

Traitement Automatique des Langues

Jean-Yves Antoine

Université François Rabelais de Tours

www.info.univ-tours.fr/~antoine

Traitement Automatique des Langues

GRAMMAIRES ATTRIBUEES ET
GRAMMAIRES D'UNIFICATION

Grammaires de traits (*features grammars*)

Limites des grammaires hors contexte

- pouvoir de génération insuffisant mais (presque) seul utilisable en pratique
- grammaires CFG réelles trop lourdes pour être maintenables.

Exemple : gestion des accords

gestion des accords en CFG : une catégorie par genre et nombre !

<SN_m_sg> → <DET_m_sg> <NC_m_sg> *le bateau*
<SN_f_sg> → <DET_f_sg> <NC_f_sg> *la fille*
lexique *les* : <DET_m_pl> , *les* : <DET_m_pl>

Grammaires de traits

[Knuth 1968]

- **Grammaires syntagmatiques** : structure de constituants
- Ajout aux catégories syntaxiques d'une structures de **traits attributs-valeurs** décrivant des propriétés diverses (morphologie, rection, fonctions syntaxiques...)
- Gestion des accords, de la sous-catégorisation verbale, etc.

Grammaires de traits (*features grammars*)

Lexique

- catégorie syntaxique + propriétés de l'entrée sous forme de traits

Exemple *le* (**DET** PRED=*le* NB=*sg* GENRE=*m*)

aimes : (**V** PRED=***aimer*** FORM=*ind_pst* PERS=2 NB= *sg*)

- ambiguïté : variables contraintes

aime(**V** PRED=***aimer*** FORM=*ind_pst* PERS= $X[1,3]$ NB= *sg*)

Règles

- Application dépendant de la satisfaction de contraintes sur certains traits
- Factorisation des situations à l'aide de valeurs variables

Exemple

(**SN** NB= X GENRE= Y) \rightarrow (**DET** NB= X GENRE= Y) (**NC** NB= X GENRE= Y)

- **Remarque** : les règles peuvent utiliser des constantes comme attributs

Grammaires de traits

Analyse : déclenchement des règles

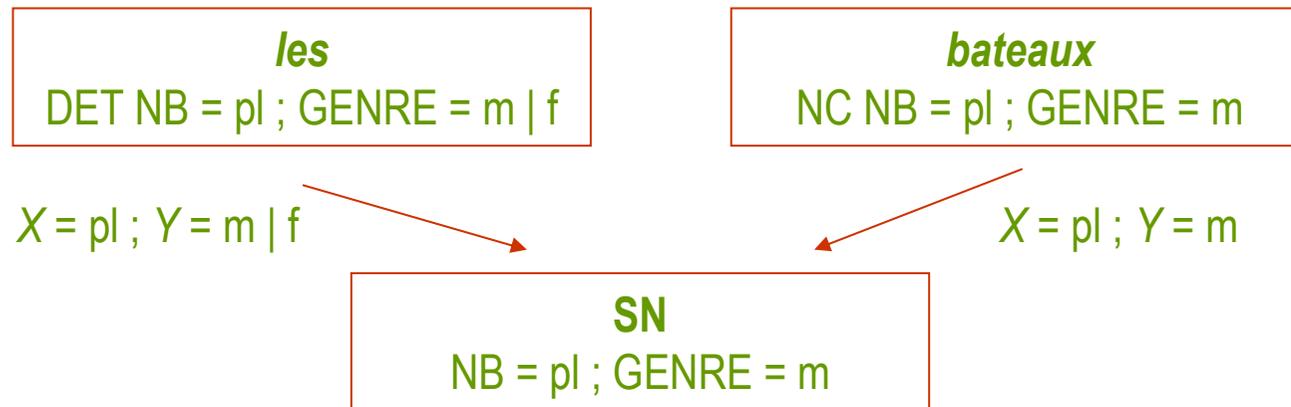
- Basée sur l'analyse des syntagmes du corps (analyse ascendante) ou de la tête (analyse descendante) comme pour les grammaires hors-contexte
- Vérification supplémentaire des contraintes sur les traits : **unification**

Exemple *les bateaux*

règle (**SN** NB=X GENRE=Y) → (**DET** NB=X GENRE=Y) (**NC** NB=X GENRE=Y)

lexique *les* (**DET** PRED=*le* NB=*pl* GENRE= Z[m,f])

bateaux (**NC** PRED=*bateau* NB=*pl* GENRE=*m*)

