

Grammaires locales

Sébastien Paumier

paumier@univ-mlv.fr

Définitions

Définition (1)

Grammaire (pratique d'une langue):

« Ensemble de règles conventionnelles (variables suivant les époques) qui déterminent un emploi correct (ou bon usage) de la langue parlée et de la langue écrite. »

Trésor de la langue française informatisé

<http://atilf.inalf.fr/>

Définition (2)

Grammaire (étude d'une langue):

« Étude objective et systématique des éléments (phonèmes, morphèmes, mots) et des procédés (de formation, de construction, d'expression) qui constituent et caractérisent le système d'une langue naturelle. »

Trésor de la langue française informatisé

<http://atilf.inalf.fr/>

Définition (3)

Grammaire générative, transformationnelle:

« Mécanisme, constitué par un ensemble de règles, qui permet de produire et de décrire toutes et rien que les phrases grammaticales d'une langue. »

Trésor de la langue française informatisé

<http://atilf.inalf.fr/>

Définition (4)

Grammaire formelle:

« Le but que se propose la linguistique moderne est de construire des grammaires pour des langues naturelles, qui soient entièrement explicitées sous forme d'automate, ou de construction algébrique. »

Trésor de la langue française informatisé

<http://atilf.inalf.fr/>

Linguistique formelle

Langages formels et naturels

- Depuis Harris, les langages naturels sont abordés comme les langages formels
- Chomsky montre que les transducteurs ne suffisent pas car le langage est récursif:

L'homme qui a vu l'homme qui a vu l'homme ... qui a vu l'ours

- Il montre également que les grammaires algébriques (hors-contexte) ne suffisent pas non plus:

Dérivation contextuelle de X en Y: $Z X W ? Z Y W$

Grammaire générative de Chomsky

On passe d'une phrase A à une phrase B par une ou plusieurs transformations orientées:

Paul aime Marie ? ^{passivation} Marie est aimée par Paul

? ^{extraction} c'est Marie qui est aimée par Paul

Problème:

Que faire quand une forme est accidentellement manquante ?

Classes d'équivalence

Si deux phrases sont équivalentes, on explicite le phénomène en les regroupant dans une même classe et non en cherchant à fournir un mécanisme opératoire permettant de passer de l'une à l'autre.

Au lieu de:

$$A ? A' ? \dots ? B$$

on a:

$$C ? A | B$$

Avantages

- Les accidents lexicaux ne sont plus un obstacle
- Plus besoin de choisir une structure canonique
- Plus besoin d'un formalisme compliqué pour décrire les mouvements et modifications que subissent les composants

Résultats empiriques

Le lexique-grammaire montre qu'il n'existe pas deux éléments ayant le même comportement syntaxique, d'où:

- impossible d'établir des règles générales qui expliquent la langue
- on doit accumuler des descriptions de phénomènes particuliers

?

Le modèle des classes d'équivalence permet de rendre compte de ces faits

Le fini et l'algébrique

Finitude du langage

- Chomsky: le langage est récursif
- Contre-argument n°1:
 - une phrase trop récursive est incompréhensible pour un humain: *Luc dont Léa dont Jean a dit du bien a applaudi le discours est venu* (seulement 2 récursions)
- Contre-argument n°2:
 - on peut borner la taille des phrases: phrase la plus longue pouvant être écrite/lue/prononcée pendant toute une vie
- Conclusion: le langage est fini, on peut donc se contenter du formalisme simple des transducteurs à états finis

Adéquation transducteurs/langage

Les transducteurs permettent de représenter simplement divers objets et phénomènes:

- Grammaires de flexion
- Variantes phonétiques: *excuse* ? [eKsKyz] ou [esKyz]
- Compression de lexique
- Variantes lexicales : *microscope* (à balayage+e) (*électronique*+e)
- Représentation des ambiguïtés d'un texte

