5 - Analyse syntaxique

1. Syntagmes et dépendances
2. Grammaires algébriques
3. Analyse syntaxique tabulaire

5.1 - Syntagmes et dépendances

- Les grammaires syntagmatiques sont fondées sur l'idée que les mots d'une phrase se regroupent en **syntagmes** qui constituent des entités autonomes.

- Pour une langue donnée, la délimitation des syntagmes dans les énoncés et leur classement par catégories sont fondés sur des critères **distributionnels**.

- La syntaxe d'une phrase se présente alors sous forme d'un arbre de syntagmes ou **arbre syntagmatique**.

5.1 - Syntagmes et dépendances

- Critère d'interchangeabilité :

  Jean vient.

  Le printemps vient.

  Le garçon vient.

  Un garçon vient.

  Jean voit un garçon.

  Jean connaît un garçon.

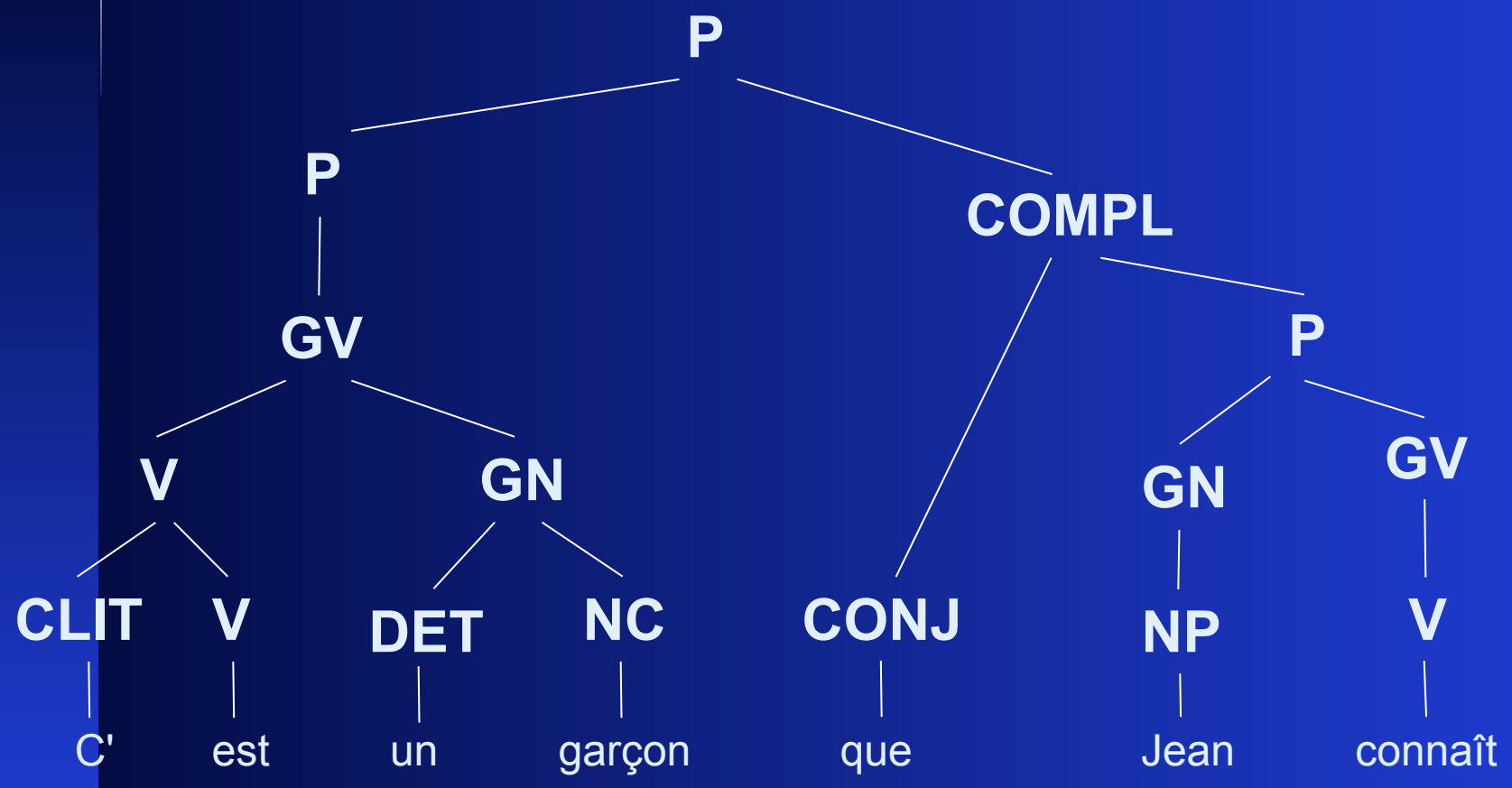
- Critère d'antéposition et de postposition :

  C'est un garçon que Jean connaît.

