Grammaires formelles : Grammaires hors-contexte

Karën Fort

karen.fort@sorbonne-universite.fr / https://members.loria.fr/KFort

27 novembre 2020

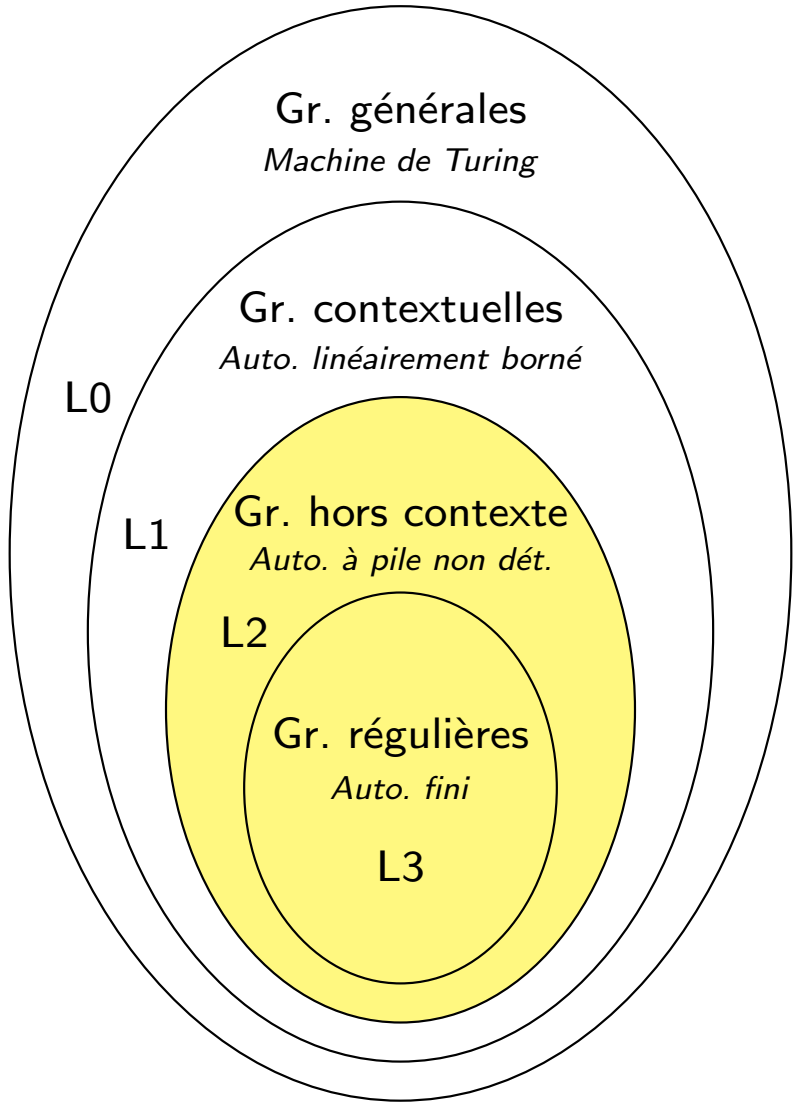
LETTRES SORBONNE UNIVERSITÉ

Quelques sources d'inspiration

par ordre d'importance décroissant

- cours de D. Battistelli (Paris 3), grâce aux notes de C. Riquier (Master 2, Paris 4)
- cours d'A. Rozenknop (Paris 13)
- cours de B. Habert (ex ENS de Lyon)
- M. Cori (Nanterre)
- *Language as a cognitive process - Syntax* (Terry Winograd) – Addison Wesley
- *Introduction à la calculabilité* (Pierre Wolper) – InterEditions, 1991
- cours en ligne de C. Touratier (U. de Provence) http://christian.touratier.pagesperso-orange.fr/html/Formalisation%20Analyse%20CI.htm
- Wikipédia sur les automates à pile (en anglais) http://en.wikipedia.org/wiki/Pushdown_automaton

Place dans la hiérarchie



Dénomination

Les grammaires de type 2 sont aussi appelées :

- hors-contexte
- indépendantes du contexte
- algébriques
- libre du contexte (horrible anglicisme)
- *context-free*

