

Cours sur le traitement automatique des langues (IV)

Violaine Prince
 Université de Montpellier 2
 LIRMM-CNRS

Grammaires d'unification

- Grammaire catégorielle
- Grammaire syntagmatique généralisée (GPSG)
- Les grammaires de tête (comme HPSG)
- Grammaire lexicale fonctionnelle (LFG)
- Grammaire d'unification fonctionnelle (FUG)
- Grammaire à clauses définies (DCG).
- Elles peuvent s'exprimer dans le formalisme PATR-II du Stanford Research Institute

Éléments de base

- Définition de traits et de valeurs (variable et domaine de définition).
- Structure de traits associés aux éléments syntaxiques.
 - ◆ Différentes variantes :
 - + Structure f de LFG
 - + Matrice de traits et catégories de GPSG
 - + Structure fonctionnelle dans FUG
 - + Terme (DCG)
 - + Directed acyclic graph ou DAG dans PATR-II
 - ◆ But commun : extension du formalisme HC

Les structures de traits

- Une représentation graphique, avec un nom de structure
- Type de structure : complexe, atomique
- Un chemin de traits : suite de traits permettant de sélectionner une partie spécifique d'une structure
- Relation entre structures

Représentation graphique

Elle est associée aux structures possédant plus d'un trait (structures complexes).
Exemple :

Nombre :	nom : D_{3sg}
singulier	
Personne :	
troisième	

Cette notation est récursive. Les valeurs de traits peuvent être aussi structurées.

Représentation graphique

Structure récursive intégrant la structure D_{3sg} dans une autre.
Exemple : représentation d'un SN pronom sujet.

Cat : SN	Nombre :	nom : D_{SN3sg}
	singulier	
Accord :	Personne :	
	troisième	

La structure précédente devenant un trait est nommée comme telle.

Concepts de base

- Les structures **complexes** comme D_{3sg} ou D_{SN3sg} sont des fonctions partielles des traits dans l'ensemble des valeurs.
- $D(f)$ dénotera la valeur associée au trait f de la structure D .
 - ◆ $D_{3sg}(\text{nombre}) = \text{singulier}$
- Une structure de traits peut avoir un domaine vide. Elle est appelée structure vide ou variable.

Propriétés : la réentrance

Une structure est dite **réentrante** si deux traits de cette structure ont une valeur en commun.
Représentation graphique, conditions de base :

F:	H: a	F et G ont deux valeurs égales.
	K: b	
G:	H: a	
	I: c	

Représentation de la réentrance

F: \square^1 | H: a | F et G on la même valeur.

G: \square^1

Autre représentation où la notation entre les valeurs communes est tracée explicitement.

F: | H: a |

G: |

Propriétés : chemin (de traits)

Exemple : la structure $D_{SN\ 3sg}$
 On veut sélectionner la valeur du trait « nombre » de la structure et le trait associé au trait « accord »

Cat : SN

Accord : | Nombre : singulier |

Personne : troisième | nom : $D_{SN\ 3sg}$

Chemin <Accord Nombre> dans $D_{SN\ 3sg}$ s'écrit :
 $D_{SN\ 3sg}(\langle \text{Accord Nombre} \rangle) = \text{singulier}$

Relation entre structures

Une structure comme $\text{Cat: SN} \mid D_{SN}$ est atomique.

Entre structures nommées il existe une relation dite de **subsumption**.

Une structure de traits D subsume une structure D', noté $D \subseteq D'$ si et seulement si:

- soit D et D' sont atomiques et $D=D'$
- soit $D(l) \subseteq D'(l), \forall l \in \text{dom}(D)$
- \forall les chemins p et q: $D'(p) = D'(q) \Rightarrow D(p) = D(q)$

On montrera la subsumption sur l'exemple des structures du syntagme nominal SN.

Remarque : Toute structure vide est subsumée par toute structure non vide.

D_{var} est subsumée par toutes les autres puisque : $D_{var} \subseteq D_{SN}$

$D_{SN} \subseteq D_{SNsg} \subseteq D_{SN3sg}$

Subsumption: relation d'ordre Partiel sur l'ensemble des structures.

L'unification

- Définition formelle:
 - ◆ Soit deux structures de traits D_1 et D_2 . L'unification de ces structures est définie par une structure D_3 telle que :
 - ✦ $D_1 \subseteq D_3$
 - ✦ $D_2 \subseteq D_3$
 - ✦ Si $D_1 \subseteq D_4$ et $D_2 \subseteq D_4$ alors $D_3 \subseteq D_4$.
 - ✦ Si D_3 existe alors D_1 et D_2 sont unifiables.
 - ◆ Notation : $D_3 = D_1 \cup D_2$

Exemple

$$\left[\begin{array}{l} \text{Cat : SN} \end{array} \right] \cup \left[\begin{array}{l} \text{Accord :} \\ \text{Nombre :} \\ \text{Singular} \end{array} \right] = \left[\begin{array}{l} \text{Cat : SN} \\ \text{Accord :} \\ \text{Nombre :} \\ \text{Singular} \end{array} \right]$$

propriétés

- L'unification ajoute de l'information.
- Elle est idempotente :
 - Si $D_1 \subseteq D_2$ alors $D_2 = D_1 \cup D_2$
- ▼ Nature et interprétation graphique
 - ◆ Graphe acyclique :
 - ✦ Arcs étiquetés par des noms de traits
 - ✦ Valeurs = points terminaux du graphe.

