A parsing system for balanced parentheses in NL texts*

Gabriel G. Bès †
Université Blaise-Pascal/GRIL

Veronica Dahl ‡
Computing Sciences Department
Simon Fraser University

Daniel Guillot §
CEDIA Consultores

Lionel Lamadon ¶
Université Blaise-Pascal/GRIL

Ioana Milutinovici ‖
Université Blaise-Pascal/GRIL

Joana.Paulo **
L2F-INESC-ID, IST-UTL

Abstract

The general architecture of the parsing system sharing in the balanced parentheses property of NL texts is presented. It consists of Modules I and II related by an interface.

1 Balanced parentheses

Many formal languages - but not all of them - use balanced parentheses: at the end of a well formed expression, $N(lp) = N(rp)$, where *N:number, lp:left parentheses, rp:right parentheses*, and, at any point of the expression, $N(lp) \leq N(rp)$.

Parentheses stipulation is not natural (Hintikka, 1994). It has been pointed out (Bès and Dahl, 2003) that, in French sentences, balanced parentheses can indeed be deduced - and not stipulated - by an adequate analysis of a subset of grammatical morphemes - the introducers - and of inflected verbal chunks. Given balanced parentheses associated to a sentence, it is possible, in its syntax/semantic interface, to point out the verb of the root sentence,'R', and to associate to the sentence a set of pairs : 'Cl' (Cl[osing pairs]), and 'C-sv' (C[oordination pairs of]sv). Thus, from (i), balanced parentheses in (ii) and the partial graph in (iii) are obtained. 'ili' (i[nitial]li[mit])and 'fp' (f[inal]p[oint]) are assumed at the beginning and the end of each sentence.

i Si la fille dit que Jacques chante et parle, c'est vrai.

ii ((ili (si la fille dit) (que Jacques (chante et parle)), c'est) vrai fp)

iii Cl = <c'est, ili>, <si, dit>, <que, parle>; C-sv = <chante, parle>; R = c'est.

2 The parsing system

The parsing system consists of Module I and Module II related by an interface. The guiding strategy underlying the whole system is to achieve the most with the least.

2.1 Module I

The general function of Module I is to point out *int[troducers]* and to specify verbal chunks, making use of very poor morphological information. The interface erases all the information not needed in Module II, leaving only tags in $V = \{ili, int, v, v1, v2, ot, fp\}$, where English *ili[initial limit], ot[ther], fp[final point]* correspond to French *li[limite initiale], au[tre], pf[point final]*, respectively. Module II specifies either the elements of the partial graph obtained by *ALGOF-C* or the *G[enerated] C[onstraints]* obtained by CHR rules, from which partial functions - not yet implemented today - are intended to specify the elements of the partial graph. The two Modules and their internal elements can be demonstrated.

The relevant features of the parsing flow of the system are summarized in the following,

*Thanks are given to Caroline Hagège for extended and enligthtening discussions in the preliminaries of this work, and to François Trouilleux for his advices on the structure of the testing device for measuring the engineering effectiveness of our linguistic hypothesis.

†Gabriel.Bes@univ-bpclermont.fr; 34 Ave. Carnot, F 63037 Clermont-Fd Cedex.

‡veronica@sfu.ca; 8888 University Dr. Burnaby B.C. V5A 1S6 Canada.

§dangui@lanet.com.ar; Chacras de Coria, Mendoza, Argentine.

¶lionellamadon@univ-bpclermont.fr; 34 Ave. Carnot, F 63037 Clermont-Fd Cedex.

‖ioana.milutinovici@univ-bpclermont.fr; 34 Ave. Carnot, F 63037 Clermont-Fd Cedex.

**Joana.Paulo@12f.inesc-id.pt; R. Alves Redol, 1000-029 Lisboa, Portugal.

with comments on (1) to (14) in Fig. 1. (i) is assumed as the basic illustration.

(i) Marie les a regardées hier. Elle juge que le projet que Jacques vienne est difficile.

[(1)] File with all and only Ascii codes.