Type 2 : grammaires hors-contexte

Définition

Une grammaire de type 2 est une grammaire dont les parties gauches des règles contiennent un unique non-terminal, indépendamment du contexte dans lequel il apparaît :

$$A \longrightarrow \alpha \text{ avec } \begin{cases} A \in V_N \\ \alpha \in V^* \end{cases}$$

 On peut avoir autant de terminaux et de non-terminaux que voulu à droite

 On ne peut remplacer qu'un élément non terminal lors d'une dérivation, les dérivations engendrent donc des chaînes qui ne diminuent jamais

Exemple de grammaire hors-contexte

Traitement des relatives emboîtées [Desclés 1973]

"Noémon qui adore Marie qui admire Jean ... est vieux et sage"

$$
P = \begin{cases}
S \longrightarrow \text{SN SV} & (1) \\
SN \longrightarrow \text{Nm REL} & (2) \\
SV \longrightarrow \text{est ATT} \mid \text{Vb SN} \mid \text{Vb Nm} & (3) \\
REL \longrightarrow \text{QU SV} & (4) \\
ATT \longrightarrow \text{ADJ ATT} \mid \text{et ADJ} & (5) \\
QU \longrightarrow \text{qui} & (6) \\
Nm \longrightarrow \text{Noémon} \mid \text{Marie} \mid \text{Jean} & (7) \\
Vb \longrightarrow \text{adore} \mid \text{admire} & (8) \\
ADJ \longrightarrow \text{vieux} \mid \text{sage} & (9)
\end{cases}
$$

Exemples de grammaires hors-contexte

Le langage $\{a^n b^n | n > 0\}$:

$$P = \begin{cases} S \longrightarrow aSb & (1) \\ S \longrightarrow ab & (2) \end{cases}$$

Exemples de grammaires hors-contexte

Le langage $\{a^n b^n | n > 0\}$:

$$P = \begin{cases} S \longrightarrow aSb & (1) \\ S \longrightarrow ab & (2) \end{cases}$$

Le langage des palindromes :

$$P = \begin{cases} S \longrightarrow \varepsilon & (1) \\ S \longrightarrow aSa & (2) \\ S \longrightarrow bSb & (3) \end{cases}$$

Exemples de grammaires hors-contexte

Le langage $\{a^n b^n | n > 0\}$:

$$P = \begin{cases} S \longrightarrow aSb & (1) \\ S \longrightarrow ab & (2) \end{cases}$$

Le langage des palindromes :

$$P = \begin{cases} S \longrightarrow \varepsilon & (1) \\ S \longrightarrow aSa & (2) \\ S \longrightarrow bSb & (3) \end{cases}$$

Le langage des mots contenant autant de a que de b :

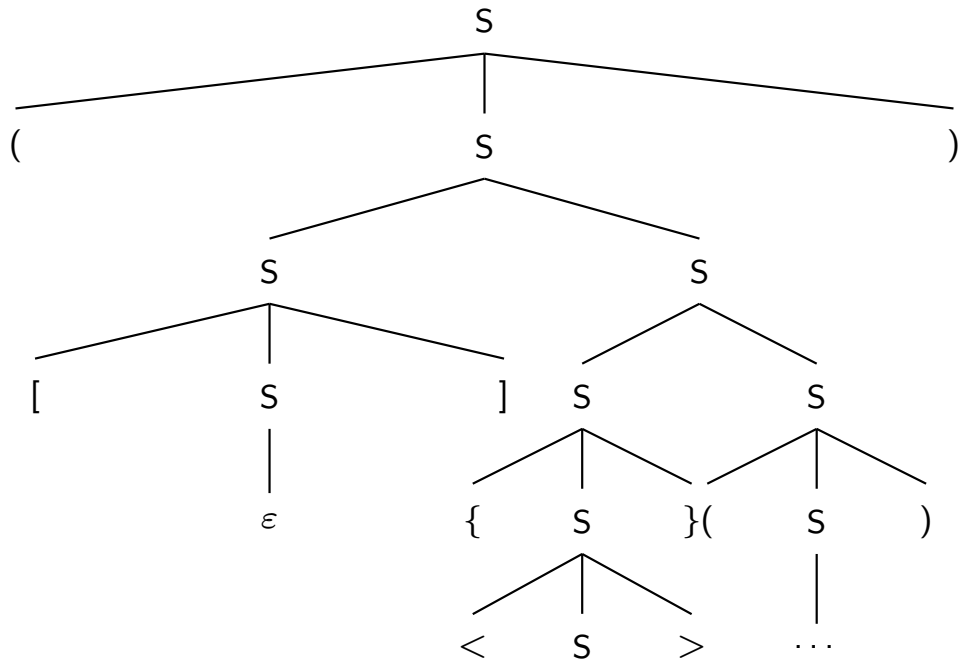
$$P = \begin{cases} S \longrightarrow SS & (1) \\ S \longrightarrow aSb & (2) \\ S \longrightarrow bSa & (3) \\ S \longrightarrow \varepsilon & (4) \end{cases}$$

