

# Formalismes pour l'analyse syntaxique

Cours ENST – TALN  
Vendredi 1er Mars 2002

Éric de la Clergerie  
[Eric.De\\_La\\_Clergerie@inria.fr](mailto:Eric.De_La_Clergerie@inria.fr)

Transparents disponibles sur <http://atoll.inria.fr/~clerger> ► Enseignement

# La syntaxe existe-t-elle ?

A priori oui :

Un schtroumpf sachant schtroumpfer

"And hast thou slain the Jabberwock ?  
Come to my arms, my beamish boy !  
O frabjous day ! Callooh ! Callay !"  
He chortled in his joy.

'Twas brillig, and the slithy toves  
Did gyre and gimble in the wabe ;  
All mimsy were the borogoves,  
And the mome raths outgrabe.

« **Jabberwocky** » (Lewis Carroll)

Colorless green ideas sleep furiously [Noam Chomsky]

Mais débat sur

- la nature des constituants : Nom, Verbe, Determinant Syntagme Nominal, Syntagme Verbal
- les relations entre constituants : arborescentes, graphe, ...
- les fonctions linguistiques : sujet, objet, objet indirect, ...

# Théories linguistiques

Explication des phénomènes linguistiques

**ordre des mots** SOV VOS

**passif** le livre est donné à Paul par Anne

**extraction** Jean demande quel livre Paul est en train de lire

**Relatives** Jean connais l'homme **que** Marie regarde  $\epsilon$ .

**Interrogative** Jean demande **quel homme** Marie regarde  $\epsilon$ .

**Clivées** C'est Jean **que** Marie regarde  $\epsilon$ .

**Topicalisées** **Le chocolat amer**, Marie aime  $\epsilon$ .

**coordination** Jean mange une pomme et Paul une poire

Existence d'une théorie « universelle » (couvrant tout les language) ?

Liens avec la sémantique ?

# Formalismes

Cadre mathématique pour faire fonctionner les théories

Compromis entre expressivité et complexité (pour un traitement automatique)

Hiérarchie de Chomsky :

**Type 3** grammaires régulières (FSA)  $\Rightarrow$  linéaire

**Type 2** grammaires hors-contextes (PDA)  $\Rightarrow$  polynomiale

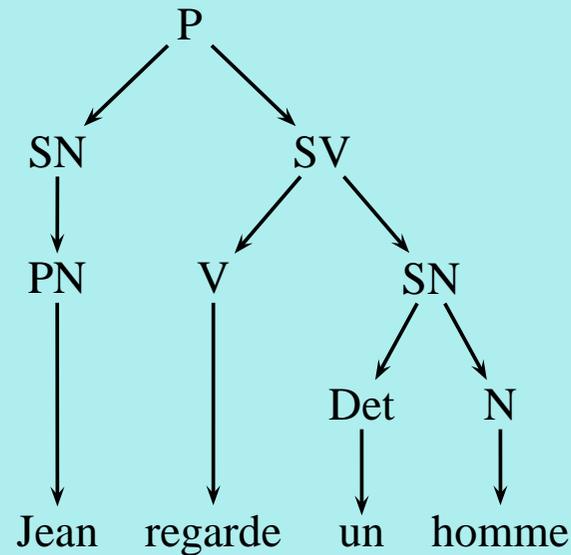
**Type 1** grammaires contextuelles (**Context-sensitive grammars**)

$uXv \rightarrow uyv$  y séquence non vide.  $\Rightarrow$  exponentielle

**Type 0** grammaires non contraintes (langage récursivement énumérable)  $\Rightarrow$  indécidable

# CFG

P --> SN SV  
SN --> NP  
SN --> Det N  
SV --> V SN  
NP --> 'Jean'  
V --> 'regarde'  
Det --> 'un'  
N --> 'homme'



Difficultés de gestion des accords (nombre, genre, ... ) : \*Jean mange une homme

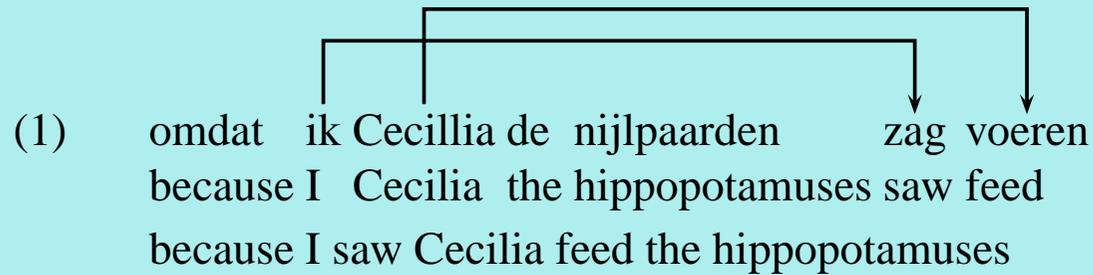
Mais possible :

SN\_masc\_sing --> Det\_masc\_sing N\_masc\_sing

SN\_masc\_plur --> Det\_masc\_plur N\_masc\_plur

## Autres limites

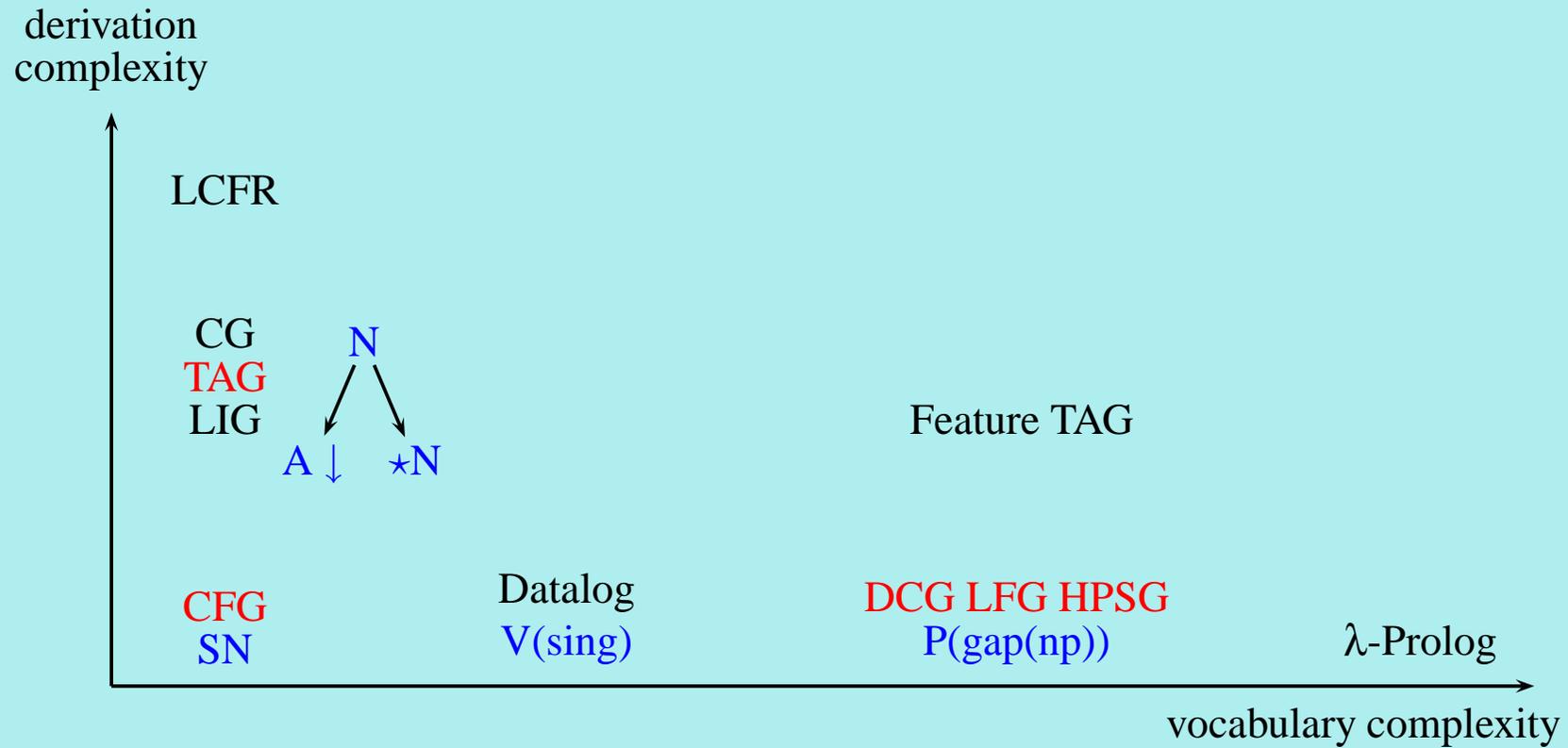
Les CFG manquent de puissance générative, par exemple pour les **croisements**



Autres difficultés :

- dépendances non bornées, extraction (topicalisation, relative, ... )
- ordre « libre » des mots (**word scrambling**)

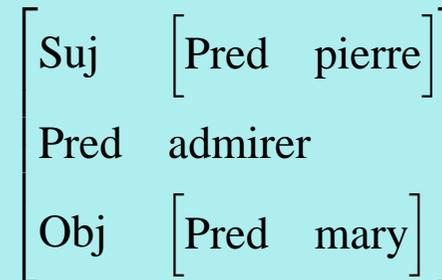
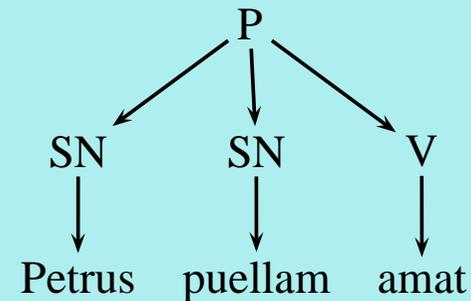
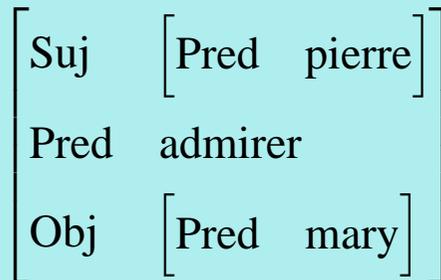
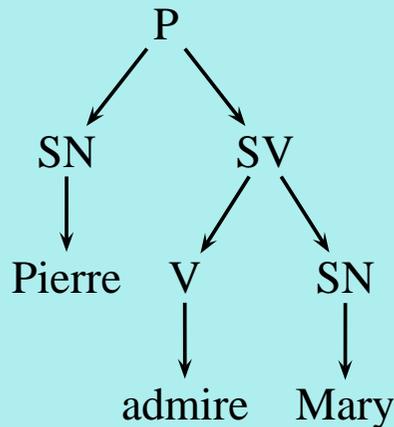
# Nature des formalismes grammaticaux



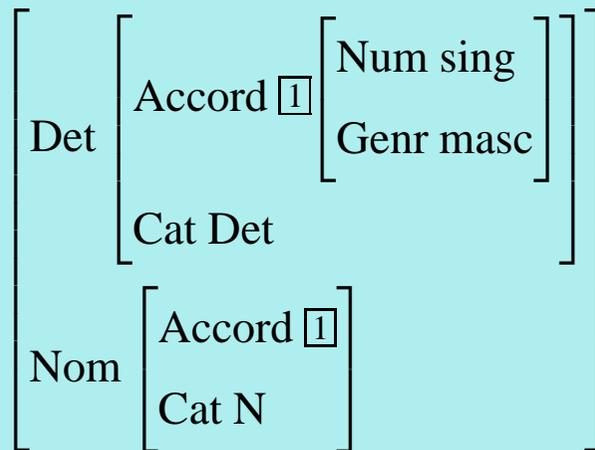
# Lexical Functional Grammars [LFG]

Bresnam et Kaplan (1982) **The mental representation of grammatical representation**

Théorie : Associer structures de constituants (**c-structures**) et structures fonctionnelles (**f-structure**) :



