

Analyse syntaxique  
et  
application aux langues naturelles

*Grammaires Lexicales Fonctionnelles (LFG)*

Jacques Farré

et

Sylvain Schmitz

# Insuffisance des grammaires non contextuelles

---

- Le problème de l'analyse syntaxique est le suivant
  - Les différents composants d'une phrase doivent s'accorder de manière cohérente
  - Il faut établir une correspondance entre les relations entre ces composants et *l'organisation de surface* de ces composants  
*Paul m'a donné ce livre / Ce livre m'a été donné par Paul*
- En ce sens, les grammaires non contextuelles (CFG) sont insuffisantes pour modéliser les langues naturelles
  - elles expriment seulement l'organisation de surface
- Il faut un modèle avec 2 niveaux de description : c'est ce que font les grammaires lexicales fonctionnelles (LFG)

# Structure de composants et structure fonctionnelle

---

- La structure de composants (*c-structure*) est définie en terme de mots (de forme fléchie de lemme), de catégories lexicales (*verbe, nom,...*) et de leurs relations syntaxiques
  - Ces relations syntaxiques se représentent à l'aide d'un arbre
  - Les CFG (mais aussi d'autres formalismes grammaticaux) sont un bon outil pour décrire ce niveau
- La structure fonctionnelle (*f-structure*, on parle aussi de *traits*) permet de représenter des notions comme *sujet, objet, temps, genre,...* et leurs relations grammaticales
  - Une f-structure est un ensemble ordonné de couples (*attribut, valeur*)

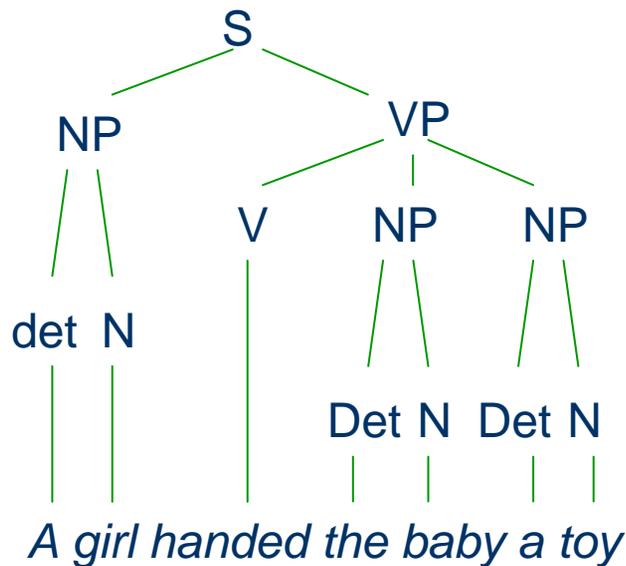
# Structure fonctionnelle

---

- Un attribut est le nom d'un concept grammatical : par ex. SUBJ (sujet), OBJ (objet), TENSE (temps), NUM (singulier/pluriel), SPEC (spécificateur/article),...
- Les valeurs des attributs sont soit
  - des symboles : par ex. PAST (pour le temps), SG (pour singulier)
  - des formes sémantiques : par ex. une entrée dans le lexique, un prédicat et ses arguments
  - des f-structures secondaires (les f-structures peuvent donc aussi être vues comme des arbres –ou plutôt des graphes orientés)
  - des ensembles d'un des 3 types précédents

# Exemple de c-structure et de f-structure

- La grammaire :  $S \rightarrow NP VP$ ,  $NP \rightarrow det N$ ,  $VP \rightarrow V NP NP$  permet de produire la phrase *A girl handed the baby a toy*



c-structure



f-structure

# Formalisme de description

---

- La grammaire est enrichie pour décrire le calcul des f-structures : ce sont des *équations* qui décrivent les valeurs obtenues par l'application de fonctions
- Ces équations s'expriment en termes de f-structure des composants de la c-structure, notés  $\uparrow$  et  $\downarrow$
- Les équations de la grammaire permettent « d'instancier » les f-structures associées aux c-structures :
- Par exemple :  $s \rightarrow \begin{array}{cc} NP & VP \\ (\uparrow \text{SUBJ}) = \downarrow & \uparrow = \downarrow \end{array}$

le sujet de S est égal à la f-structure de NP, et la f-structure de VP est « copiée » directement dans la f-structure de S