Langage des systèmes bien parenthésés

$$P = \begin{cases} S \longrightarrow SS & (1) \\ S \longrightarrow (S) & (2) \\ S \longrightarrow [S] & (3) \\ S \longrightarrow \{S\} & (4) \\ S \longrightarrow < S > & (5) \\ S \longrightarrow \varepsilon & (6) \end{cases}$$

 Dériver "([]{<<>>}([(){<>}]))". Comparer cette grammaire avec la grammaire vue lors du cours précédent

Structure des grammaires hors-contexte



Structure en arbre dont les éléments sont en position contiguë (adjacente)

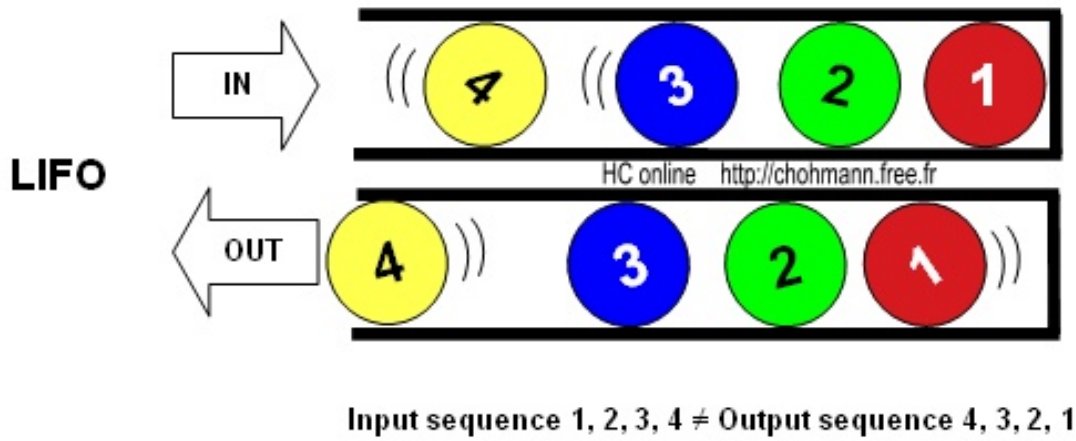
Vers l(a mémoire) infini(e)

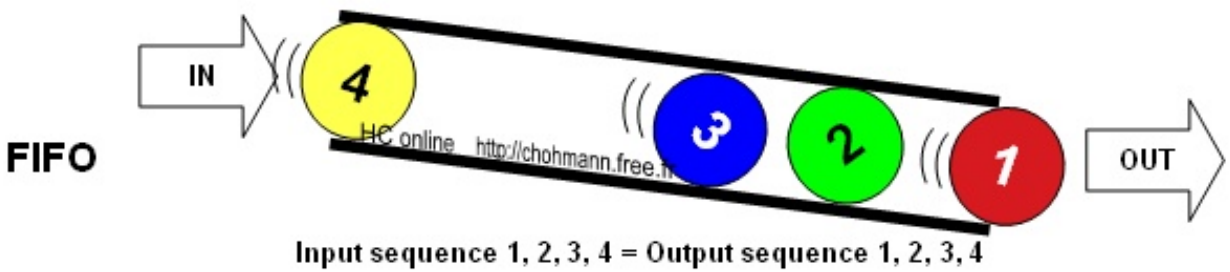
Automates à pile

Les automates à piles sont des automates finis auxquels on a adjoint une mémoire de capacité non bornée.

