# Chaînes de traitement syntaxique

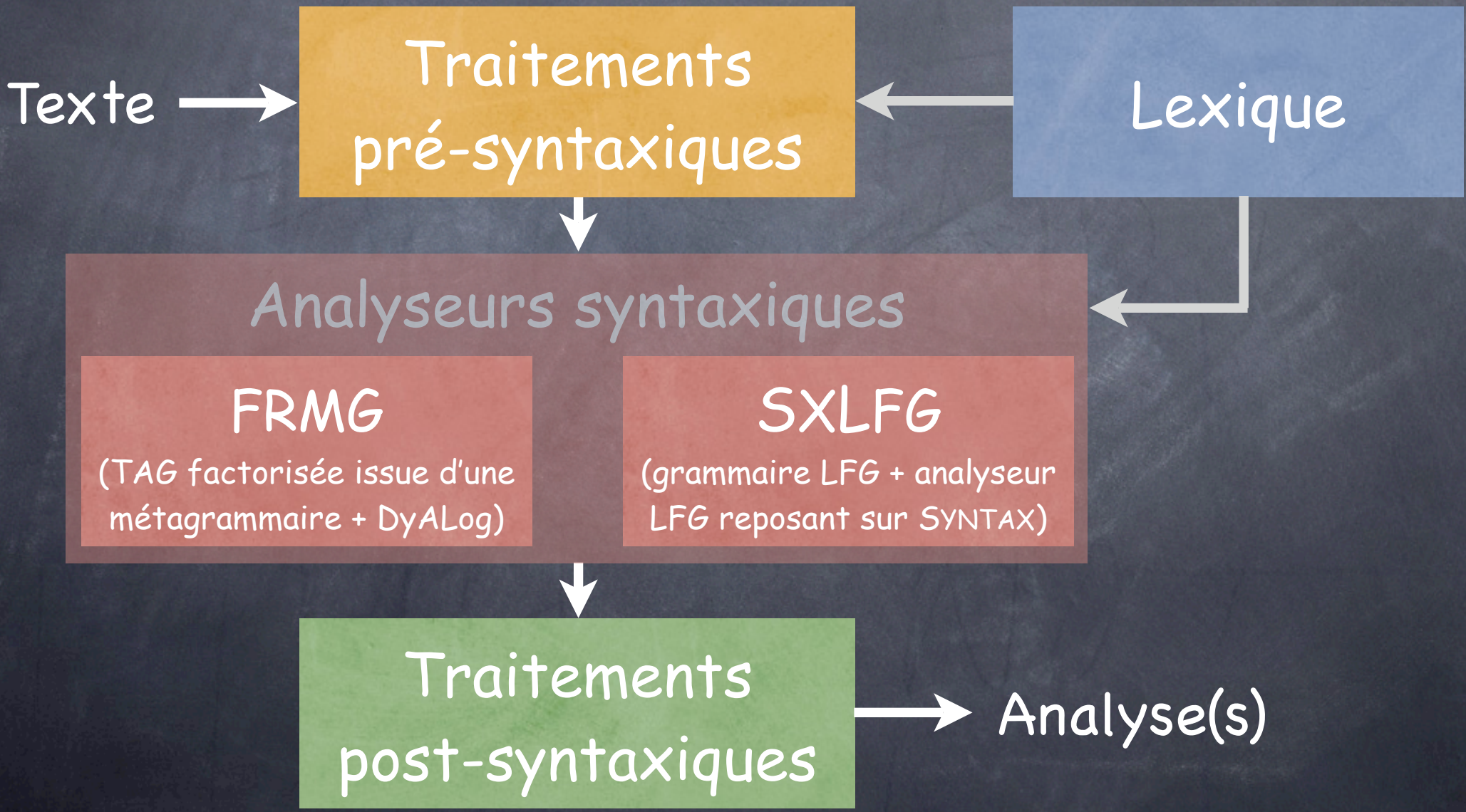
P. Boullier, L. Clément, B. Sagot, É. de La Clergerie
INRIA Rocquencourt - Projet ATOLL

TALN 2005 - Dourdan

# Introduction

- Le projet Atoll de l'INRIA développe des outils pour le TAL, et en particulier
  - des analyseurs non-probabilistes ambigus,
  - plus récemment, des grammaires, un lexique et une chaîne de traitement pré-syntaxique
- Deux chaînes d'analyse syntaxique profonde sont déployables à grande échelle
  → participation à la campagne EASy.

