Analyse syntaxique et application aux langues naturelles

Jacques Farré

et

Sylvain Schmitz

Objectifs du cours

- Présenter les principaux formalismes grammaticaux permettant de modéliser les langues naturelles
  - Puissance descriptive des formalismes
  - Efficacité des analyseurs
- Faire manipuler et évaluer ces formalismes

Plan du cours

1. Introduction, rappels, retour sur les grammaires non contextuelles (CFG)
2. Lexiques et Grammaires Lexicales Fonctionnelles (LFG)
3. Méthodes tabulées pour CFG : CYK, Earley, GLR
4. TP d'utilisation d'outils (GLR, TAG)
5. Formalismes faiblement contextuels (Grammaires d'Adjonction d'Arbres) et Grammaires Catégorielles

Evaluation des connaissances

- Examen avec des questions portant
  - Sur le cours
    - écriture de fragments de grammaires
    - construction de partie d'automates
    - etc.
  - Sur l'article distribué
    - compréhension de la motivation
    - sur la partie technique
- Examen début janvier 2007

Bibliographie sommaire (analyse syntaxique)

- Parsing Techniques – A Practical Guide, 2ème edition, D. Grune & C.J.H. Jacobs, Springer Verlag, 2007 (1ère ed: ftp://ftp.cs.vu.nl/pub/dick/PTAPG/BookBody.pdf)
- R. Grishman, Computational Linguistic: an introduction, Cambridge University Press, 1986
- Les nouvelles syntaxes : grammaires d'unification et analyse du français, A. Abeillé, Armand Colin, 1993
- L'intelligence artificielle et le langage naturel, G. Sabah, Hermès, 1988-1989 (2 volumes)
- Parsing Theory, S. Sippu & E. Soisalon-Soinnen, Springer Verlag, 1988

Principaux journaux

- Computational Linguistics
- Computers and the Humanities
- Computer, Speech & Language
- Computer Assisted Language Learning
- Grammars
- Journal of Language and Computation (logic, linguistics, formal grammar, and computational linguistics)
- Linguistics
- Literary and Linguistic Computing
- Natural Language Engineering
- Machine Translation
- …

Quelques conférences

- Sous l'égide de l' (European) Association for Computational Linguistic : ACL
- COLING (Int'l Conf. on Comp. Ling.)
- CICLing
- LATIN (Latin American Theoretical Informatics)
- ANLP (Applied Nat. Lang. Processing)
- IEEE Int'l Conf. on Natural Language Processing and Knowledge Engineering
- TALN (Traitement automatique du langage naturel)
- CIAA (Implementation & Application of Automata)
- …

Domaines d'application (1)

- Traitement documentaire
  - Traduction (semi)-automatique
    - Bons traducteurs dans un domaine spécialisé pour préparer le terrain à une traduction par un humain
    - Mémoires de traduction
  - Recherche de documents
    - Veille scientifique
    - Routage, indexation de documents
  - Analyse de documents
    - Graphe de relations entre termes d'un document

Domaines d’application (2)

- Production de documents
  - Correcteurs d’orthographe, de syntaxe, …
  - Correcteurs stylistiques (bonnes pratiques rédactionnelles dans un domaine donné)
  - Génération automatique à partir de spécifications plus ou moins formelles (documents techniques, juridiques, …) de documents finalisés par des humains

Domaines d’application (3)

- Interfaces homme-machine
  - Interrogation de bases de données
    - Traduction langage naturel → SQL
  - E-learning
  - Interfaces vocales
    - Téléphonie
    - Ordinateurs (et autres machines) mains libres
    - Marché de plusieurs milliards de dollars

Un peu d'histoire (1)

- A l'origine, les militaires (comme souvent)
  - Années 50 : traduction de documents russes (nucléaire, recherche spatiale, …)
  - Concentré sur l'élaboration de dictionnaires bilingues
  - Traduction mot à mot pour l'essentiel
    - Exemple célèbre de traduction anglais→russe→anglais :
      The spirit is willing but the flesh is weak →
      The vodka is strong but the meat is rotten
  - ⇒ élaborer des modèles plus riches aux niveaux
    - syntaxique
    - sémantique

Un peu d'histoire (2)

