

Notes du module de grammaires lexicalisées

Hugo Mougard

17 septembre 2013

0.1 Grammaires lexicalisées

Une grammaire lexicalisée est une grammaire dont toutes les règles contiennent au moins un terminal.

Forme normale de Greibach

Concerne les langages propres (sans ϵ) dont les grammaires (propres elles aussi, ou réduites) sont sous la forme

$$V \rightarrow XV^*$$

où X est terminal et V non-terminal.

Par exemple, voici une grammaire lexicale :

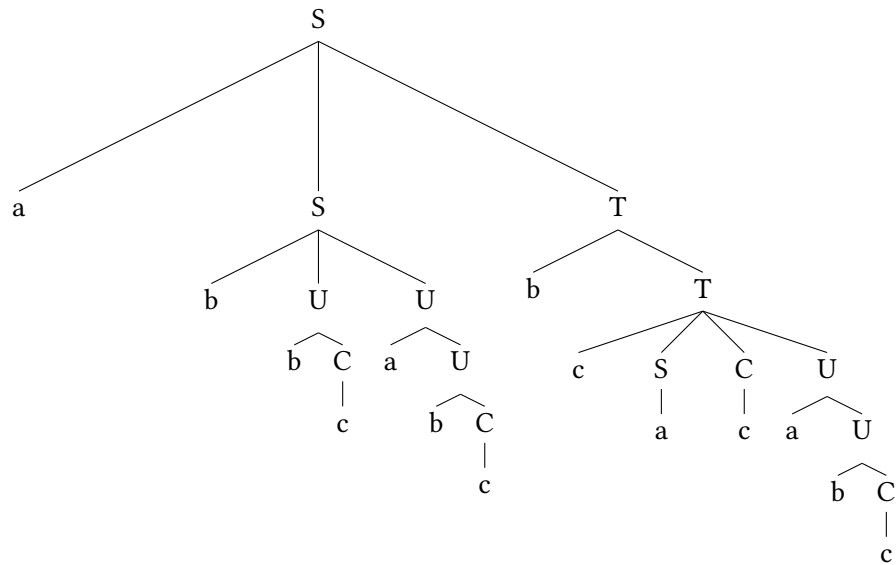
$$S \rightarrow aST \mid bUU \mid a$$

$$T \rightarrow bT \mid cSCU \mid cTU$$

$$U \rightarrow aU \mid bC$$

$$C \rightarrow c$$

Et l'arbre de parsing de la phrase `abbcabcabcacabc` en utilisant cette grammaire :



On peut aussi « inverser » la grammaire pour la centrer sur les terminaux :

$a \rightarrow S$
 $aST \rightarrow S$
 $aU \rightarrow U$
 $bUU \rightarrow S$
 $bT \rightarrow T$
 $bC \rightarrow U$
 $cSCU \rightarrow T$
 $cTU \rightarrow T$
 $c \rightarrow C$

Traduction en automate à piles :

En utilisant la grammaire inversée ci-dessus, on peut produire le tableau ci-dessous :

	S	T	U	C
a	ε		U	
	ST			
b	UU	T	C	
c	SCU			ε
	TU			

Le tableau se lit, par exemple pour la case (a, S) : on peut dépiler S si on lit a et qu'on empile ϵ ou T puis S.

Exemple d'exécution :

bande : abb, pile S
 \rightarrow lit a
 \rightarrow bande : bb, dépile S, empile ST, pile ST
 \rightarrow lit b
 \rightarrow bande : b, dépile S, empile UU, pile UUT
 \rightarrow lit b
 \rightarrow bande : #, dépile U, empile C, pile CUT

Transformation en forme normale de Greibach :

Tout d'abord, il faut noter quels sont les problèmes :

- les règles produisant ϵ ne sont pas tolérées (on vise une grammaire propre)
- les règles récursives à gauche ne sont pas tolérées