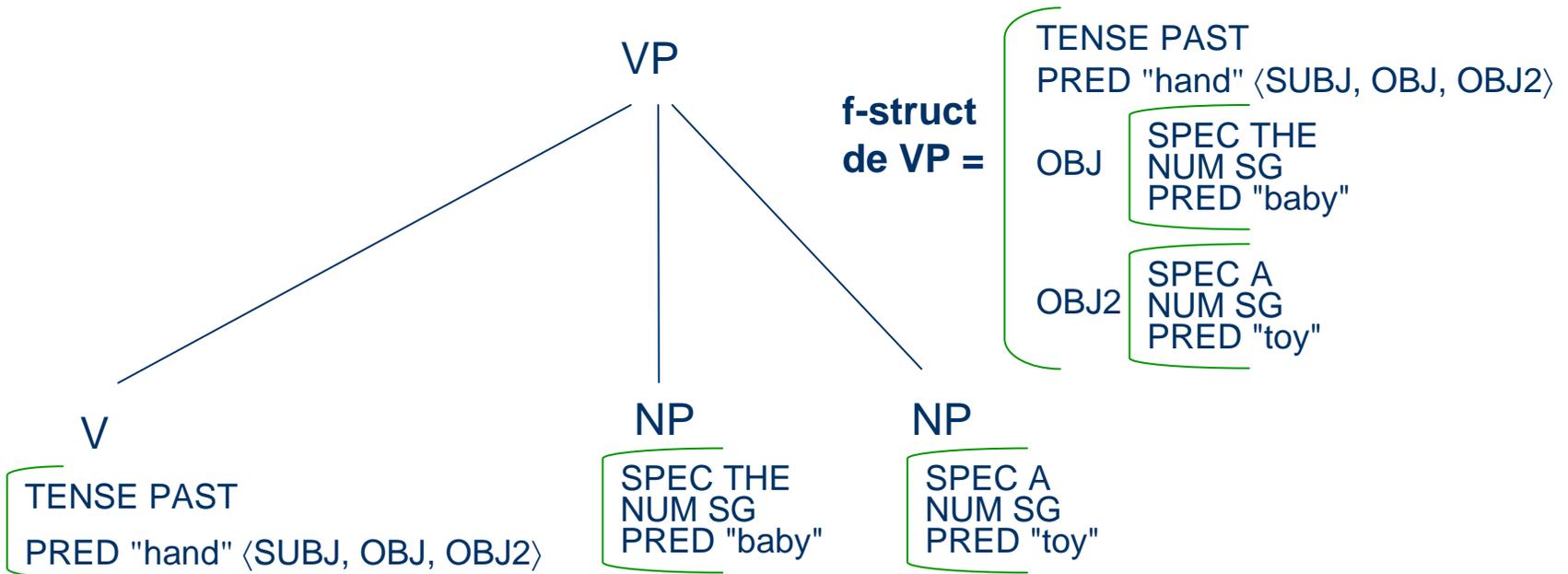
# Mécanisme de « transmission » des f-structures (1)