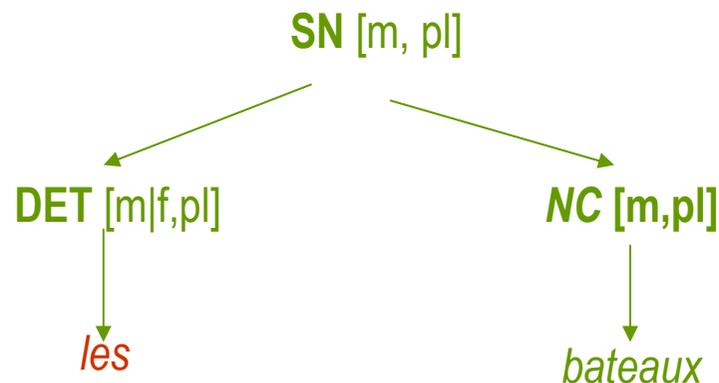


Grammaires de traits

Construction de l'arbre d'analyse

- La règle ne précise que les contraintes sur les traits, mais l'ensemble de la structure de traits des éléments du corps est héritée dans le syntagme tête
- syntagme : catégorie syntaxique + structure de traits comme pour le lexique.

les bateaux : (**SN** NOMBRE=*pl* GENRE = *m*
1 (**DET** PRED=*le* NOMBRE=*pl* GENRE=*m|f*)
2 (**NC** PRED=*bateau* NOMBRE=*pl* GENRE=*m*)
)



Grammaires de traits : analyse

Pouvoir de génération

- Équivalent à une CFG si l'ensemble des valeurs d'attribut est fini (ce qui est le cas en pratique)
- Grammaires de traits équivalentes aux CFG, mais plus maintenables

Stratégies d'analyse

- Équivalence avec les CFG : mêmes stratégies d'analyse
- Algorithmes identiques + vérification des contraintes sur les traits + instanciation des variables (unification) satisfaisant ces contraintes au cours de l'analyse

Exemples d'attributs

Gestion des accords

- SN : accord en genre et en nombre NB, GENRE
- SV : accord en nombre et personne NB, PERS

Sous-catégorisation verbale

- permet de spécifier le nombre et le type des arguments attendus
- lexicalisation de l'analyse : les syntagmes de l'énoncé dépendront du verbe

exemples	SSCAT = nil	<i>dormir, sourire, courir,...</i>
	SSCAT = (GN)	<i>regarder, manger, ...</i>
	SSCAT = (GN,GP:à)	<i>donner, prendre, ...</i>

règles	(SV) → (V SSCAT= <i>nil</i>)
	(SV) → (V SSCAT=(<i>GN</i>)) (GN)
	(SV) → (V SSCAT=((<i>GN, GP:à</i>)) (GN) (GP PREP= <i>à</i>))

- ⇒ chaque règle correspondra à une grande classe de verbes et sera activée uniquement par le lexique correspondant.

Fonction syntaxique

- Objet, Patient...
⇒ Grammaires Lexicales-Fonctionnelles (LFG)

Élaboration de grammaire réelle

Valeurs par défaut

- Afin de limiter le travail du concepteur de grammaire, on peut associer des valeurs de traits par défaut à chaque catégorie syntaxique
- Valeur utilisée uniquement si aucune précision n'est donnée sur l'attribut concerné dans la définition de l'entrée lexicale considérée.

Règles lexicales

- Permettent d'éviter de définir à la main les entrées de toutes les formes fléchies
- Règles morphologiques de dérivation inflexionnelle

lexique manger (V PRED=manger FORM=inf SSCAT=(SN) GROUP=1)

règle (V PRED=X SSCAT=Y GROUP=1 FORM=ind_pst PERS=2 NB= sg)
 → (V PRED=X SSCAT=Y GROUP=1 FORM=inf) + (S)

- Règles par défaut : en cas d'exception, on rentre la définition précise

Bibliographie

Articles ou ouvrages d'entrée

- **Abeillé A.** (1991) Quand l'arbre ne cache pas la forêt : analyse du français à l'aide d'une grammaire d'arbres adjoints. *TA Informations*, 32(2), 51-70.
- **Abeillé A.** (1993) Les nouvelles syntaxes : grammaires d'unification et analyse du français.
- **Allen J.** (1995) *Natural Language Understanding*. Benjamin / Cummings Publ. Comp. Redwood City, CA.
- **Grosz B., Jones K., Webber B.** (Eds.) (1986) *Readings in Natural Language Processing*. Morgan Kaufmann Publishers Inc.