# Formalisme : Structures de traits



# Unification de structures de traits

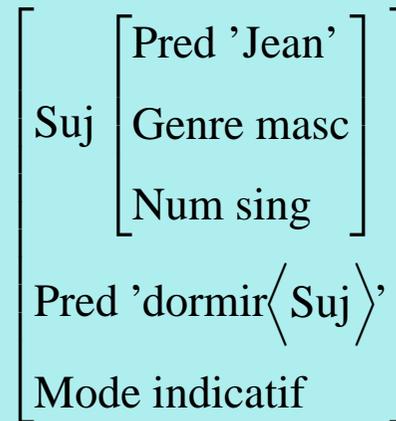
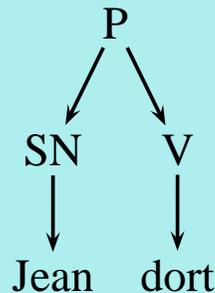
$$\left[ \begin{array}{l} \text{Det} \left[ \text{Accord} \left[ \text{Num sing} \right] \right] \\ \text{Nom} \left[ \begin{array}{l} \text{Accord} \left[ \text{Num sing} \right] \\ \text{Cat N} \end{array} \right] \end{array} \right] \sqcup \left[ \text{Det} \left[ \text{Accord} \left[ \text{Genre Masc} \right] \right] \right] = \left[ \begin{array}{l} \text{Det} \left[ \text{Accord} \left[ \begin{array}{l} \text{Num sing} \\ \text{Genre masc} \end{array} \right] \right] \\ \text{Nom} \left[ \begin{array}{l} \text{Accord} \left[ \text{Num sing} \right] \\ \text{Cat N} \end{array} \right] \end{array} \right]$$

$$\left[ \begin{array}{l} \text{Det} \left[ \text{Accord} \boxed{1} \left[ \text{Num sing} \right] \right] \\ \text{Nom} \left[ \begin{array}{l} \text{Accord} \boxed{1} \\ \text{Cat N} \end{array} \right] \end{array} \right] \sqcup \left[ \text{Det} \left[ \text{Accord} \left[ \text{Genre Masc} \right] \right] \right] = \left[ \begin{array}{l} \text{Det} \left[ \text{Accord} \boxed{1} \left[ \begin{array}{l} \text{Num sing} \\ \text{Genre masc} \end{array} \right] \right] \\ \text{Nom} \left[ \begin{array}{l} \text{Accord} \boxed{1} \\ \text{Cat N} \end{array} \right] \end{array} \right]$$

# Productions et équations fonctionnelles

Une grammaire est donnée par des productions CFG décorées par des équations fonctionnelles.

$P \longrightarrow$	SN	V	$SN \longrightarrow$	Jean	$V \longrightarrow$	dort
	$(\uparrow\text{Suj})=\downarrow$	$\uparrow=\downarrow$		$(\uparrow\text{Num})=\text{sing}$		$(\uparrow\text{Suj Num})=\text{sing}$
				$(\uparrow\text{Genre})=\text{masc}$		$(\uparrow\text{Suj Pers})=3$
				$(\uparrow\text{Pred})=\text{'Jean'}$		$(\uparrow\text{Mode})=\text{indicatif}$
						$(\uparrow\text{Pred})=\text{'dormir<Suj>'}$



# Variations sur les équations

## Contraintes existentielles :

$$\begin{array}{ccccccc}
 \text{SN} \longrightarrow & (\text{Det}) & \text{N} & \text{N} \longrightarrow & \text{Jean} & \text{N} \longrightarrow & \text{chien} & \text{Det} \longrightarrow & \text{le} \\
 & \uparrow=\downarrow & \uparrow=\downarrow & & \sim(\uparrow\text{Det}) & & (\uparrow\text{Det}) & & (\uparrow\text{Det})=\text{le}
 \end{array}$$

## Equations « contraintes »

$$\begin{array}{ccc}
 \text{P}' \longrightarrow & \text{SN} & \text{P} \\
 & (\downarrow\text{Qu})=_{c+} & (\uparrow\text{Qu})=+ \\
 & & \uparrow=\downarrow
 \end{array}$$

## Equations ensemblistes

$$\begin{array}{ccc}
 \text{SV} \longrightarrow & \text{V} & (\text{SN}) & & (\text{SP})^* \\
 & \uparrow=\downarrow & \uparrow\text{Obj}=\downarrow \text{ ou } \uparrow\text{Ajout}\ni\downarrow & & \uparrow\text{Ajout}\ni\downarrow
 \end{array}$$

(Jean dort le matin.

Jean mange le gateau

Jean mange ce gateau avec Anne)

# Fonctions grammaticales

Fonctions catégorisables : Sujet, Objet, Comp(létive), XComp (infinitives et participiales), Prep-Obj (compléments prépositionnels, obliques)

Fonctions non-catégorisables : XAjout, X étant la catégorie

**Vcomp** Jean veut **partir à Rio**.

**Acomp** Jean devient **fou**.

**Ncomp** Ils ont élu Jean **président**

**Vajout** **Partant en voyage**, Marie se prépare

**Aajout** Paul est parti **content**

**Prep-Obj** Paul ressemble **à Jean**

# Sous-catégorisation

Le champ Pred indique les fonctions attendues par un mot.

**manger** ( $\uparrow$ Pred)='manger<Suj,Obj>'

**donner** ( $\uparrow$ Pred)='donner<Suj,Obj,A-Obj>'

**falloir** ( $\uparrow$ Pred)='falloir<Obj>Suj' et ( $\uparrow$ Suj Form) =<sub>c</sub> il

**vouloir** ( $\uparrow$ Pred)='vouloir<Suj,Vcomp>' et ( $\uparrow$ Suj)=( $\uparrow$ Vcomp Suj) Jean veut venir

**proposer** ( $\uparrow$ Pred)='proposer<Suj,A-Obj,Vcomp>' et ( $\uparrow$ Vcomp Suj)=( $\uparrow$ Suj)/( $\uparrow$ A-Obj)

Jean propose à Jean de venir

**destruction** ( $\uparrow$ Pred)='destruction<De-Obj,Par-Obj>' Destruction de la maison par les promoteurs

## Sous-catégorisation (2)

SP	→	Prep	SN		SP	→	Prep	SV		Prep	→	à
		↑=↓	(↑Obj)=↓				↑=↓	(↑Vcomp)=↓				(↑Pcas)=A
								(↓Mode=inf				

SV	→	V	(SN)	(SP)*	(SP)*
		↑=↓	(↑Obj)=↓	(↑(↓PCas)-Obj)=↓	(↑(↓PCas)-Vcomp)=↓
				(↓Obj)	(↓Vcomp)

# Dérivations lexicales : Analyse du passif

Certaines « transformations » syntaxiques analysables par des entrées lexicales dérivées  
(approche **lexicale**, alternative aux approches **transformationnelles**)

$$\begin{aligned} \text{Suj} &\Rightarrow \emptyset/\text{Par} - \text{Obj} \\ \text{Passif : Obj} &\Rightarrow \text{Suj} \\ \text{Vinf} &\Rightarrow \text{Vppart} \end{aligned}$$

donner, V :  $\uparrow\text{Pred} = \text{'donner}\langle\text{Suj,Obj,A-Obj}\rangle\text{'}$   
 $\uparrow\text{Mode} = \text{inf}$

donné, V :  $\uparrow\text{Pred} = \text{'donner}\langle\emptyset/\text{Par-Obj,Suj,A-Obj}\rangle\text{'}$   
 $\uparrow\text{Mode} = \text{ppart}$   
 $\uparrow\text{Passif} = +$

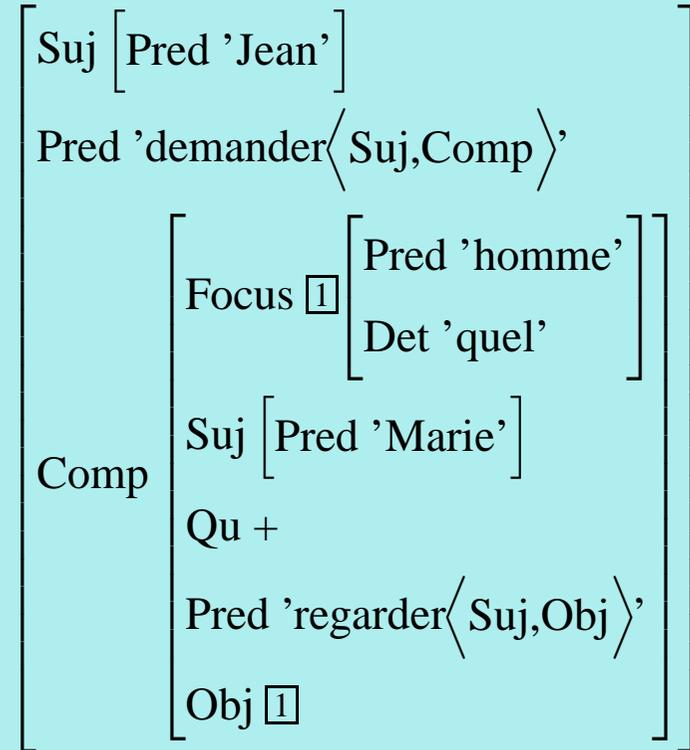
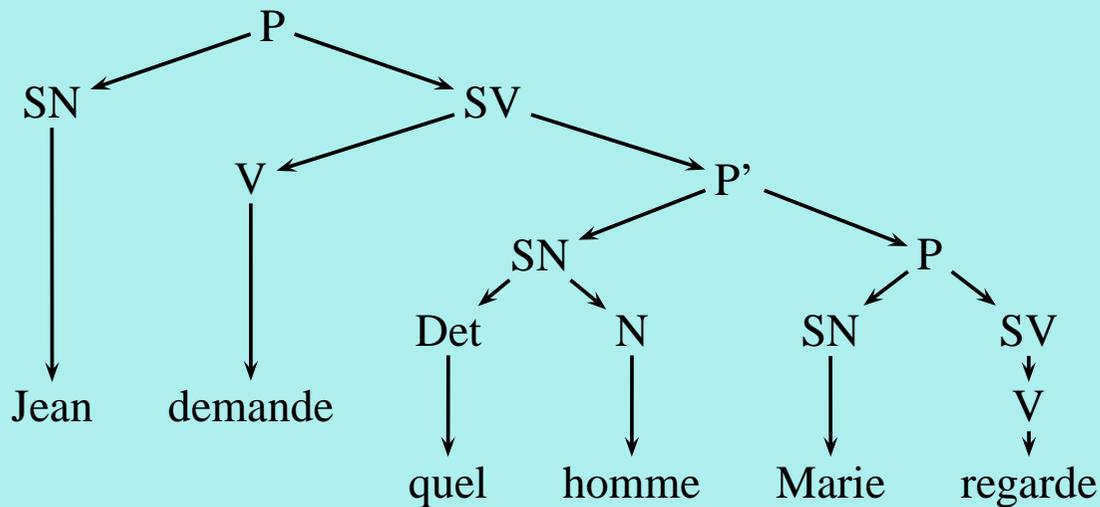
Non passivable : coûter :  $\uparrow\text{Pred} = \text{'couter}\langle\text{Suj,Ncomp}\rangle\text{'}$

# Extractions

$P' \longrightarrow SN \quad P$   
 $(\downarrow Qu) =_c + \quad \uparrow = \downarrow$   
 $(\uparrow Focus) = \uparrow \quad (\downarrow Qu) = +$   
 $(\uparrow Focus) = (\uparrow Obj)$

demande, V :  $(\uparrow Pred) = \text{'demander' } \langle \text{Suj, Comp} \rangle$   
 $(\uparrow Comp Qu) =_c +$

quel, Det :  $(\uparrow Det) = \text{'quel'}$   
 $(\uparrow Qu) = +$



# dépendances non bornées

Nombre arbitraire d'enchâssements entre un constituant extrait et le prédicat dont il dépend :

Jean demande [quel homme Paul pense [que Marie regarde  $\epsilon$ ]]