Voici maintenant l'algorithme pour transformer les règles récursives à gauche en règles récursives à droite :

Pour tout V_i non terminal, on introduit V_i' puis, on transforme tout règle de la forme :

$$V_k \rightarrow V_k M_1 \mid \dots \mid V_k M_n \mid W_1 \mid \dots \mid W_p$$

en

$$\begin{aligned} V_k &\rightarrow W_1 V_k' \mid \dots \mid W_p V_k' \mid W_1 \mid \dots \mid W_p \\ V_k' &\rightarrow M_1 V_k' \mid \dots \mid M_n V_k' \mid M_1 \mid \dots \mid M_n \end{aligned}$$

puis

$$\begin{aligned} V_j &\rightarrow V_k M \\ V_j &\rightarrow W_1 V_k' M \mid \dots \mid W_p V_k' M \mid W_{1M} \mid \dots \mid W_{pM} \end{aligned}$$

devient

$$V_j \rightarrow W_1 V_k' m \mid \dots \mid W_p V_k' M \mid W_1 M \mid \dots \mid W_p M$$

Exemple :

$$\begin{aligned} A &\rightarrow BC \\ B &\rightarrow AB \mid a \\ C &\rightarrow AC \mid b \end{aligned}$$

devient

$$\begin{aligned} A &\rightarrow BC \\ B &\rightarrow BCB \mid a \\ C &\rightarrow BCC \mid b \end{aligned}$$

puis

$$\begin{aligned} B &\rightarrow aB' \mid a \\ B' &\rightarrow CBB' \mid CB \\ A &\rightarrow aB'C \mid aC \\ C &\rightarrow aB'CC \mid aCC \mid b \end{aligned}$$

puis

$$\begin{aligned} B &\rightarrow aB' \mid a \\ B' &\rightarrow aB'CCBB' \mid aCCBB' \mid bBB' \mid AB'CCB \mid aCCB \mid bB \\ A &\rightarrow aB'C \mid aC \\ C &\rightarrow aB'CC \mid aCC \mid b \end{aligned}$$

Exercice

1. Mettre sous forme lexicale :

$$\begin{aligned} A &\rightarrow AaB \mid BA \mid b \\ B &\rightarrow Bd \mid BAa \mid aA \mid c \end{aligned}$$

2. Mettre sous forme lexicale :

$$\begin{aligned} S &\rightarrow AA \mid a \\ A &\rightarrow SS \mid b \end{aligned}$$

3. Tout langage algébrique propre peut s'écrire avec une grammaire sous forme normale de Greibach double :

$$\begin{aligned} V &\rightarrow X \\ V &\rightarrow XV^*X \end{aligned}$$

Apporter un argument.

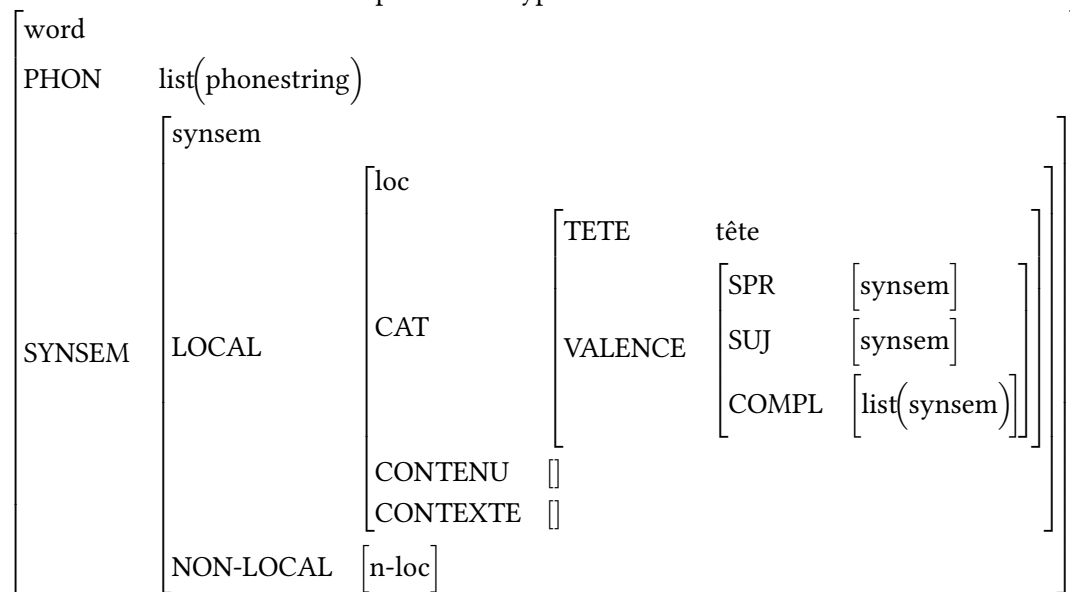
0.2 Head-driven Phrase Structure Grammars