- Travaux de Chomsky (fin des années 50) sur les grammaires formelles et leurs relations avec les langues naturelles $\Rightarrow$ hiérarchie de Chomsky
- Parallèlement, travaux sur l'intelligence artificielle (Mc Carthy, Minsky, …)
  - Système ELIZA (MIT, 1966) : simulation d'un dialogue entre un psy et son patient (application de modèles reprenant des mot-clés du patient) : voir par exemple *psychiatrist* sous emacs

Un peu d'histoire (3)

- Travaux sur la représentation des connaissances dans les années 70 (Minsky, Shank, …)
  - Réseaux sémantiques, graphes conceptuels, *frames*
  - La sémantique prime, la syntaxe est jugée secondaire
- Mais aussi développement de l'analyse syntaxique (dans le cadre des langages de programmation, récupéré en partie et étendu pour les langues naturelles)

Un peu d'histoire (4)

- Actuellement, nombreux formalismes (non contextuels enrichis, faiblement contextuels, d'unification, probabilistes, …)
- Quel que soit le formalisme, difficulté d'écrire une grammaire (nombre trop important de règles)
  - $\Rightarrow$ méta grammaires
  - $\Rightarrow$ apprentissage à partir d'un corpus (par exemple inférence grammaticale, fouille d'erreur)

Problématique du TALN: quelques exemples

- Difficultés de plusieurs ordres, notamment
  - Ambiguïtés
    - Des terminaisons : que marque un *s* final ?
    - Des lexèmes : *les poules du couvent couvent*
    - Des formes grammaticales : *il poursuit les filles à vélo*
  - Emploi irrégulier des genres lexicaux
    - *La mer rouge (rouge = adjectif)*
    - *Il boit du gros rouge (rouge = adjectif en place d'un nom)*
  - Implicite/contextuel : à qui/quoi réfère *il*
    - Le prof a saqué cet élève parce
      - qu'*il* ne peut pas le sentir (il, prof)
      - qu'*il* lui a cassé les pieds toute l'année (il, élève)

Niveaux de traitement et outils

- Lexical : découper le texte en mots et calculer leur genre (adjectif/verbe/nom/préposition…)
  - Dictionnaires, automates à états finis
- Syntaxique : trouver la structure de la phrase (quelque chose comme *sujet verbe complément)*
  - Grammaires, arbres, forêts (partagées), graphes
- Sémantique : “donner” un sens, lever les ambiguïtés
  - Graphes conceptuels, prédicats logiques :
    ce chien est curieux, il poursuit les [filles à vélo]
- Pragmatique : juger la pertinence selon le contexte
  - *J’ai froid → ferme la fenêtre*

Traitement lexical : buts (1)

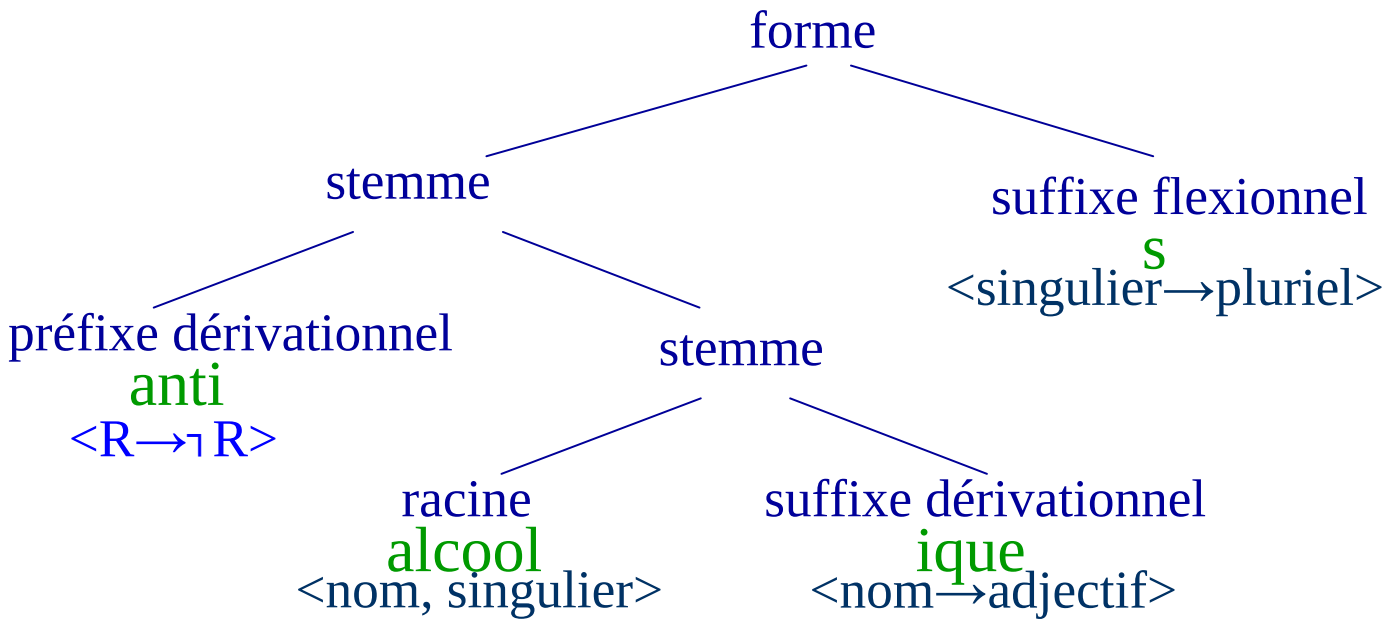
- Décomposition en unités lexicales (lexème)
  - Trouver les *mots* : ambiguïté des séparateurs, par ex. le point (point final, abréviation, sigles comme S.N.C.F.)
  - Un segmenteur (tokenizer) doit connaître les règles d'usage des signes de ponctuation de la langue
  - Une forme enrichie des textes (HTML…) peut être une aide
  - Taux d'erreur des meilleurs segmenteurs pour reconnaître la fin d'une phrase ≈ 1%