Articles ou ouvrages de référence

- **Joshi A.** (1987) Introduction to Tree Adjoining Grammar. In Ramer M. (Ed.) *The mathematics of language*. J. Benjamins, Amsterdam, Pays-Bas.
- **Knuth D.E.** (1968) Semantics for context-free languages. *Mathematical Systems Theory*, 2. 127-145.
- **Kaplan R., Bresnan J.** (1982) Lexical-Functional Grammar : a formal system for grammatical representation. In Bresnan J. (Ed.) *The mental representation of grammatical relations*. MIT Press, Cambridge, MA. 173-281.

Traitement Automatique des Langues

GRAMMAIRES ATTRIBUEES ET
GRAMMAIRES D'UNIFICATION

*Compléments hors programme :
formalismes d'analyse syntaxique*

Grammaires de traits : analyse

Pouvoir de génération

- Équivalent à une CFG si l'ensemble des valeurs d'attribut est fini (ce qui est le cas en pratique)
- Grammaires de traits équivalentes aux CFG, mais plus maintenables

Stratégies d'analyse

- Équivalence avec les CFG : mêmes stratégies d'analyse
- Algorithmes identiques + vérification des contraintes sur les traits + instantiation des variables (unification) satisfaisant ces contraintes au cours de l'analyse

Exemple : analyse tabulaire ascendante

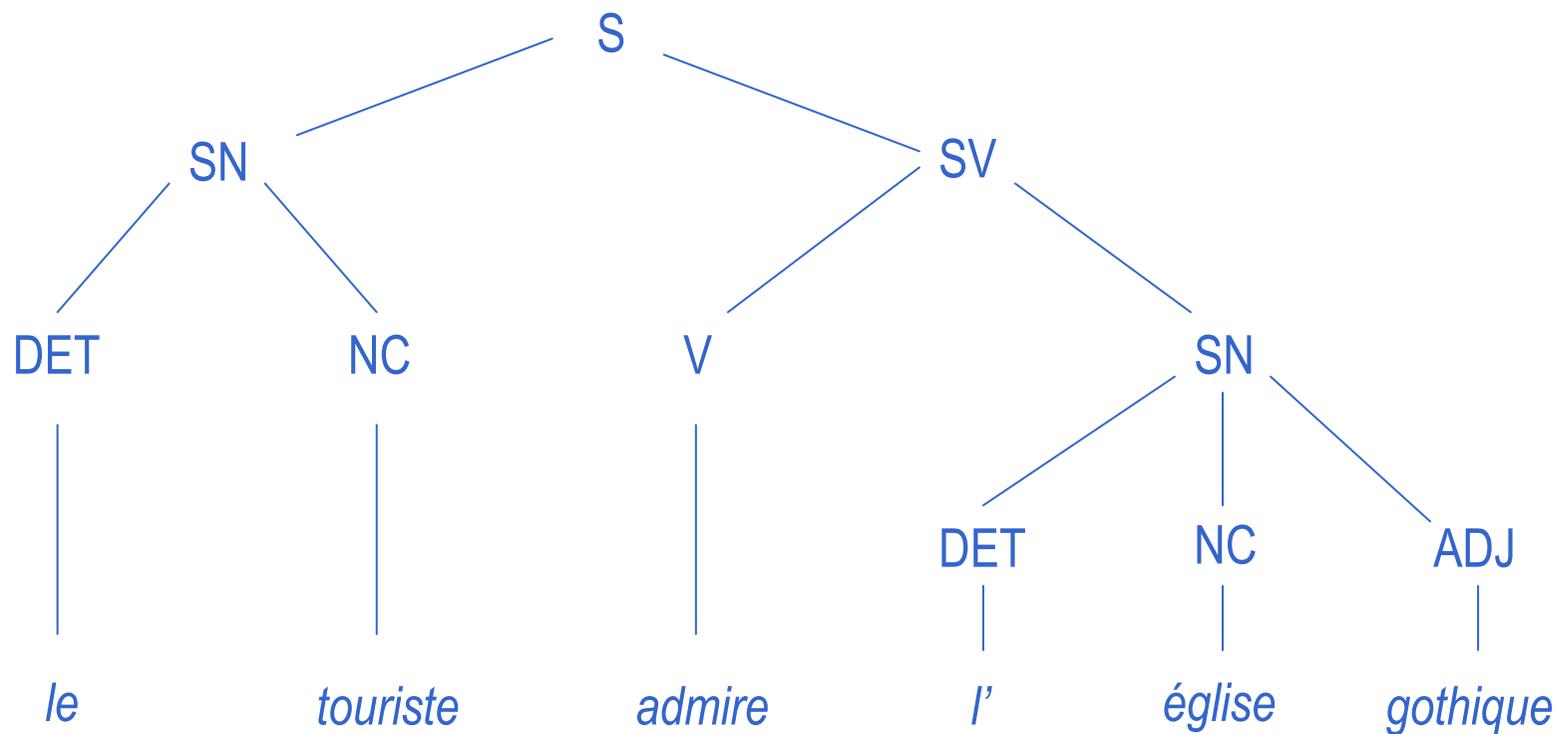
- **Extension d'arc** lors de l'extension, on instancie les variables en fonction des valeurs d'attributs que l'on vient de traiter.
- **Ambiguïté** instantiation du syntagme en construction (tête) et non pas de ses sous-constituants : la valeur de leurs traits peut avoir une autre interprétation sur une analyse alternative.