Grammaires de constituants. Basées sur les principes suivants :

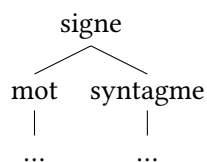
- on ne veut pas de transformation
- grammaires axées sur les têtes des syntagmes, laissant certaines constructions au lexique (comme la cliticisation)
- utilise des structures de traits et AVM (Attribute Value Matrix)

Structures de traits

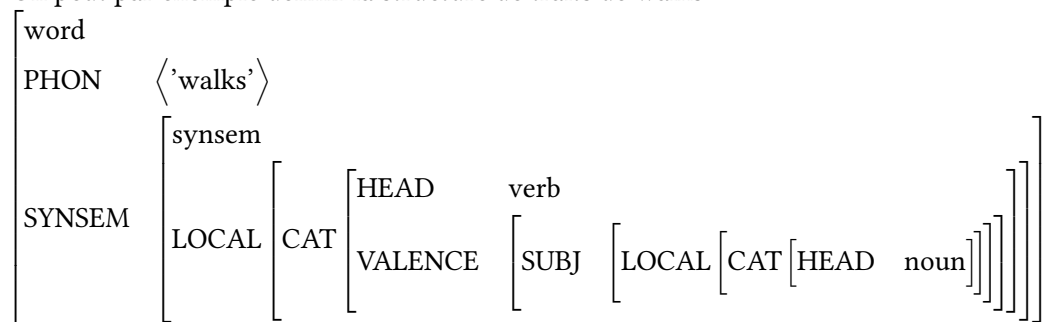
Une structure de traits est composée d'un type et d'une structure de traits :



Les structures de traits s'obtiennent en instanciant signe :



On peut par exemple définir la structure de traits de walks :



Et she :

word									
PHON	$\langle \text{'she'} \rangle$								
SYNSEM	<table> <tr> <td>synsem</td><td></td></tr> <tr> <td>LOCAL</td><td> <table> <tr> <td>CAT</td><td>$[\text{HEAD noun}]$</td></tr> <tr> <td>CONT</td><td>...</td></tr> </table> </td></tr> </table>	synsem		LOCAL	<table> <tr> <td>CAT</td><td>$[\text{HEAD noun}]$</td></tr> <tr> <td>CONT</td><td>...</td></tr> </table>	CAT	$[\text{HEAD noun}]$	CONT	...
synsem									
LOCAL	<table> <tr> <td>CAT</td><td>$[\text{HEAD noun}]$</td></tr> <tr> <td>CONT</td><td>...</td></tr> </table>	CAT	$[\text{HEAD noun}]$	CONT	...				
CAT	$[\text{HEAD noun}]$								
CONT	...								