Cette mémoire a la structure d'une pile (*stack*), c'est-à-dire d'une file à accès LIFO (*Last In, First Out*, dernier arrivé, premier parti)

Parenthèse : LIFO *vs* FIFO

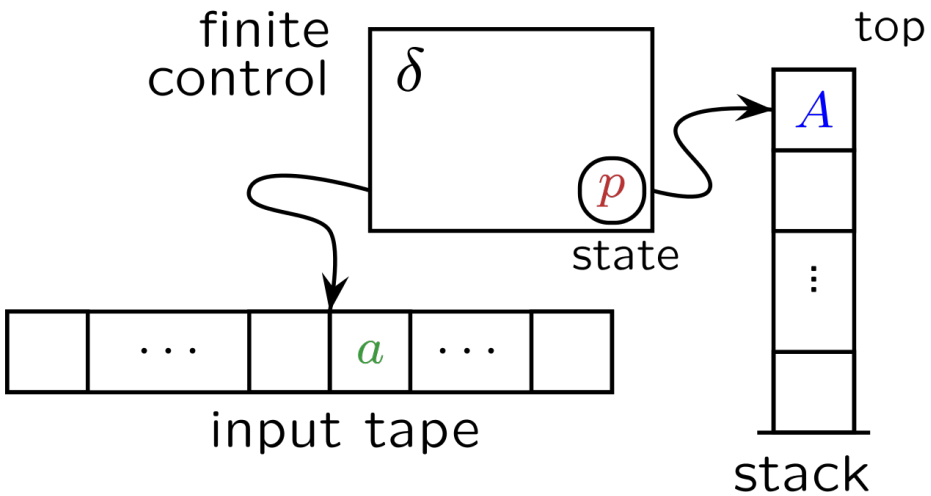




[© Christian Hohmann avec son accord]

Automate à pile (LIFO) ou PDA

non déterministe



[Jochgem, CC BY-SA]

- un ensemble fini d'états
- un alphabet d'entrée
- un alphabet de pile
- un symbole initial de pile
- un état initial
- un ou des états accepteurs (ou finaux)
- une relation de transition

Les transitions sont semblables à celles d'un automate fini non déterministe, et spécifient en plus la manipulation de la pile

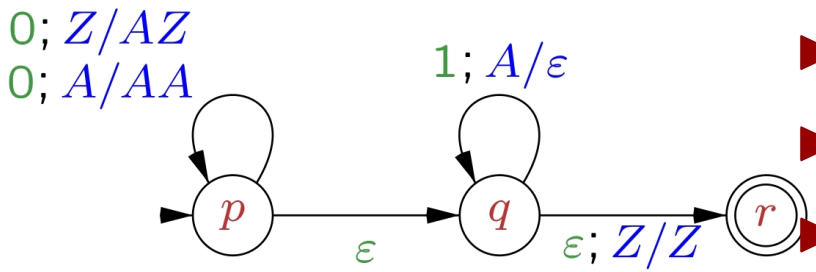
Configurations acceptantes

- Reconnaissance par pile vide : arriver à vider entièrement la pile au moment où on termine la lecture du mot
- Reconnaissance par état final : la pile n'est pas nécessairement vide à la fin de la lecture du mot
- Reconnaissance par pile vide et état final : la pile est vide à la fin de la lecture du mot

Exemple d'automate à pile

pour le langage $\{0^n 1^n | n \geq 0\}$

(pas de représentation standard)



[Jochgem, CC BY-SA]

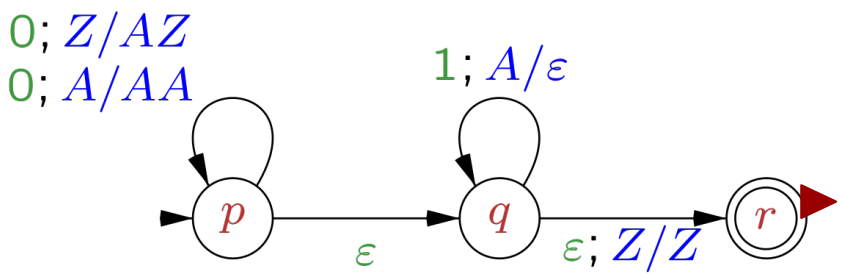
- états : $Q = \{p, q, r\}$
- alphabet d'entrée : $\Sigma = \{0, 1\}$
- alphabet de pile : $\Gamma = \{A, Z\}$
- état initial : $q_0 = p$
- symbole de fond de pile : Z
- état final :$F = \{r\}$