Grammaires lexicales-fonctionnelles

- **Grammaire LFG** (*Lexical-Functional Grammar*) [Bresnan & Kaplan 1982]
 - Grammaire hors-contexte augmentée par des structures de trait
 - Extension à la syntaxe des règles lexicales (passif ...)
 - Remarque** : abandonné dans les version ultérieures du formalisme
 - Fonctions syntaxiques : rapprochement de la sémantique
- **Représentations de l'énoncé**
 - Plusieurs arbres d'analyse ou représentations logiques construites en parallèle au cours de l'analyse
 - **Structure de constituants** (*c-structure*) structure syntagmatique classique
 - **Structure fonctionnelle** (*f-structure*) fonction syntaxique des éléments
 - **Structure prédicative** relations prédicat-argument proche d'une formule sémantico-logique

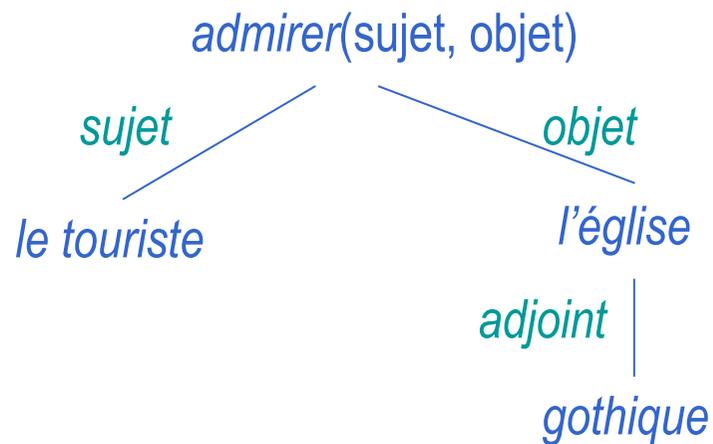
Grammaires lexicale-fonctionnelles

- C-structure



Grammaires lexicale-fonctionnelles

- **F-structure** (simplifiée)



PRED = admirer < (↑ **SUJ**) (↑ **OBJ**) >
 MODE = indicatif
 TPS = present
 PRES = 3

SUJ = {

PRED = touriste

GR = m NB = sg

DEF = +

PERS = 3
 }

OBJ = {

PRED = église

GR = f NB = sg

DEF = +

ADJT = { PRED = gothique }
 }

- **Structure prédicative** (simplifiée)

<admirer

arg-1 : <touriste>

arg-2 : <église adjoint-1 : <gothique>>

>

LFG : principales fonctions syntaxiques

- **Arguments : compléments sous-catégorisés**

- **SUJ** sujet
- **OBJ** objet
- **P_OBJ** compléments prépositionnels : *dans_OBJ, par_OBJ...*
- **COMP** complétive en *que* *je crois **que tu ne m'aimes pas***
- **VCOMP** complétive infinitive *Jean veut **manger de la soupe***
- **ACOMP** complément adjectival *Jean devient **dangereux***

- **Adjoints : compléments non sous-catégorisés**

- **ADJT** adjectifs épithètes, adverbes, groupes prépositionnels circonstanciels
- **VADJT** propositions subordonnées

- **Référence**

- **TOPIC** rappel du référent du pronom relatif
- **FOCUS** rappel du référent antéposé dans les extractions ou les questions