- Reprenons la grammaire, maintenant enrichie d'équations

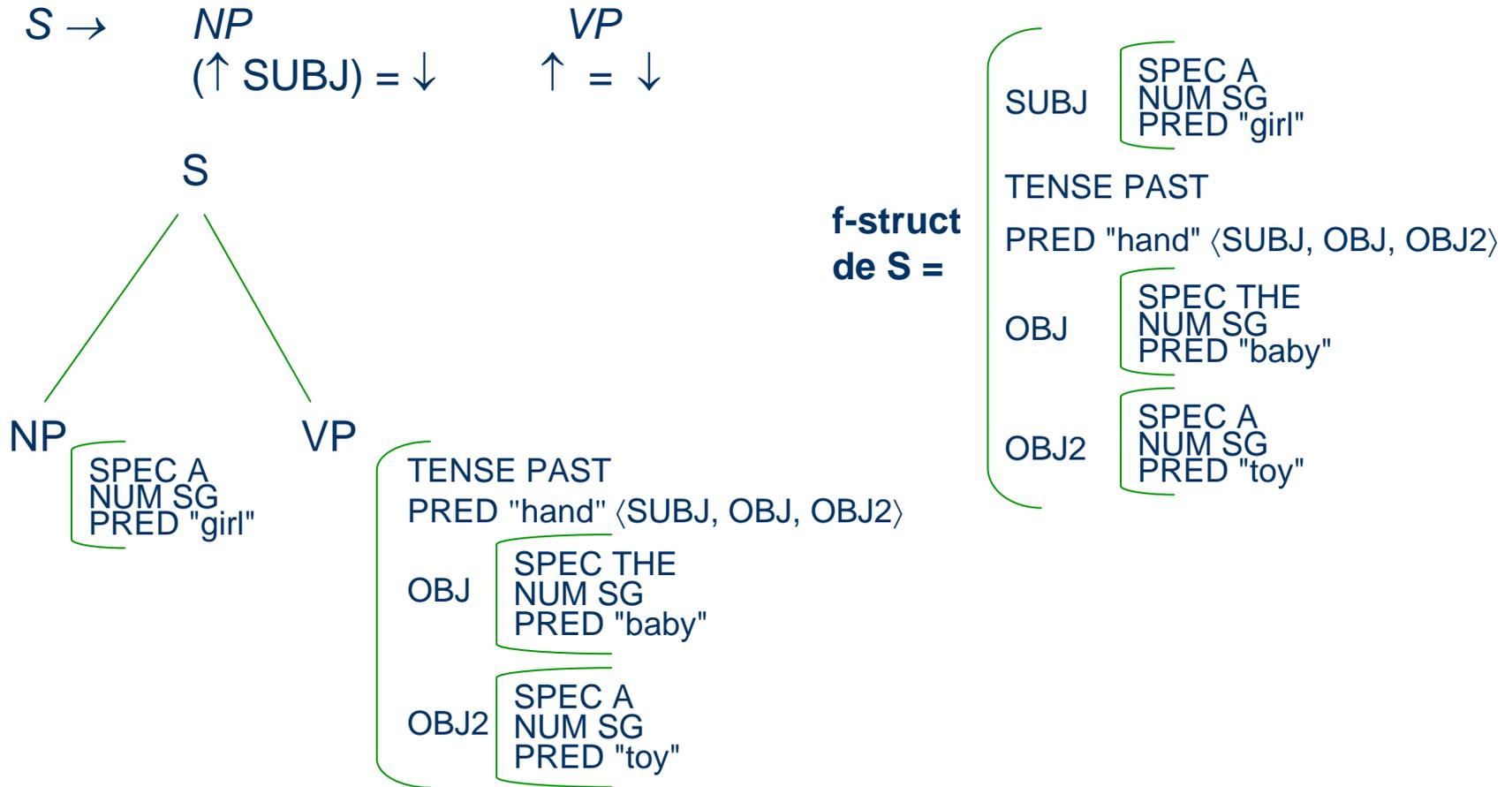
$VP \rightarrow V$   
 $\uparrow = \downarrow$