Maintenant une structure syntagmatique :

head-subj-phrase																													
PHON	$\langle \boxed{1} + \boxed{2} \rangle$																												
SYNSEM	<table> <tr> <td>synsem</td><td></td></tr> <tr> <td>LOCAL</td><td> <table> <tr> <td>CAT</td><td> <table> <tr> <td>HEAD</td><td>$\boxed{3}$</td></tr> <tr> <td>VALENCE</td><td>$[\text{SUBJ } \langle \rangle]$</td></tr> </table> </td></tr> <tr> <td>...</td><td></td></tr> </table> </td></tr> <tr> <td>HEAD-DTR</td><td> <table> <tr> <td>PHON</td><td>$\boxed{2}$</td></tr> <tr> <td>SYNSEM</td><td> <table> <tr> <td>LOCAL</td><td> <table> <tr> <td>CAT</td><td>$[\text{HEAD } \boxed{3}]$</td></tr> </table> </td></tr> </table> </td></tr> <tr> <td>NON-HEAD-DTR</td><td> <table> <tr> <td>PHON</td><td>$\boxed{1}$</td></tr> <tr> <td>...</td><td></td></tr> </table> </td></tr> </table> </td></tr> </table>	synsem		LOCAL	<table> <tr> <td>CAT</td><td> <table> <tr> <td>HEAD</td><td>$\boxed{3}$</td></tr> <tr> <td>VALENCE</td><td>$[\text{SUBJ } \langle \rangle]$</td></tr> </table> </td></tr> <tr> <td>...</td><td></td></tr> </table>	CAT	<table> <tr> <td>HEAD</td><td>$\boxed{3}$</td></tr> <tr> <td>VALENCE</td><td>$[\text{SUBJ } \langle \rangle]$</td></tr> </table>	HEAD	$\boxed{3}$	VALENCE	$[\text{SUBJ } \langle \rangle]$...		HEAD-DTR	<table> <tr> <td>PHON</td><td>$\boxed{2}$</td></tr> <tr> <td>SYNSEM</td><td> <table> <tr> <td>LOCAL</td><td> <table> <tr> <td>CAT</td><td>$[\text{HEAD } \boxed{3}]$</td></tr> </table> </td></tr> </table> </td></tr> <tr> <td>NON-HEAD-DTR</td><td> <table> <tr> <td>PHON</td><td>$\boxed{1}$</td></tr> <tr> <td>...</td><td></td></tr> </table> </td></tr> </table>	PHON	$\boxed{2}$	SYNSEM	<table> <tr> <td>LOCAL</td><td> <table> <tr> <td>CAT</td><td>$[\text{HEAD } \boxed{3}]$</td></tr> </table> </td></tr> </table>	LOCAL	<table> <tr> <td>CAT</td><td>$[\text{HEAD } \boxed{3}]$</td></tr> </table>	CAT	$[\text{HEAD } \boxed{3}]$	NON-HEAD-DTR	<table> <tr> <td>PHON</td><td>$\boxed{1}$</td></tr> <tr> <td>...</td><td></td></tr> </table>	PHON	$\boxed{1}$...	
synsem																													
LOCAL	<table> <tr> <td>CAT</td><td> <table> <tr> <td>HEAD</td><td>$\boxed{3}$</td></tr> <tr> <td>VALENCE</td><td>$[\text{SUBJ } \langle \rangle]$</td></tr> </table> </td></tr> <tr> <td>...</td><td></td></tr> </table>	CAT	<table> <tr> <td>HEAD</td><td>$\boxed{3}$</td></tr> <tr> <td>VALENCE</td><td>$[\text{SUBJ } \langle \rangle]$</td></tr> </table>	HEAD	$\boxed{3}$	VALENCE	$[\text{SUBJ } \langle \rangle]$...																					
CAT	<table> <tr> <td>HEAD</td><td>$\boxed{3}$</td></tr> <tr> <td>VALENCE</td><td>$[\text{SUBJ } \langle \rangle]$</td></tr> </table>	HEAD	$\boxed{3}$	VALENCE	$[\text{SUBJ } \langle \rangle]$																								
HEAD	$\boxed{3}$																												
VALENCE	$[\text{SUBJ } \langle \rangle]$																												
...																													
HEAD-DTR	<table> <tr> <td>PHON</td><td>$\boxed{2}$</td></tr> <tr> <td>SYNSEM</td><td> <table> <tr> <td>LOCAL</td><td> <table> <tr> <td>CAT</td><td>$[\text{HEAD } \boxed{3}]$</td></tr> </table> </td></tr> </table> </td></tr> <tr> <td>NON-HEAD-DTR</td><td> <table> <tr> <td>PHON</td><td>$\boxed{1}$</td></tr> <tr> <td>...</td><td></td></tr> </table> </td></tr> </table>	PHON	$\boxed{2}$	SYNSEM	<table> <tr> <td>LOCAL</td><td> <table> <tr> <td>CAT</td><td>$[\text{HEAD } \boxed{3}]$</td></tr> </table> </td></tr> </table>	LOCAL	<table> <tr> <td>CAT</td><td>$[\text{HEAD } \boxed{3}]$</td></tr> </table>	CAT	$[\text{HEAD } \boxed{3}]$	NON-HEAD-DTR	<table> <tr> <td>PHON</td><td>$\boxed{1}$</td></tr> <tr> <td>...</td><td></td></tr> </table>	PHON	$\boxed{1}$...															
PHON	$\boxed{2}$																												
SYNSEM	<table> <tr> <td>LOCAL</td><td> <table> <tr> <td>CAT</td><td>$[\text{HEAD } \boxed{3}]$</td></tr> </table> </td></tr> </table>	LOCAL	<table> <tr> <td>CAT</td><td>$[\text{HEAD } \boxed{3}]$</td></tr> </table>	CAT	$[\text{HEAD } \boxed{3}]$																								
LOCAL	<table> <tr> <td>CAT</td><td>$[\text{HEAD } \boxed{3}]$</td></tr> </table>	CAT	$[\text{HEAD } \boxed{3}]$																										
CAT	$[\text{HEAD } \boxed{3}]$																												
NON-HEAD-DTR	<table> <tr> <td>PHON</td><td>$\boxed{1}$</td></tr> <tr> <td>...</td><td></td></tr> </table>	PHON	$\boxed{1}$...																									
PHON	$\boxed{1}$																												
...																													

