Computer Science Journal of Moldova, vol.**14**, no.2(41), 2006

Local and Global Parsing with Functional X-bar Theory and SCD Linguistic Strategy (II.)
Part II. Functional Generative Capacity, SCD Marker Classes, and Local / Global Segmentation / Parsing Algorithms

Neculai Curteanu

Abstract

This paper surveys latest developments of SCD (Segmentation-Cohesion-Dependency) linguistic strategy, with its basic components: FX-bar theory with local and (two extensions to) global structures, the hierarchy graph of SCD marker classes, and improved versions of SCD algorithms for segmentation and parsing of local and global text structures. Briefly, **Part I** brings theoretical support (predicational feature and semantic diathesis) for handing down the predication from syntactic to lexical level, introduces the new local / global FX-bar schemes (graphs) for clause-level and discourse-level, the (global extension of) dependency graph for SCD marker classes, the problem of (direct and inverse) local FX-bar projection of the verbal group (verbal complex), and the FX-bar global projections, with the special case of sub-clausal discourse segments. **Part II** discusses the implications of the functional generativity concept for local and global markers, with a novel understanding on the taxonomy of text parsing algorithms, specifies the SCD marker classes, both at clause and discourse level, and presents (variants of) SCD local and global segmentation / parsing algorithms, along with their latest running results.

Notice. This is a paper in two parts, preserving a unitary numbering of the sections, and the unitary set and system of references along both parts.

6 Functional Generativity of Local and Global Markers

The aim of Part II of the paper is to use the results of Part I for designing improved theoretical mechanisms and text segmentation / parsing techniques based on proper linguistic *marker classes* incorporated into (D)FX-bar theory and SCD linguistic strategy. We discuss the development of *segmentation* and (*dependency establishing*) *parsing algorithms*, especially for global, clause-level and discourse-level text structures, using the newly defined notions of *strong* (for lexical-level phrase markers) and *weak* (for class-level phrase markers) *functional generative capacity* [16].

A whole *class of segmentation / parsing algorithms* is described within the SCD strategy, by refining the SCD marker classes (and hierarchy) towards the lexical level of the contained markers as (weak vs. strong) *functional generativity* (using a generalization of lexical marker database in [10], both at the M3 and M4 levels of the SCD marker hierarchy [17]).

A comparison between (versions of) *SCD segmentation algorithms* [9], [13] and *Marcu*'s *segmentation algorithm* [25], [26] was realized in [14], [17], and two directions to approach the segmentation and parsing processes at the *global text structures* are explored: (G1) clause-level (syntactic) parsing, and (G2) discourse (rhetoric relation) parsing.

6.1 Functional Generativity for Classifying Parsing Algorithms

In [16], we defined the *functional generative capacity* for *phrase markers*, such as those in SCD or in Marcu's segmentation and / or parsing algorithms, as follows: when applied at *lexical level*, the phrase markers provide *strong functional generativity*; when applied at (marker) *class* *level*, they provide *weak functional generativity.* An observation is necessary: while a structure built from lexical preterminals N, V, and/or A is *strongly generated* and a sequence of lexical categories (words) is *weakly generated* in the classical sense of *categorial generativity* [22], [27], we consider *lexical markers* of *strong functional generativy* since the sharper *functional meaning* of *a lexical marker* entails (is stronger than) the *functional meaning* of a whole *class of markers.*

For example, the categorial strongly generated structure Det A N "implies" (its less informative meaning subsumes the one of) the weakly generated sequence *the beautiful flower*, and also the functional strongly generated *and*(XG_1, XG_2) implies (in the partial, reverse ordering of semantic meanings) the functional weakly generated *conjunction*(XG_1, XG_2), since the *more informative* meaning of the lexical conjunction "*and*" subsumes the inherently *less informative* meaning of the class-depending *conjunction* marker. To further support the proposed definition of functional generativity, we observe that the 'strong' lexical marker "*dacă*" (*if*) entails the 'weaker'-level phrase-marker *conjunction* (a marker class comprising several lexical conjunctions), since the information it holds is richer (e.g., in the sense of subordinate type determination) than the information held by the less informative *conjunction* class (which can only determine the subordinate, but not its type).

Thus, these definitions preserve the general entailment "*strong* implies *weak generativity*", but with the essential *difference* that while "strong categorial implies weak lexical generativity", we need a "strong *lexical marker*" to be *functionally* applied to an utterance to entail a "weak *marker class*" that is applied to the same utterance similarly.

Related to the manner in which markers or marker classes are applied to local or global text structures, the concept of *functional generativity* has immediate consequences on the FX-bar *projections* of local and global text structures, hence within the segmentation / parsing algorithms whose task is to handle the recognition / generation of these entities efficiently.