LFG : lexique

- **Entrée lexicale**

- structure de traits : attributs morphologies + sous-catégorisation

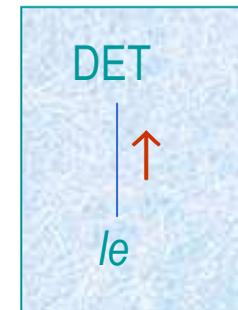
pain (NC PRED=*pain* GR=*m* NB=*sg* DEF=X[+,-] PERS=3)

le (DET PRED=*le* GR=*m* NB=*sg* DEF=+)

mange (V PRED=*manger* SSCAT=(*SUJ,OBJ*) NB=*sg* PERS=3 MOD=*ind* TPS=*pres*)

- **Représentation en LFG**

- $\uparrow X$: attribut X du nœud père. Dans le cas d'une entrée lexicale, cela correspondra au nœud où est inséré le mot dans l'énoncé



- La gestion des accords est ainsi réalisée par le lexique (*cf infra*)

pain (NC (\uparrow PRED) = *pain*, (\uparrow GR) = *m*, (\uparrow NB) = *sg*, (\uparrow PERS) = 3)

mange (V (\uparrow PRED) = *manger* < (\uparrow **SUJ**) (\uparrow **OBJ**) > (\uparrow MOD) = *ind* (\uparrow TPS) = *pres*)

le (DET (\uparrow DEF) = + GR = *m* NB = *sg*)

déterminants non prédicatifs

LFG : grammaire

- Règles

- grammaire hors-contexte + équations fonctionnelles sur les traits

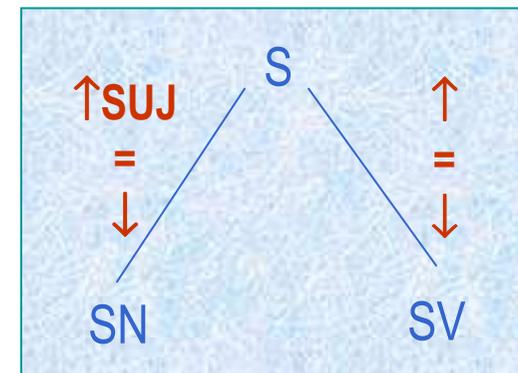
S → SN
(↑ SUJ) = ↓

*L'attribut sujet de la f-structure du constituant S (↑ SUJ) est **égal** à la f-structure (↓) du sous-constituant SN*

SV
↑ = ↓

*la f-structure du constituant S (↑) est **égale** à celle du sous-constituant concerné (↓), ici le SV*

- ↑ : nœud père, i.e. de la tête de règle
- ↓ : nœud fils correspondant au constituant concerné du corps de règle.



LFG : grammaire

- **Équations fonctionnelles**

- ① règles de construction de la f-structure
- ② contraintes sur le déclenchement de la règle : les attributs doivent s'unifier

- **Exemple : gestion des accords**

- **Lexique** (simplifié)

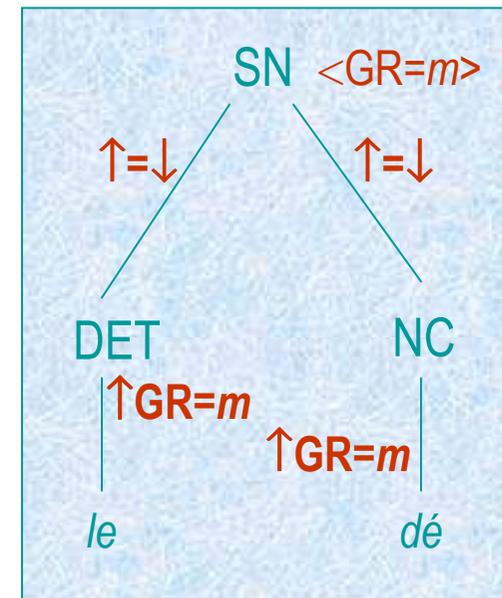
le : (DET (\uparrow GR) = *m* (\uparrow NB) = *sg*)

dé : (NC, (\uparrow PRED) = *dé* (\uparrow GR) = *m* (\uparrow NB) = *sg*)

- **règle**

SN	→	DET		NC
		$\uparrow = \downarrow$		$\uparrow = \downarrow$

- **gestion explicite** (équation sur les attributs de genre, nombre ou personne) dans d'autres cas



LFG : grammaire

- Exemple

- lexique

Jean (NP (↑PRED) = *Jean* (↑GR) = *m* (↑ NB) = *sg* (↑DEF) = + (↑PERS) = 3)

dort (V, (↑PRED) = *dormir*< (↑SUJ) > (↑MOD) = *ind* (↑TPS) = *pres* (↑ NB) = *sg* (↑PERS)=3)