Traitement lexical : buts (2)

- Caractériser les lexèmes
  - Nom, adjectif, verbe, …
  - Singulier, pluriel, diminutif, abréviation, …
- Difficile de ranger tous les mots possibles d'une langue dans un dictionnaire :
  - conjugaison des verbes, composition de noms, néologismes, …
- Avoir une forme de dérivation des mots à partir d'une racine

Traitement lexical : moyens (1)

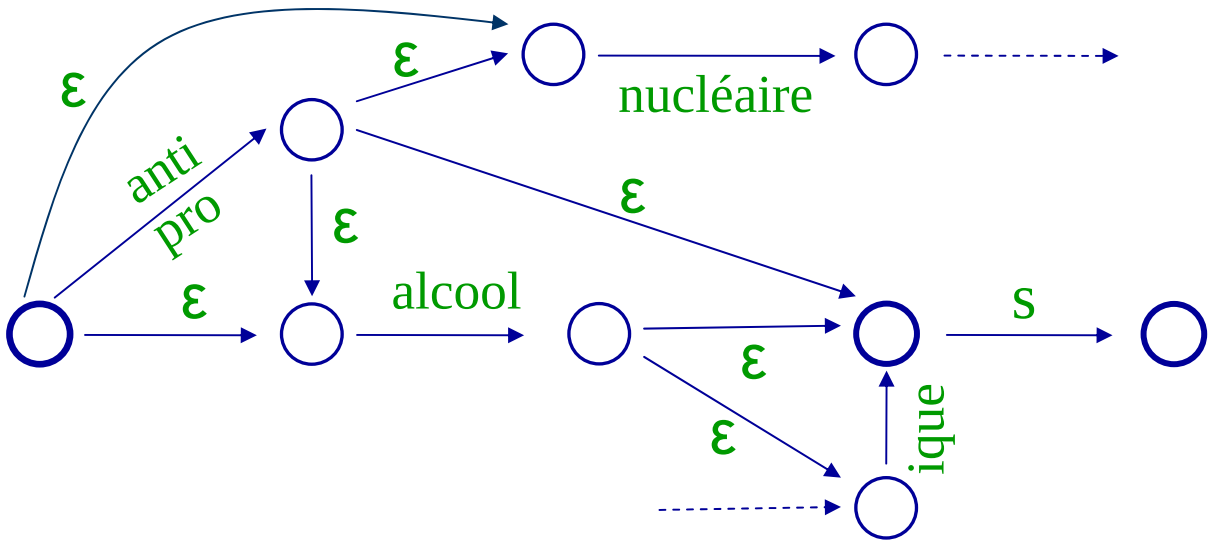
- Exemple de règle de formation des mots (français)



antialcooliques : <adjectif, pluriel, (idée d'opposition à alcool)>

Traitement lexical : moyens (2)

- Par un automate d'état fini (non déterministe)



Les pro-alcooliques et les antis ?