[(2), (3), (4)] SMORPH (French S[egmentation] et MORPH[ologie]), in (2), specified and implemented in C++ by Salah Aït-Mokhtar (Aït-Mokhtar, 1998), runs in UNIX/linux platforms. In general, it segments Ascii codes strings, associates to each string one or more lemmas and features values. In (3), its information source, it is possible to declarativelly specify different kinds of Ascii codes, and relations between them (e.g. between upper and lower case letters), morphological endings, prefixes, suffixes and infixes, morphological rules and lexical entries, these pointing either to the morphological rules or having idiosyncratic behaviour.

Driving from the guiding strategy, the declarative sources of Smorph are specified in (3) with strongly poor information. For instance, *les* is not associated to two different lemmas - an article and a clitic -, which is possible to do within Smorph fonctionnalities, but to a potentially ambiguous clitic (different thus from the non ambiguous clitic *y*); by the same token, *juge* is a potentially ambiguous inflected verb, and not a verb and a noun. Potentially ambiguous introducers, as *que*, are labelled *inta*, while non-ambiguous ones, as *lorsque*, are labelled *int*. The lemmatized string in (4), associated to Ascii codes underlying *... hier. Elle juge que...* in (i) is (ii), with feature values relevant for the further discusion : *mi* for unknown forms (French *m(ot) i(nconnu)*), *pron* for pronouns, *v* for verbs, *fl* for inflected, *amv* for verbal ambiguity and *inta* for introducer ambiguity. Observe that sentences in (i) are not segmented in (ii).

```
(ii)
'hier'.
[ 'hier', mi].
'.'.
[ 'point', 'TPAS','pf'].
'Elle'.
[ 'elle', 'TFG','pron'].
'juge'.
[ 'juger', 'TVE','v', 'MOD','fl',
'TFGB','amv'].
'que'.
[ 'que', 'TPAS','inta'].
```

[(5), (6), (7)] PASMO (Portuguese Pós AnáliSe MOrfológica), in (5), specified in (Paulo et al., 2001) and implemented in C++ by Joana Paulo, runs on UNIX/Linux platforms. It obtains the string of enumerated sentences, introduces *li*s (French *limite initiale*) as the first element of each sentence, disambiguates, in terms of contextual local relations, the expressions that are potentially ambiguous, and specifies verbal chunks, labelling the inflected ones with one of the *v, v1, v2* tags of *V*, inasmuch they are preceded or not by commas and/or coordination morphemes. Three kinds of rules are in the declarative source of PASMO (i.e. (6) in Fig. 1): rules which segment the input (4) into sentences and enumerate them, rules which recompose segmented strings in (4) obtaining chunks, and rules which modify feature values in (4). At the present stage, there is no potential ambiguity in PASMO output (i.e. (7) in Fig. 1). For instance, the two *que*s of the second sentence in (i), that are *inta* in (4), become *int* in (7). The first one, because it is immediately to the right of the verb *juge*. This, in turn, is an inflected verb, and not a noun, because *Elle* to its left is in the nominative function. The second *que* becomes an *int*, because it is to the right of *projet*, which is a noun accepting embedded wh-sentences as non relative modifiers. PASMO output in (7), associated to (ii), is (iii). Observe that the unknown *hier*, i.e. *mi* in (ii), is *au* in (iii).

```
(iii)
[ F1
  [ A1
    'li', ['li','li '],
    'Marie', ['Marie','au'],
    'les a regardees', ['regarder','v'],
    'hier', ['hier','au'],
    '.', ['point','pf'],
    'xxx', ['xxx',' ']
  ] A1
] F1  %fim da frase numero 1
[ F2
  [ A1
    'li', ['li','li '],
    'Elle', ['elle','au'],
    'juge', ['juger','v'],
    'que', ['que','int'],
    'le', ['le','au'],
    'projet', ['projet','au'],
    'que', ['que','int'],
```