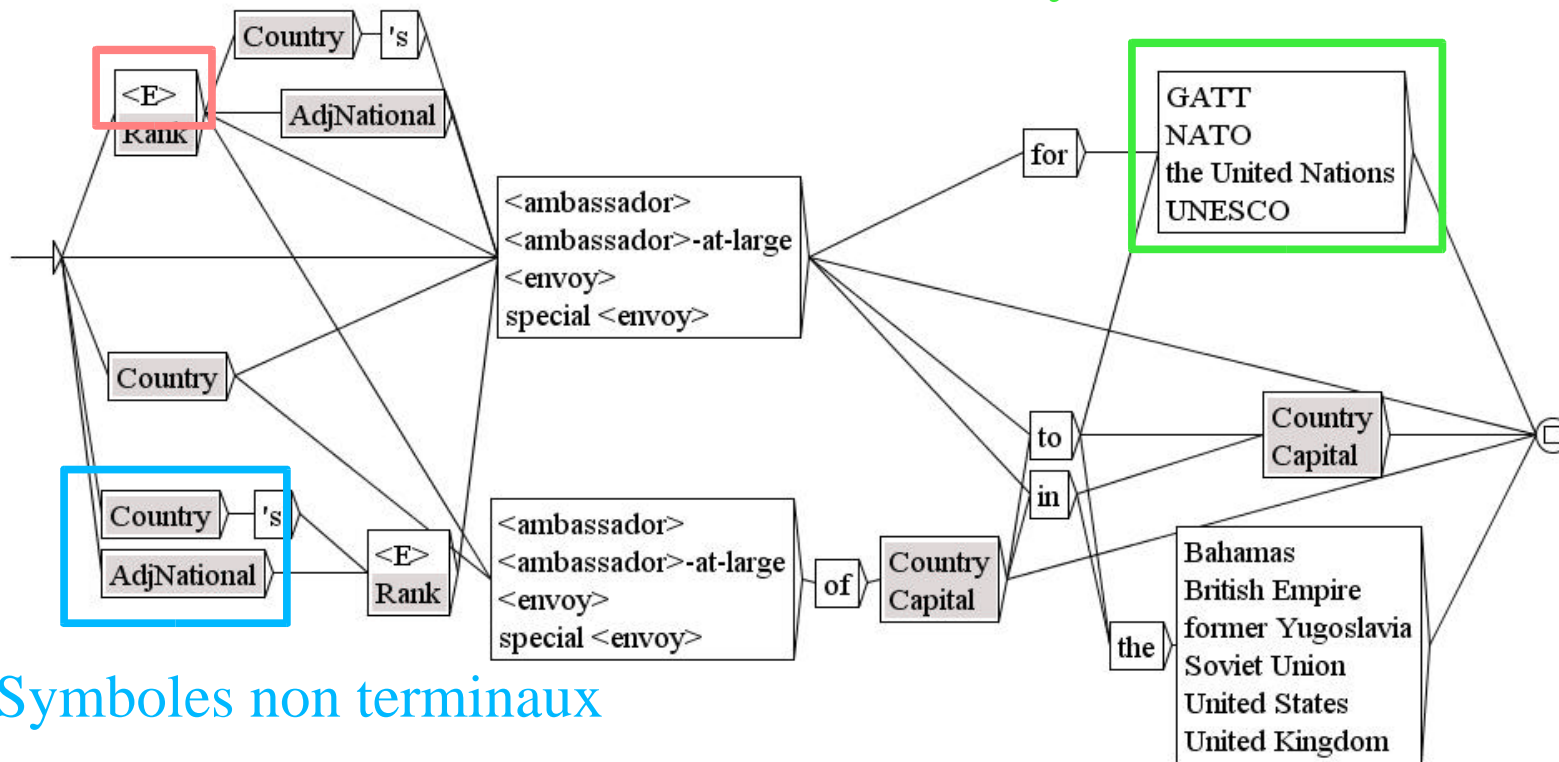
Les types de graphes

- grammaires de flexion
- grammaires de découpage en phrase
- grammaires de normalisation
- grammaires de levée d'ambiguïtés
- grammaires locales

Une vision formelle des graphes

Mot vide e

Symboles terminaux

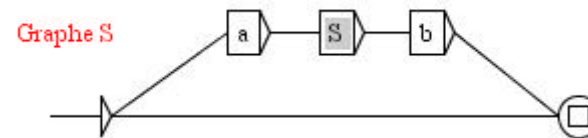


Symboles non terminaux

Graphes = objets algébriques

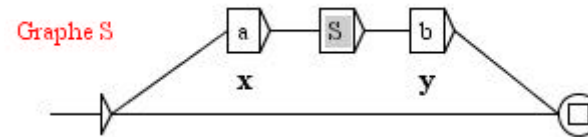
Grammaires algébriques:

$S ? a S b | e \quad ?$

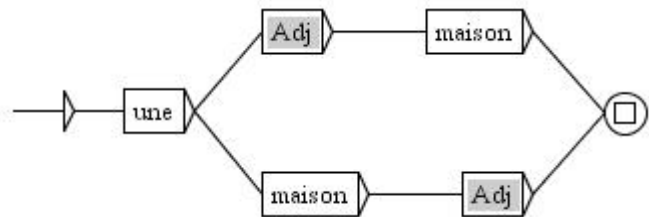


Transducteurs algébriques:

$S ? a/x S b/y | e \quad ?$



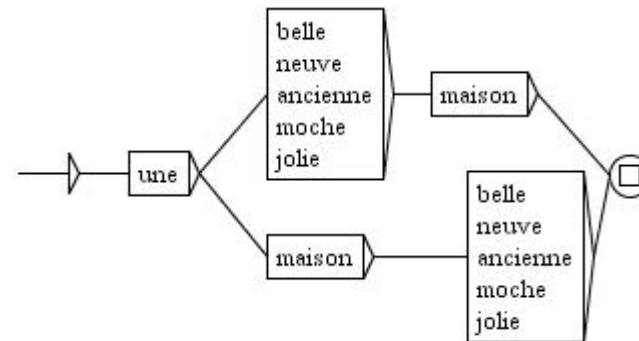
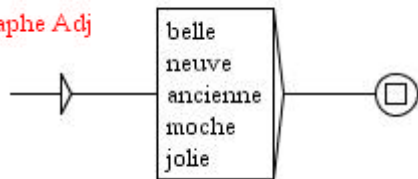
Des objets algébriques aux objets à états finis



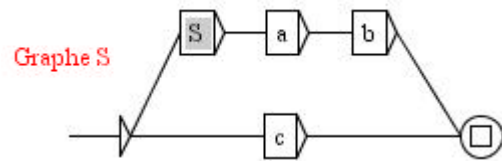
+

?

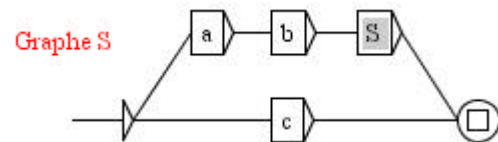
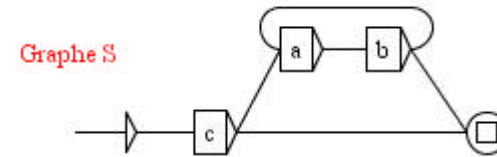
Graphe Adj



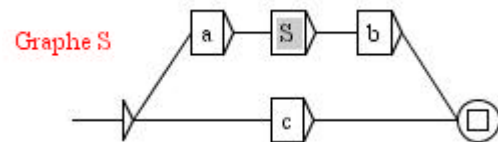
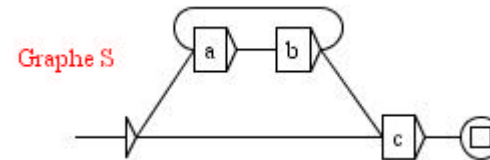
Problème de récursivité



?



