compl, P	objet d'une préposition
P	- compl, T/V	objet prépositionnel d'un verbe
	- compl, A	objet prépositionnel d'un adjectif
	- compl, N	objet prépositionnel d'un nom
A/Adv	- compl T,V	ajout

Outre le repérage d'un élément-*wh* et la construction des chaînes, le traitement des éléments $\overline{A}$ fait intervenir la vérification de la bonne formation des structures contenant des chaînes $\overline{A}$. Le problème, ici, est de vérifier que toutes les chaînes d'une structure se terminent. En effet, on ne peut pas établir *a priori* la longueur d'une chaîne. D'autre part, étant donné la nature de notre algorithme, une chaîne est créée aussitôt que sa tête apparaît. Il faut ensuite garantir la présence de la queue de la chaîne, qui encore une fois, peut survenir beaucoup plus tard. Concrètement, il s'agit de marquer (37a) comme bien-formé et (37b) comme malformé :

(37)a. who$_i$ has John loved e$_i$

b. who$_i$ has John loved

La condition utilisée ici est que toutes les chaînes se terminent, autrement dit que la liste des élément $\overline{A}$ associée à un constituant soit vide.

3.11 Le problème du filtrage

Le problème du filtrage est particulièrement aigu étant donné la nature modulaire de l'analyseur. En effet, le module $\overline{X}$ est fort peu contraint dans son application, ce qui entraîne une abondante surgénération. Il est par conséquent nécessaire de filtrer ces nombreuses structures pour éviter une explosion combinatoire (point de vue computationnel) ainsi que la génération d'alternatives non-motivées (point de vue linguistique et psycholinguistique).

La sélection de structures se fait à deux niveaux différents et sur la base de principes tout à fait distincts. Il y a d'une part la sélection grammaticale, c'est-à-dire la sélection effectuée par l'application d'autres modules syntaxiques, tels que le module des cas ou celui des fonctions thématiques. Ces modules fonctionnent en effet comme des contraintes de bonne formation sur l'ensemble des structures engendrées par le module $\overline{X}$, et de ce fait fonctionnent comme des filtres, rejetant toutes les alternatives qui ne satisfont pas certaines contraintes. Par exemple, le module des cas ou le module thématique rejetteront les structures qui ne satisfont pas respectivement au filtre casuel et au critère-θ. Ces modules n'ont pas encore été implémentés.

A côté de cette sélection grammaticale, il est un autre type de sélection, qui ne repose pas sur des critères purement grammaticaux mais davantage sur des heuristiques de nature psycholinguistique. C'est ce deuxième type de sélection qui permet d'établir des préférences parmi des structures bien formées syntaxiquement.

Ainsi, en l'absence de marques contrastives, on préférera sans doute la lecture (b) à la lecture (c) des exemples ci-dessous :

(38)a. Ils aiment tous les bons vins

b. Ils aiment [$_{DP}$ tous les bons vins]

c. Ils aiment [$_{Adv}$ tous] [$_{DP}$ les bons vins]

Ce deuxième type de sélection n'a pas encore été implémenté.

Bibliographie

Berwick, R. 1987. "Principle-Based Parsing", Technical report, MIT AI Lab.

Clark, R. 1990a. "The Auxiliary System of English", notes techniques 90/1, projet IPS Université de Genève.

Clark, R. 1990b. "Some Determiners and Partitives", notes techniques 90/2, projet IPS, Université de Genève.

Clark, R. 1991a. "Chain Formation I", notes techniques 91/1, projet IPS, Université de Genève.

Clark, R. 1991b. "Some Aspects of Sentential Complementation in English", notes techniques 91/2, projet IPS, Université de Genève.

Clark, R. 1991c. "Possessives, Gerunds and Contractions", notes techniques 91/3, projet IPS, Université de Genève.

Clark, R. 1991d. "A Simple Procedure for Coordination", notes techniques 91/4, projet IPS, Université de Genève.

Clark, R. 1991e. "Considering Comparatives", notes techniques 91/5, projet IPS, Université de Genève.

Clark, R. 1991f. "Relative Clauses", notes techniques 91/6, projet IPS, Université de Genève.

Marcus, M. 1980. *A Theory of Syntactic Recognition for Natural Langague*, MIT Press.

Rizzi, L. 1986. "On Chain Formation" in H. Borer (éd.) *The Syntax of Pronominal Clitics*, Academic Press.

Walther, C. 1991. "The lexical database for English", notes techniques 91/8, projet IPS, Université de Genève.

Wehrli, E. 1988. "Parsing with a GB Grammar" in U. Reyle and C. Rohrer (eds.) *Natural Language Parsing and Linguistic Theories*, Reidel, 1988, 177-201.