# Architecture générale



# 1. Lexique

- Lefff 2 (Lexique des formes fléchies du français) [Sagot et al. 2005]
  - 600 000 formes fléchies
  - 400 000 entrées, certaines factorisées
  - informations morphologiques et syntaxiques:
    - sous-catégorisation des verbes
    - informations syntaxiques diverses
- Acquis en partie à partir d'une analyse statistique automatique de gros corpus [Clément, Sagot et Lang 2004]

# 2. Traitements pré-syntaxiques

- Corpus bruts (et ceux d'EASy en particulier) = loins des phrases forgées par des linguistes

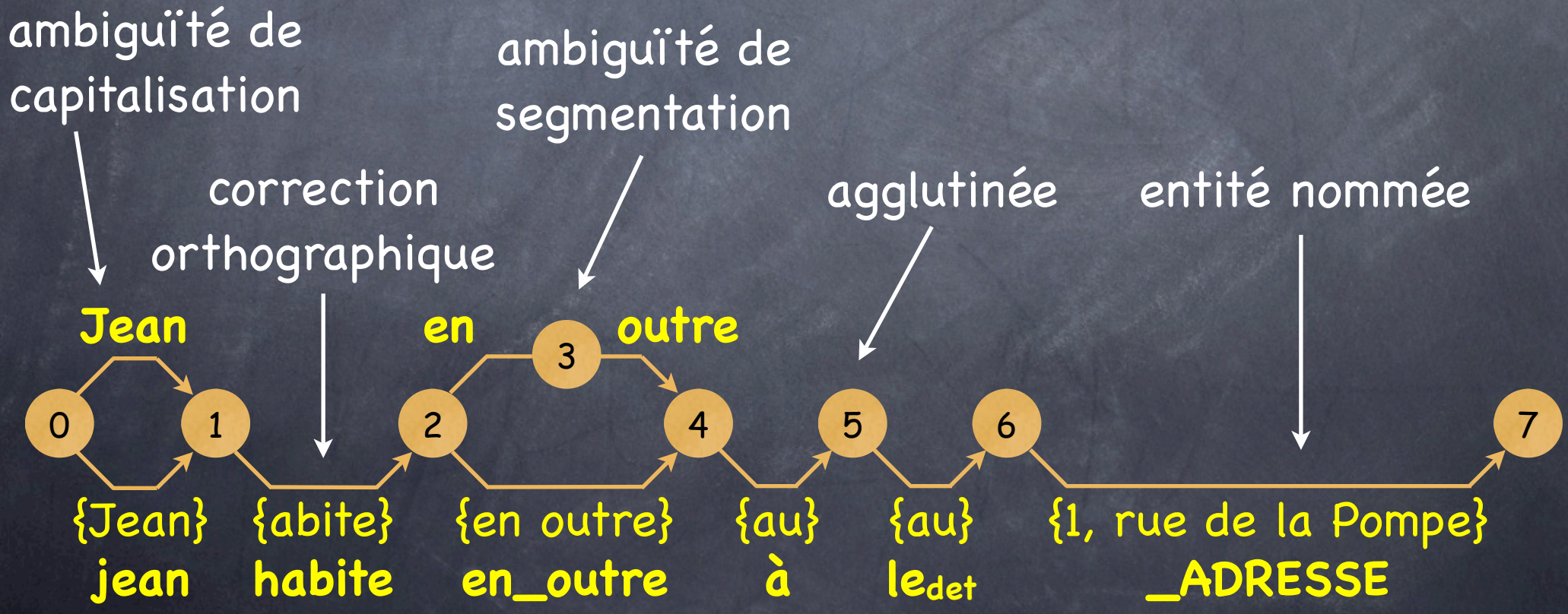
→ Traitements non-déterministes indispensables en préalable à l'analyse syntaxique
  - entités nommées
  - segmentation en phrases et en mots
  - correction orthographique