$NP$   
 $(\uparrow OBJ) = \downarrow$

$NP$   
 $(\uparrow OBJ2) = \downarrow$



# Mécanisme de « transmission » des f-structures (2)



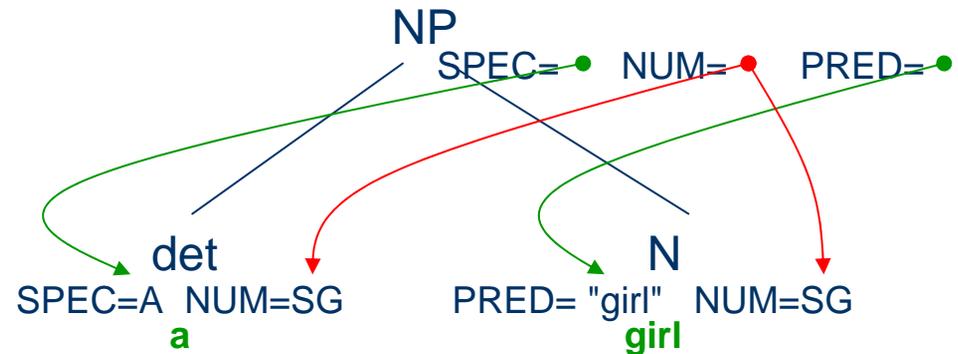
# f-structures et lexique

---

- Le lexique contient des équations de f-structure (ou des informations les simulant) :

a	det	(↑SPEC)=A	(↑NUM)=SG
girl	N	(↑PRED)="girl"	(↑NUM)=SG

- Avec la règle

$$NP \rightarrow \begin{matrix} det & N \\ \uparrow = \downarrow & \uparrow = \downarrow \end{matrix}$$


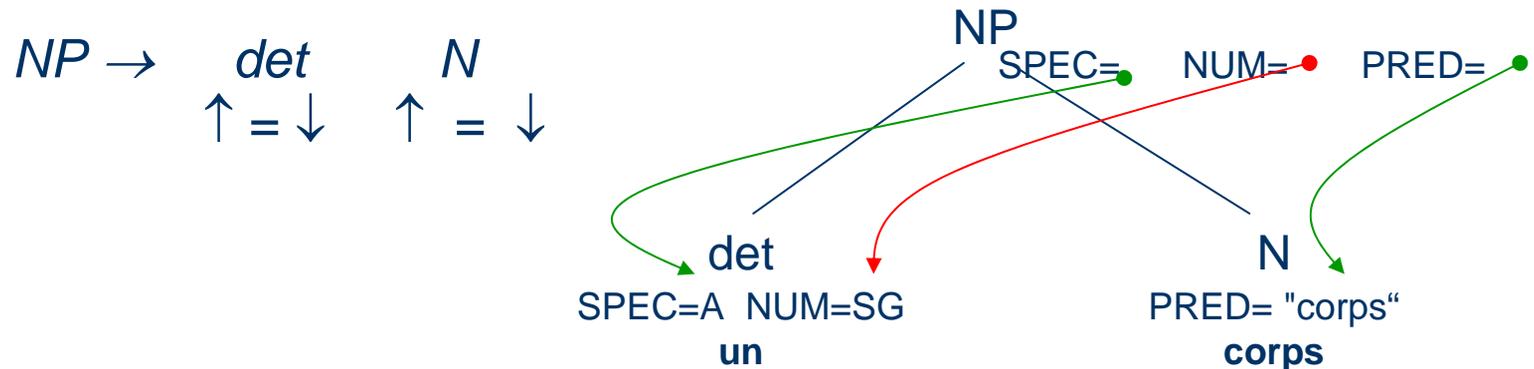
- On peut voir des analogies avec les grammaires attribuées : en yacc  
 $NP : det N \{ \$.SPEC = \$1.SPEC; \$.PRED = \$2.PRED; \$.NUM = ???; \}$

# Règles d'unification

---

- L'unification permet d'ajouter des traits non spécifiés

un	det	(↑SPEC)=A	(↑NUM)=SG
corps	N	(↑PRED)="corps"	<del>(↑NUM)=</del>



- Les valeurs unifiées doivent bien sûr être égales

# Formes « sémantiques »

---

- Les valeurs de l'attribut PRED, par exemple l'expression prédicat-argument "*hand*"  $\langle SUBJ, OBJ, OBJ2 \rangle$ , sont assez proches des termes du langage Prolog, et sont traitées par unification
  - On a l'équivalent du terme *hand* (*agent, thème, but*)
  - Chaque *argument* sera unifié avec les interprétations de, respectivement, SUBJ, OBJ, OBJ2, qui devront être présents dans la f-structure qui contient cette attribut PRED
  - Donc *agent* = le lemme "girl" au singulier précédé d'un article indéfini
- Toute f-structure doit avoir un (et un seul) attribut PRED, et tous ses arguments doivent être instanciés
- Il n'y a pas unification des attributs PRED