?



?

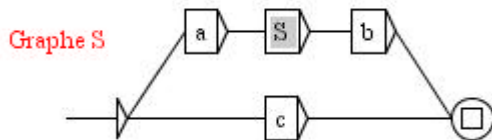
pas de solution exacte

Approximation

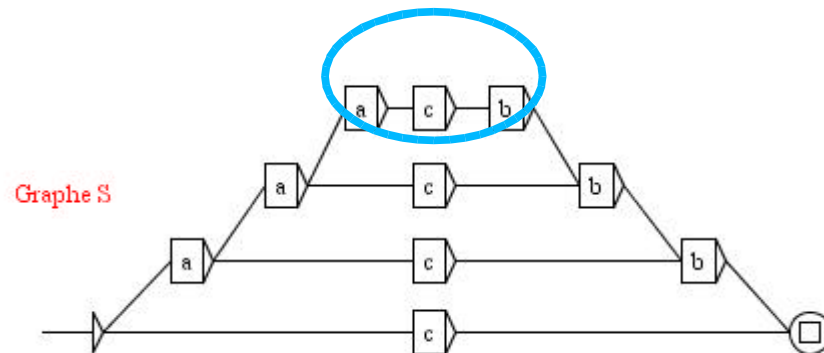
On effectue un remplacement jusqu'à une certaine profondeur:

Exemple: profondeur=3

Au dernier niveau, l'appel récuratif est supprimé

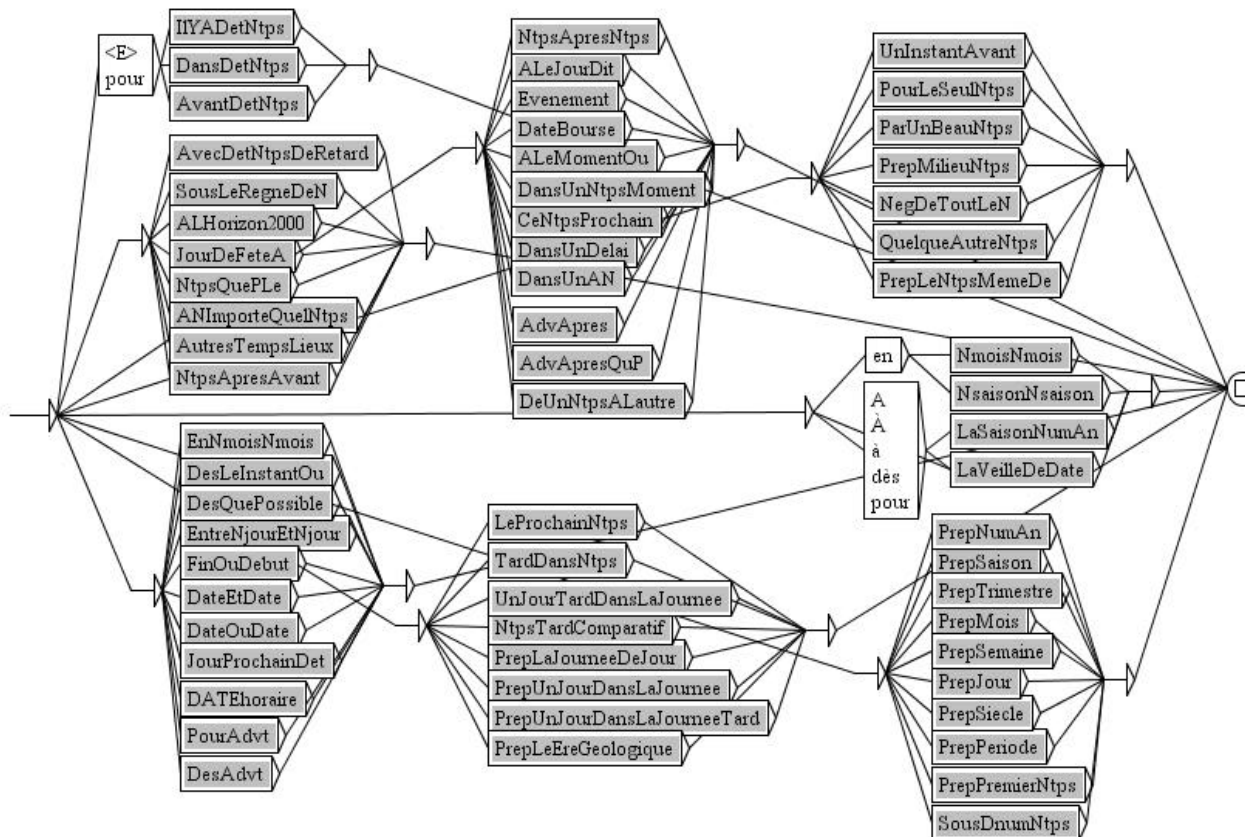


?



Problème d'explosion combinatoire

Exemple des dates: 237 graphes ? 50 méga-octets



Faire du fini avec de l'algébrique

Même si le langage est théoriquement fini, il est plus commode de travailler avec des outils algébriques.