- Nous avons mis en place une chaîne pour effectuer ces traitements, appelée SxPipe [Sagot et Boullier 2005]

# 2. Traitements pré-syntaxiques

- Sur un exemple :

  **Jean abite en outre au 1, rue de la Pompe**



# 2. Traitements pré-syntaxiques

- Évaluation partielle de SxPipe

| | Rappel | Précision | Volume du corpus |
|---|---|---|---|
| URL | 100% | 100% | $1{,}1.10^6$ mots |
| adresses physiques | 100% | 100% | $1{,}1.10^6$ mots |
| expressions en langue étrangère | 88% | 83% | 2000 phrases |
| segmentation | 100% | 100% | 400 phrases |
| correction orthographique | - | 91% | $1{,}1.10^6$ mots |

# 3. Analyseurs syntaxiques

- Pour EASy, nous avons mis en place 2 analyseurs syntaxiques:
  - FRMG [Thomasset et de La Clergerie 2005]
  - SXLFG [Boullier, Sagot et Clément 2005]
- Ils diffèrent par:
  - le formalisme utilisé (TAG avec décorations / LFG),
  - la grammaire,
  - le générateur d'analyseurs (DyALog / Syntax+SXLFG)

# 3. Analyseurs syntaxiques
## a. FRMG

- S'appuie sur une grammaire TAG avec décorations générée à partir d'une méta-grammaire par le système DyALog
- La grammaire obtenue est très compacte, car ses quasi-arbres TAG sont factorisés:
  - disjonctions entre nœuds,
  - répétitions de nœuds,
  - nœuds optionnels contrôlés par des gardes
- L'ancrage lexical se fait par des hypertags