  **C'est un que Jean connaît garçon.*

5.1 - Syntagmes et dépendances

- Arbre syntagmatique de la phrase *"c'est un garçon que Jean connaît "*



5.1 - Syntagmes et dépendances

- Les grammaires de dépendances sont fondées sur la notion de **dépendance syntaxique** entre mots.

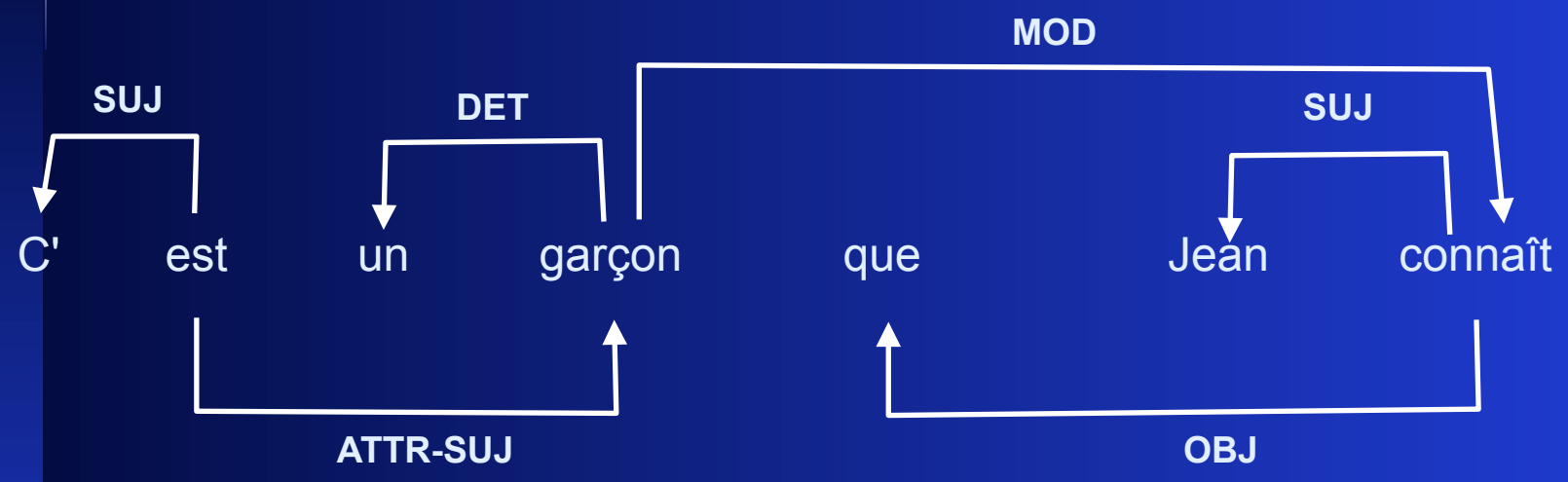
- Une dépendance syntaxique est formée d'un **gouverneur** et d'un **dépendant** et elle est étiquetée par une fonction syntaxique.

- La syntaxe d'une phrase se présente alors comme un **graphe de dépendances** qui est le plus souvent un **arbre de dépendances**.

- L'arbre est **projectif** quand les dépendances ne se croisent pas selon l'ordre des mots dans la phrase.

5.1 - Syntagmes et dépendances

- Arbre de dépendances de la phrase *"c'est un garçon que Jean connaît "*



5.2 - Grammaires algébriques

- Dans une vision syntagmatique de la syntaxe, la question est de proposer un formalisme qui permette d'exprimer les règles de construction des arbres syntagmatiques.