? Conservation de la structure en sous-graphes :

- + pas d'explosion combinatoire
- + pas d'approximation
- objets moins faciles à gérer que les transducteurs à états finis

Définition et construction des grammaires locales

Les degrés du figement

Mots composés

? *pomme de terre*

Expressions figées

? *prendre le taureau par les cornes*

Expressions semi-figées

? *perdre la (boule+tête+raison+boussole+...)*

Syntagmes libres

Grammaires locales



Qu'est-ce qu'une grammaire locale ?

Certains mots ont une distribution contrainte:

*président de la (compagnie X+république+France+*chaise+*théorie+...)*

(* = interdiction)

?

On peut donc construire des grammaires qui décrivent ces contraintes
locales

Plus de contexte = moins d'ambiguïté

...président...

? *ambiguïté totale*

...ils président de...

...éliront-ils président de la France...

? *ambiguïté très faible*

Qui éliront-ils président de (Guy de) la France ou de (Gérard) Brissonière ?

? *ambiguïté théorique, très peu probable*

Recherche du plus grand contexte

Principe: élargir le contexte pour lever l'ambiguïté

Conséquence: la grammaire peut reconnaître des séquences de structure variable

Paul est président de la société X ? GN

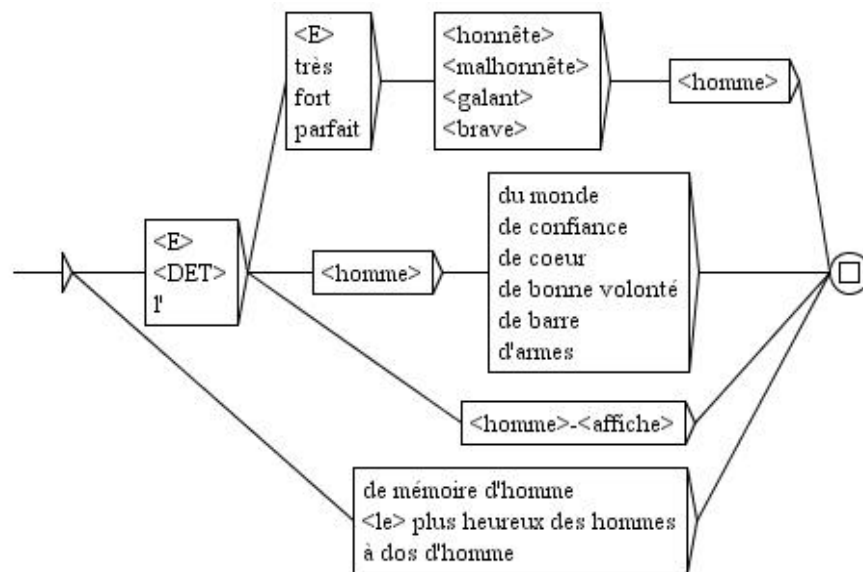
Le peuple élit le président ? V+GN

Ils l'ont élu président à l'unanimité ? Phrase avec adverbe

Construction d'une grammaire

Rechercher un mot seul dans un corpus, puis examiner de quelle façon on peut élargir le contexte

Exemple: étude de *homme* dans *Le tour du monde en 80 jours*

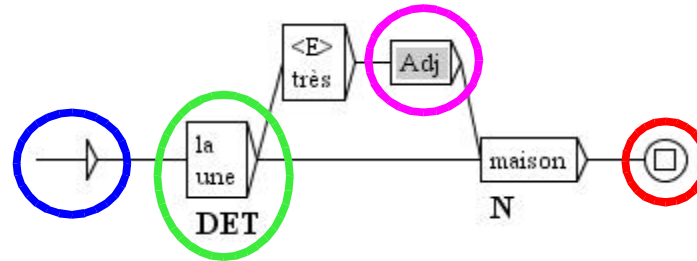


e c'était [un fort galant homme](#) et l'un des plus
olument à [un honnête homme](#). _ Monsieur le con
a l'air d'[un parfait honnête homme](#) ! _ Possib
purser ", [l'homme de confiance](#) de la Compagni
alanquin, [à dos d'homme](#), en coach, etc.{S} Ma
vous êtes [un homme de coeur](#) ! dit Sir Francis
vous êtes [un brave homme](#) ! " {S}Le pilote ne
aître est [un honnête homme](#), et que, quand il
t le plus [honnête homme](#) du monde ! _ Qu'en sa
s honnête [homme du monde](#) ! _ Qu'en savez-vous
que c'est [un honnête homme](#) ! _ Oui ! oui ! Ré
procher à [l'homme de barre](#) !{S} On n'eût pas
rcussion, [hommes d'armes](#) du mikado, ensachés
Il suivit [l'homme-affiche](#), et, à sa suite, il
le croyez [un honnête homme](#) ? _ Non, répondit
inel ou d'[un honnête homme](#) ! " {S}Passepartou
rable.{S} [Des hommes-affiches](#) circulaient au
r !{S}... [Trente hommes de bonne volonté](#) ! "
'aspect d'[un homme du monde](#). " Le capitaine ?
insi : {S}[Honnête homme](#), Phileas Fogg était r
ruiné. {S}[Malhonnête homme](#), il était pris. {S
vu courir [de mémoire d'homme](#), renversant les
le rendit [le plus heureux des hommes](#) ! {S}En

Implémentation des grammaires

Les graphes

```
#Unigraph
SIZE 1188 840
FONT Times New Roman: 10
OFONT Times New Roman:B 12
BCOLOR 16777215
FCOLOR 0
ACOLOR 13487565
SCOLOR 16711680
CCOLOR 255
DBOXES y
DFRAME y
DDATE y
DFILE y
DDIR n
DRIG n
DRST n
FITS 100
PORIENT L
#
6
"<E>" 70 200 1 2
"" 364 200 0
"la+une/DET" 135 200 2 4 5 => x=135, Y=200, 2 transitions sortantes vers 4 et 5
":Adj" 237 151 1 1
"maison/N" 282 200 1 1
"<E>+très" 183 151 1 3
```