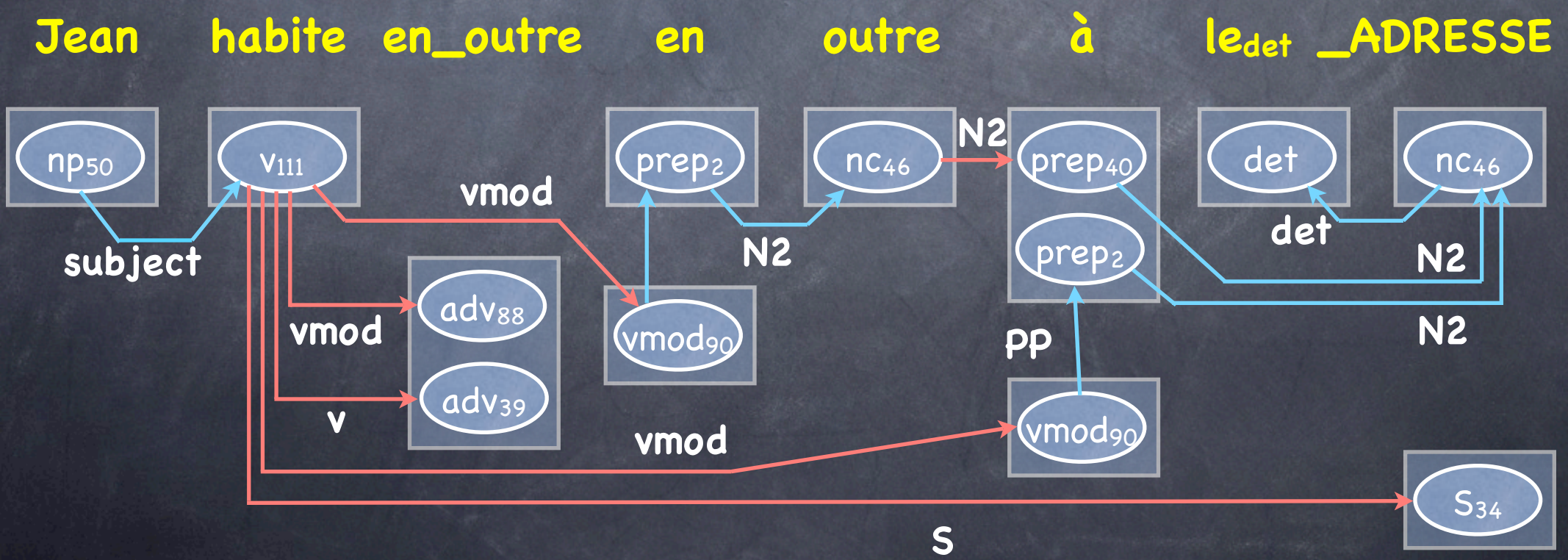
# 3. Analyseurs syntaxiques
## a. FRMG

- La grammaire est compilée dans le système DyALog en un analyseur hybride TAG/TIG tabulaire, qui peut prendre en entrée un "treillis" de mots tel que généré par SxPipe
- En cas d'échec, l'analyseur passe en mode "robuste": il cherche un ensemble d'analyses partielles couvrant au mieux la phrase
- Le résultat est donné sous la forme d'une forêt de dérivation

# 3. Analyseurs syntaxiques

## a. FRMG

- Exemple de forêt de dépendances obtenue à partir de la sortie de FRMG



# 3. Analyseurs syntaxiques
# b. SXLFG

- SXLFG repose sur une grammaire LFG issue de celle utilisée par XLFG [Clément et Kinyon 2001]
- Comme toute grammaire LFG, elle est constituée d'une grammaire CFG support décorée par des équations fonctionnelles
- La grammaire CFG sous-jacente est compilée par le système Syntax en un analyseur à la Earley qui produit une forêt partagée

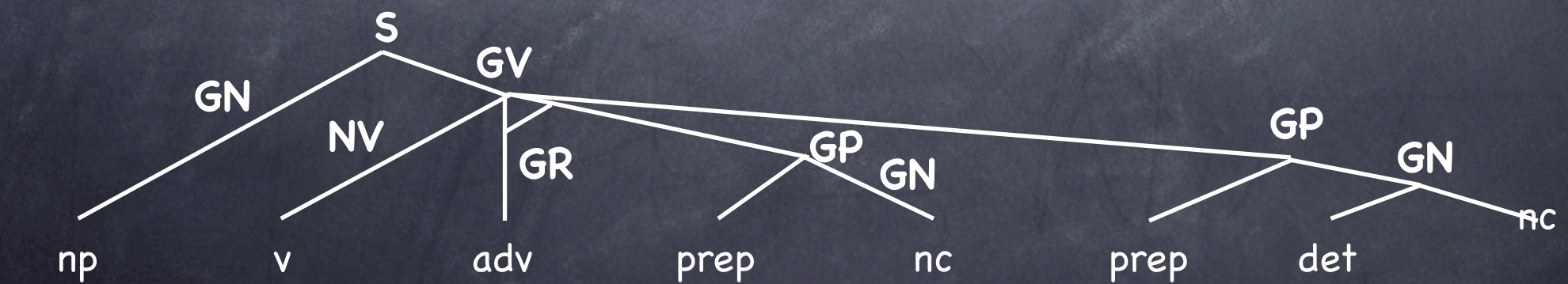
# 3. Analyseurs syntaxiques
## b. SXLFG

- Sur cette forêt sont évaluées les équations fonctionnelles selon une stratégie bottom-up
- La forêt est alors filtrée pour ne garder que les structures en constituants (analyses CFG) correspondant à des structures fonctionnelles correctes
- Des mécanismes de rattrapage d'erreur à tous les niveaux, ainsi qu'un mécanisme de sur-segmentation de la phrase d'entrée en cas d'échec global, en font un analyseur robuste