Relation de transition (6 instructions) :
$(p, 0, Z, p, AZ)$, $(p, 0, A, p, AA)$, (p, ϵ, Z, q, Z), (p, ϵ, A, q, A), $(q, 1, A, q, \epsilon)$, et (q, ϵ, Z, r, Z)

Exemple d'automate à pile

pour le langage $\{0^n 1^n | n \geq 0\}$

(pas de représentation standard)

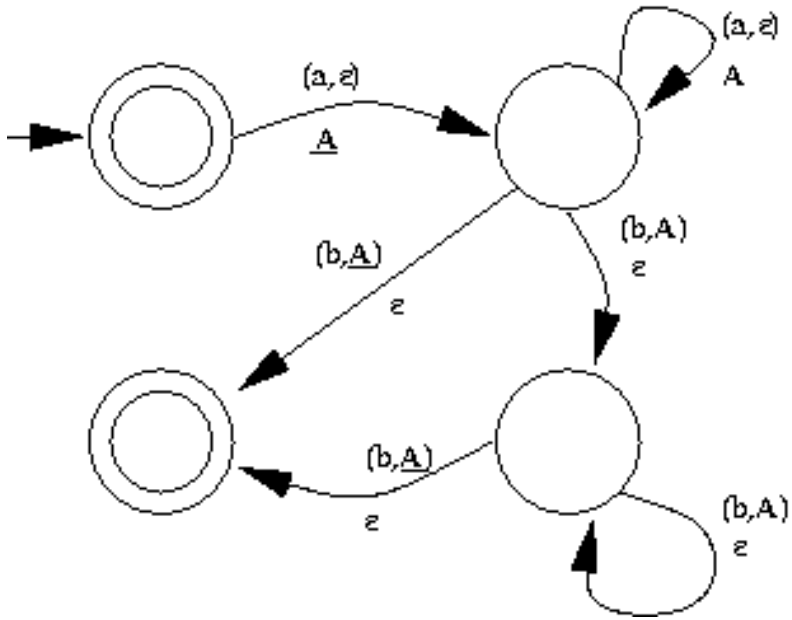


[Jochgem, CC BY-SA]

- instructions 1 et 2 : à l'état p, quand 0 est lu, 1 A est ajouté sur la pile
- instructions 3 et 4 : l'automate peut passer à n'importe quel moment de l'état p à l'état q
- instruction 5 : dans l'état q quand 1 est lu, 1 A est enlevé de la pile
- instruction 6 : l'automate ne peut passer de l'état q à l'état r que lorsque la pile ne contient plus qu'un Z

Relation de transition (6 instructions) :
$(p, 0, Z, p, AZ)$, $(p, 0, A, p, AA)$, (p, ϵ, Z, q, Z), (p, ϵ, A, q, A), $(q, 1, A, q, \epsilon)$, et (q, ϵ, Z, r, Z)

Exercice (avec une autre représentation)



[Vincent Balat, CC BY-SA]

Quel langage représente cet automate ?

Déterminisme *vs* non déterminisme

Attention

≪ Contrairement à ce qui se passe pour les automates finis, il n'est pas possible en général de trouver un automate à pile déterministe qui reconnaît le même langage qu'un automate à pile non-déterministe donné ≫ [JF Perrot, cours 18]

Conséquence

Il existe des langages hors-contexte qui ne sont acceptés par aucun automate à pile déterministe. Les langages acceptés par les automates à pile déterministes forment donc une sous-classe des langages hors-contexte (langages hors-contexte déterministes). Les grammaires dites LR décrivent de tels langages.

Reformulation de la définition

≪ Ce genre d'analyse est taxinomique, puisque chaque "histoire de production" d'une phase reste classificatoire : les symboles des catégories (syntagmatiques) représentent des classes d'équivalence d'objets observables [...]. L'histoire de production est en fait une hiérarchie de partitions de plus en plus fines, chaque partition représentant une segmentation classificatoire de la phrase. ≫

[Desclés, 1982, p 92]