- Presque simultanément, Chomsky (1956) et Backus (1959) ont conçu un formalisme, les **grammaires algébriques**, qui a été utilisé pour représenter les grammaires syntagmatiques.

- Dans ce formalisme, une grammaire se présente comme un ensemble de **règles de réécriture** de suite de syntagmes. Ces règles vont permettre d'engendrer tous les arbres syntagmatiques des phrases de la langue.

5.2 - Grammaires algébriques

- Une grammaire algébrique est un quadruplet (N, T, S, R) tel que:
  - ✓ N est un alphabet fini de symboles non terminaux.
  - ✓ T est un alphabet fini de symboles terminaux.
  - ✓ S est un élément particulier de N, le symbole de départ.
  - ✓ R est un ensemble fini de règles de production de la forme $A \rightarrow \alpha$, où A est un non terminal et α un mot de $(N \cup T)^*$.

5.2 - Grammaires algébriques

- La relation de **dérivation** $\Rightarrow$ entre mots de $(N \cup T)^*$ est définie ainsi : $\alpha_1 A \alpha_2 \Rightarrow \alpha_1 \alpha \alpha_2$ si $A \rightarrow \alpha \in R$. Sa clôture réflexive et transitive est notée : $\Rightarrow^*$

- Le langage engendré par la grammaire est l'ensemble des mots α de T^* tels que : $S \Rightarrow^* \alpha$. Un langage engendré par une grammaire algébrique est appelé un **langage algébrique**.

5.2 - Grammaires algébriques

- Grammaire algébrique définie par les règles ci-dessous et le symbole de départ P:

P → GN Vintr

P → GN Vtr GN

P → NP Vc Compl

P → P GP

GN → Det N

GN → GN GP

Compl → Conj P

GP → Prep GN

GN → jean

Det → le

N → bébé

N → berceau

Vintr → dort

Vtr → porte

Vc → pense

Prep → dans

Conj → que

5.2 - Grammaires algébriques

- Une **dérivation** d'un mot α du langage est une suite $S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha$.

- Une dérivation est représentée de manière compacte par un **arbre d'analyse,** qui fait abstraction de l'ordre d'application des règles.

- Un arbre d'analyse est un arbre ordonné étiqueté par des symboles de $N \cup T$. La racine est étiquetée par S. Les feuilles sont étiquetés par des symboles terminaux. Chaque nœud qui n'est pas une feuille est étiqueté par un non terminal A et ses fils ordonnés sont étiquetés par des symboles constituant un mot α tel que $A \rightarrow \alpha$ est une règle de production de la grammaire.

5.2 - Grammaires algébriques

- Le **langage d'arbres** engendré par une grammaire est l'ensemble des arbres de dérivation de cette grammaire.

- Deux grammaires sont **faiblement équivalentes** si elles engendrent le même langage de mots.

- Deux grammaires sont **fortement équivalentes** si elles engendrent le même langage d'arbres, au renommage des non terminaux près.

- Une grammaire est **ambiguë** si elle permet de produire deux arbres d'analyse différents pour le même mot.

5.2 - Grammaires algébriques

1. **De la grammaire au langage**

   Déterminer les langages engendrés par les grammaires suivantes (dans chacun des cas, S est le symbole de départ). S'ils sont ambigus, le montrer à l'aide d'un exemple.

   a) $S \rightarrow \varepsilon \mid aaaS$

   b) $S \rightarrow SS \mid a$

   c) $S \rightarrow aA \mid Bb \qquad A \rightarrow Ab \mid b \qquad B \rightarrow aB \mid a$

   d) $S \rightarrow \varepsilon \mid aSa \mid bSb \mid cSc$

2. **Du langage à la grammaire**

   Définir des grammaires algébriques engendrant les langages suivants.

   a) $L_1 = \{ a^n b^p \mid 0 < p < n \}$

   b) $L_2 = \{ a^n b^n c^m d^m \mid n, m \in \mathbb{N} \}$

   c) $L_3 = \{ a^n b^m c^p \mid n = m \text{ or } p = m \}$

5.2 - Grammaires algébriques

3. **Grammaire du français**

Etendre la grammaire du français du cours et avec cette grammaire, analyser les phrases suivantes :

a) *Le beau bébé dans le berceau dort profondément.*

b) *Jean voit le bébé endormi dans le berceau dans la chambre.*

c) *Jean pense que dans le berceau le bébé dort profondément.*

d) *Que voit Jean ?*

e) *Que voit Jean dans la chambre ?*

f) *Le beau bébé que Jean voit dort.*

g) *Le beau bébé que Jean pense que Marie garde dort dans la chambre.*

Discuter de la pertinence de la grammaire construite par rapport au problème de la surgénération

5.3 - Analyse syntaxique tabulaire

- Le but de l'analyse syntaxique est de trouver tous les arbres d'analyse pour une phrase et une grammaire données.