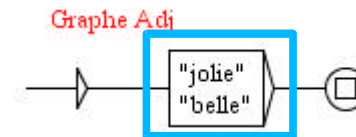
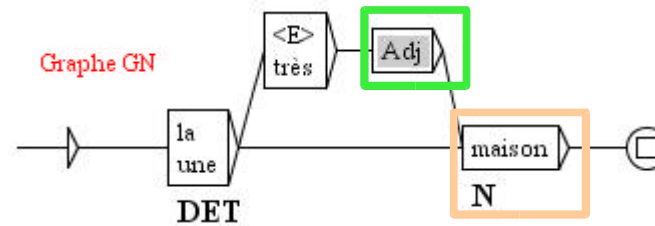


Le format FST2

000000002 ? 2 sous-graphes

```

-1 GN
: 2 1 1 1
: 3 4 -2 3 4 2
: -2 3
: 3 4
t
f
-2 Adj
: 6 1 5 1
t
f
%<E>
%la/DET
%une/DET
%maison/N
%très
@jolie
@belle
f
    
```



Les RTN

RTN = Recursive Transition Network

Définition: RTN = n -uplet (E, I, F, S, S, d)

- E = ensemble des états
- I = ensemble des états sous-initiaux (*état sous-initial* = état qui étiquette au moins une transition du RTN, ce qui représente un appel récursif au sous-RTN obtenu en prenant cet état comme état initial; par convention, $S \in I$)
- F = ensemble des états terminaux
- S = alphabet d'entrée
- S = état initial du RTN (axiome de la grammaire)
- d = fonction de transition; on a:

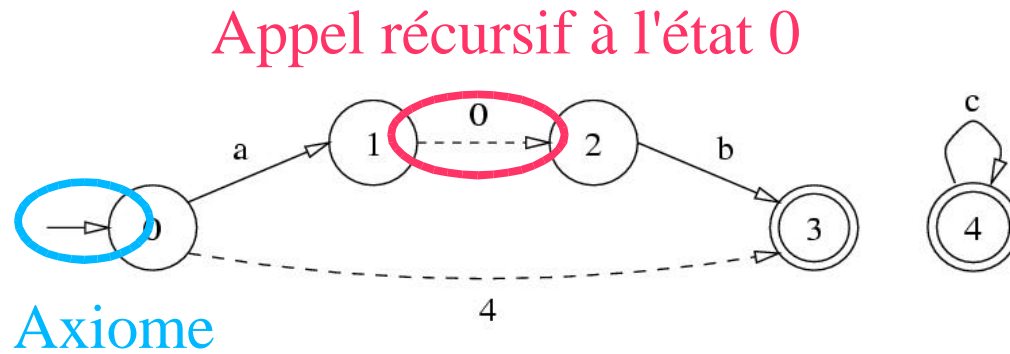
$$d: E \times (I \cup S \cup \{\varepsilon\}) \rightarrow E$$

Exemple de RTN

$S ? a S b | C$

$C ? e / c C$

?



• $E = \{0,1,2,3,4\}$

• $I = \{0,4\}$

• $F = \{3,4\}$

• $S = \{a,b,c\}$

• $S = 0$

Exemple de mot reconnu: $a c c b$

Chemin = 0 a 1 0 2 b 3

[0 4 3]

[4 c 4 c 4]

Transducteurs algébriques étendus

Définition: n -uplet (E, I, F, S, O, S, d)

- O = alphabet de sortie
- d = fonction de transition; on a:

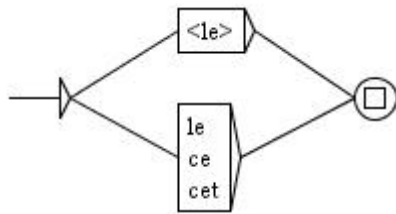
$$d: E \times I \cup [(S \cup \{\varepsilon\}) \times (O \cup \{\varepsilon\})] \rightarrow E$$

Application des grammaires aux textes

Non déterminisme

Les grammaires sont non déterministes pour 2 raisons:

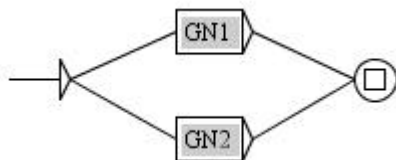
- interprétations des étiquettes ? non déterminisme d'interprétation



?

- pas déterministe si <le> peut reconnaître *le*, *ce* ou *cet*
- MAIS: déterministe si on n'interprète pas <le>

- appels aux sous-graphes ? non déterminisme de structure



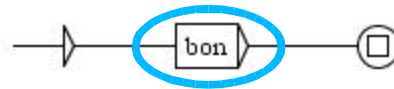
?

- pas déterministe car on doit nécessairement rentrer dans les 2 sous-graphes

Interprétations des étiquettes

- Variantes de casse:

bon, Bon, BON, ...



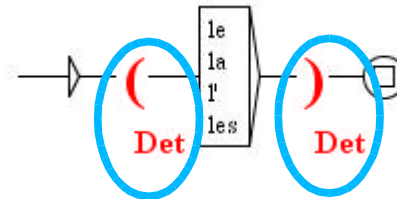
- Filtrés lexicaux:

*tout adjectif au féminin
singulier*



- Utilisation de variables:

*début et fin de déclaration d'une
variable*

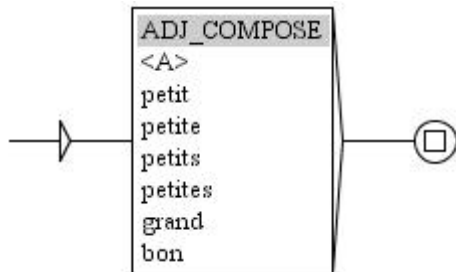


Adaptation de la grammaire au texte



Texte = **Grand** principe et **petite** vertu.

