Generating French with the LKB

Jesse Tseng

CNRS, Loria

Nancy, France

Linguistic Engineering Group

IPIPAN, Warsaw

17 March 2006

Outline

I. grammar overview

II. Matrix for French

III. grammar modifications

IV. demo

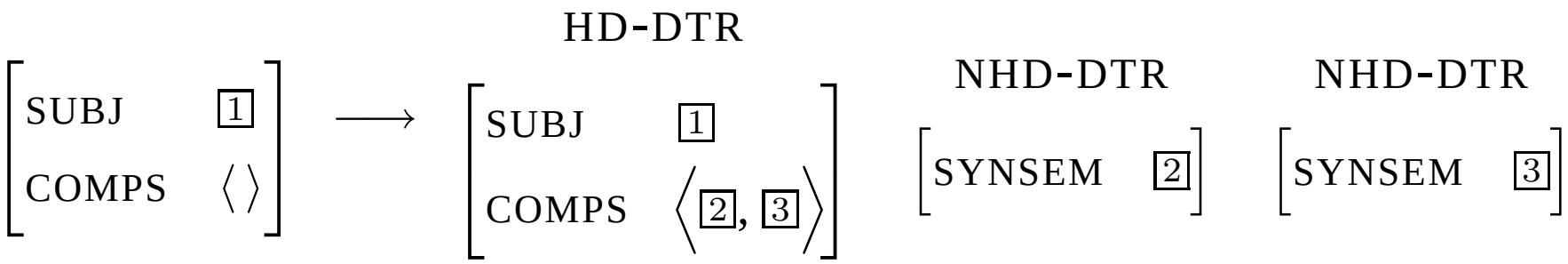
Standard HPSG: ARG-ST and VALENCE

$$\begin{bmatrix} \text{ARG-ST} & \textit{list(synsem)} \text{ (“initial subcategorization”)} \\ \text{VALENCE} & \begin{bmatrix} \text{(“surface subcategorization”)} & \\ \text{SUBJECT} & \textit{list(synsem)} \\ \text{SPECIFIER} & \textit{list(synsem)} \\ \text{COMPLEMENTS} & \textit{list(synsem)} \end{bmatrix} \end{bmatrix}$$

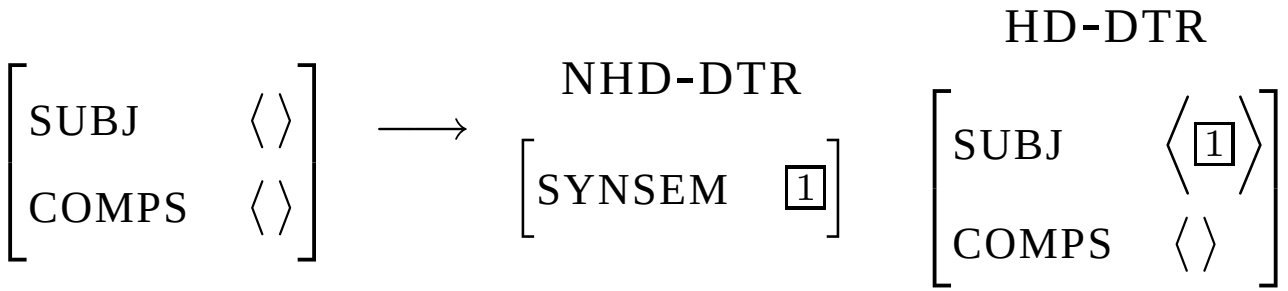
- *synsem* = collection of syntactic and semantic properties (category, bar level, semantic index, predicate)
- three subtypes: *canonical* (ordinary syntactic realization), *gap* (extracted), *affix* (clitic)
- VALENCE lists (underespecified in lexical entries) instantiated by ARG-ST to VAL mapping constraints

Valence and syntactic combination

- Head-Complements Schema



- Head-Subject Schema



Difficulties for implementation

- lack of “bottom-up” type inference and automatic type resolution
- no complex antecedents, disjunction, negation
- limited list manipulation functions (no generalized `append`, `member`, `shuffle`)

theoretical approach can be approximated in implementation but highly inefficient

- multiplication of recursive lexical rules
- massive lexical ambiguity

Modified approach to VALENCE

valence	
SUBJ	*list-of-synsems*
SPR	*list-of-synsems*
VCOMPS	*list-of-words*
DO	*list-of-synsems*
A-OBJ	*list-of-synsems*
DE-OBJ	*list-of-synsems*
XCOMP	*list-of-synsems*

ARG-ST maintained for SLASH amalgamation, etc.