# Définitions un peu plus formelles : opérations sur les f-structures

---

- Une f-structure  $A$  est une extension de  $B$  ( $A \supset B$ ) ssi:
  - Tout trait atomique de  $B$  est présent dans  $A$  avec la même valeur
  - Toute sous f-structure  $F$  de  $A$  est une extension de la sous f-struct  $F$  de  $B$
- L'unification  $C$  de deux f-structures  $A$  et  $B$  ( $C = A \cup B$ ) est la f-structure minimale telle que  $C \supset A$  et  $C \supset B$ 
  - L'unification peut donc échouer, noté  $\perp$
- La généralisation  $C$  de deux f-struct  $A$  et  $B$  ( $C = A \mathbf{G} B$ ) est la f-structure maximale telle que  $A \supset C$  et  $B \supset C$
- On peut faire une analogie avec le plus petit multiple commun et le plus grand diviseur commun de 2 nombres

# Définitions un peu plus formelles : équations fonctionnelles

---

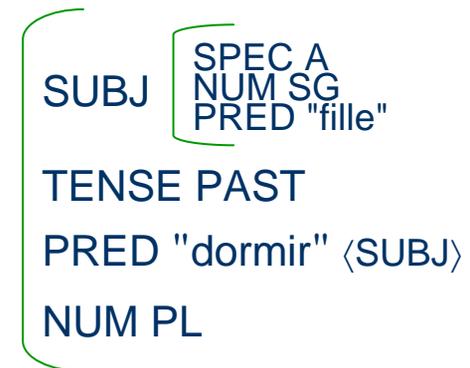
- Pour un attribut  $a$  d'une f-structure
  - $(\downarrow a) = \alpha$  sous un composant de c-structure  $X$  unifie les traits de la f-struct de  $X$  par  $a = \alpha$   
Exemple pour *le chien se léche les babines*  
 $S \rightarrow NP \quad pr\text{-réfl} \quad VP$   
 $(\downarrow CAT) = \text{verbe-pronominal}$
  - $(\uparrow a) = \alpha$  dans une règle  $Y \rightarrow \dots$  unifie les traits de la f-struct de  $Y$  par  $a = \alpha$
  - Autre opérations possibles : inexistence, appartenance à un ensemble  $\sim(\downarrow a), (\uparrow a') \in \downarrow a''$
  - Possibilité de nommer un attribut d'une sous f-structure :  
 $(\uparrow SUBJ NUM) = \alpha$
  - ...

# Unification comme moyen de contrôle contextuel

---

- Supposons qu'un verbe a aussi un attribut NUM qui indique s'il est au singulier ou au pluriel

- Dans  $s \rightarrow$   $\begin{array}{cc} NP & VP \\ (\uparrow \text{SUBJ}) = \downarrow & \uparrow = \downarrow \end{array}$   
on ne vérifie pas qu'il y a accord entre le sujet et le verbe,  
ce qui autorise *une fille dormions* puisque les 2 attributs NUM ne sont pas unifiés (ils ne sont pas au même niveau)



- On peut ajouter une équation  $(\uparrow \text{SUBJ NUM}) = (\uparrow \text{NUM})$  pour faire la vérification (et donc considérer la phrase ci-dessus comme incorrecte)

# Contraintes existentielles

---

- On peut contraindre à ce qu'un attribut soit présent (ou absent), sans vouloir préciser sa valeur
- Par exemple un nom peut être un nom commun (ayant un attribut ART) et doit être précédé d'un article, ou un nom propre (sans attribut ART), non précédé d'un article :

$$S \rightarrow \begin{array}{cccc|cc} N & & VP & & \text{det} & N & VP \\ & \sim(\downarrow\text{ART}) & & & & (\downarrow\text{ART}) & \end{array}$$

permet d'accepter *la fille dort* ou *Jean dort*,  
et de rejeter *fille dort* ou *le Jean dort*