?



?



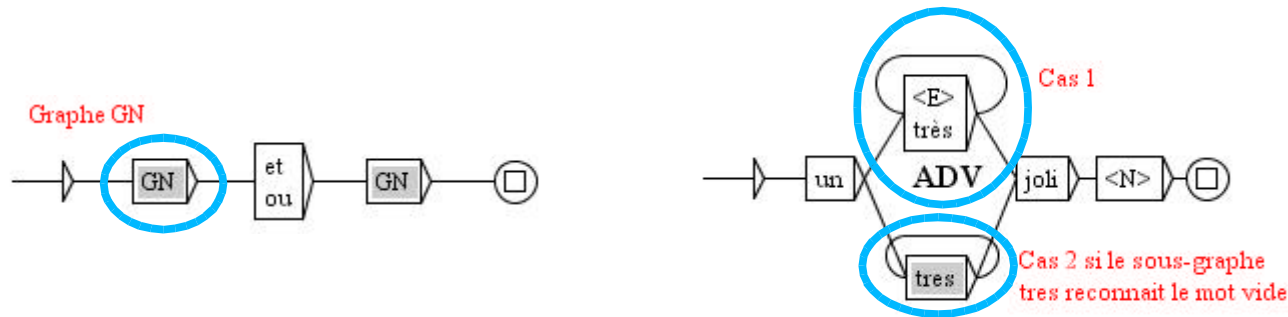
remplacement des filtres lexicaux comportant un lemme par les mots qu'ils peuvent reconnaître (au plus 39 en français)

remplacement des mots par les unités lexicales réellement présentes dans le texte, en tenant compte des variantes de casse

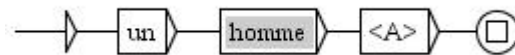
Application des grammaires

Les grammaires sont appliquées par analyse descendante:

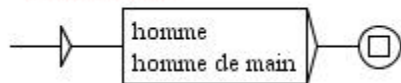
- contrainte sur la récursivité à gauche et sur les boucles vides:



- remontée non déterministe des appels récursifs:



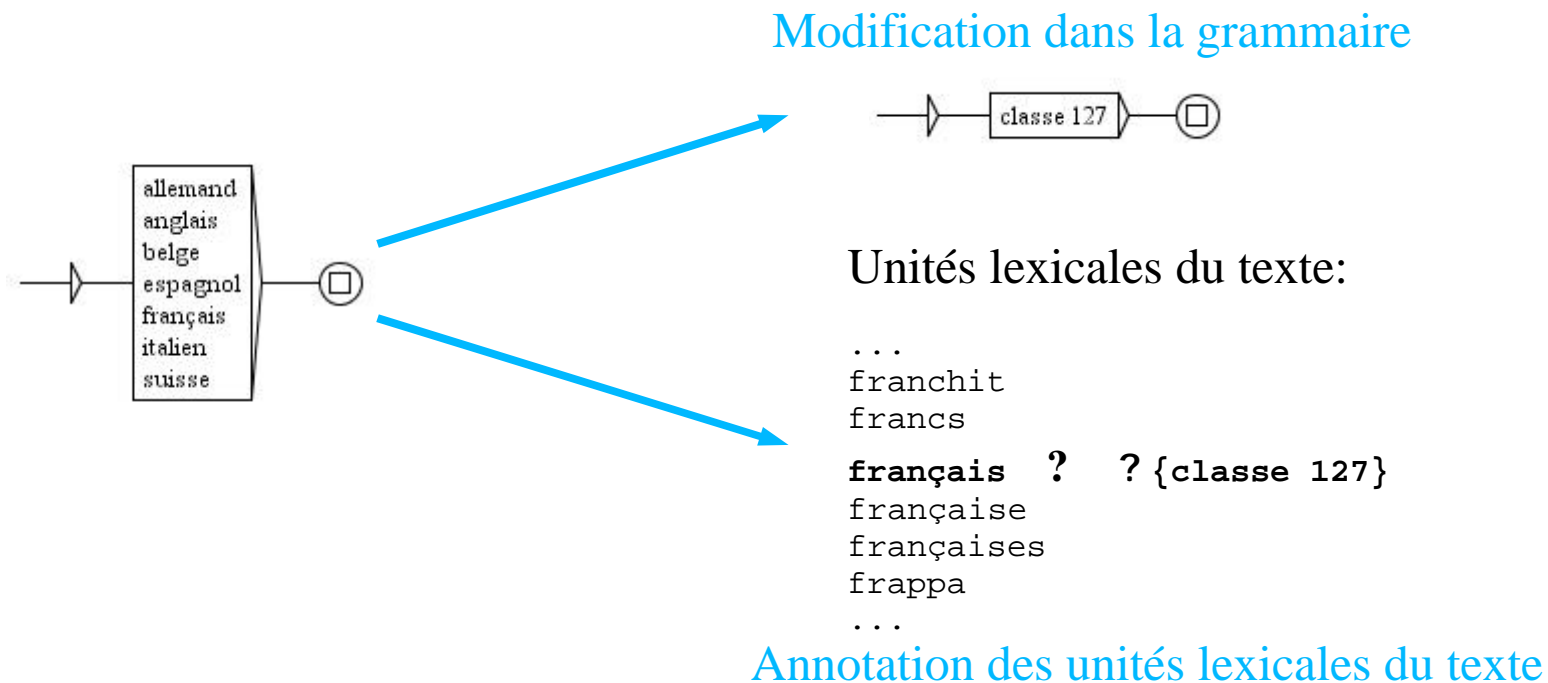
Graphe homme



{ ... un homme de main cruel...