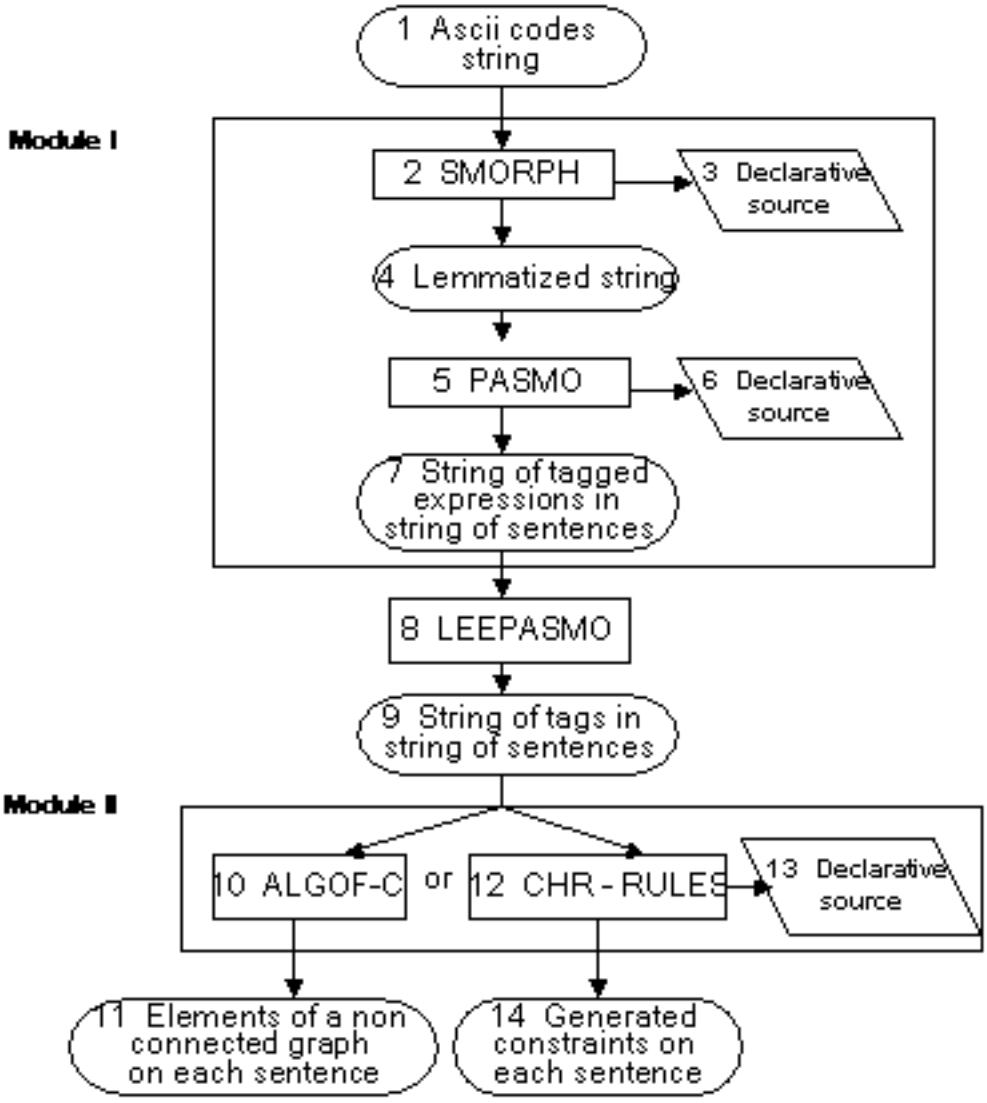

Figure 1: Parsing system architecture

```
    'Jacques', ['Jacques','au'],
    'vienne', ['venir','v'],
    'est', ['etre','v'],
    'difficile', ['difficile','au'],
    '.', ['point','pf']
  ] A1
] F2  %fim da frase numero 2
```

[(8), (9)] LEEPASMO (Spanish *Reads Pasmo*), in (8), specified and implemented by Daniel Guillot in C++, runs by the present on Windows platforms only. It erases superfluous information in (7) leaving only tags from V. Thus the complete (i) becomes (iv) in (9).

```
(iv)
E1 li au v au pf
E2 li au v int au au int au v v au pf
```

2.2 Module II

There are two ways for obtaining the partial graph interface to the semantics, both sharing in the balanced parentheses property; these are [10, 11] and [12, 13, 14] in the Module II of Fig. 1.