# 3. Analyseurs syntaxiques

## b. SXLFG

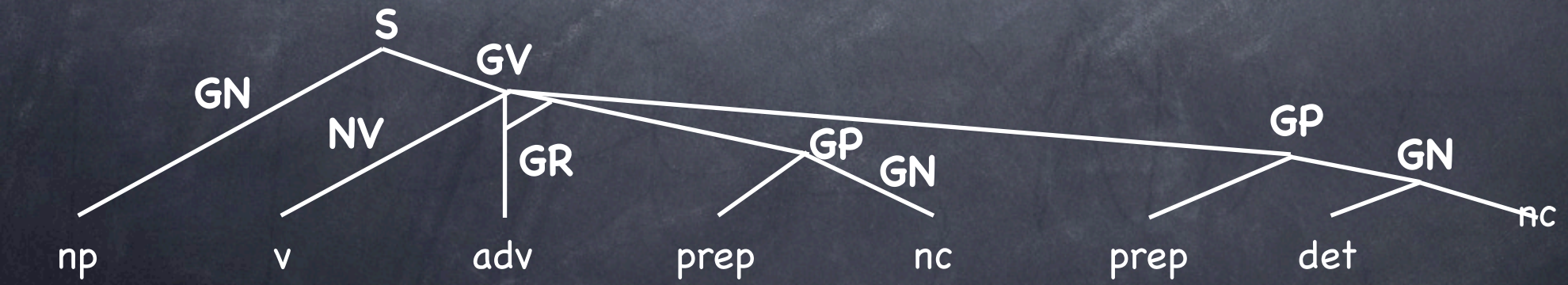
```
[ pred="habiter<subj,(obj)>"
  subj=[pred="Jean"]
  number=sing ; person=3
  mode=indicative|subjonctive
  tense=present
  adjunct= { [ pred="en<obj>"
               obj= [ pred="outre"
                      ... ] ],
             [ pred="à<obj>"
               obj= [ pred="_ADRESSE"
                      ... ] ] } ]
```



# 3. Analyseurs syntaxiques

## b. SXLFG

```
[ pred="habiter<subj,(obj)>"
  subj=[pred="Jean"]
  number=sing ; person=3
  mode=indicative|subjonctive
  tense=present
  adjunct= { [pred="en_outre"] , [ pred="à<obj>"
                                   obj= [ pred="_ADRESSE"
                                          ... ] ] } ]
```




Jean habite en_outre en outre à le$_{det}$ _ADRESSE

# 4. Traitements post-syntaxiques

- Désambiguïsation par heuristiques
  - Dans FRMG, elle se fait sur la forêt de dépendances (arbre de dérivation), en attribuant à chaque arc un certain poids, et en cherchant l'ensemble d'arcs de poids minimal couvrant la phrase de façon cohérente (contraintes topologiques)
  - Dans SXLFG, elle se fait sur les structures fonctionnelles à l'aide d'heuristiques appliquées en cascade. Elle est suivie d'un élagage de la forêt d'analyse CFG pour ne garder que la structure en constituants correspondant à la structure fonctionnelle choisie

# 4. Traitements post-syntaxiques

- Conversion au format EASy
  Paradoxalement, c'est une des étapes induisant le plus d'erreurs... mais le manque de préparation en est le responsable

| GN1 | NV2 | GR3 | | GP4 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Jean** | **abite** | **en** | **outre** | **au** | **1** | **,** | **rue** | **de** | **la** | **Pompe** |
| F1 | F2 | F3 | F4 | F5 | F6 | F7 | F8 | F9 | F10 | F11 |

| sujet | verbe |
|---|---|
| GN1 | NV2 |

| complmt | verbe |
|---|---|
| GP4 | NV2 |

| modifieur | verbe |
|---|---|
| GR3 | NV2 |

# 5. Mise en œuvre et résultats

- Nous avons utilisé ces deux chaînes d'analyse syntaxique pendant la campagne EASy (35 000 phrases de corpus brut divers et de qualité variée à analyser)
- Nous avons également testé nos analyseurs sur des corpus tels que TSNLP ou Eurotra
- Nous utilisons également ces chaînes pour l'extraction d'informations à partir de corpus spécialisé de botanique (projet BIOTIM)