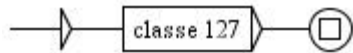
Optimisation par classes de mots (1)

Principe: regrouper dans une classe toutes les transitions étiquetées par des mots qui vont d'un même état vers un même état



Optimisation par classes de mots (2)

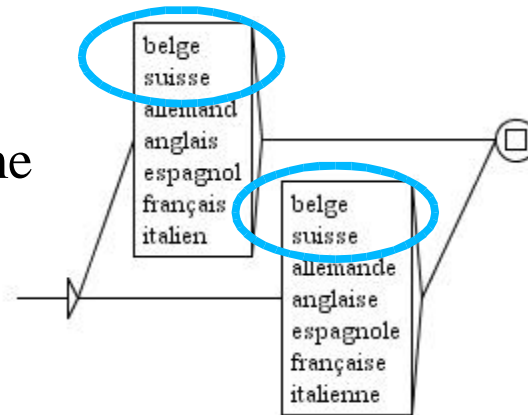
Gain: la comparaison entre l'unité du texte et la grammaire se fait en un seul test



... un **français** qui ... ? Est-ce que **français** ? {classe 127} ?

Coût: on doit stocker pour chaque unité lexicale à quelles classes elle appartient

Inconvénient: pas d'impact sur le non déterminisme



Optimisation par dichotomie (1)

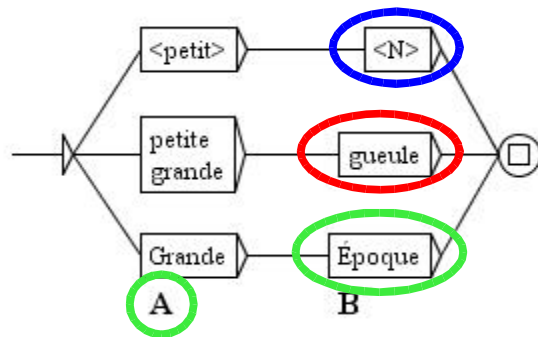
D'un état peuvent partir 4 types de transitions:

- appels à des sous-graphes
- symboles spéciaux (<E>, #, \$a(, ...)
- motifs ou filtres lexicaux (<DIC>, <N:mp>, ...)
- unités lexicales (joli, Car, ...)

Remarque: il faut explorer les 3 types entièrement, mais par construction, au plus une unité lexicale peut coïncider avec celle du texte

Optimisation par dichotomie (2)

Idée: trier les unités lexicales, et tester par dichotomie si celle du texte est présente dans le tableau



Grande	{2, «»}	{3, «A»}
petit	{1, «»}	
Petites	{1, «»}	
PETITE	{1, «»}	{2, «»}

- Comme les unités sont représentées par des entiers, le test est très rapide
- Test déterministe, car chaque unité lexicale distincte est testée au plus une fois; mais l'exploration peut être non déterministe (**Grande** ou **PETITE**)