$P' \longrightarrow SN$

$(\downarrow Qu) =_c +$

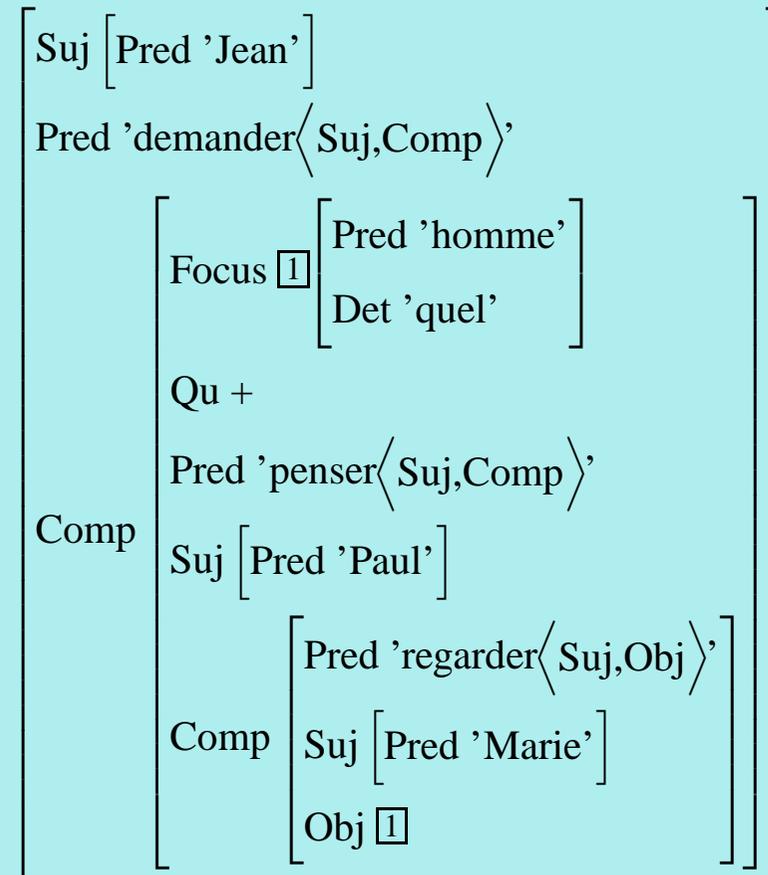
$(\uparrow Focus) = \uparrow$

$(\uparrow Focus) = \uparrow (Comp)^* Obj$

$P$

$\uparrow = \downarrow$

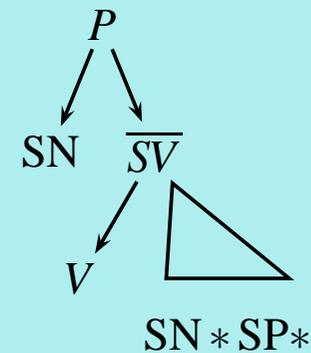
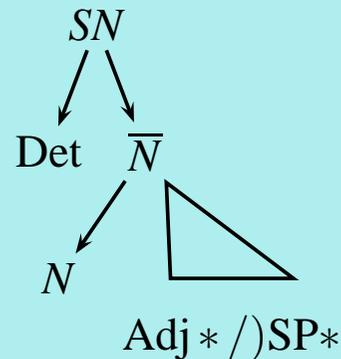
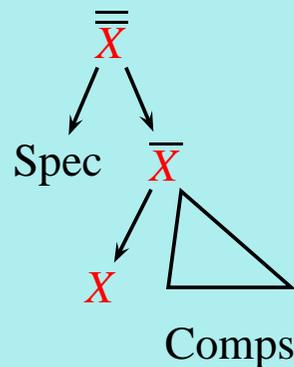
$(\downarrow Qu) = +$



# Head-driven Phrase Structure Grammars [HPSG]

**Pollard et Sag (1987)** *Information-based syntax and semantic*

- Théorie linguistique complète : phonologie, lexical, syntaxe, sémantique et pragmatique
- Approche lexicale, avec importance de la sous-catégorisation
- Notion de **tête linguistique** (Théorie X-Bar)

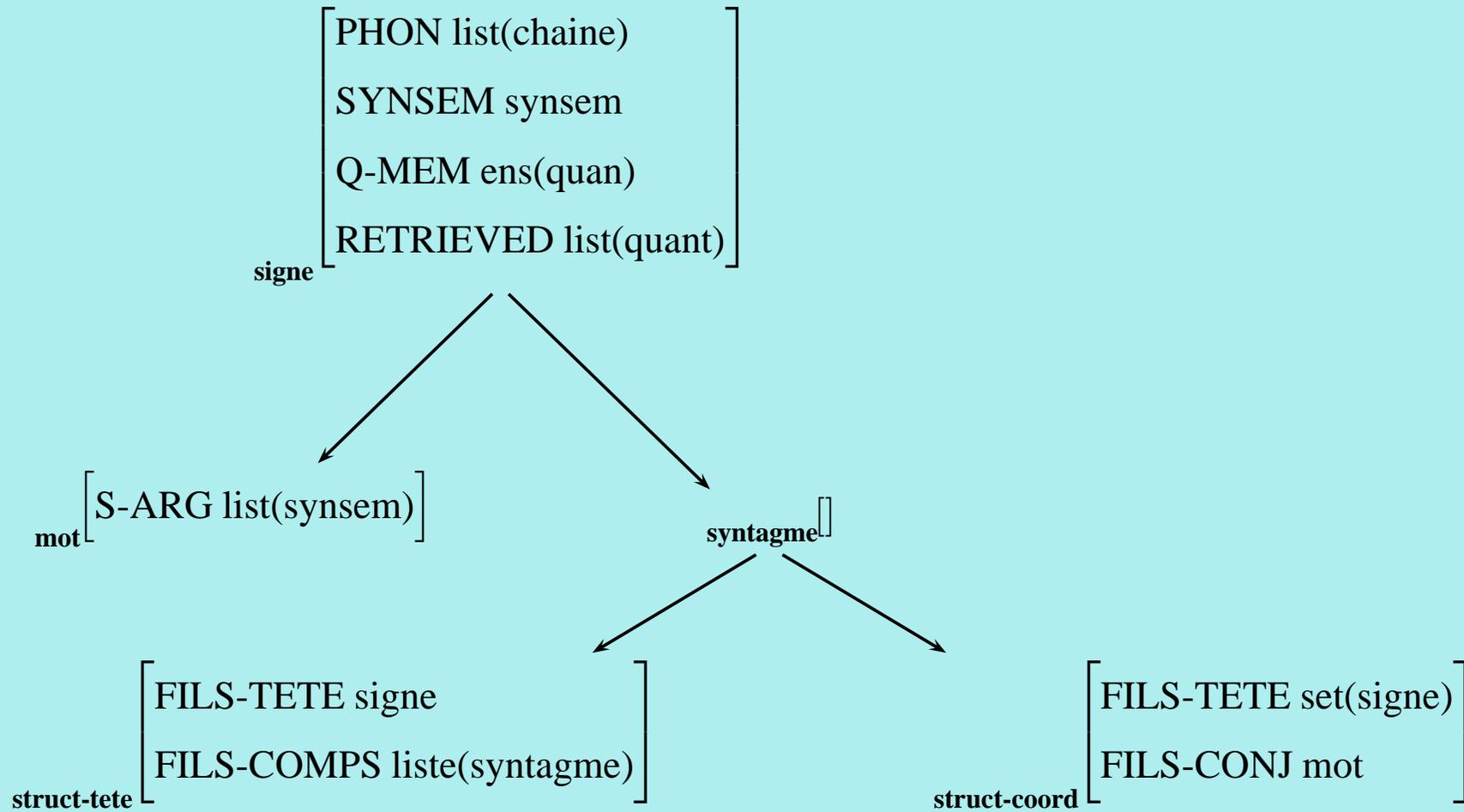


- Théorie en évolution (HPSG III)
- S'appuie sur un formalisme puissant et homogène : structures de traits typées [TFS]

**Bob Carpenter (1992)** *The logic of typed feature structures*

Structures de traits possédant un type défini dans une hiérarchie

# Hiérarchie de type



# Hiérarchie de type

Notation à la Prolog, utilisée dans ALE et DyALog.

bot **sub** [ string , list , cat , synsem].

string **escape** symbol.

cat **sub** [ s , np , vp , det , n ].

s **sub** []. np **sub** []. vp **sub** [].

det **sub** []. n **sub** [].

synsem **sub** [ phrase , lexeme ] **intro** [ cat : cat ] .

phrase **sub** [ root ] **intro** [ args : list ] .

root **sub** [] **intro** [ cat : s ] .

lexeme **sub** [] **intro** [ orth : string ] .

list **sub** [ ne\_list , e\_list ] .

ne\_list **sub** [] **intro** [ hd : bot , tl : list ] .

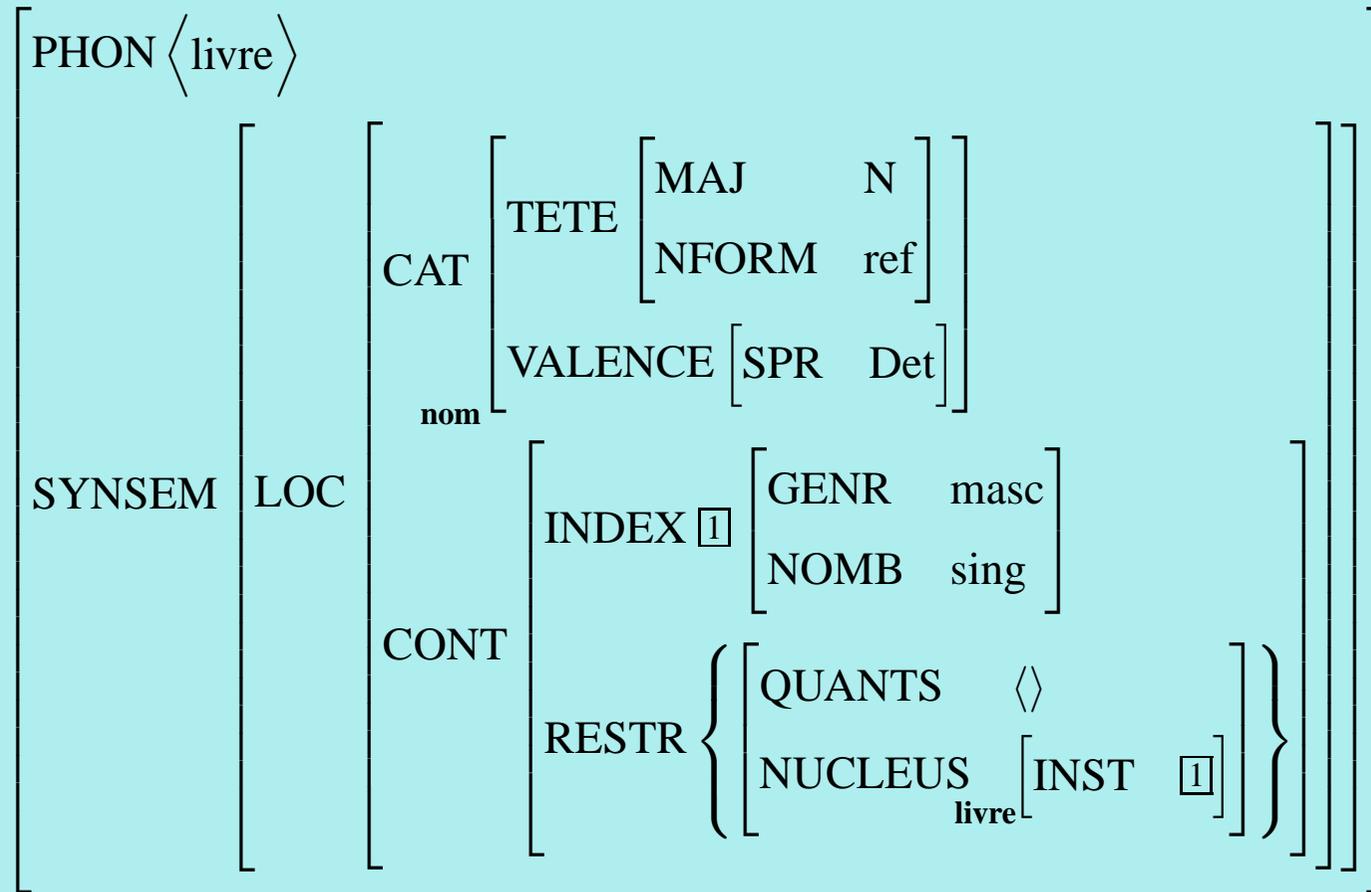
e\_list **sub** [] .

