SYSTEM DEMONSTRATION OVERVIEW OF GBGEN[*]

Thierry Etchegoyhen
LATL
University of Geneva
etchegoyhen@latl.unige.ch

Thomas Wehrle
FPSE
University of Geneva
wehrle@fapse.unige.ch

1 Introduction

This paper presents an overview of the GBGen system, a sentence realizer currently developped for French. The system is strictly deterministic, i.e. it maps semantic structures to surface forms without either simulating parallelism or using backtracking, and the performances are accordingly extremely satisfying. It is procedural, based on Government & Binding Theory (Chomsky 1981): several levels of syntactic representation are defined, on which configurational searches and transformations apply.

GBGen is large-scale, based on a lexicon of approximately 185.000 entries (more or less 24.000 lexemes together with inflected word forms). The system covers simple and complex sentences, complex grammatical phenomena like unbounded dependencies, raising and control structures, intrasentential coreference, cliticization, modifiers (both clausal and prepositional) and main cases of coordination. It also computes several morphosyntactic phenomena like agreement, contractions or pronoun lexicalization. In what follows, we present the general characteristics of the software and detail its majors components.

2 Overview of the system

Two main components form the GBGen system. The pseudo-semantic component, which defines the semantic input of the generation process and the syntactic component, which produces a sentence (in written or spoken format) from the pseudo-semantic specifications. We describe the main aspects of these components in the following sections.

2.1 Pseudo-semantics

The input of the generation process is dubbed Pseudo-Semantics. A pseudo-semantic structure (PSS) contains both lexical and abstract information (whence the term *pseudo*). A PSS can be one of the following four semantic objects: CLS, DPS, SLS and CHS.

CLSs (clause structures) represent events and states. They contain a predicate (usually a verb or an adjective), functional information such as Tense and Aspect, and other PSS objects that participate in the interpretation of the CLS (e.g., elements bearing the thematic roles assigned by the predicate, etc.). DPSs (DP structures) semantically characterize noun phrases. They consist

[*] The GBGen project is supported by grant n^0 12-50797.97 from the Fonds National Suisse pour la Recherche Scientifique. We are grateful to the members of the LATL, especially Eric Wehrli and Christopher Laenzlinger, for comments and feedback during the development of the system. Special thanks are due to Juri Mengon, whose work since he joined the project has led to major developments of the system.

of a nominal Property along with a semantic Operator, phi-features, and a referential index used for Binding resolution. SLSs (Semantic Label Structures) consist of a semantic label/function and an associated PSS. Roughly, these objects are used to characterize thematic-role bearing elements, modifiers, or the semantic function of adverbs and adjectives. Finally, CHSs (Characteristic Structures) are used to represent adjectives and adverbs. All these elements can be combined to obtain the desired semantic representation, but can also be used autonomously (a useful characteristic for the use of pseudo-semantics for machine translation).

As an illustration, the (slightly simplified) PSS for the sentence (1a) is (1b):

(1)a. A big dog was probably killed in this bed

```
b. PSS[
     CLS[
       Mood           : real
       Tense          : E < S
       Aspect         : perfective
       Voice          : passive
       Negation       : not negated
       Clause type    : declaration
       Predicate      : kill
       Satellites     :
             SLS[  (theme)
                   DPS[ Property         : dog
                        Operator         : some individual
                        Satellites       :
                           SLS[ |set_restriction|
                CHS[ characteristic   : big ]CHS ]SLS ]DPS ]SLS

             SLS[  |eval_truth|
                   CHS[ characteristic   : probably ]CHS ]SLS

             SLS[  |in|
                   DPS[ Property         : bed
                        Operator         : demonstrative ]DPS ]SLS
     ]CLS ]PSS
```