- règle

S

→

NP

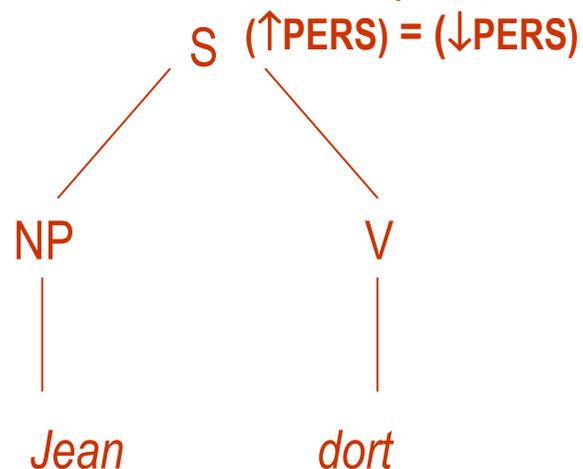
(↑SUJ) = ↓

(↑NB) = (↓NB)

(↑PERS) = (↓PERS)

V

↑ = ↓



PRED = *dormir*<↑(suj)>
 MODE = *indicatif*
 TPS = *present*
 PERS = 3 NB = *sg*
 SUJ = $\left[\begin{array}{l} \text{PRED} = \textit{Jean} \\ \text{GR} = \textit{m} \text{ NB} = \textit{sg} \\ \text{DEF} = + \\ \text{PERS} = 3 \end{array} \right]$

Unification

- **Substitution**

- On appelle substitution (finie) toute application qui associe à un ensemble (fini) de valeurs variables à chaque fois une valeur constante ou une autre valeur variable de domaine de variation compatible.

- **Exemple**

PERS=X[1,2,3]	substituable par PERS=2 ou PERS= Y[2,3]
DEF= +	non substituable par DEF= -
MODE=X[ind,subj]	non substituable par MODE=X[ind,cond]

- **Unification**

- Soient deux structures de traits S1 et S2. On dit que S1 et S2 sont unifiables s'il existe une structure de trait S (appelée **unificateur**) qui contient S1 et S2 ou une de leur substitution.

- **Exemple**

(NC PERS=2 NB=sg)	unifiable avec (NC PERS=X NB=sg DEF=+)
unificateur	(NC PERS=2 NB=sg DEF=+)

LFG et unification

- Unification

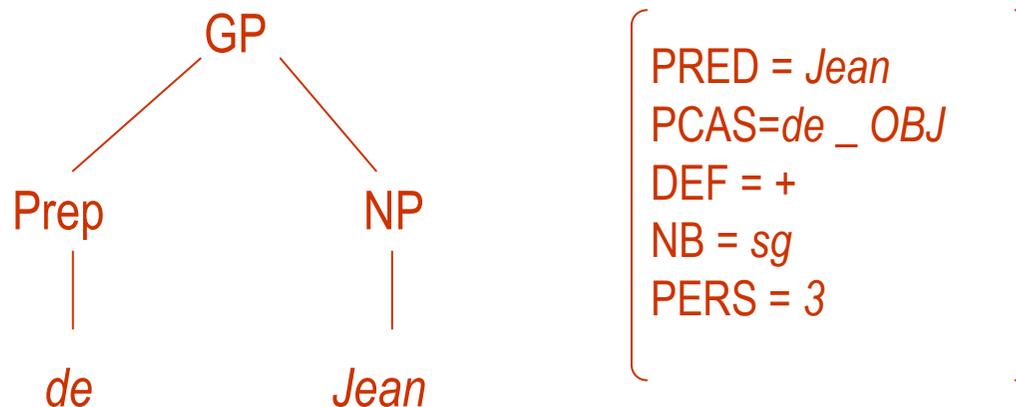
- **équations fonctionnelles**: unification et non égalité des f-structures
- **exemple** : représentation de la « tête double » (préposition + GN) des GP

GP → PREP NP
 ↑ = ↓ ↑ = ↓

et *de* : (PREP (↑ PCAS) = *de_OBJ*)

Jean : (NP (↑ PRED) = *Jean* (↑ PERS) = 3 (↑ NB) = *sg* (↑ DEF) = +)

de Jean :



LFG : analyse

- **Stratégie générale**

- grammaire hors-contexte (construction *c-structure*) + vérification des équations fonctionnelles (construction *f-structure*)
- *f-structure* construction en parallèle ou en post-filtrage de la *c-structure*
respects de principes de bonne formation
- algorithmes classiques utilisables : *chart parser* [Zweigenbaum 1991]