- Un type peut introduire des sous-types
- Un type peut introduire ou préciser un trait et le type de sa valeur
- héritage multiple possible
- les types introduisant un trait  $f$  ont un type plus général

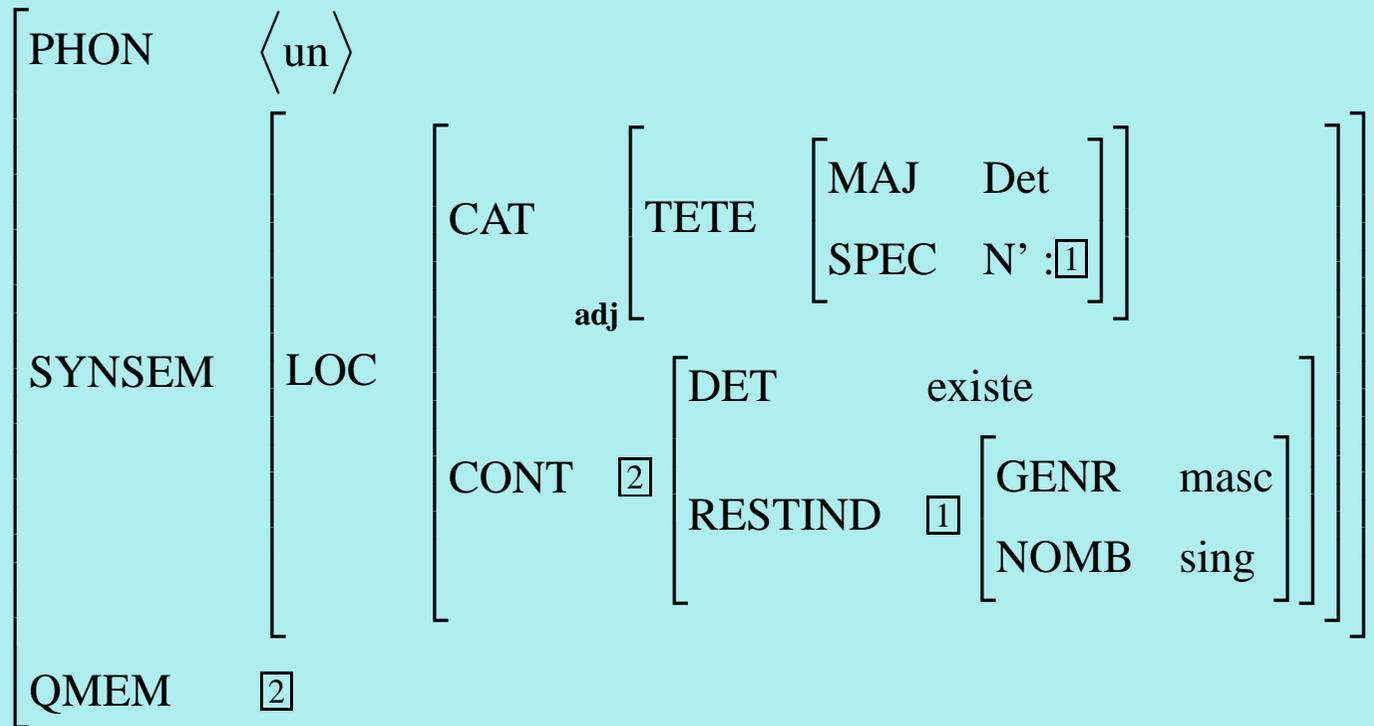
# Architecture HPSG

- **entrées lexicales**  
exploitant la notion de sous-catégorisation
- **règles lexicales** de dérivation de nouvelles entrées (ex. le passif)
- des **schémas** de construction des constituants
- des **principes** de bonne construction