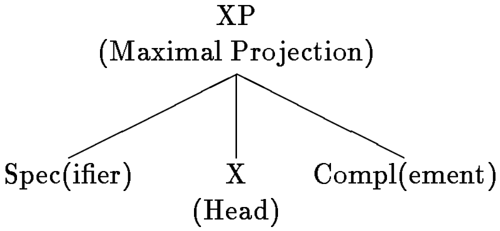
Let us briefly detail the components of the above PSS. The main object is a CLS with the predicate *kill*. Tense is represented through a modified version of Reichenbach's analysis ([Reichenbach 47]), where E is the event time point and S the speech time point, the two points being either equal or ordered with a precedence relation. Combining Tense with non-lexical aspect (progressive, perfective) leads to verbal tenses. The other functional information states that the sentence to be generated is a declarative, positive and passive one. The other elements that form part of the event are (unorderly) listed in the Satellites list. The first one is an SLS with a thematic role *Theme* and a DPS bearing this role. The DPS has a lexical Property *dog* and an Operator *some_ individual* (the interpretation of DPSs follows the generalized quantifiers analysis, see [Barwise & Cooper 81]). A CHS appears in the Satellite list of the DPS, restricting the set denotation of the property. The second SLS in the above representation contains a semantic label *Eval_ truth* and an "adverbial" CHS. The label states that the semantic function of the CHS is an evaluation of the truth of the statement expressed in the CLS. Finally, a spatial SLS is present in the Satellite list, with a spatial label *In* and a DPS with a lexical Property *bed* and an Operator *demonstrative*.

Notice, to conclude this section, that the PSSs are not syntactic in nature. They are unordered, closed class elements are abstractly represented, and recursiveness in these structures represents no more than minimal semantic scope. Hence, the efficiency of the system does not come from the fact that the input contains syntactic information, but rather from the way syntactic realization is done.

2.2 Syntactic Component

The syntactic processing has three main steps. First, we map the pseudo-semantic information into a D-structure. This is achieved by the *projection* subcomponent. Briefly, each element of the PSS which has a categorial feature (X=V,N,...) is mapped into a local tree, as in (2):

(2)



The Head/Projection distinction should be seen as a convenient presentational device. Actually, a Projection is a record of the properties of the lexical item. Thus, combination of XPs to create bigger structures can be done by using properties of heads (e.g., subcategorization). Spec and Compl are ordered lists which serve to combine all the subtrees created in the *projection* component, according to the properties of the subtrees. To give a concrete example, assuming the PSS in (1b), the system creates the D-structure in (3):

(3) [$_{\mathrm{CP}}$ [$_{\mathrm{TP}}$ {past} [$_{\mathrm{VP}}$ [$_{\mathrm{AdvP}}$ probably] [$_{\mathrm{V}}$ kill (perf.; passive)] [$_{\mathrm{DP}}$ a [$_{\mathrm{NP}}$ [$_{\mathrm{AdjP}}$ big] dog]] [$_{\mathrm{AdvP}}$ [$_{\mathrm{PP}}$ in [$_{\mathrm{DP}}$ this [$_{\mathrm{NP}}$ bed]]]]]]]

CP is the top node of each sentence and always takes a TP as its complement, which contains tense information and the subject of the clause in most cases (in the example, a passive sentence, the subject is omitted). VP contains the verb, so-called VP-adverbs in its Spec list, and complements/adjuncts in its Compl list. In our example, the latter list contains the *theme* noun phrase and an adjunct (marked with an AdvP), which is the prepositional phrase *in this bed.* Nous phrases are formed with an NP, which contains the noun, its complements and adjectives, and a DP, the projection of determiners, which subcategorizes for NPs.