Exemple (d'un sous-ensemble) des insultes du Capitaine Haddock

reformulation

$$
P = \begin{cases}
S \longrightarrow \text{GN-non-determiné} & (1) \\
\text{GN-non-determiné} \longrightarrow \text{N} \mid \text{N ADJ} \mid \text{N Gprep} \mid \text{N ADJ Gprep} & (2) \\
\text{Gprep} \longrightarrow \text{Prep-de GN-non-déterminé} \mid \text{Prep-à GN-déterminé} & (3) \\
\text{GN-déterminé} \longrightarrow \text{Det N} \mid \text{Det N Gprep} & (4) \\
\text{Prep-de} \longrightarrow \text{de} \mid \text{d'} & (5) \\
\text{Prep-à} \longrightarrow \text{à} & (6) \\
\text{N} \longrightarrow \text{amiral} \mid \text{analphabète} \mid \text{guano} \mid \text{athlète} \mid \text{bougre} \mid \text{Vercingétorix} \mid \ldots & (7) \\
\text{ADJ} \longrightarrow \text{complet} \mid \text{simili} \mid \text{tartare} \mid \text{faux} \mid \text{diplômé} & (8) \\
\text{Det} \longrightarrow \text{la} & (9)
\end{cases}
$$

Le recours à une grammaire hors-contexte permet de rendre compte des proximités structurales et de la récursivité des insultes.

Exemple (d'un sous-ensemble) des insultes du Capitaine Haddock

reformulation

$$P = \begin{cases} S \longrightarrow \text{GN-non-determiné} & (1) \\ \text{GN-non-determiné} \longrightarrow \text{N} \mid \text{N ADJ} \mid \text{N Gprep} \mid \text{N ADJ Gprep} & (2) \\ \text{Gprep} \longrightarrow \text{Prep-de GN-non-déterminé} \mid \text{Prep-à GN-déterminé} & (3) \\ \text{GN-déterminé} \longrightarrow \text{Det N} \mid \text{Det N Gprep} & (4) \\ \text{Prep-de} \longrightarrow \text{de} \mid \text{d'} & (5) \\ \text{Prep-à} \longrightarrow \text{à} & (6) \\ \text{N} \longrightarrow \text{amiral} \mid \text{analphabète} \mid \text{guano} \mid \text{athlète} \mid \text{bougre} \mid \text{Vercingétorix} \mid \ldots & (7) \\ \text{ADJ} \longrightarrow \text{complet} \mid \text{simili} \mid \text{tartare} \mid \text{faux} \mid \text{diplômé} & (8) \\ \text{Det} \longrightarrow \text{la} & (9) \end{cases}$$

Construire les structures pour les insultes : "bougre d'extrait d'hydrocarbure" et "extrait de guano diplômé"

Parenthèse sur la récursivité

- Certaines formes de récursivité peuvent être réglées par une grammaire régulière :

  ≪ nom de dieu de nom de dieu de... ≫

- Expression régulière :
  nom de dieu (de nom de dieu)*

- Grammaire :
  nom-de-coton $\longrightarrow$ nom de dieu | nom de dieu de nom-de-coton

$\rightarrow$ capacités limitées des grammaires régulières

Expressivité des grammaires hors-contexte

Les grammaires hors-contexte sont adaptées pour :

- les chaînes $a^n b^n$ de type $a_1 \dots a_n b_n \dots b_1$ (expressions parenthésées, structures enchâssées)

  $$\begin{array}{rcl} \text{P} & \longrightarrow & \text{SN SV} \\ \text{SN} & \longrightarrow & \text{SN (P)} \end{array}$$ peut analyser :

  The dog the stick the fire burned beat bit the cat

Expressivité des grammaires hors-contexte

Les grammaires hors-contexte sont adaptées pour :

- les chaînes $a^n b^n$ de type $a_1 \ldots a_n b_n \ldots b_1$ (expressions parenthésées, structures enchâssées)
- les chaînes *abac*

  $soit \ldots \ soit \ldots \ (X \longrightarrow soit \ Y \ soit \ Y)$

  $ni \ldots \ ni \ldots \ (X \longrightarrow ni \ Y \ ni \ Y)$

Applications

Principalement la syntaxe :

- des langues naturelles
- des langages de programmation

Analyse syntaxique

Le problème de l'analyse syntaxique est de déterminer si un mot (une phrase) est dans le langage généré par la grammaire hors-contexte et d'établir comment ce mot (cette phrase) a été généré.

Ce problème est soluble par un algorithme pour n'importe quelle grammaire hors-contexte. Toutefois, pour obtenir des algorithmes d'analyse efficaces il faut imposer des restrictions sur le type de grammaire hors-contexte que l'on peut utiliser.