For instance, using clause markers at *lexical* level in a segmentation task entails a weaker *categorial* generativity and a higher complexity of the algorithms. The same task, worked with *classes* of clause markers, increases the expressional generativity and decreases the algorithm complexity.

At discourse level, and especially for the parsing (dependency-establishing) task, it is more profitably to use the *lexical* markers in order to obtain a stronger *functional* generativity. The usage of marker classes at this level, either for segmentation or parsing task, involves a certain degree of generality-ambiguity in determining the discourse units and rhetorical relations, resolved by the use of markers at lexical level (see [25] and the more general parsing tables proposed in Section 8.2).

In [17] we analyzed and classified several classes of local and global segmentation / parsing algorithms, based on such criteria as: **(a) categorial generative capacity** (or categorial generativity); the *strong* and *weak generativity* of major (N, V, A) preterminal and lexical categories [22], [27]; **(b) functional generative capacity** (or functional generativity); the *new concept*, introduced in [16], of *strong* and *weak functional generativity* of lexical and, respectively, classes of clause / discourse markers, as the *functional* counterpart to the corresponding notions of *categorial generative capacity* defined for major lexical categories [22], [27]; **(c) processing task**: *segmentation* or (dependency-establishing) *parsing*; **(d) the output structure targeted**: *clause* level or *discourse segment* (clause-like) level, *i.e.* local and/or global structures as outcome.

Finally, to notice that our concept of *functional generativity* differs essentially from what in [27; p.140-141] is called *derivational generative capacity*, a notion that is related to the derivation trees of functor-argument clause-type in TAGs (*e.g.* [21] and A. Joshi's previous papers on this well-known grammatical formalism). We see *derivational generativity* as a generalized form of strong categorial generativity applied to (derivation) trees instead of simpler categories (or lexical preterminals) N, V, A. The essential distinction between *categorial* (with the more general *derivational*) *generativity* and our concept of *functional generativity* is that they represent different components of the mathematical function object $f(X)$: the first concept corresponds to the argument X, while the second concept corresponds to the function name and role f.

6.2 SCD Variants for Local / Global Segmentation / Parsing

SCD algorithms are discussed at segmentation and parsing level. The segmentation is realized using inter-clausal marker classes (M3 class, see §7), to obtain finite clauses. Clause parsing is realized using lexical markers (of M3 class, described in a database which contains the information from each marker of the class, and a set of correction rules for the partial trees obtained in the first step of dependency determination).

Discourse segmentation and establishing the rhetorical relations between the discourse segments, as well as discourse tree building, is achieved using lexical markers from the M4 (discourse-level) class.

(1) In (SCD) *clause-level segmentation / parsing*, the following steps can be distinguished: SCD automated annotation (see §7.1.d and §7.2), clause segmentation, resolving the dependencies between clauses based on marker classes (superordinate – subordinate clause type) or on lexical markers (specifying the type of the subordination). **(2)** *Discourse-level parsing* can be done using marker classes of clause-like markers and structures [25], establishing of the discourse segments based on the lexical markers from the M4 level, determination of the rhetorical relations between the obtained discourse segments, based on the lexical discourse markers [25], [26], (extending lexical markers from M3 to M4-level lexical markers, Tables 8.1-8.2, §8).

Revealing the *discourse segments* is realized using clause segmentation and the lexical markers from the M3 level, while estimating the rhetorical relations between discourse segments is based on the discursive interpretation of the lexical discourse markers.

7 SCD Marker Classes and Algorithms

The SCD parsing strategy extends from *three* to *four* the representation level of marker classes, providing functions for setting the boundaries of the main syntactic structures, XG (X = N, V, A), clause, inter-clause, and discourse elementary unit (segment) (see Fig. 2.1-2.2, Part I).

The *first marker class*, denoted **M0** (or M00), is applied to the *word dictionary form*, is represented by the *functional role* of *morpho-grammatical inflection*, and corresponds to the lexical level of each word.

a) M1 Class = **{***markers delimiting (introducing) XG structures***}**.