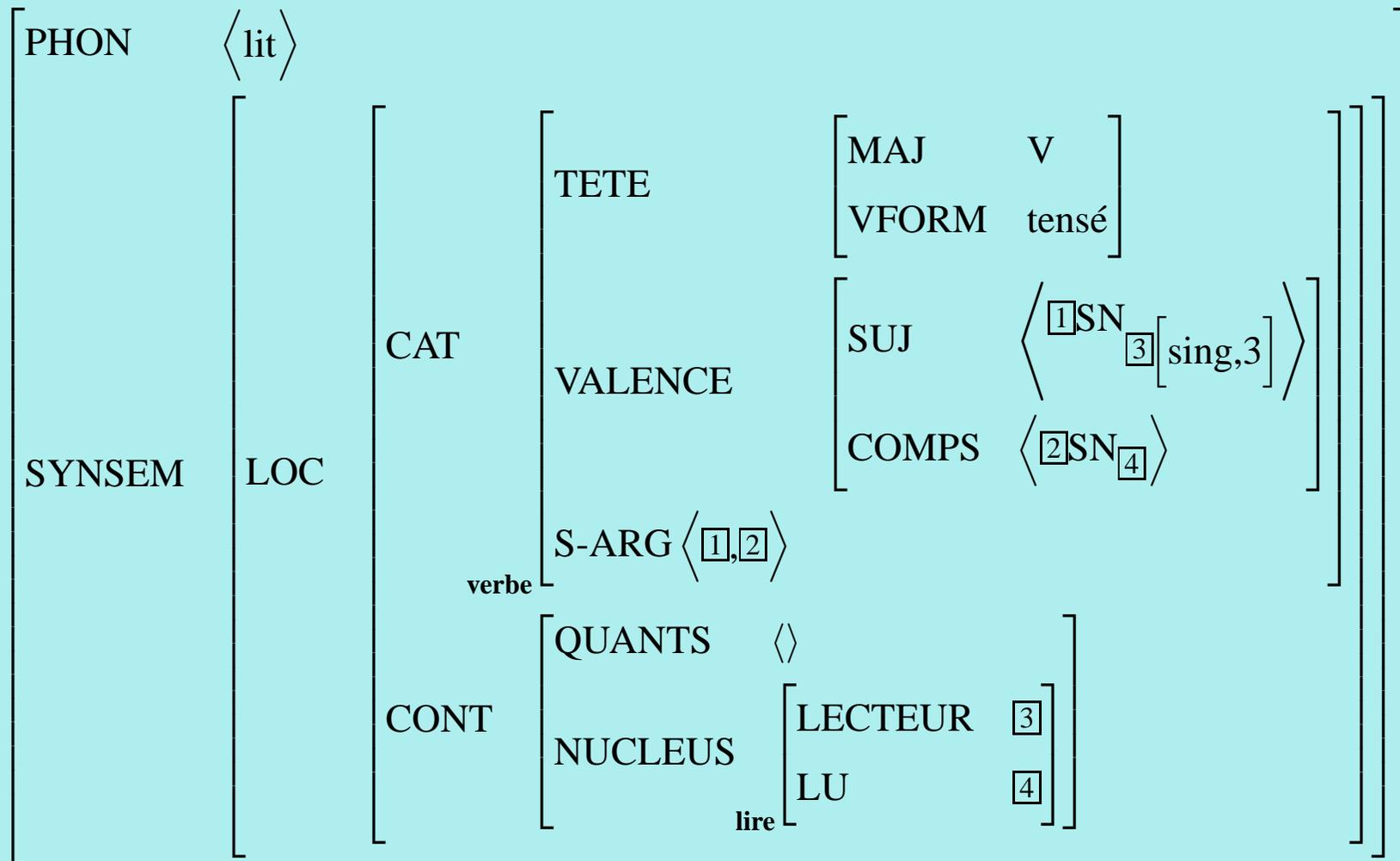
# Nom



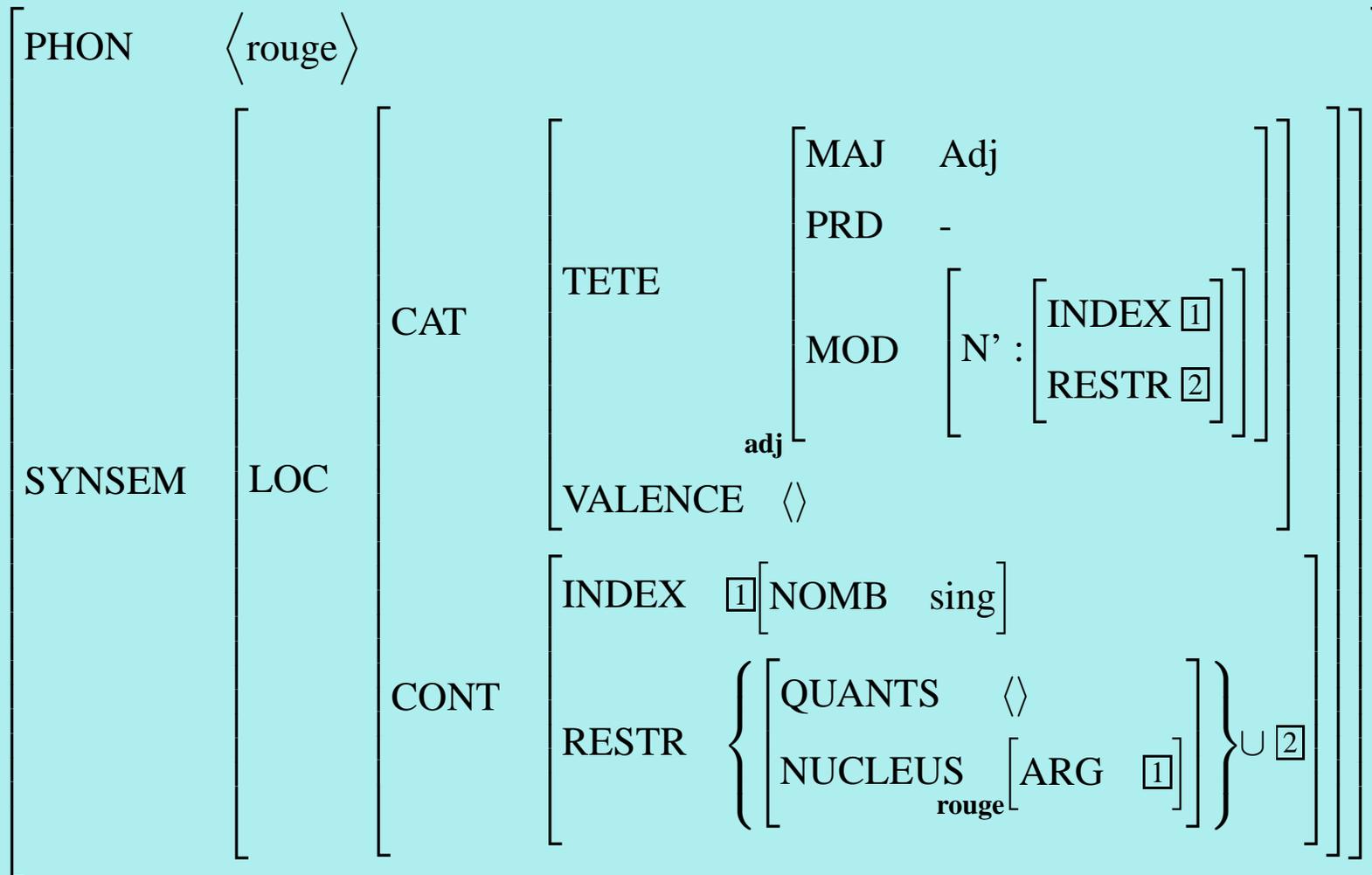
# Article



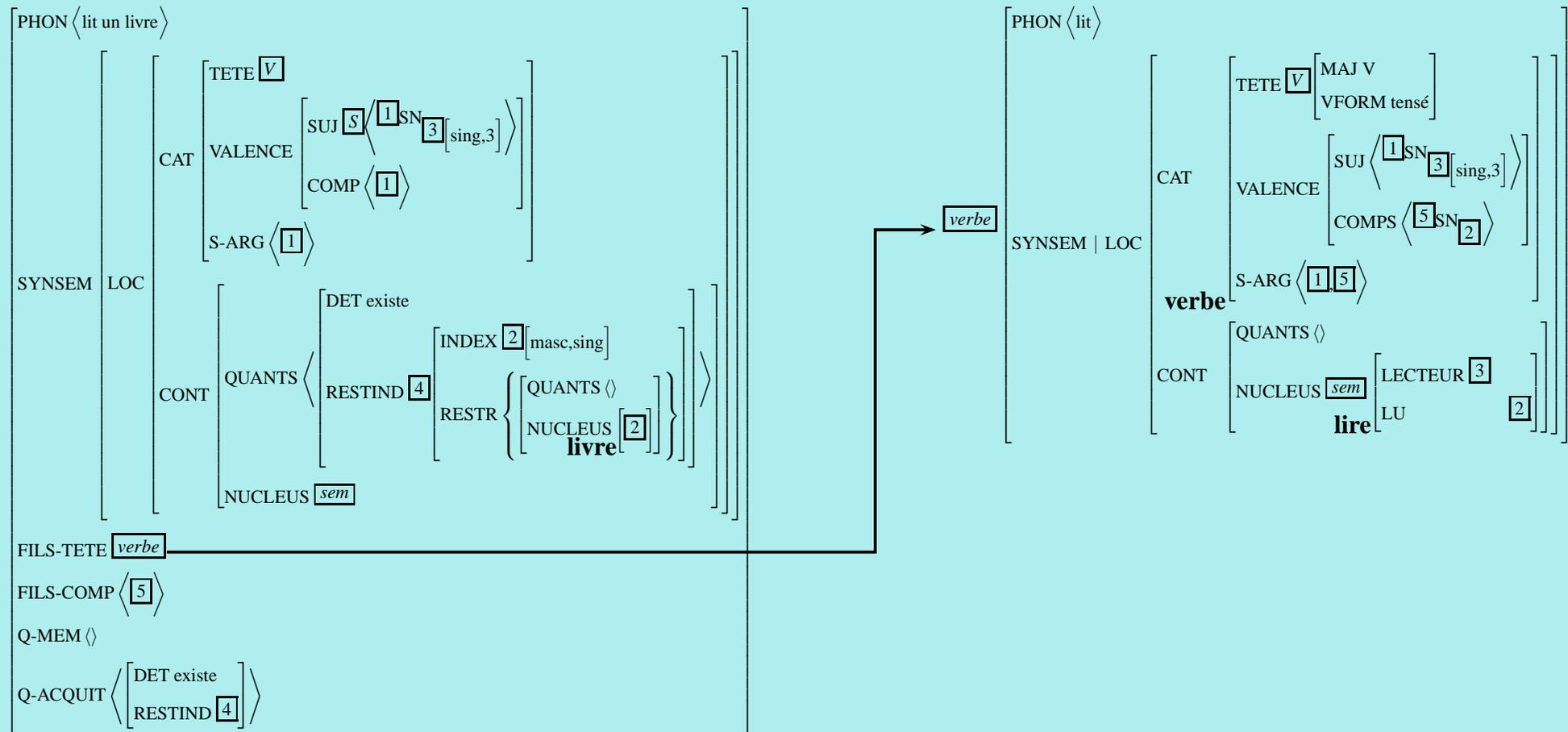
# Verbe



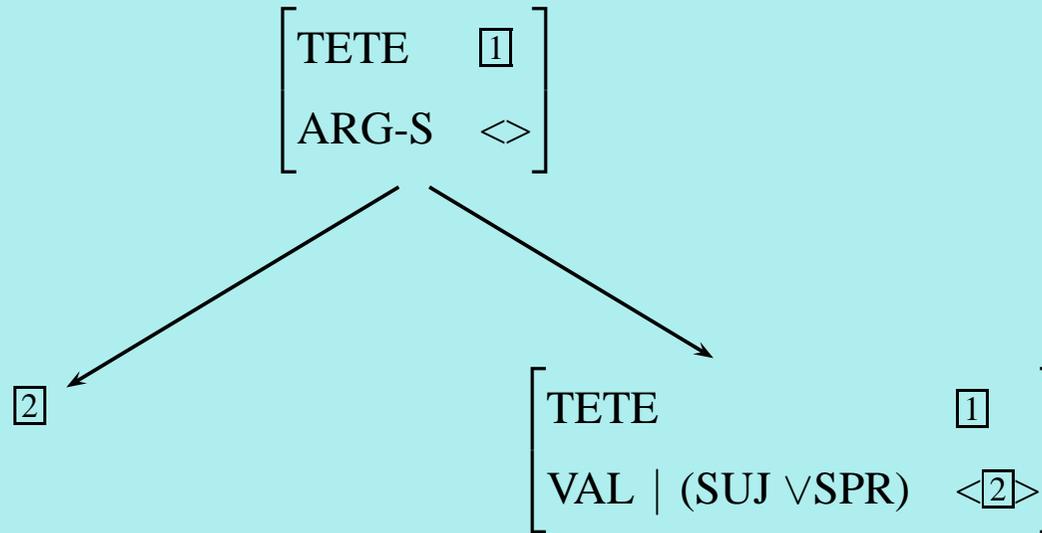
# Adjectif



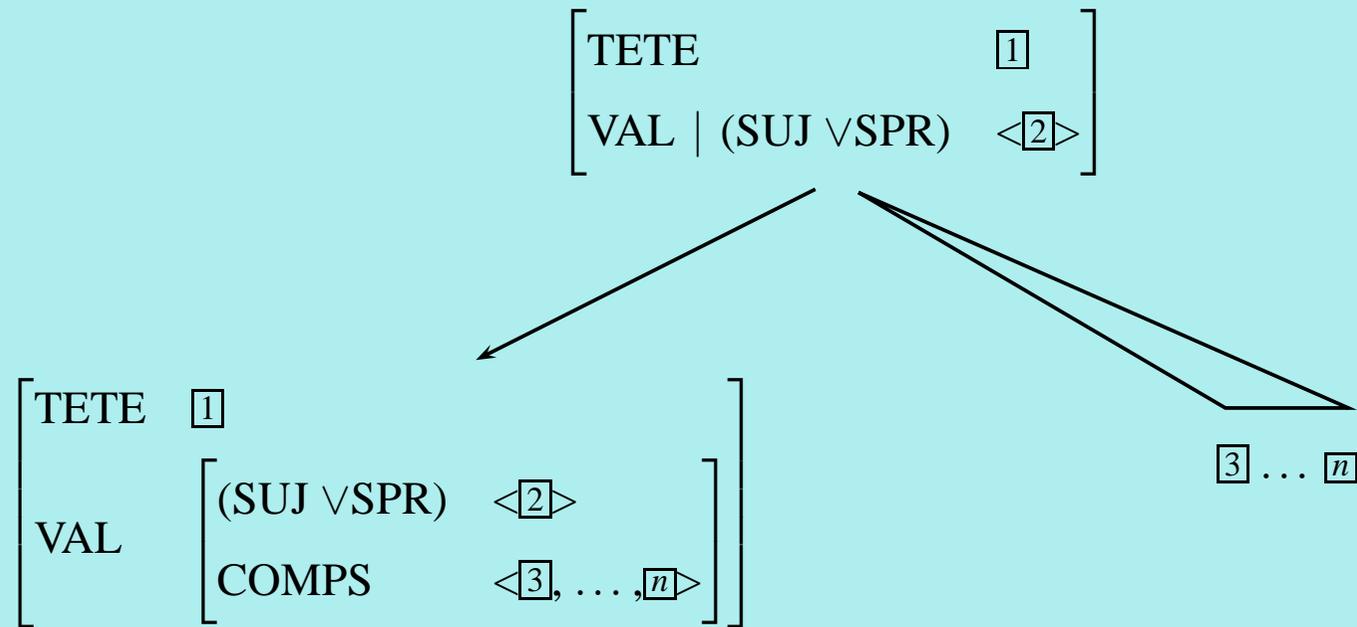
# Fragment de phrase (Syntagme Verbal)



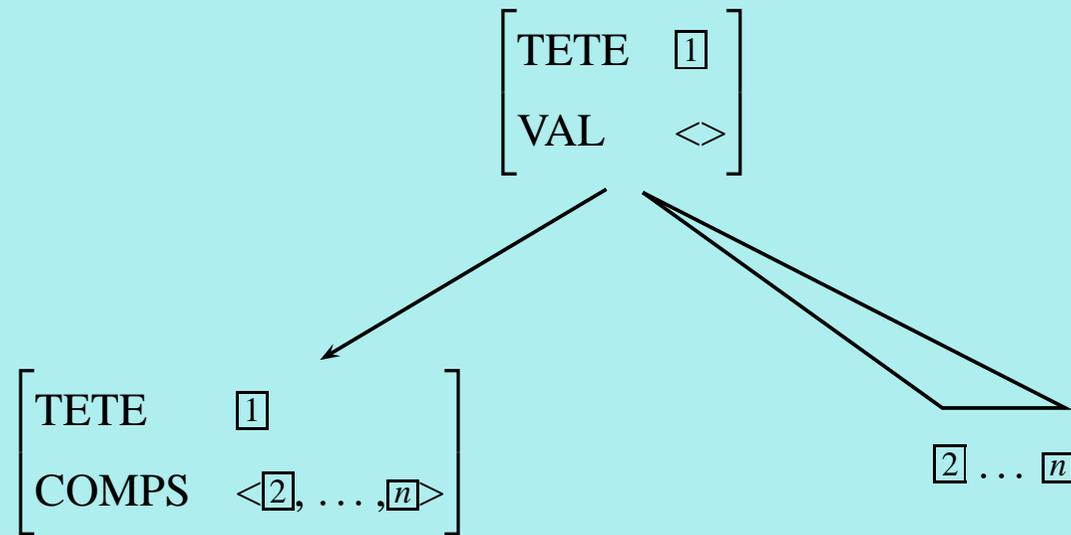
# Schéma 1 : Constituants saturés



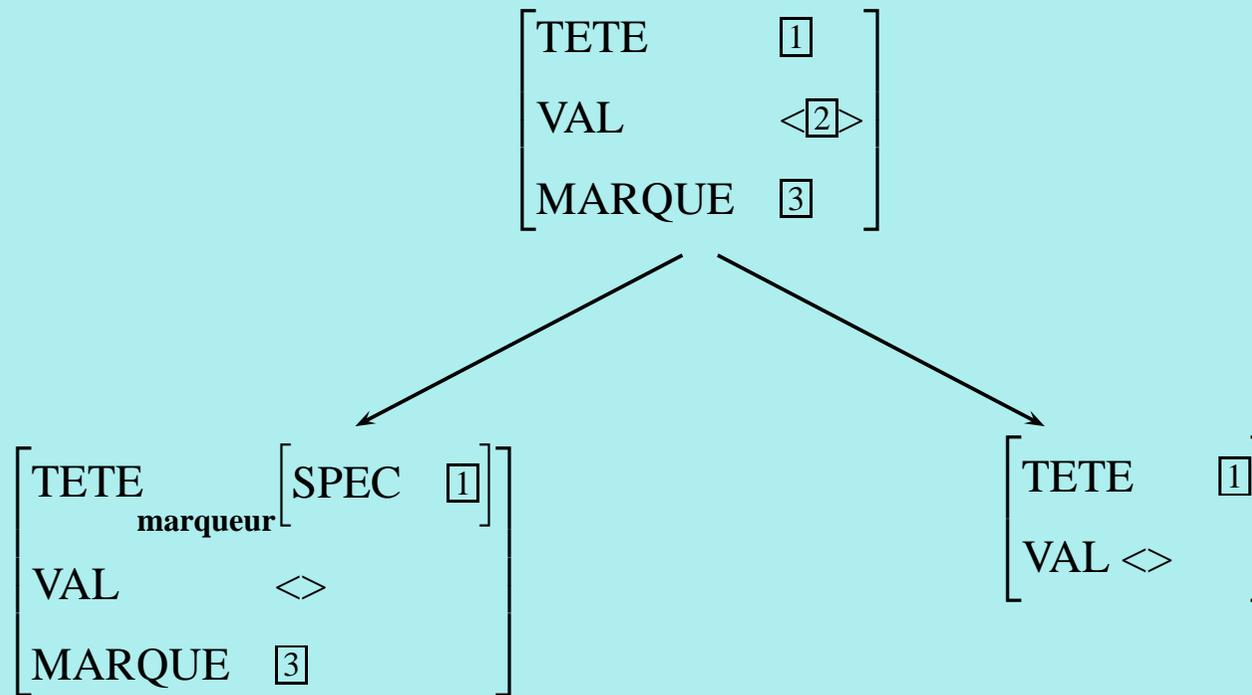
## Schéma 2 : Constituants non spécifiés