Principe de construction : on peut par exemple combiner she et walks en tant que NON-HEAD-DTR et HEAD-DTR respectivement grâce au syntagme head-subj-phrase.

Exemple : Félix aime les chatouilles

– Félix :

TETE	nom
SPR	$\langle \rangle$
COMPS	$\langle \rangle$

– aime :

TETE	verbe $\boxed{3}$
SUJ	$\boxed{1}$
COMPS	$\boxed{2}$

– les :

TETE	det
------	-----

– chatouilles :

$$\begin{bmatrix} \text{TETE} & \text{nom } \boxed{4} \\ \text{SPR} & \langle \boxed{5} \rangle \\ \text{COMPS} & \langle \rangle \end{bmatrix}$$

- puis on peut combiner les et chatouilles en une structure spécifieur - syntagme :

$$\begin{bmatrix} \text{TETE} & \text{nom } \boxed{4} \\ \text{SPR} & \langle \rangle \\ \text{COMPS} & \langle \rangle \end{bmatrix}$$

- puis on peut combiner aime et les chatouilles en une structure complément - syntagme :

$$\begin{bmatrix} \text{TETE} & \boxed{3} \\ \text{SPR} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle \rangle \end{bmatrix}$$

- et enfin Félix et aime les chatouilles :

$$\begin{bmatrix} \text{TETE} & \boxed{3} \\ \text{SPR} & \langle \rangle \\ \text{COMPS} & \langle \rangle \end{bmatrix}$$