The M1 class of markers *consists of* X1-*level markers*, (X = N, V, A), *i.e.* markers to be applied to the X1-*level syntactic constructions* (also denoted XG, and called X *groups*). These syntactic constructions consist basically of a *semantic head* (N, V, A category) surrounded by *modifiers* (adjectives or adverbs), and/or by (generalized) *quantifiers* (this includes determiners, negation, etc), *modal modifiers* of level 1-bar (*e.g.* the A1 adverb "*poate*" (*maybe*)) or 2-bar (*e.g.* the V2 modal verb "*a putea*"(*can-may*)), and/or functionally marked by *pre-positions* (in English, French, Romanian) or post-positions (in German or English) that express the *case* (for N), *aspect* or *meaning* (for V), etc. The main elements of an XG structure provide also the marker subclasses of M1. It is important to mention a certain linguistic (but not linear) *order* of these components of the XG, coming from the *distance* of these elements to the left or to the right of their semantic head, *e.g.* for the noun: the closest to the head are the modifiers, followed by quantifiers, the farthest to the head being the pre- or post-position functional particles. For VG (or verbal complex), the *predicational marking* and FX-bar *projections* are by far more elaborated operators and operations.

M1 can be split into subclasses of markers that are useful in delimiting the XG (X1) substructures, X = N, A, accordingly to criteria such as the above-mentioned *distance* to the X0 semantic head of the surrounding elements, a head which ultimately is always an (overt or covert) objectual common noun, proper noun, or personalized (no-named) noun.

M11 = {M11N, M11P}

M11N = {the occurrence of an objectual, non-predicational common noun, or of a proper noun}

M11P = {the occurrence of an accentuated or non-accentuated pronominal form}

M12 = {M12N, M12V}

M12N = {the occurrence of a noun modifier (adjective, pronominal adjective)}

M12V = {the occurrence of a verb modifier (adverb)}

M13 = {the occurrence of a (generalized) quantifier}

M14 = {pre-positions or post-positions expressing the case (for N), aspect or meaning (for V), etc.}

b) M2 Class = **{***markers that introduce a (finite or non-finite) clause, or a syntactic category group phrase with the semantic head N, V, A***}**. XG syntactic compound, (X = N, V, A), may be assimilated with a (degenerated) non-finite clause for X = N, A. M2 is split into the following subclasses (in decreasing order of priority when introducing dependency relations):

M25 = {markers that introduce the *relative clause*}.

The explanation for M25 *tag* (and its place in the dependency graph of Fig. 2.3, Part I) is that the relative clause represents the most complex syntactic compound playing the role of a modifier, to be applied to its NG head argument. The relative clause is an A2-level modifier in the FX-bar scheme, *i.e.* a modifier of 2-bar (clause) level of FX-bar projection.

M24 = {the occurrence of a *finite verbal group* (FVG) or, simply, the occurrence of the FINIte feature value assigned to a verb, introducing a *finite* VG, thus *clause*}.

The whole VG may inherit the FINIte feature value if its (predicational V) *semantic head*, or its (auxiliary V) *syntactic head* for complex tenses, bears this feature value.

M23 = {the occurrence of PREDF = PROCess non-lexical feature value assigned to any of the major categories N, V, A (since the lexicon encoding), thus introducing a clause}.

M22 = {the occurrence of the TENSe=NonFINite feature value assigned to the category V}. See [2], [28], [30], [23], [18] for various analyses of the *verbal complex*, *i.e.* VG in FX-bar terms.

M25, M24, M23 and M22 marker classes introduce X2-level structures, *viz.* finite or non-finite clauses, made up of an X1 phrase (or XG group, X = N, V, A) that represents the semantic (either finite, non-finite or predicational) head of the X2-level structure, *followed* by the corresponding NG-type (including prepositional-headed) arguments and/or adjuncts within the same clause. Some of the arguments, such as the classical case of the *grammatical subject* (or *all* the arguments, as it is possible in German), may *precede* the X1-type semantic head of the clause to which they belong [12; p.73]. Note that there exists a *systemic* (*canonical*) *order* [37] of the clause compounds, or '*actants*' (Arguments and Adjuncts) in a (finite or non-finite) clause: ACT(or), PAT(ient), ADDR(essee), ORIG(ine), LOC(ation), etc. The systemic order of the arguments within a clause is (a *theta-order*) specific to each NL, being obtained as a result of a very careful linguistic and statistic research.

M21 = {markers that introduce JOIN-type relations, *i.e. conjunctions* of the type "*and*", "*or*", "*as_well_ as*", "*together_with*"}.

M20 = {COMMA}.

Classes M21 and M20 comprise markers with an important degree of ambiguity since they may introduce any structure of type X1 (XG groups, X = N, V, A) or X2 (finite or nonfinite clauses).

c) M3 Class = {*inter-clausal (discourse) markers***}**.

The M3 class markers are functions, or relations (when correlated), having as arguments two or several finite (some of them may be non-finite) clauses. These markers are what [25], [16], [17] and other approaches are calling inter-clause, 'clause-like', or discourse markers, and apply to the X2 = CL1 syntactic projections of clause-type in the FX-bar scheme(s).