Movement and Binding algorithms apply once the D-structure has been created. They merely consist in searches in the tree and the movement operation is the generic *Move* α instruction, familiar to GB practicioners. In this respect, syntactic processing in the system is configurational. Going back to our example, the object of the passive verbal form is moved to the first (Spec of) TP with finite T, leaving a coindexed empty category, and the obtained S-structure is the following one:

(4) [$_{\mathrm{CP}}$ [$_{\mathrm{TP}}$ [$_{\mathrm{DP}_i}$ a [$_{\mathrm{NP}}$ [$_{\mathrm{AdjP}}$ big] dog]] {past} [$_{\mathrm{VP}}$ [$_{\mathrm{AdvP}}$ probably] kill (perf.; passive) [$_{\mathrm{DP}_i}$ e] [$_{\mathrm{AdvP}}$ [$_{\mathrm{PP}}$ in [$_{\mathrm{DP}}$ this [$_{\mathrm{NP}}$ bed]]]]]]]

Finally, we apply the morphological procedure, which computes agreement, selects the correct verbal inflected forms, and treats other phenomena like determiner contraction or pronoun lexicalization. In our simple example, we would obtain the final sentence in (a1).

3 Concluding Remarks

GBGen is written in Modula-2, developed under Open VMS on a DEC-Alpha system, and also runs on PC-Windows. The system is being used (or will be used in the near future) in the following systems/projects:

- ITS3 - a multilingual machine translation system [Etchegoyhen & Wehrli 98]. This system uses the IPS parser ([Wehrli 92]) to parse English, French, German or Italian inputs and GBGen to generate into the target language. The French-to-French version of the system, used as a test tool for GBGen, is available on the web.[1]
- CSTAR-II speech to speech machine translation project.[2] The aim of the project is to produce on line translation of dialogs in the domain of hotel reservation and travel information. GBGen takes as input the interlingua developed for the project and produces French spoken output.
- GENE. This is the interactive version of GBGen, in which the user interactively creates pseudo-semantic inputs. The system will soon be part of the SAFRAN project ([Hamel & Wehrli 97], [Hamel & Vandeventer 98]), a toolbox for computer assisted language learning.

We presented an overview of GBGen, a large-scale domain-independent syntactic generator. At present, the system covers a large part of French grammar and deals with complex grammatical phenomena in a highly efficient way. The system is also strongly generic, which means that its extension to other languages should not require major changes in the procedures. A tentative orientation to English generation has shown that the system needs only small parametric variations in the procedures to generate major constructions of this language. Given the promising results of the approach to surface realization we have choosen, we will pursue the development of the GBGen system by extending its grammatical coverage and adding several languages to it.

References

[Barwise & Cooper 81] Barwise, J. & Cooper, R. 1981. “Generalized quantifiers and natural language”, Linguistics and Philosophy, 4 :159-219.

[Chomsky 81] Chomsky, N. 1981. *Lectures on Government and Binding*. Dordrecht, Foris, The Netherlands.

[Etchegoyhen & Wehrli 98] Etchegoyhen, T. & Wehrli, E. 1998. “Traduction automatique et structures d’interface”. *Proceedings of Traitement Automatique du Langage Naturel (TALN98)*, Paris, France.

[Hamel & Vandeventer 98] Hamel, M.-J. & Vandeventer, A. “SAFRAN-Grammaire”. To appear in the *Proceedings of the International Conference on Natural Language Processing and Industrial Applications (NLP+IA98)*, Moncton, New-Brunswick, Canada, August 1998.

[Hamel & Wehrli 97] Hamel, M.-J. & Wehrli, E. “Outils de TALN en EIAO. Le projet SAFRAN”. *Proceedings of the 1res Journées Scientifiques et Techniques (JST97)*, Avignon, France.

[Reichenbach 47] Reichenbach, H. 1947. *Elements of Symbolic Logic*, Free Press, New York.

[Wehrli 92] Wehrli, E. 1992. “The IPS System”. *Proceedings of COLING-92*, Nantes, France.

[1] At *http://latl.unige.ch/gbgen.html.* Note that the program does not make use of all the capabilities of GBGen, since not all the relevant information is at present extracted from the parse. Major syntactic constructions are nonetheless treated, and the system gives a representative picture of the generator.

[2] Information on the CSTAR-II project can be found at *http://www.is.cs.cmu.edu/cstar/CSTAR-II.html.*