Antimilita(i)r(e)iste → règles d'ajustement phonologique

Pour synthétiser les caractéristiques, il faut un *transducteur*

Traitement lexical : moyens (3)

- Représentation intentionnelle et extentionnelle
  - Intentionnelle : la racine et ses dérivations possibles, organisées une hiérarchie de classes

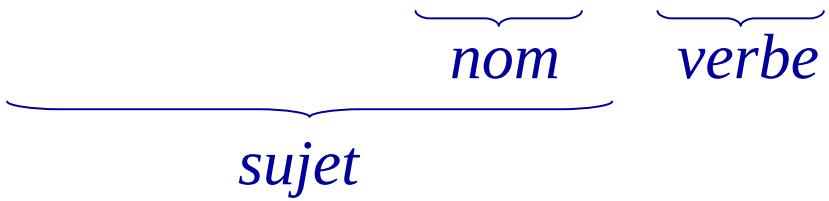
    blanc:adjectif@couleur
    >nominalisé // laisser un blanc
    >verbe-causatif: -(h)ir // blanchir
    >...

  - Extensionnelle : production de tous les mots déductibles de la forme intentionnelle

    blanc:adjectif,masculin,singulier@couleur
    blanches:adjectif,féminin,pluriel[blanc]
    blanchir:verbe-causatif,infinitif[blanc]
    blanchira:verbe[blanchir],futur,singulier,3ème pers

Traitement syntaxique : buts

- Vérifier la validité d'une suite de lexèmes
  - Permet de lever des ambiguïtés lexicales comme dans *les poules du couvent couvent*



  - Vérification de fautes d'accord (mais grammaire très lourde) :
    groupe-nominal → *article-pluriel nom-pluriel*
    | …
- Dégage une organisation (structuration) hiérarchique : *sujet, verbe, complément, …*

Traitement syntaxique : outils

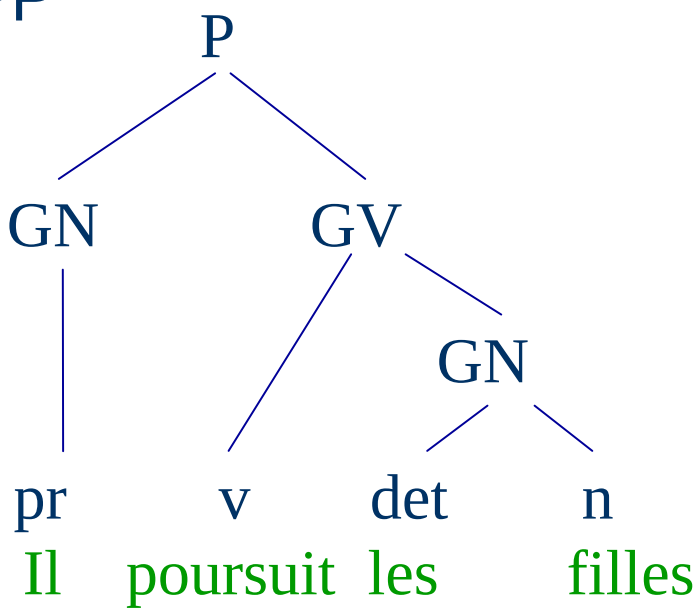
- Grammaires non contextuelles (context free)
  - Plusieurs (dizaines de) milliers de règles
  - Nécessairement ambiguës, et reconnaissent un sur-langage
    - Quelle que soit la méthode d'analyse employée, construction d'une forêt partagée d'arbres syntaxiques : les phases sémantique/pragmatique choisiront l'arbre final
- Autres formalismes :
  - Grammaires lexicales fonctionnelles (CFG enrichies)
  - Grammaires d'adjonction d'arbre (en pratique lexicalisées)
  - Grammaires Catégorielles (lexicalisées par nature)
  - Grammaires Probabilistes, Head-driven Phrase Structure Grammars, Range Concatenation Grammars…