- Une **analyse ascendante** ou guidée par les données cherche à construire les arbres d'analyse des feuilles à la racine. Les règles de la grammaire sont utilisées de droite à gauche.

- Une **analyse descendante** ou guidée par le but cherche à construire les arbres d'analyse de la racine vers les feuilles. Les règles de la grammaire sont utilisées de gauche à droite.

- L'inconvénient d'une analyse ascendante est de construire des arbres inutiles qui n'entreront jamais dans un arbre dont la racine représente une phrase. L'inconvénient d'une analyse descendante est de construire d'autres arbres inutiles dont la liste des feuilles entre en contradiction avec la phrase à analyser.

5.3 - Analyse syntaxique tabulaire

- La **tabulation** ou **mémoïsation** est une technique caractéristique de la **programmation dynamique** qui utilise une table pour stocker les résultats intermédiaires pour éviter de les recalculer en cas de besoin.

- Les données calculées sont rangées sous forme d'items dans une **table**.

- Dans le cas de l'analyse syntaxique, les règles de calcul des items peuvent être présentées sous forme d'un **système déductif**.

- La signification des items est donnée par un **invariant** qui est préservé par les règles de déduction.

- Pour gérer la façon dont les règles sont appliquées, on utilise un **agenda** où sont stockés les items actifs, déclencheurs des règles et une stratégie d'ordonnancement des items dans l'agenda.

5.3 - Analyse syntaxique tabulaire

- L'algorithme de Cocke (1970) - Kasami (1965) - Younger (1967) (CKY ou CYK) est une méthode tabulaire d'analyse ascendante des grammaires algébriques.

- Soit $w_1\, w_2 \ldots w_n$ une phrase de n mots à analyser à l'aide d'une grammaire algébrique G.

  L'algorithme de CKY peut être traduit par un système déductif manipulant des items de la forme [i, j, A] avec l'invariant suivant : le segment $w_{i+1} \ldots w_j$ est un mot A (terminal) ou un syntagme de type A (non terminal).

  Les règles d'inférence des items sont les suivantes :

$$\frac{}{[\, i,\ i+1,\ w_{i+1}]}\ \text{Init}$$

$$\frac{[i_0, i_1, A_1]\ [i_1, i_2, A_2]\ \ldots.\ [i_{p-1}, i_p, A_p]}{[i_0, i_n, A]}\ \text{Compl} \quad \text{si} \quad A \rightarrow A_1\, A_2 \ldots.\ A_p \text{ est une règle de G}$$

- La complexité en temps de l'algorithme est de l'ordre de n^3, n étant la longueur de la phrase.

5.3 - Analyse syntaxique tabulaire

- Grammaire algébrique définie par les règles ci-dessous et le symbole de départ P:

P → GN Vintr

P → GN Vtr GN

P → NP Vc Compl

P → P GP

GN → Det N

GN → GN GP

Compl → Conj P

GP → Prep GN

GN → jean

Det → le

N → bébé

N → berceau

Vintr → dort

Vtr → porte

Vc → pense

Prep → dans

Conj → que

5.3 - Analyse syntaxique tabulaire

1. A l'aide de l'algorithme de CKY et de la grammaire construite précédemment en exercice, analyser les phrases :

   a) *Le beau bébé dans le berceau dort profondément.*

   b) *Jean voit le bébé endormi dans le berceau dans la chambre.*

   c) *Jean pense que dans le berceau le bébé dort profondément.*

2. Analyser la phrase « *le joueur de football américain arrive* » en utilisant l'algorithme de CKY avec la grammaire suivante :

S → NP VP	Adj → américain
NP → Det N	Det → le
N → N PP | N Adj	N → football | joueur
PP → Prep N	Prep → de
	VP → arrive