M3 may be partitioned into the following subclasses (in decreasing order of priority when introducing dependency relations):

M34 = {punctuation (pragmatic) markers that separate clauses, *e.g.* ".","!","?"...}

M33 = {inter-clausal / discourse markers that introduce (unambiguous) *strict super-ordination* clausal dependency }. *Strict super-ordination* means the effective *raising* of (*at least*) *one level* of clausal dependency, and is represented by such markers as "*then*", "*else*", etc.

M32 = {inter-clausal / discourse markers that introduce *superordination* clausal dependency, including *punctuation marks* such as *colon*, *semi-colon*, *closed parenthesis, second-paired dash*, etc.}. *Superordination* means *raising* one (or several) level(s) of clausal dependency, or remaining on the same dependency level within a *coordination*-type dependency. Typical examples of markers from M32 class are: "*but*", "*therefore* (*thus*)", "*even*", "*equally_*(*to*)", "*in_comparison_with* (*compared_to*)", etc.

M31 = {inter-clausal (discourse) markers introducing one (or several) *sub-ordination* clausal dependency level(s), including *punctuation marks* such as *open parenthesis, first-paired dash*, etc.}. This is a large class of discourse markers bearing various types of relations between clauses: logical, syntactic (of several types), semantic, pragmatic, etc.

As mentioned above, each of the M33, M32, and M31 classes may, at their turn, be partitioned into sub-subclasses that contain relational-type markers (expressed by correlation) as relations on clauses, or as functions (with at least two arguments) on clauses.

d) M4 Class = {*discourse markers, which determine the rhetorical relations that can be established between discourse segments***}**.

The *elementary discourse units* (EDU*s*, or *segments*) are identical to clauses in most of the cases, but exceptions can be found, that is, some segments can be constituted of several clauses and, remarkably, sub-clausal segments (non-finite clauses or groups different from the verbal one, but which still contain a covert predication) can also exist (see §4.2, Part I). Some of the *discourse markers* are also M3 level markers, i.e. they also have an inter-clausal relation determination role.

The same rhetorical marker can introduce several types of rhetorical relations, the disambiguisation being resolved by additional methods (statistical results, anaphora resolution, and lexical chains).

The M4 level markers can be classified accordingly to several criteria:

i) *According to the type of rhetorical relation introduced*;

The M4 level markers determine certain types of rhetorical relations, similar to those described in [24]. The number of these relations is approximately 25 in [24], the list being extended in [25]. The M4-level markers can be classified by the type of the established relations, as follows:

Antithesis: *dar, însă, cu toate acestea, ci, dacă nu, numai nu*;
Concession: *deşi, cu toate că, cel puţin*;
Detail: *în acelaşi mod, la fel cum, cât despre*;
Duration: *niciodată, încă o dată, după ce, în tot acest timp*;
Elaboration: *pe deasupra, şi încă, în acea perioadă, la care*;
Justify: *dar şi, însă, de asemenea*;
Purpose: *pentru că, ca să, fiindcă, cu scopul*;

ii). *According to the type of the units introduced*

The discourse markers can be classified after the type of the discourse segments they introduce, in: markers that introduce nucleus-type discourse units (*dar, însă, atunci, altfel, în primul rând*) and discourse markers that introduce satellite units (*chiar dacă, cu toate că, din cauza, dacă*).

iii). *According to the complexity of the introduced relations*;

Applying these criteria, the M4-level markers can introduce: **(a)** binary relations – most of the relations between the discourse segments are binary; for example, the *Elaboration* relation, introduced by markers like "*în plus*", "*pe lângă acestea*", "*de asemenea*", "*în afară de acestea*"; **(b)** n-ary relations, ($n >= 3$).

There are some rhetorical relations which can have as arguments more than two discourse segments. Among these are the *Joint* relation (introduced by markers like *"şi"*, "*sau*"), the *Contrast* relation (introduced by markers like "*dimpotrivă*", "*deşi*", "*ca şi cum*"), the *List* or *Sequence* relations.

An important aspect that has to be considered in establishing the rhetorical relations between discourse units is *marker correlation.* This is also used to establish dependencies between clauses, but at the discourse level it is essential if we want to build the discourse trees correctly.

An obvious example of *correlation* at the M4 level is the 3-uple (*dacă* S1) – (*atunci* S2) – (*altfel* S3) (*if-then-else* relation). In this case, the tree corresponding to the sentence must be built taking into consideration not only the relations between the S1, S2 and S3 segments, established on the markers, but also the relations between the markers, relations that determine the structure of the tree built from the discourse units.