# Schéma 3 : Constituants avec compléments catégorisés

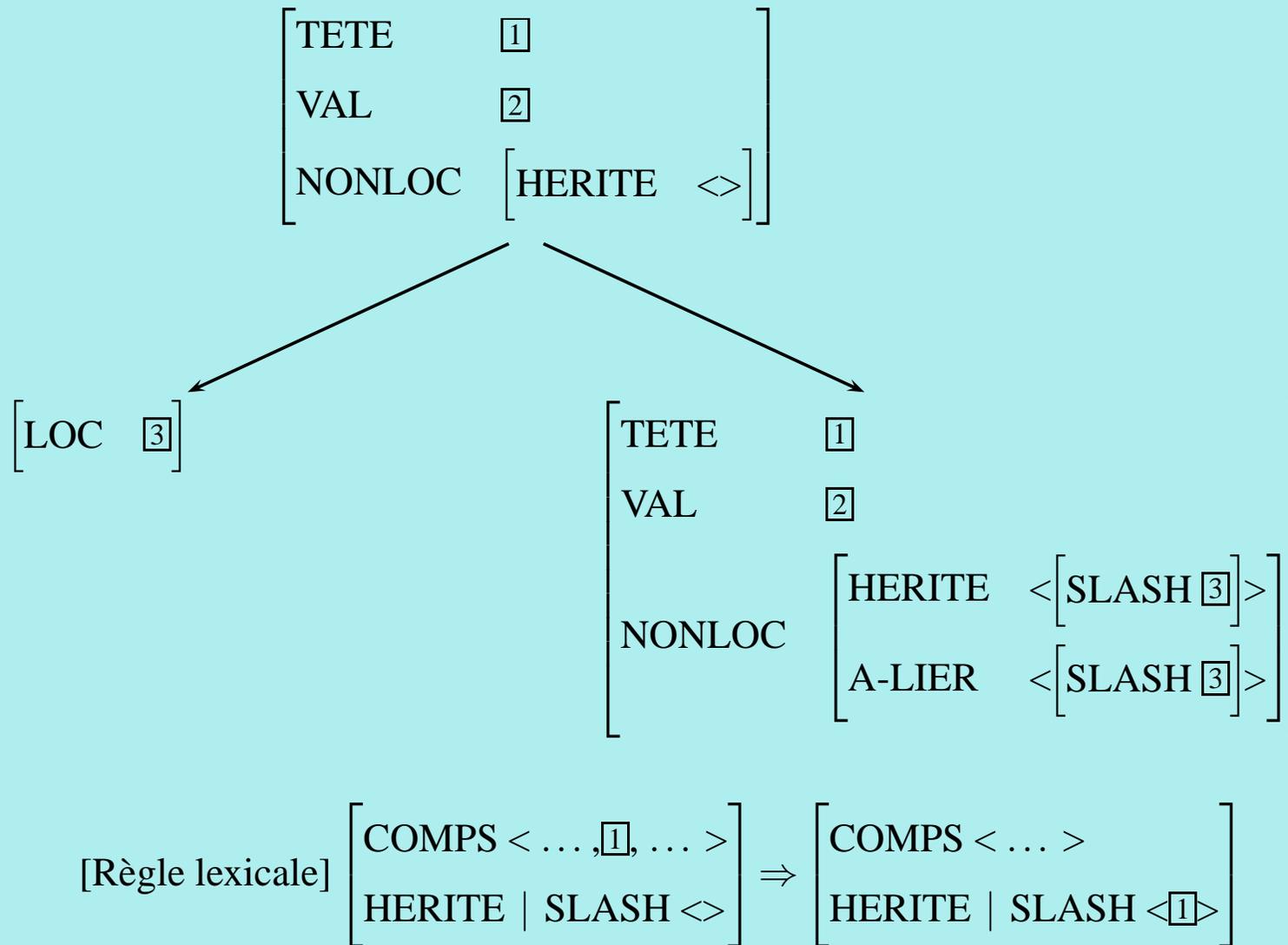


## Schéma 4 : Constituants marqués



Permet le traitement des prépositions et de phrases telles que « [que Jean parte] m'inquiète ».

## Schéma 5 : Extraction



# Les principes

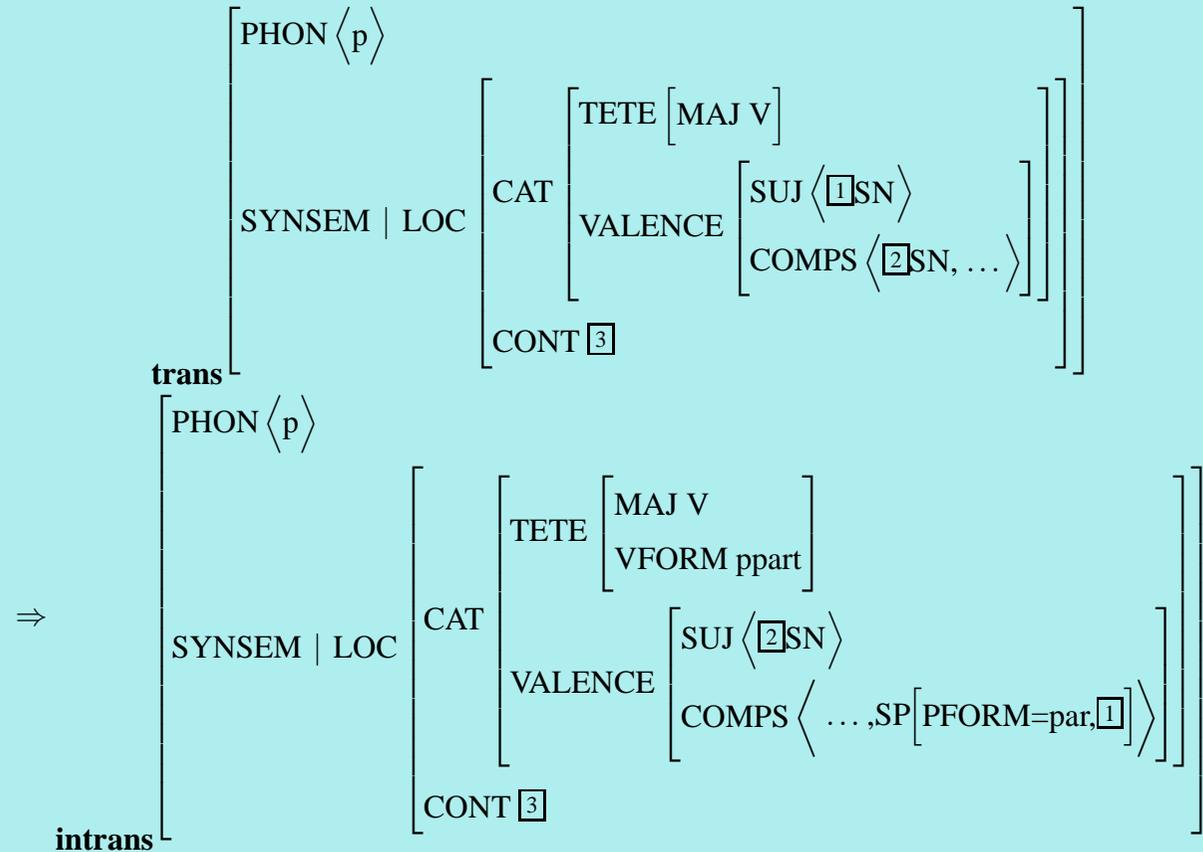
**Principe de tête** Les valeurs des traits de tête entre un syntagme et sa tête sont identiques

**Principe de valence** La valence d'un syntagme est la valence de sa tête moins la liste des fils (non tête).

**Principes des traits non locaux** La valeur des traits non locaux hérités est l'union des valeurs des traits hérités des fils, moins la valeur à lier du syntagme racine.

# Règle lexicale : le passif

Construction d'une entrée lexicale pour les verbes transitifs : manger ⇒ mangé



# Syntaxe ALE : entrée lexicale

```
persuade ---> % object equi
word,
synsem:loc:( cat :( head:( verb ,
                mod:none, vform:bse, aux:minus, inv :minus),
                subcat :[( @ np(Ind1)),
                          (@ np(Ind2), @ case(acc )),
                          (@ vp(VCont),
                           loc : cat :( head:vform:inf ,
                                         subcat :[( @ np(Ind2 ))])]),
                marking:unmarked),
cont:( nucleus:( persuading , persuader:Ind1 , persuaded:Ind2 , soa_arg:VCont),
        quants :[]),
conx:backgr: e_set ),
(@ empty_non_loc),
qstore : e_set .
```

## Syntaxe ALE : schéma

```
schema1 rule (Mother,phrase,synsem:loc:cat:subcat:[]) ==>
cat > (SubjDtr,non_word,synsem:SubjSyn),
cat > (HeadDtr,phrase),
goal > ( head_feature_principle (Mother,HeadDtr),
  inv_minus_principle (Mother),
  subcat_principle (Mother,HeadDtr,[SubjSyn]),
  marking_principle (Mother,HeadDtr),
  spec_principle (SubjDtr,HeadDtr),
  semantics_principle (Mother,HeadDtr,[SubjDtr ]),
  parochial_trace_principle (SubjDtr),
  nonlocal_feature_principle (Mother,HeadDtr,[SubjDtr ]),
  single_rel_constraint (Mother),
  clausal_rel_prohibition (Mother),
  relative_uniqueness_principle (Mother,[SubjDtr,HeadDtr]),
  conx_consistency_principle (Mother,[SubjDtr,HeadDtr]),
  deictic_cindices_principle (Mother,[SubjDtr,HeadDtr])).
```

# Grammaires Catégorielles [CG]

Calcul de **Lambek** (1958)

**Steedman** (1998) Combinatory Categorical Grammars

- Théorie plus linguistique, non centrée sur une notion de constituant  $\Rightarrow$  traitement de la coordination et constituants incomplet
- Formalisme :  $\lambda$ -calcul (et combinateurs)

# Applications

$$\frac{X/Y : f \quad Y : a}{X : fa} > \qquad \frac{Y : a \quad X/Y : f}{X : fa} <$$

$$\frac{\text{Pierre} \quad \text{mange} \quad \text{une pomme}}{\text{SN : pierre}' \quad (P \setminus \text{SN}) / \text{SN} : \lambda x. \lambda y. \text{manger}' xy \quad \text{SN : pomme}' >} \\ \frac{(P \setminus \text{SN}) : \lambda y. \text{manger}' pomme' y}{P : \text{manger}' pomme' pierre' <}$$

- Puissance générative des CFG  $\Rightarrow$  nécessaire d'ajouter des combinateurs supplémentaires
- Possibilité de décorer les non-terminaux avec des structures de traits

## Coordination (simple)

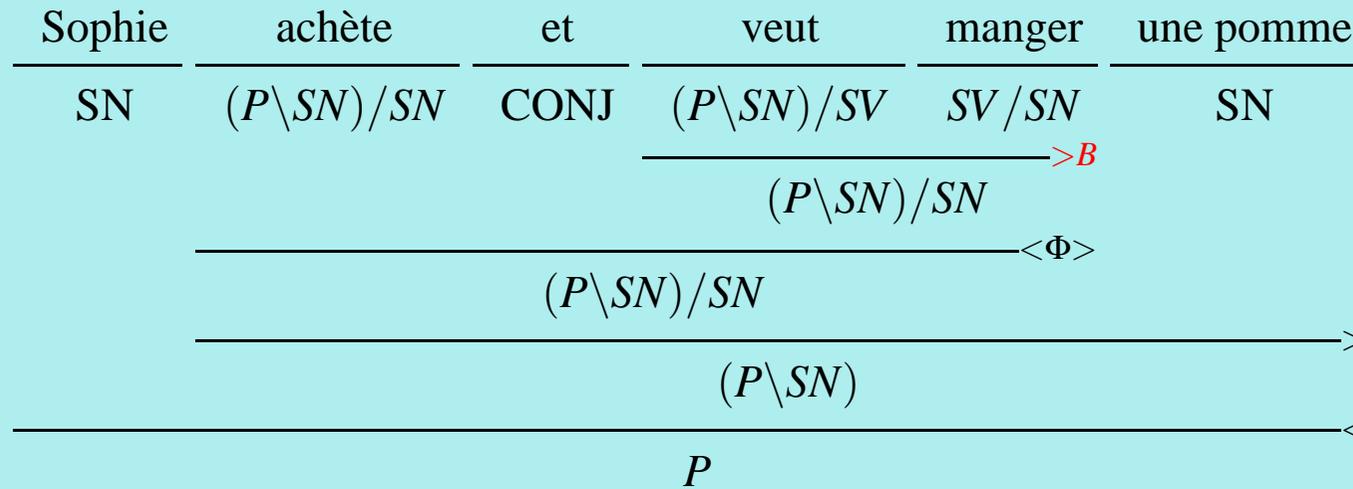
$$\frac{X \quad \text{CONJ} \quad X}{X} \langle \Phi \rangle$$