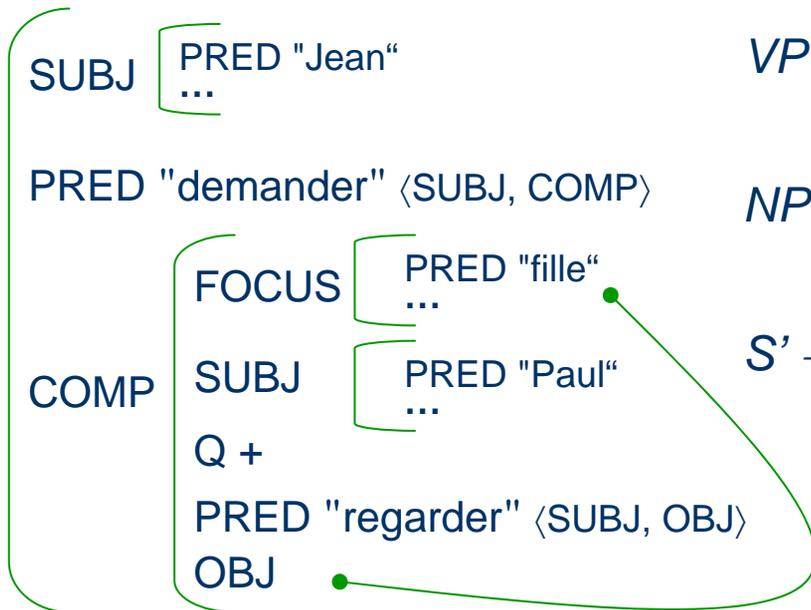
# Equations contraintes

---

- Il n'y a pas, en général, de correspondance 1-1 entre c-structure et f-structure
  - $S \rightarrow VP$  pour *Hand the baby a toy*
  - La f-structure de S contient un attribut SUBJ alors que la forme superficielle de la phrase n'a pas de sujet
  - Le verbe doit être un infinitif (attribut INF)
  - Donc  $S \rightarrow VP$ 
    - $\uparrow = \downarrow$
    - $(\uparrow \text{ INF}) =_c +$
    - $(\uparrow \text{ SUBJ PRED}) = \text{"you"}$qui exige que l'entrée lexicale de *hand* ait un trait  $(\uparrow \text{ INF}) =_c +$  puisque  $\uparrow = \downarrow$

# Partage des structures fonctionnelles

- Comment traiter *Jean demande quelle fille Paul regarde ?*  
*fille* est l'objet de *regarde*, mais aussi celui de *demande*
- Ils partagent l'objet



$S \rightarrow$	$NP$	$VP$
	$(\uparrow \text{SUBJ}) = \downarrow$	$\uparrow = \downarrow$
$VP \rightarrow$	$V$	$S'$
	$\uparrow = \downarrow$	$(\uparrow \text{COMP}) = \downarrow$
$NP \rightarrow$	$det$	$N$
	$\uparrow = \downarrow$	$\uparrow = \downarrow$
$S' \rightarrow$	$NP$	$S$
	$(\uparrow \text{FOCUS}) = \downarrow$	$\uparrow = \downarrow$
	$(\downarrow Q) =_c +$	$(\uparrow Q) = +$
	$(\uparrow \text{FOCUS}) = (\uparrow \text{OBJ})$	

# Accès « à distance »

---

- Des constituants n'ont pas de relation immédiate dans la c-structure (c-à-d père-fils) mais ont des dépendances fonctionnelles :

*Jean demande quelle fille Marie pense que Paul regarde*

- *fille* est l'objet de *regarde*, mais aussi le focus de *demande* qui n'ont pas de relation immédiate dans l'arbre
- Donc transformation de la règle S' en utilisant une dépendance à distance:

$S' \rightarrow NP$   
 $(\uparrow \text{FOCUS}) = \downarrow$   
 $(\downarrow Q) =_c +$