“Grammatical functions”

- SUBJ, SPR: no change
- DO: NP[*acc*] ou S[*fin*], typically corresponds to an accusative clitic (*le*, *la*, *les*) or partitive *en*
- A-OBJ: NP[*dat*] corresponding to dative clitic (*lui*, *leur*) or NP[*loc*] corresponding to “locative” *y*
- DE-OBJ: NP[*de*] (clitic *en*)
- XCOMP: VP[*inf*] (no corresponding clitic!) or predicative XP (sometimes *le*)
- VCOMPS: bare V in complex predicate constructions (argument inheritance with temporal and causative auxiliaries), “lite” complements (*avoir faim*, *prendre froid*)

Related changes

- `head-comp-rule` replaced by more specific rules
  $\rightsquigarrow$ `head-do`, `head-aobj-rule`, `head-xcomp-rule`, ...
- flat structures (`head-comp-rule`, `head-2comps-rule`, `head-3comps-rule`, ...)
  $\rightsquigarrow$ (mostly) binary structures
- preverbal clitics realized syntactically (no longer in morphology)
- $\longrightarrow$ improved efficiency (many fewer edges)
- extraction, passive still analyzed by lexical rule

Aspects of the analysis

- free order among complements and adjuncts achieved with few rules, but...
- VCOMPS must be realized first
  Jean a parlé [de Marie]. (*John has talked about Mary*)
  *Jean a [de Marie] parlé.
- two items on one list ⇒ order fixed
  - VCOMPS: Jean [a fait couler] le navire. (*John has made the ship sink*)
    but *Jean [a couler fait] le navire.
  - DO: Jean informe Marie [que le navire va couler]. (*John informs Mary that the ship is going to sink.*)
    but *Jean informe [que le navire va couler] Marie.

Accommodating exceptions

- VPs generally cannot be pronominalized
  - Je promets [de faire qqch] (*I promise to do sth*)
    *J'en promets.
    Je le promets. (source: Je promets qqch. '*I promise sth*')
- Rare counterexamples
  - Je peux [faire qqch] (*I can do sth*)
    Je le peux. (but no nominal source: *Je peux qqch)
    $\Rightarrow$ add frame with clitic in DO
  - Je l'empêche [de faire qqch] (*I keep her from doing sth*)
    Je l' en empêche (but no nominal source:
    *Je l'empêche [de qqch]. '*I keep her from sth*')
    $\Rightarrow$ add frame with clitic in DE-OBJ

Problems

- more lists will be needed (e.g., OBLIQUE)
- importing lexical information from external resource (e.g., SynLex)
  - predefined valence templates (intransitive, direct transitive, subject raising with bare VP, *à*-NP controlling *de*-VP, . . .) and/or
  - predictable mapping from subcat frame elements to the appropriate valence list

Generation

- HPSG grammars are bidirectional.
- LKB grammars are (more or less) bidirectional.
- Generation is now possible with the LKB.
  - depends on a particular implementation of Minimal Recursion Semantics
  - poorly documented for the moment

The Grammar Matrix

http://www.delph-in.net/

- "universal type hierarchy" (with parameters)
- quick-start tool for new grammars
- questionnaire about simple syntactic properties of the language
  - Are there determiners? Optional or obligatory? Precede or follow N?
  - Yes/no questions: indicated by marking on V, word order, clausal marker?
  - Negation: expressed as a marking on V, modifier of VP, or operator on S?
- small number of lexical entries with semantic relations

Matrix for French

$\rightsquigarrow$ tiny but fully-formed LKB grammar capable of analyzing and generating 3-4 sentences

- `chien` (dog), `Paul`, `dort` (sleeps), `voit` (sees), `le` (masc sing def art)
- accepts: `Paul dort`, `Le chien voit Paul`
- rejects: `*Paul dort le chien`, `*Chien dort`
- "overgeneration":
  `dort le chien`, `le Paul dort`, `voit Paul le chien`

$\approx$ 2000 lines (cf. $\approx$ 5000 lines in current French grammar)

MRS “extraction” and “importation”

100% non-automatic. . .

- all necessary type declarations to introduce appropriate attributes and values for semantic representation
- encoding semantics in lexical entries
- preserving/modifying semantics in lexical rules
- determining semantic composition in syntactic combinations

. . . and ongoing

Implications for development

- new constructions require both syntactic and semantic analysis
- grammar maintenance wrt both parsing and generation
- default unification: not bidirectional
- LKB morphological component *is* bidirectional
- semantically empty elements
  - huge strain on generator if unconstrained
  - solution: trigger rules licensing the hypothesis of a particular empty element

Demo

- . . .