Sophie	achete	et	mange	une pomme
SN :s'	$(P \setminus SN) / SN : \lambda x. \lambda y. \text{acheter}'xy$	CONJ :et'	$(P \setminus SN) / SN : \lambda x. \lambda y. \text{manger}'xy$	SN :p'
$\langle \Phi \rangle$				
$(P \setminus SN) / SN : \lambda x. \lambda y. \text{et}'(\text{acheter}'xy)(\text{manger}'xy)$				
$(P \setminus SN) : \lambda y. \text{et}'(\text{acheter}'p'y)(\text{manger}'p'y)$				
$P : \text{et}'(\text{acheter}'p's')(\text{manger}'p's')$				

Mais ne permet pas de traiter toutes les coordinations

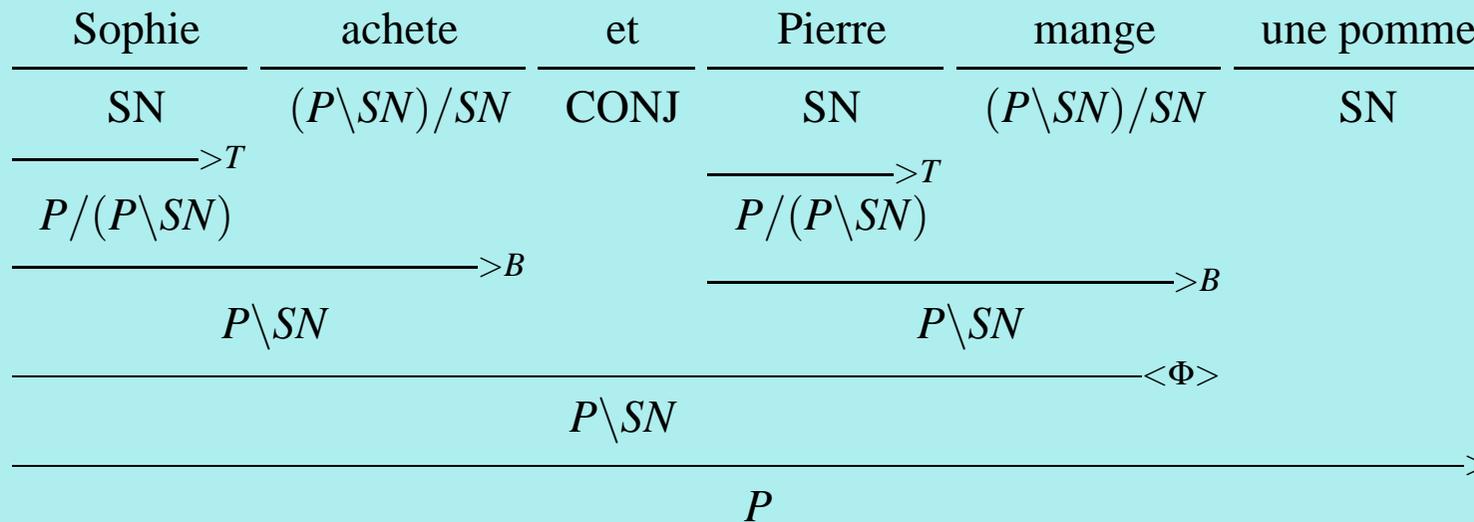
# Composition

$$\frac{X/Y : f \quad Y/Z : g}{X/Z : \lambda x.f(gx)} > B$$



# Montée de type (type-raising)

$$\frac{SN : a}{T / (T \setminus SN) : \lambda f.f a} > T$$



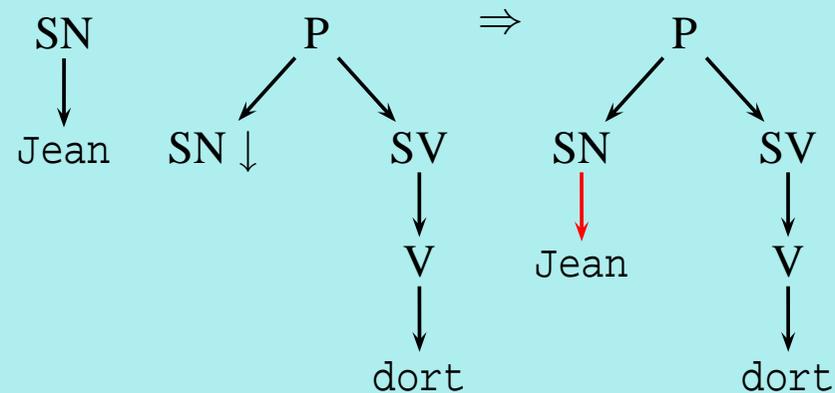
Restriction des montées de type.

# Grammaires d'arbres adjoints [TAG]

Joshi (1975) *Tree Adjunct Grammars*

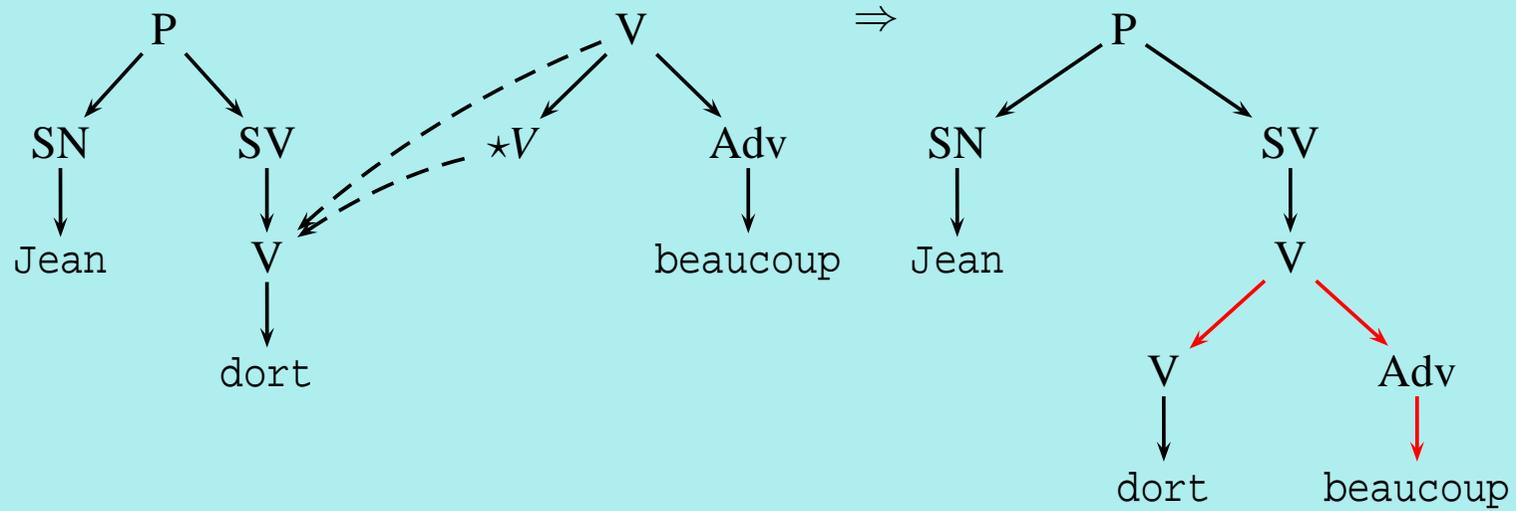
**Théorie** lexicale, catégorisation, tête, constituants

**Formalisme** manipulation d'arbres, avec les opérations de **substitution** et d'**adjonction**



# Adjonction

Permet de gérer les ajouts



# Principes de bonne formation

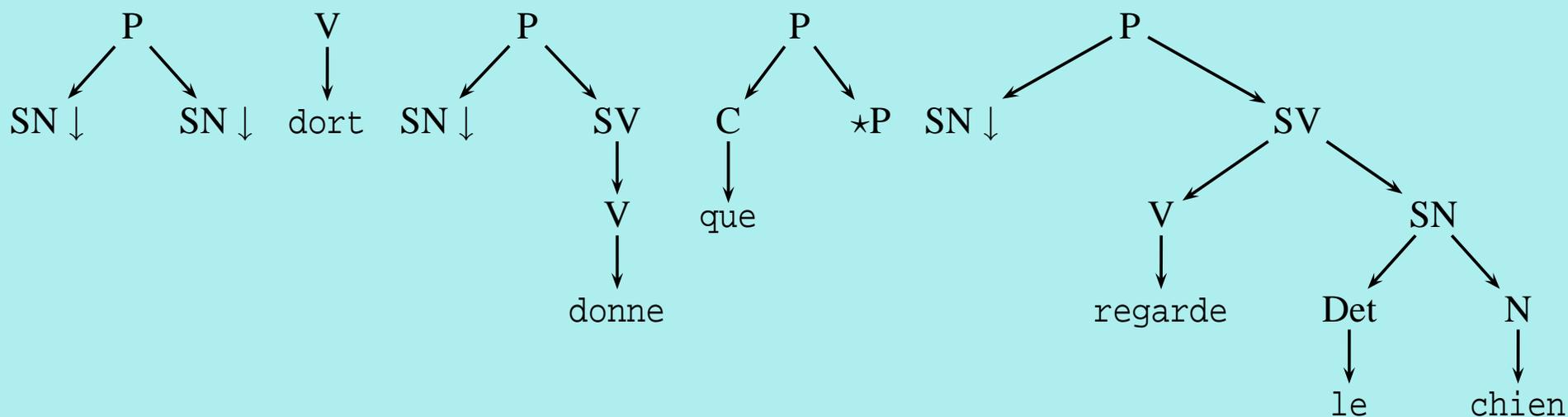
Pour tout arbre élémentaire :

**ancrage lexical** au moins une tête lexicale (non vide)

**catégorisation** un noeud pour chaque argument catégorisé par la tête (domaine de localité)

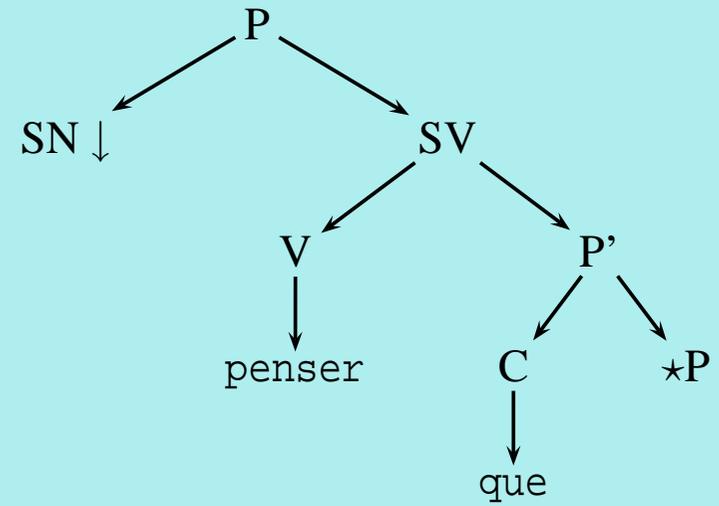
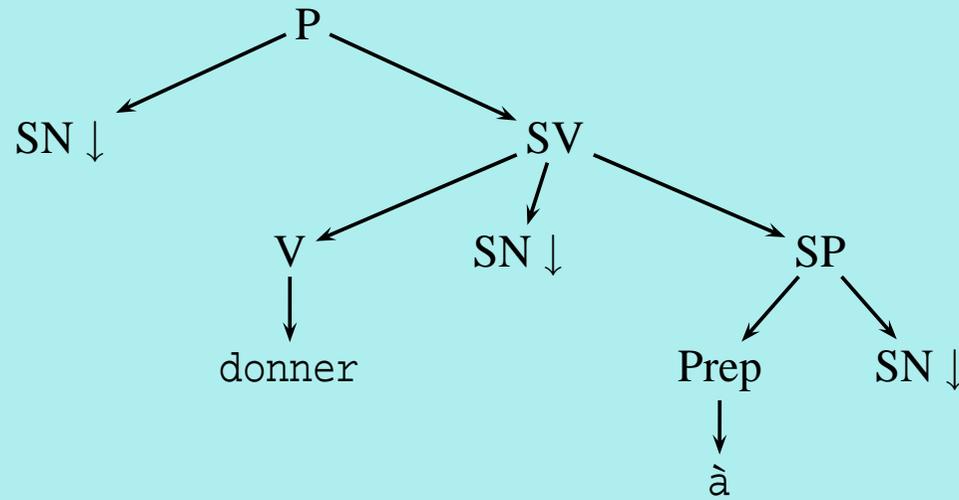
**Consistence sémantique** une association sémantique