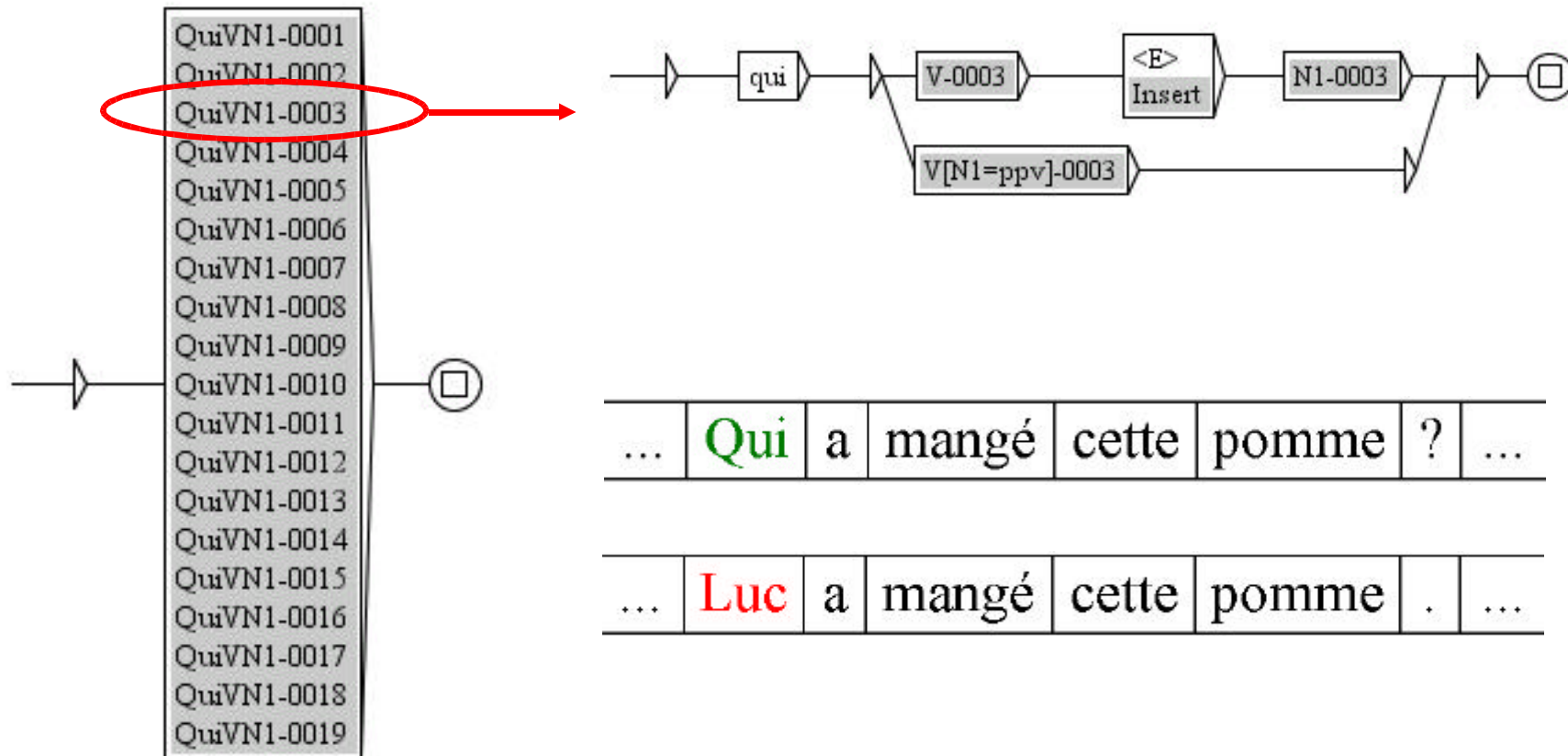
Résultats expérimentaux

Difficile de calculer la complexité, mais l'expérience montre que l'on gagne en temps:

	Expressions numériques	Syntagmes verbaux	Groupes nominaux économiques
Nombre d'états	107 000	250 000	587
Nombre de transitions	703 000	2 000 000	3 596
Taille du texte	0.4 Mo	16 Mo	120 Mo
Temps avec Intex	20 min	échec	9 min
Temps avec Unitex	22 sec	3 min 37 s	1 min 11 s

Résultats obtenus sur un PC équipé d'un Pentium 500 MHz et de 128 Mo de RAM

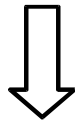
Non déterminisme de structure



Optimisation par transformation des grammaires

Idée: comparer au plus tôt la grammaire avec le texte

→ les grammaires ne doivent pas débuter par des appels à des sous-grammaires



Mise en forme normale de Greibach

Forme normale de Greibach (1)

Une grammaire est en *forme normale de Greibach* SSI toutes les productions, sauf éventuellement $S \rightarrow \epsilon$, sont de la forme:

$A \rightarrow a\beta$ ou a est un terminal et β une séquence quelconque de non terminaux

On parle de *forme normale faible* si les productions $A \rightarrow a\beta$ sont telles que β est une séquence quelconque de terminaux et/ou non terminaux

Forme normale de Greibach (2)

Un RTN est en *forme normale faible de Greibach SSI*:

- pas de ϵ -transition
- $I \cap F = \{S\}$ ou \emptyset
- ? de transition (a,x,b) telle que $a,x \in I$

? on n'explore pas un sous-graphe sans avoir au préalable exploré au moins une transition normale

Résultats expérimentaux

- FST3 = grammaire mise en forme normale de Greibach, aux *e*-productions près
- Comparaisons effectuées avec le programme AGLAE (ancêtre d'Unitex)

	Phrases économiques (281 sous-graphes, corpus = 4,7 Mo)	Expressions de pourcentages (16 sous-graphes, corpus = 125 Mo)	Expressions de date (94 sous-graphes, corpus = 125 Mo)
FST2	2 min 28 s	11 min 15 s	41 min 40 s
FST3	31 s	1 min 59 s	8 min 11 s

Résultats obtenus sur un PC équipé d'un Pentium 500 MHz et de 128 Mo de RAM

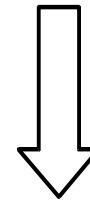
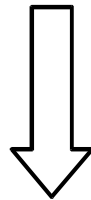
Analyse syntaxique & perspectives

Recherche de motifs

Texte

```
En cuisine, le chef commande.  
{S}ce est une belle commande de poissons.
```

Grammaire



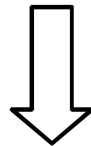
```
En cuisine, le chef commande. {S}ce est une belle commande de poiss  
En cuisine, le chef commande. {S}ce est le chef commande. de poissons.  
En cuisine, le chef commande. {S}ce est une belle commande de poissons.
```

Occurrences

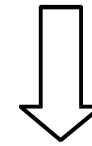
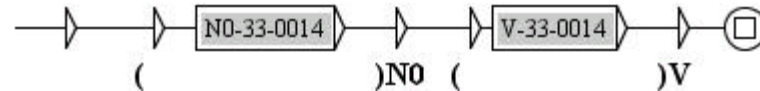
Analyse syntaxique

Texte

En cuisine, le chef commande.
{S}ce est une belle commande de poissons.



Grammaire avec sorties



En cuisine, (le chef)NO (commande)V. {S}ce est une belle commande d
En cuisine, le chef commande. {S}ce est (une belle)NO (commande)V de poissons.

Occurrences étiquetées = analyses potentielles

?

recherche de motifs ~ analyse syntaxique

Perspectives

- Optimiser l'application des grammaires:
 - construction de tables d'analyse ?
 - exploration paresseuse des grammaires ?
 - restriction par le lexique ?
- Travailler sur les automates de phrases:
 - intersection ?
 - extension des algorithmes utilisés pour les textes linéaires ?
 - que faire des transductions ?
- Utiliser des grammaires issues des tables pour l'analyse syntaxique