- **Principes généraux de bonne formation**

- **consistance** une fonction ne peut avoir qu'une seule valeur (unicité)
- **cohérence** toute fonction doit être sous-catégorisée par un prédicat
- **complétude** toutes les fonctions sous-catégorisées (*et non les obliques*) doivent être remplies

Exemple : lexique LFG

<i>le</i>	(DET (↑DEF) = + (↑GR) = <i>m</i> (↑NB) = <i>sg</i>)
<i>la</i>	(DET (↑DEF) = + (↑GR) = <i>f</i> (↑NB) = <i>sg</i>)
<i>pain</i>	(NC (↑PRED) = <i>pain</i> (↑GR) = <i>m</i> (↑NB) = <i>sg</i> (↑PERS) = 3)
<i>fil</i>	(NC (↑PRED) = <i>fil</i> (↑GR) = <i>m</i> (↑PERS) = 3)
<i>Jean</i>	(NP (↑PRED) = <i>Jean</i> (↑GR) = <i>m</i> (↑NB) = <i>sg</i> (↑PERS) = 3)
<i>bon</i>	(ADJ (↑PRED) = <i>bon</i> <(↑ SUJ)> (↑ATTR) = + (↑GR) = <i>m</i> (↑NB) = <i>sg</i>)
<i>mange</i>	(V, (↑PRED) = <i>manger</i> < (↑SUJ) (↑OBJ) > (↑MOD) = <i>ind</i> (↑TPS) = <i>pres</i>)
<i>est</i>	(V, (↑PRED) = <i>être</i> < (↑ SUJ) (↑ ACOMP) > (↑MOD) = <i>ind</i> (↑TPS) = <i>pres</i>)
<i>qui</i>	(PREL (↑TOPIC)=(↑SUJ))
<i>que</i>	(PREL (↑TOPIC)=(↑OBJ))
<i>de</i>	(PREP (↑ PCAS) = <i>de_OBJ</i>)

Exemple : grammaire LFG

S	→	SN	SV	
		(↑SUJ) = ↓ (↑NB) = (↓NB) (↑PERS) = (↓PERS)	↑ = ↓	
SV	→	V	SN	{ GP }
		↑ = ↓	(↑OBJ) = ↓	(↑ ADJT) ∃ ↓
SV	→	V	ADJ	
		↑ = ↓	(↑ACOMP) = ↓	
SN	→	NP		
		↑ = ↓		
SN	→	GN	{ GP }	
		↑ = ↓	↑(ADJT) ∃ ↓ ; (↓ PCAS) = <i>de_OBJ</i>	
GN	→	DET	{ ADJ }	NC
		↑ = ↓	(↑ ADJT) ∃ ↓ (↓ SUJ) = ↑ (↑NB)=(↓NB)(↑GR)=(↓GR)	↑ = ↓
				{ SUBREL }
				(↑ ADJT) ∃ ↓ (↓ TOPIC) = ↑
GP	→	Prep	SN	
		↑ = ↓	↑ = ↓	
SUBREL	→	Prel	S	
		↑ = ↓	↑ = ↓	

LFG : exemple

PRED = *manger*<(↑ SUJ)(↑ OBJ)>
MODE = *indicatif*
TPS = *present*
PERS = 3 PERS = *sg*

SUJ = PRED = *films*
GR = *m* NB = *sg*
DEF = +
PERS = 3

1 = OBJ = PRED = *pain*
GR = *m* NB = *sg*
DEF = +
ADJT1 = PRED = *bon*<(↑SUJ)>
ATTR = +
SUJ = 1
ADJT2 = PRED = *Jean*
GR = *m* NB = *sg* PERS = 3
PCAS = *de_Obj*

*le fils mange
le bon pain
de Jean*

LFG : exemple

- *le fils mange le pain qui est bon*

PRED = *manger* < (↑ SUJ) (↑ OBJ) >
MODE = *indicatif*
TPS = *present*
PERS = 3

SUJ = PRED = *fils*
GR = *m* NB = *sg*
DEF = +
PERS = 3

1 = OBJ = PRED = *pain*
GR = *m* NB = *sg*
DEF = +
ADJT = *à vous de jouer....*

- *la fils mange le bon pain*
- *le fils mange*
- *le fils est le pain*

⇒ échec car problème d'accord

⇒ échec car principe de complétude non satisfait

⇒ échec car principe de complétude et de cohérence non satisfait

LFG : exemple

- *le fils mange le pain la viande* ⇒ échec dès la c-structure. Auto-correction ?
- *la fils mange le bon pain* ⇒ échec car problème d'accord
- *le fils est le bon pain* ⇒ échec car structure prédicative incompatible
- *le fils mange* ⇒ échec car principe de complétude non satisfait.
Ajout *manger* intransitif dans le lexique ?