[(10), (11)] ALGOF-C (French Algo[rithme] de f[ermeture]-c[oordination]) is the not declarative way, specified in (Bès and Abaidi, 2003) and implemented by Daniel

Guillot in C++, which runs by the present on Windows platforms only. It obtains as output in (11) different sets of arrowing pairs, and the verb position of the root (*rac* in (v)). From (iv), ALGOF-C obtains (v) in (11).

```
(v)
<E1      ; F = [<2,0>], C-sv = [],
C-r = [], rac = 2>
<E2      ; F = [<2,0><6,8><3,9>],
C-sv = [], C-r = [], rac = 2>
```

[(12), (13), (14)] CHR-RULES is a CHR program implemented by Veronica Dahl on Sicstus Prolog, running in UNIX/Linux and Windows platforms. It is a direct implementation of a specification grammar (Bès, 2003) of type 1 in the Chomsky hierarchy in (13), which, for now, has little less expressive power than (10)[1].

CHR-RULES can directly mirror grammar rules. They generate the set *GC* (Generated Constraints), bottom up and left to right, implementing grammar rules.

The format of the input of CHR-RULES is

il ... x_1 ... x_n ... x_m ... fp

where x_i is either an *int* or an inflected nuclear verbal phrase - i.e. a verbal chunk -, tagged with a *v* or *v1* or *v2* tag, and '...' is, either a sequence of *au*'s or *e(mpty)*. The basic challenge of CHR-RULES is to specify the constraints in *GC* from which arcs in the partial graph can be obtained. The whole set *GC* is not needed for obtaining arcs. *GC* is thus the domain of partial functions specifiying pairs of the resulting graph in its range. **[(12), (13)]** obtain in (14) *G[enerated] C[onstraints]*, from which partial functions are intended to obtain the same kind of output as in (11).

3 Ongoing work

Neither all *int*s or all verbal chunks can be obtained by Module I, nor all verbal coordination of inflected forms be handled by Module II. Thus, ongoing work concernes automatic evaluation of results against manually annoted corpus. We want to obtain not only global results in terms of recall and precision, but also a detailed evaluation of them (i.e. a "glass box" evaluation). Futhermore, ongoing work relates to the improvement of the expressive power of the grammar, of ALGOF-C and of CHR-RULES program, searching two goals : the extension of the coverage of the system to verbal coordination of infinitive and participial forms in French, and to Spanish verbal coordination.

[1] The grammar and the CHR-rules program can be provided on demand.

References

Salah Aït-Mokhtar. 1998. *L'analyse présyntaxique en une seule étape.* Ph.D. thesis, Université Blaise-Pascal/GRIL, Clermont-Fd.

Gabriel G. Bès and Mohammed Abaidi. 2003. Algorithme de fermeture et de coordination pour les indicateurs dans les chaînes. Technical report, Université Blaise-Pascal/GRIL, Clermont-Fd.

Gabriel G. Bès and Veronica Dahl. 2003. Balanced parentheses in NL texts: a useful cue in the syntax/semantics interface. In *Proceedings of the Lorraine-Saarland Workshop Series*, Nancy (France).

Gabriel G. Bès. 2003. Spécification fermeture et coordination sous forme d'une grammaire de type 1. Technical report, Université Blaise-Pascal/GRIL, Clermont-Fd.

Jaako Hintikka. 1994. *Fondements d'une théorie du langage.* PUF.

Joana L. Paulo, Nuno J. Mamede, and Caroline Hagège. 2001. PAsMO - Pós-AnáliSe MOrfológica. Technical report, L2F - INESC-ID, Lisboa.