# 5. Mise en œuvre et résultats

- Résultats pour FRMG (timeout de 100s)

| Corpus | #phr. | couv. | temps d'analyse | | | amb. |
|---|---|---|---|---|---|---|
| | | | moy. | méd. | <1s | |
| Eurotra | 334 | 95,8% | 1,8s | 1,3s | 38% | 0,7 |
| TSNLP | 1661 | 93,4% | 0,7s | 0,6s | 78% | 0,4 |
| EASy | 34438 | 42,5% | 5,5s | 1,6s | 36% | 0,6 |

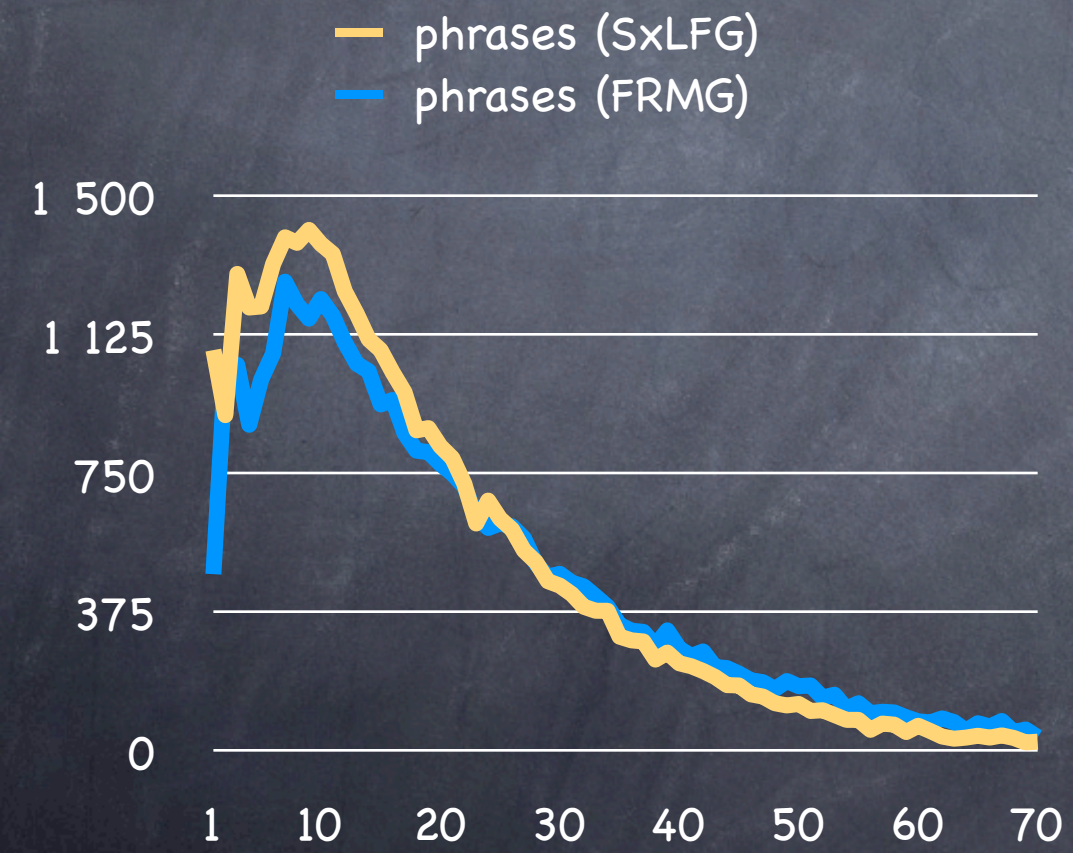
# 5. Mise en œuvre et résultats

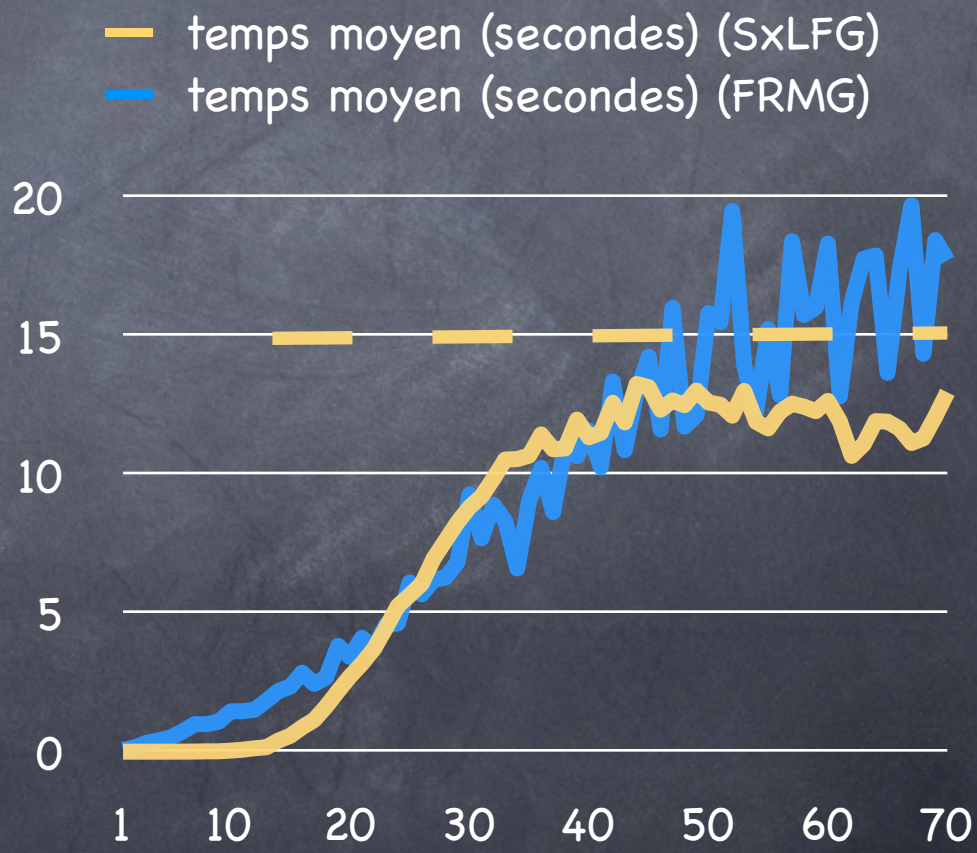
- Résultats pour SXLFG (timeout de 15s)

| Corpus | #phr. | couv. (sans vérif. de cohérence) | couv. (avec vérif. de cohérence) | temps d'analyse | | |
|---|---|---|---|---|---|---|
| | | | | moy. | méd. | <1s |
| Eurotra | 334 | 94,6% | 84,4% | 0,33s | 0,02s | 94% |
| TSNLP | 1661 | 98,5% | 79,1% | 0,03s | 0,0s | 99% |
| EASy | 40859 | 66,7% | 42,0% | 3,3s | 0,03s | 71% |

# 5. Mise en œuvre et résultats

- Comparaison selon la longueur de la phrase



# Conclusion

- Différence considérable entre le développement d'un analyseur syntaxique et le développement d'une chaîne complète d'analyse syntaxique
- Très forte interaction entre les différents composants (lexique, grammaire et analyseur en particulier, mais aussi pré-traitement et post-traitement)

# Perspectives

- Exploitation des résultats de la campagne EASy
- Exploitation de la disponibilité de deux systèmes d'analyse différents.
  En particulier:
  - amélioration des ressources lexicales
  - détection d'erreurs ou de manques dans les grammaires et la chaîne de traitement pré-syntaxique
- Extraction d'informations à grande échelle