Grammaires d'unification

- **Principes généraux**

- Logique des **grammaires de traits** poussées à son terme : la catégorie syntaxique est elle-même considérée comme un attribut.
- Entrées lexicales se limitant à une structure de traits

pain (**CAT=NC** PRED = *pain* GR = *m* NB = *sg* DEF = *X[+,-]* PERS =3)

est (**CAT=V** PRED=*être* SSCAT=(*SUJ,OBJ*) NB=*sg* PERS=3 MOD=*ind* TPS=*pres*)

- Analyse : **unification des structures** de traits correspondantes.

- **Pouvoir de génération**

- Les grammaires d'unifications « standard » restent dans la classe des grammaires hors-contexte
- **Justification** : on peut toujours créer autant de classes syntaxiques qu'il y a de combinaisons (finies) de structures de traits possibles.

Grammaires d'unification

- **Grammaires lexicalisées**

- En se basant sur les propriétés de sous-catégorisation des entrées lexicales, il est possible de réduire au strict minimum, voir de se passer de la notion de règle: **connaissance syntaxique majoritaire dans le lexique**

- **Exemple** (artificiel) $aime(CAT = V \text{ PRED}=\textit{aimer} \text{ ARGSUJ}=\text{NP} \text{ ARGOBJ}=\text{NP})$
 $Jean (CAT = NP \text{ PRED}=\textit{Jean})$
 $Marie(CAT = NP \text{ PRED}=\textit{Marie})$

Jean aime Marie

$$\left[\begin{array}{l} \text{PRED} = \textit{aimer} \\ \text{CAT} = V \\ \text{ARGSUJ} = \left\{ \begin{array}{l} \text{PRED} = \textit{Jean} \\ \text{CAT} = \text{NP} \end{array} \right\} \\ \text{ARGOBJ} = \left\{ \begin{array}{l} \text{PRED} = \textit{Marie} \\ \text{CAT} = \text{NP} \end{array} \right\} \end{array} \right]$$

- **Grammaires lexicalisées**

TAG

[Joshi *et al.* 1987]

HPSG

[Pollard, Sag 1993]

link grammars (dépendances) [Sleator, Temperley 91]

Grammaires TAG

- **Grammaires d'arbres adjoints** (*Tree Adjoining Grammars*)
 - Opération d'**unification** mais aussi d'**adjonction** (non étudiée dans ce cours)
- **Pouvoir de génération**
 - L'**adjonction** étend légèrement le pouvoir de **génération faible** des TAG par rapport aux grammaires hors-contexte :
 - $(a^n b^n c^n d^n)$ est un langage pouvant être généré par une grammaires TAG
 - $(a^n b^n c^n d^n e^n)$ ne peut être généré par une grammaire TAG
 - Classe des grammaires **légèrement sensibles au contexte**
 - $\text{CFG} < \text{TAG} < \text{grammaires contextuelles}$ [Joshi et al. 1991]
- **Combinatoire d'analyse**
 - On reste dans la classe des grammaires analysables en **temps polynomial**
 - Algorithme d'Earley adaptable aux grammaires TAG [Shabes 1990]

Bibliographie

Travaux cités

- **Abeillé A.** (1989) L'unification dans les grammaires d'arbres d'arbres adjoints : quelques exemples en syntaxe française. *TA Informations*, 30 (1-2), 69-112.
- **Joshi A., Weir D., Vijay-Shanker K..** (1991) The convergence of mildly context-sensitive formalisms. In Sells P., Shieber S., Wasow T. (Eds.) *Foundational issues in NLP*. MIT Press, Cambridge, MA
- **Pollard C., Sag I.** (1993) *Head-driven Phrase Structure Grammars*. CSLI, University of Chicago Press, Ma.
- **Schabes Y.** (1990) Computational and mathematical properties of lexicalized grammars. PhD Thesis. CIS, U. of Pennsylvania, Philadelphia.
- **Sleator D., Temperley D.** (1991) Parsing English with a link grammar. Rapport recherche CMU-CS-91-196. School of Computer Science. CMU. Pittsburg. USA.
- **Zweigenbaum P.** (1991) Un analyseur syntaxique pour grammaires lexicales fonctionnelles. *TA Informations*, (32)2, 19-34.