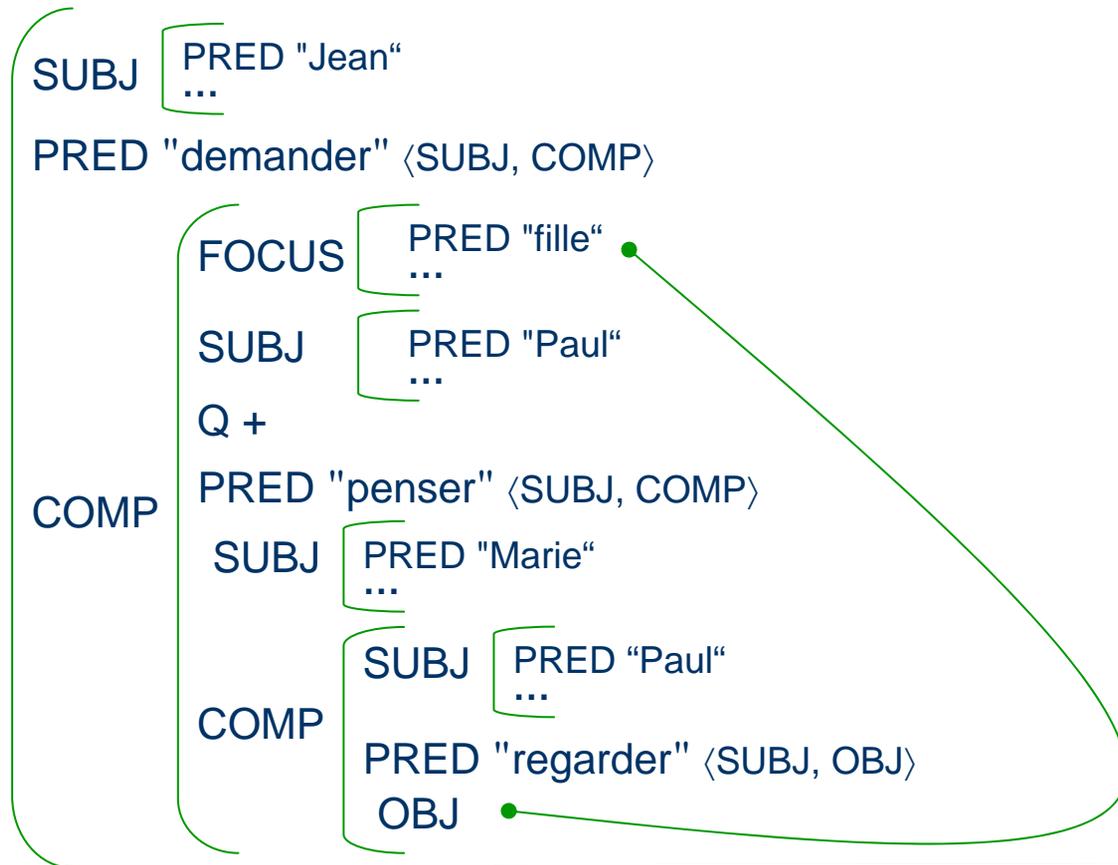
$S$   
 $\uparrow = \downarrow$   
 $(\uparrow Q) = +$

$(\uparrow \text{FOCUS}) = \uparrow(\text{COMP})^* \text{OBJ}$  on « descend » dans les f-structures jusqu'à un attribut COMP ayant un attribut OBJ

# Accès « à distance »

---

*Jean demande quelle fille Marie pense que Paul regarde*



# Bonne formation des f-structures

---

- Pour être bien formée, une f-structure doit respecter les principes suivants :
  - *Unicité* : un même attribut ne peut y être présent qu'une fois avec une seule valeur à un même niveau
  - *Complétude* : ses sous-structures doivent être complètes, c-à-d tous les arguments obligatoires du prédicat doivent être présents  
*Ce problème concerne Jean* ( $\Rightarrow$ OK) / *Ce problème concerne* ( $\Rightarrow$ KO)
  - *Cohérence* : ses sous-structures doivent être cohérentes, c-à-d tous les arguments du prédicat doivent avoir des valeurs correctes  
*Jean dort (pendant) le jour* ( $\Rightarrow$ OK) / *Jean dort (pendant) la pomme* ( $\Rightarrow$ KO)
- f-structure malformée  $\Rightarrow$  phrase incorrecte