Limitations

- Les grammaires hors-contexte traitent avec difficulté le rejet en fin de phrase des prépositions (en anglais) ou des particules séparables (en allemand).

  Solution : dupliquer les règles pour chaque préposition ou particule

Limitations

- Les grammaires hors-contexte traitent avec difficulté le rejet
- Elles ne peuvent traiter les dépendances à longue distance (interrogatives, clivées. . .) , le problème étant par exemple de contraindre un accord selon un syntagme qui sort du contexte de cet accord :
  Jean veut savoir quelle fille Marie croit que Paul a vue.

Limitations

- Les grammaires hors-contexte traitent avec difficulté le rejet
- Elles ne peuvent traiter les dépendances à longue distance (interrogatives, clivées...)
- Ni les rares langues qui ont des structures de type $a^n b^m c^n d^m$

Limitations

- Les grammaires hors-contexte traitent avec difficulté le rejet
- Elles ne peuvent traiter les dépendances à longue distance (interrogatives, clivées...)
- Ni les rares langues qui ont des structures de type $a^n b^m c^n d^m$
- Ni les chaînes $a^n b^n$ de type $a_1 \ldots a_n b_1 \ldots b_n$ ("respectivement")

  Henri et Sophie sont respectivement indifférent et séduite par le film.

Limitations

- Les grammaires hors-contexte traitent avec difficulté le rejet
- Elles ne peuvent traiter les dépendances à longue distance (interrogatives, clivées...)
- Ni les rares langues qui ont des structures de type $a^n b^m c^n d^m$
- Ni les chaînes $a^n b^n$ de type $a_1 \dots a_n b_1 \dots b_n$ ("respectivement")

Ces arguments ne sont pas vraiment décisifs pour mettre de côté ces grammaires pour le TAL, car en pratique, les ^{n} et ^{m} ne sont jamais grands.

Capacité générative faible *vs* forte

≪ Étant donné une théorie descriptive de la structure linguistique, nous pouvons distinguer sa capacité générative au sens faible et sa capacité générative au sens fort : nous dirons qu'une grammaire engendre au sens faible un ensemble de phrases et engendre au sens fort un ensemble de descriptions structurales, chaque description structurale spécifiant une phrase de manière unique, mais l'inverse n'étant pas nécessairement vrai ≫

[Chomsky, 1971, p 86-87]

Capacité générative faible *vs* forte

> *« Étant donné une théorie descriptive de la structure linguistique, nous pouvons distinguer sa capacité générative au sens faible et sa capacité générative au sens fort : nous dirons qu'une grammaire engendre au sens faible un ensemble de phrases et engendre au sens fort un ensemble de descriptions structurales, chaque description structurale spécifiant une phrase de manière unique, mais l'inverse n'étant pas nécessairement vrai »*
>
> *[Chomsky, 1971, p 86-87]*

Une grammaire formelle peut avoir une capacité générative faible acceptable, mais une capacité générative forte insuffisante

Capacité générative forte

≪ la capacité générative au sens faible est d'un intérêt assez marginal pour la linguistique ≫

[Chomsky, 1971, p 87]

Capacité générative forte

> ≪ *la capacité générative au sens faible est d'un intérêt assez marginal pour la linguistique* ≫
>
> *[Chomsky, 1971, p 87]*

 ≪ Démontrer qu'une grammaire donnée ne peut pas engendrer la description structurale de certaines phrases d'une langue particulière est une condamnation de cette grammaire ; car cela revient à dire qu'elle est incapable de rendre compte de la structure syntaxique des phrases de cette langue ≫ [C. Touratier]

Grammaires hors-contexte :

- définition
- structure, limites
- récursivité

Automates à pile :

- définition
- fonctionnement

Capacité générative

Exercice (tableau) : de quel type sont les règles

$S \longrightarrow D\ N$
$Wa \longrightarrow D$
$S \longrightarrow aW$
$DN \longrightarrow a$
$a \longrightarrow A$
$A \longrightarrow a$

Exercice : structures

Donnez, pour chacune de ces expressions, une structure les représentant et une grammaire permettant de les générer :

le départ de la fille de ma voisine
le départ de ma fille de la maison

Est-il possible de les représenter avec un autre type de grammaire, avec une autre structure ? (Dé)montrez-le.