Arbres de dérivation

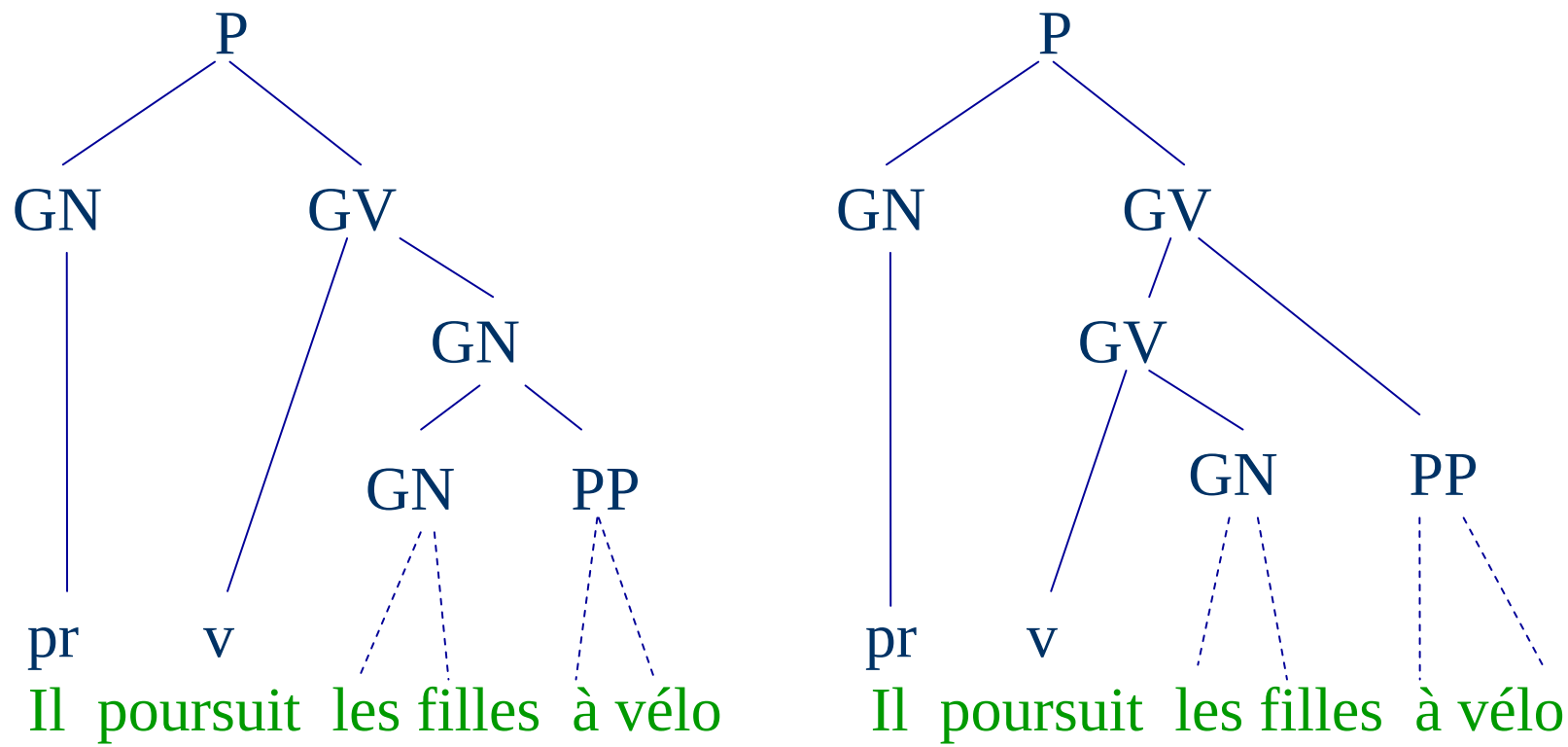
Soit la grammaire

P → GN GV
GN → pr | det n | GN PP
GV → v GN | GV PP
PP → p GN

Arbre de dérivation pour la phrase
Il poursuit les filles

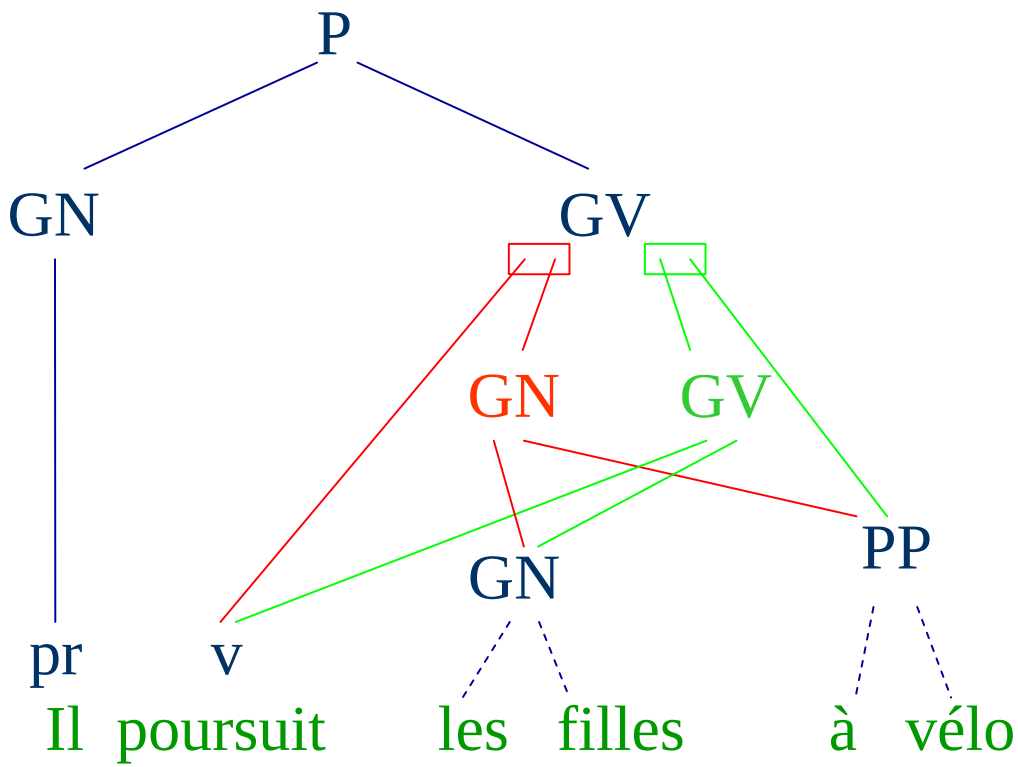


Arbres de dérivation et phrases ambiguës



Ambiguïté non soluble en l'absence d'informations sémantiques contextuelles

Forêt partagée



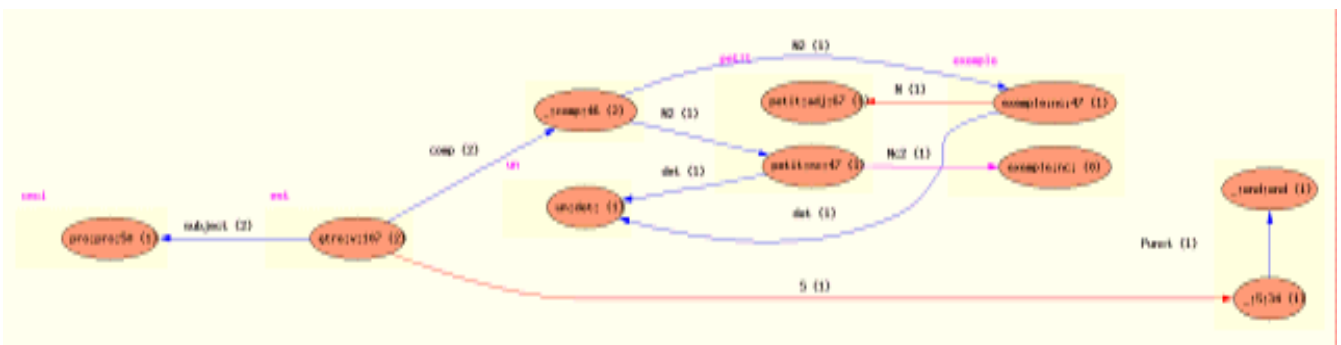
Grammaire décrivant la forêt

$P_0 \rightarrow GN_0\ GV_0$
$GN_0 \rightarrow$ pr (il)
$GV_0 \rightarrow$ v(poursuit) GN_1
$GV_0 \rightarrow GV_1\ PP_0$
$GN_1 \rightarrow GN_2\ PP_0$
$GV_1 \rightarrow$ v(poursuit) GN_2
$GN_2 \rightarrow$ det(les) n(filles)
$PP_0 \rightarrow$ p(à) n(vélo)

cf article à lire

Grammaires et structures fonctionnelles

- En fait, il n'y a pas superposition des fonctions grammaticales et des positions syntaxiques
  - Les positions syntaxiques induisent un arbre : structure de constituants
  - Les fonctions grammaticales induisent un graphe de dépendance : structure fonctionnelle