# Bonne formation des f-structures (exemple)

---

- Reprenons  $S \rightarrow VP$ ,  $VP \rightarrow NP$   $NP$   
 $\uparrow = \downarrow$   $(\uparrow\text{OBJ}) = \downarrow$   $(\uparrow\text{OBJ2}) = \downarrow$   
qui permet de produire *Hand the baby a toy*
- L'équation  $\uparrow = \downarrow$  permet de définir *l'ancre* (on dit aussi la *tête*) de S, c-à-d quelle f-structure sera celle de S (donc ici celle de VP)
- Il y aura dans le lexique une entrée pour *hand* en tant que verbe, mais aussi une en tant que nom commun
  - l'entrée verbe aura un PRED "hand"  $\langle \text{SUBJ}, \text{OBJ}, \text{OBJ2} \rangle \Rightarrow \text{OK}$
  - L'entrée nom n'aura pas de prédicat, ou pas un d'arguments  $\langle \text{SUBJ}, \text{OBJ}, \text{OBJ2} \rangle \Rightarrow \text{KO}$

# Applicabilité des LFG

---

- Un 1<sup>er</sup> résultat capital : ***on peut décider si une phrase appartient ou non au langage d'une LFG***  
(c'est beau de concevoir un formalisme, encore faut-il qu'il soit opératoire !)
- Éléments de preuve : on connaît des algorithmes pour savoir si une phrase appartient au langage d'une CFG, et une fois la c-structure construite, on sait vérifier que sa f-structure est bien formée
  - Si on exclut les grammaires cycliques, c-à-d telles que  $\exists A, A \Rightarrow^+ A$  (donc avec des règles du genre  $A \rightarrow A_1, A_1 \rightarrow A_2, \dots, A_{n-1} \rightarrow A_n, A_n \rightarrow A$ )

# Puissance de description des LFG

---

- Les langages LF contiennent les langages CF (context-free), et sont inclus dans les langages CS (context-sensitive)
- Le langage  $a^n b^n c^n$  n'est pas CF, mais il est LF :

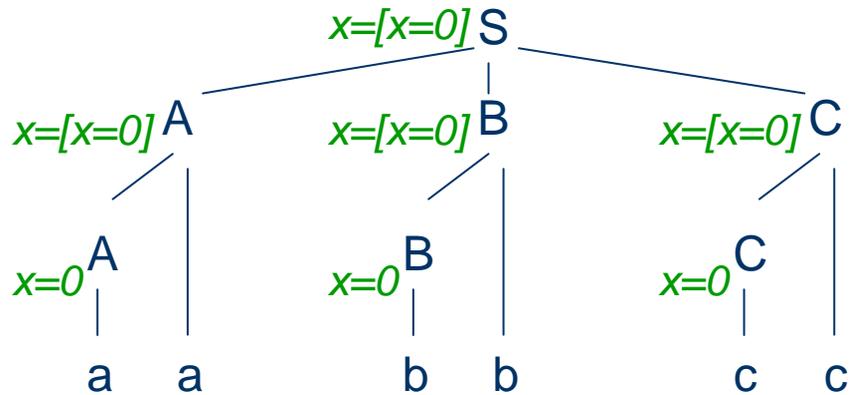
- Grammaire LF :
$$\begin{array}{l} S \rightarrow \quad A \quad B \quad C \\ \quad \quad \uparrow=\downarrow \quad \uparrow=\downarrow \quad \uparrow=\downarrow \\ A \rightarrow \quad a \\ \quad \quad (\uparrow x) = 0 \\ A \rightarrow \quad A \quad a \\ \quad \quad (\uparrow x) = \downarrow \end{array}$$

mêmes équations pour  $B \rightarrow b \mid B b$ ,  $C \rightarrow c \mid C c$

# Puissance de description des LFG (suite)

- Pour la phrase aabbcc, on a

⇒ OK



- Mais pour aabcc, on a

⇒ KO