**non-composition** une seule unité sémantique



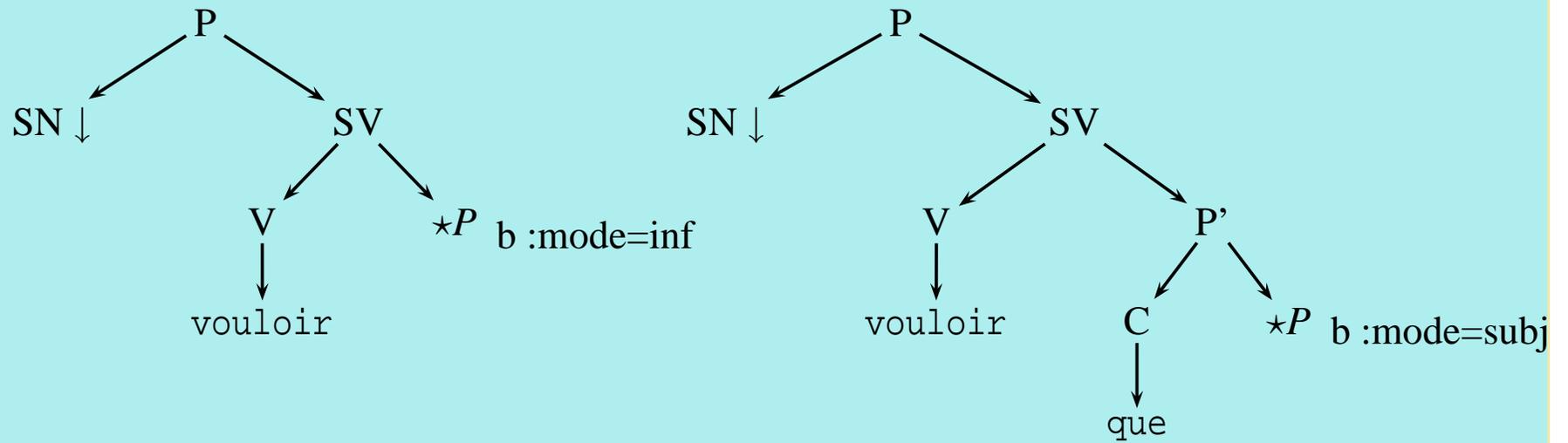
# Catégorisation

Immédiate dans la structure des arbres



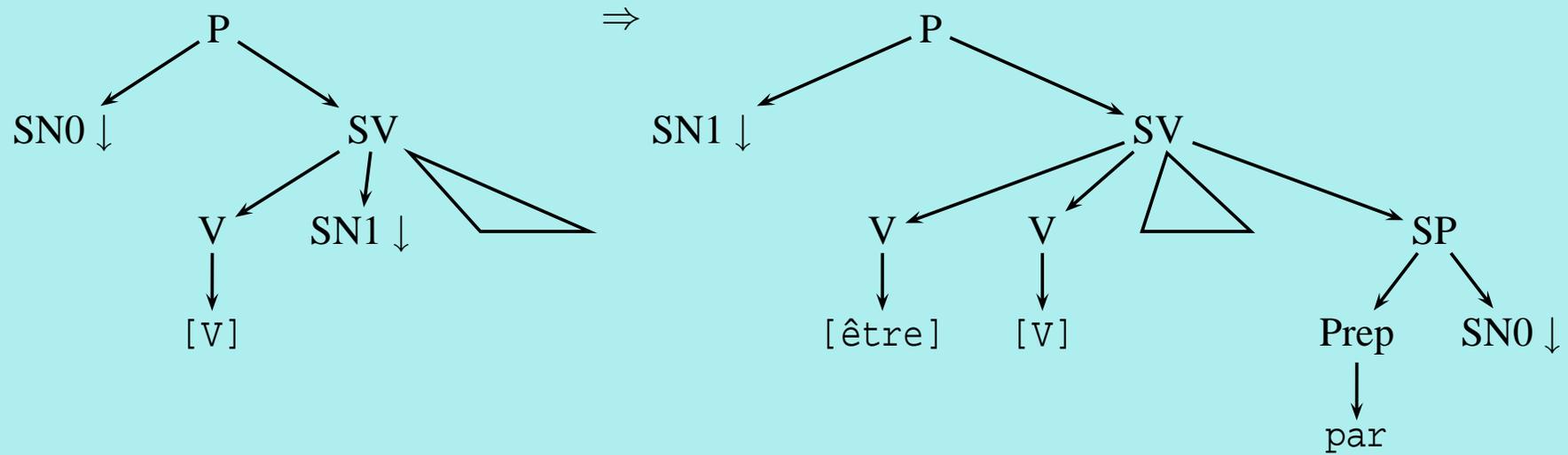
# TAG avec traits

Structures de traits top et bot associables aux noeuds :



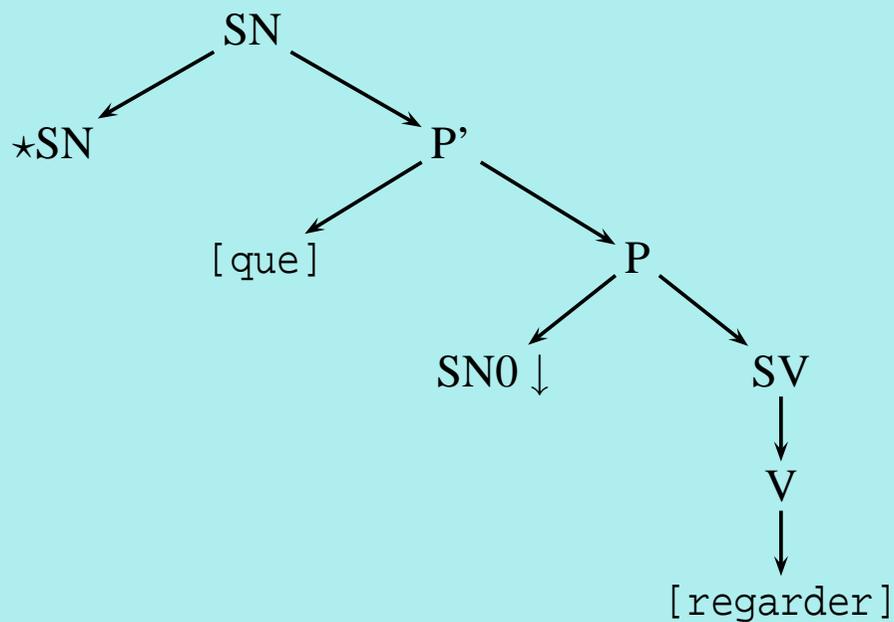
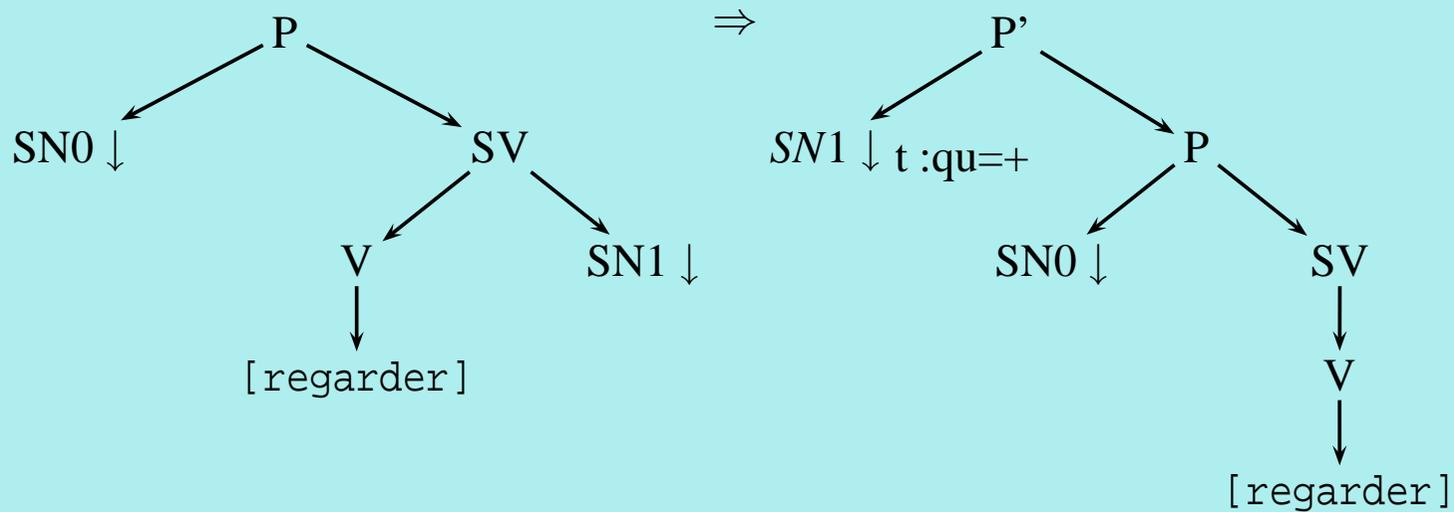
# Familles

Transformations lexicales pour gérer le passif :



L'ensemble des arbres obtenus est une **famille**

# Extraction



# Dépendances non bornées

Une succession d'adjonctions peut séparer un constituant extrait de son prédicat tête.

# Arbres de dérivation et sémantique

# Formalismes TAG dérivés

**TIG** Adjonction réduite (non enveloppante)

**Multi-Component Tag** insertion par substitution et/ou adjonction d'un ensemble d'arbres élémentaires dans un arbre.

**Dominance-Tree** Quasi arbres avec relations parents et ancêtres

Formalismes faiblement dépendant du contexte (MCS)

⇒ complexité polynomiale.

# Grammaires de dépendance [DG]

Théorie :

- Remise en cause de la notion de constituant et d'arborescence des constituants
- Relations (**dépendances**) entre mots ( $\Rightarrow$  graphe de dépendance)
- Contraintes sur les dépendances

# Bibliographie

**Les nouvelles syntaxes – Grammaires d'unification et analyse du français** Anne Abeillé (1993).

**Une introduction à HPSG** Philippe Blache

<http://www.lpl.univ-aix.fr/~blache/papers/Intro-hpsg.ps.gz>

**The combinatory manifesto** Mark Steedman (Draft, novembre 2000) et transparents associés.

<http://www.cogsci.ed.ac.uk/~steedman/>

**Ressources HPSG** <http://lingo.stanford.edu/>

**Categorial Grammars Home Pages** <http://www.cs.man.ac.uk/ai/CG/>

**XTAG Group** <http://www.cis.upenn.edu/~xtag/>

**Dependency Grammars Home Page** <http://ufal.mff.cuni.cz/dg/dgmain.html>

**Les grammaires de dépendance** Revue T.A.L., volume 41 (2000), Sylvain Kahane éditeur.

<http://www.atala.org/tal/tal.html>