Transformation en graphe

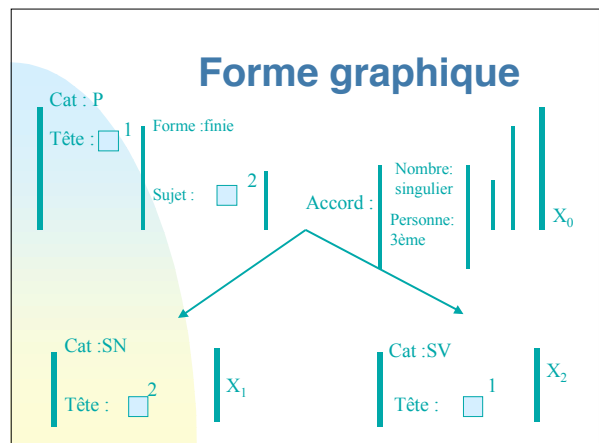
- L'unification se réduit à un traitement de graphe.
- On peut définir d'autres opérations sur les structures de traits :
 - ◆ La généralisation
 - ◆ La disjonction
 - ◆ La négation

Règles combinatoires

- La structure de traits définit l'information associée aux chaînes.
- Les règles combinatoires permettent de dériver des structures pour représenter un problème donné.
- Une règle PATR-II est définie par :
 - ◆ Une partie formelle : une règle de grammaire hors contexte associant des structures de traits
 - ◆ Une suite de conditions portant sur les structures de traits associées.

Exemple

- $X_0 \rightarrow X_1 X_2$
- Conditions :
- $\langle X_0 \text{ cat} \rangle = P$
- $\langle X_1 \text{ cat} \rangle = SN$
- $\langle X_2 \text{ cat} \rangle = SV$
- $\langle X_0 \text{ tête} \rangle = \langle X_2 \text{ tête} \rangle$
- $\langle X_0 \text{ tête sujet} \rangle = \langle X_1 \text{ tête} \rangle$



remarques

- L'application d'une règle se fait en accord avec les relations de subsumption.
- Les règles définissent également un processus d'analyse qui va utiliser le fonctionnement inverse des règles avec des relations d'identité et d'unification.
 - ◆ Les contraintes d'identité peuvent être vues comme des instructions de remplacement de sous-structures par leurs unifications. Cette opération est appelée unification par abus de langage (et relation avec l'unification logique).

Simplification des règles

- $X_0 \rightarrow X_1 X_2$
- $\langle X_0 \text{ cat} \rangle = P$
- $\langle X_1 \text{ cat} \rangle = SN$
- $\langle X_2 \text{ cat} \rangle = SV$
- $\langle X_0 \text{ tête} \rangle = \langle X_2 \text{ tête} \rangle$
- $\langle X_0 \text{ tête sujet} \rangle = \langle X_1 \text{ tête} \rangle$ devient:
- $P \rightarrow SN SV$
- $\langle P \text{ tête} \rangle = \langle SV \text{ tête} \rangle$
- $\langle P \text{ tête sujet} \rangle = \langle SN \text{ tête} \rangle$

Des règles de grammaire du LN

- Aux précédentes règles on peut rajouter :
- $SV \rightarrow V$
- $\langle SV \text{ tête} \rangle = \langle V \text{ tête} \rangle$
- La grammaire ainsi définie permet en fonction des traits de structure
 - ◆ D'accepter 'Jean dort'
 - ◆ Et de refuser 'Jean dorment'.

Mais

- L'information associée aux éléments du lexique remonte vers la racine de la structure syntaxique.
- => il faut au maximum spécifier, au niveau des structures, les comportements des éléments.
- => sous-catégorisation : représenter les comportements distincts des éléments d'une même catégorie.

Exemple

- Le comportement des verbes renverser et persuader est distinct.
- Jean renverse le pot d'Arthur.
- Jean persuade Arthur de partir.
- Dans un cas :
 - SV -> V SN
 - <SV tête> = <V tête>
- Dans l'autre
 - SV1 -> V SN SV2
 - <SV1 tête> = <V tête>
 - < SV2 tête forme> = infinitif

La sous-catégorisation

- L'idée est de dire que :
 - ◆ Chaque verbe a un trait qui s'appelle « sous-catégorie » qui va définir une contrainte supplémentaire .
 - ◆ SV -> V SN « renverser »
 - ◆ < SV tête> = < V tête>
 - ◆ < V souscat = SN>
 - ◆ Renverser prend un syntagme nominal en sous-catégorie

La sous-catégorisation

- ◆ SV1 -> V SN SV2 « persuader »
- ◆ < SV1 tête> = < V tête>
- ◆ <SV2 tête forme> = infinitif
- ◆ < V souscat = SNINF>
- ◆ L'objet de « persuader » est un SN suivi d'un infinitif.
- La sous-catégorisation permet de définir les compléments de verbe.