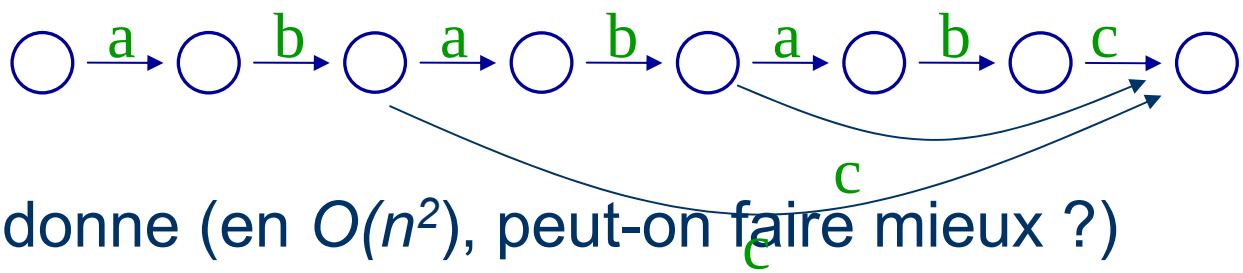
d'après http://atoll.inria.fr/parserdemo

Traitement lexical : moyens (3)

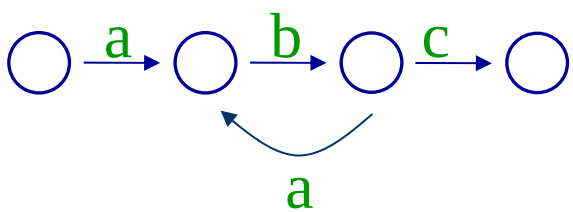
- Relations entre concepts
  - Synonymie : X est équivalent à Y
  - Antonymie : X est l'opposé de Y
  - Hyponymie : X est une spécialisation de Y
  - Hyperonymie : X est une généralisation de Y
- Mécanismes de composition de mots
  - Nom + adjectif (*système distribué*)
  - Nom + préposition + nom (*réseau de neurones*)

Traitement lexical : quelques problèmes

- Minimisation des automates
  - Par un automate reconnaissant un langage plus grand (une *couverture*)

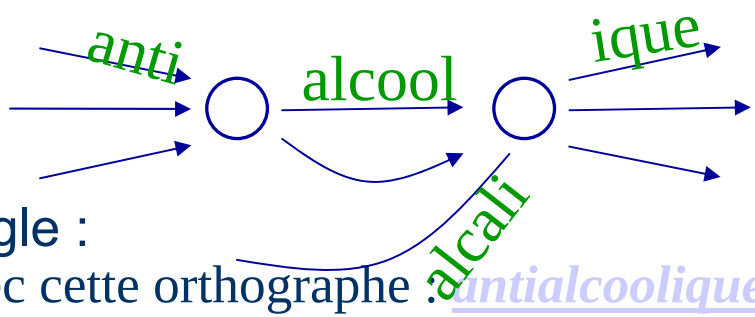


donne (en $O(n^2)$, peut-on faire mieux ?)



Traitement lexical : quelques problèmes

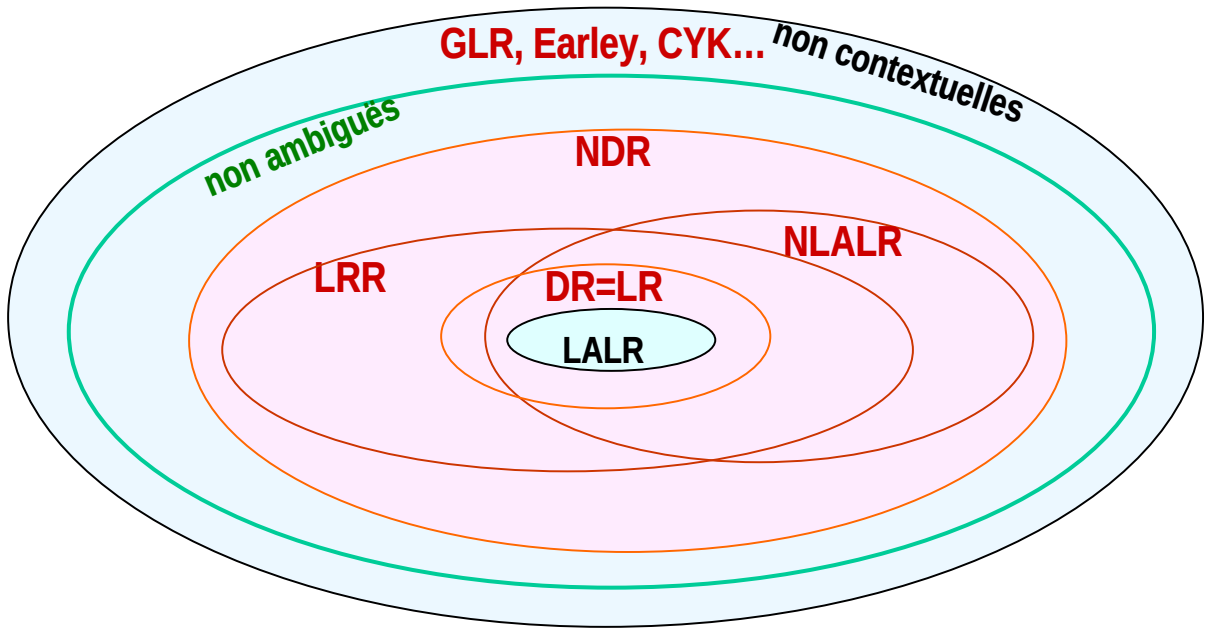
- Tolérance aux erreurs (fautes d'orthographe)
  - Notion de région dans laquelle on cherche à corriger l'erreur : pour antialcalique, chercher à corriger par un mot de la région



  - Essai Google :
    Essayez avec cette orthographe : *antialcoolique*
    Aucun document ne correspond aux termes de recherche spécifiés (**antialcalique**)

Traitement syntaxique : problèmes

- Analyseurs courants basés sur LR(0), SLR(1) ou LALR(1)
  - Provoquent des conflits là où il n'y a pas d'ambiguïté
  - Élargir la taille de la fenêtre ?



Grammaires : quelques conventions d'écriture

- Une lettre minuscule du début de l'alphabet latin représente un élément du vocabulaire terminal T : a,b,c…
- Une lettre minuscule de la fin de l'alphabet latin représente un mot du vocabulaire terminal (dans T*) : x,y,z…
- Une lettre majuscule du début de l'alphabet latin représente un élément du vocabulaire non terminal N : A,B,C…
- Une lettre majuscule de la fin de l'alphabet latin représente un élément du vocabulaire $V = T \cup N$: X,Y,Z…
- Une lettre grecque représente un mot du vocabulaire (dans V*) : $\alpha,\beta,\gamma\ldots$; ε représente le mot vide
- Les productions, par exemple { $S \rightarrow c$, $S \rightarrow A\ S\ b$, $A \rightarrow a$ }, qui permettent d'engendrer le langage (sur l'exemple : $a^n\ c\ b^n, n \geq 0$)
- Une chaîne de dérivations : $S \Rightarrow A\ S\ b \Rightarrow A\ A\ S\ b\ b \Rightarrow \ldots \Rightarrow A^k\ S\ b^k$

Hiérarchie de Chomsky (1)

- Type 3 : grammaires régulières (ou rationnelles)
  régles de la forme $A \rightarrow a$ ou $A \rightarrow a\ B$ (ou $A \rightarrow B\ a$)
  - automates à états finis, analyse linéaire
  - langages clos par intersection, concaténation, complémentation

- Type 2 : grammaires non contextuelles
  régles de la forme $A \rightarrow \alpha$
  - automates à pile, analyse polynomiale ($O(n^3)$ au pire)
  - langages clos par intersection

Hiérarchie de Chomsky (2)

- Type 1 : grammaires contextuelles
  régles de la forme $\alpha \rightarrow \beta$, $|\beta| \geq |\alpha|$
  - Machine de Turing, analyse exponentielle
  - langages clos par intersection, concaténation (et complémentation?)
  - Des formalismes faiblement contextuels permettent une analyse polynomiale ($O(n^6)$)
- Type 0 : grammaires générales
  régles sans restriction
  - Machine de Turing mais appartenance d'un mot au langage indécidable