Fig. 2.3 (Section 2.3, Part I) presents the hyper-graph hierarchy of the SCD marker classes. This hierarchy is considered to be valid for Romanian. Certain modifications could be necessary from a NL to another. When we situate within restrained field of Indo-European languages (such as French, English, German, possibly Russian), one can appreciate that the proposed marker classes and structures (in Fig. 2.3) remain very similar, possibly submitted to slight modifications from a particular NL to another.

7.1 SCD Segmentation / Parsing Algorithms

The SCD segmentation algorithm presented here has a *breadth-first* (or *sequential-linear*) processing form, using as input a morphologically tagged text, and obtaining the finite (and non-finite) clauses, and the XG-structures (*e.g.* [9], [13] for a *depth-first*, *recursive* version of the SCD segmentation-parsing algorithm). The XML standard is used for data representation, and the implementation of the algorithm is made in Java.

Steps of the SCD algorithm:

a) Marker recognition for local text structures (M1, M2, M3 classes)

This step is realized automatically, except for the *predicationality feature* PREDF := PROCess, which cannot be assigned in the same manner, and has to be done manually. A sample set of rules used to realize the automated SCD *annotation* is presented in [14; p.75].

M11N markers are associated to nouns; M11P markers to pronouns, M12N is being associated to adjectives and pronominal adjectives; M12V markers to adverbs; M13 subclass contains the quantifiers and negation, and M14 subclass contains prepositions and post-positions.

M20 marker is represented by comma, M21 is *“şi”* (*and*) coordinating conjunction; subclass M22 represents an occurrence of the NonFINite feature associated to verbs, M23 marker is associated to a V, N, or A that bears the *predicationality* feature PREDF := PROC, M24 marker is associated to the lexical elements tagged as verbs at a finite mode (TENSe = FINIte), and M25 subclass is assigned to relative pronouns.

M3 class markers are associated at this stage to lexical elements that have coordination, sub-ordination or super-ordination role, as well as sentence boundaries. The ‘real’ M3 markers, which can contain multiple lexical elements, as “*aşa cum*” (*such as*), “*chiar dacă*” (*even whether*) etc. are recognized in a subsequent stage.

b) Recognition and Structure of the Verbal Group Kernel [FVGIN tag]

The verbal group (VG), as XG structure in the BAR = 1 projection level of the FX-bar scheme, contains a *semantic head* verb, *around* which one can find pronouns (only in unaccentuated forms, *i.e.* clitics), special adverbs, auxiliaries, modal verbs (or adverbs), negation. VG is also better known under the label of *verbal complex* (see [28], [29], [2]), and constitutes what is traditionally called *verbal predicate* for the classical clause (proposition). The VG Kernel (VGK) was initially introduced in [17, p.175] (under the name of *default verbal kernel*), and represents a basic substructure in the VG parsing. The *typical difference* between VG and VGK is that VGK is missing the *proper adverb* of VG (that may syntactically commute with VGK to accomplish the VG).

c) Recognition of the inter-clause markers

M3 (M3n, $n = 1,\ldots,4$) and M25 class markers are recognized using the database described below (Table 8.1).

d) Text segmentation into finite clauses

Using the outcome of the precedent steps, the algorithm determines the clauses of every sentence in the text, based on the marker classes. A pseudo-code description of the algorithm is given below:

```
SCD Tagging
Input: morphological tagged text
```

Output: SCD tagged text

1.1 Recognition of the M1 class and M22 markers

Recognition of the FVGIN structure (M24 marker)

Recognition of the MRK structure (M3 class, M20, M21 and M25 markers)

Manual annotation of the M23 marker (predicational feature)

SCD meta-algorithms

2.1 Segmentation Algorithm

Input: SCD tagged text

Output: The syntactic structures like: finite-clauses, nonfinite-clauses, noun groups and verbal groups.

2.1.1 Finite-clause recognition

Input: sentence S with the SCD markers

Output: finite-clauses of the S sentence +

```
index_fvg1 := -1, index_fvg2 := -1;
index_mrk := -1;
nr_fvg = 1;
index_fvg1 := findFVG(S, nr_fvg);
while(index_fvg1 != -1)
{
   index_fvg2 := findFVG(S, nr_fvg+1);
   if(index_fvg2 != -1)
   {
        index_mrk = findMRK(index_fvg1, index_fgv2,
   “M3” OR “M25”);
        if(index_mrk != -1)
        {
            insert_boundary(index_mrk);
            continue;
        }
        else
        {
            index_mrk = findMRK(index_fvg1,
   index_fvg2, “M20” OR “M21”);
```

```
      if(index_mrk != -1)
      {
        insert_boundary(index_mrk);
        continue;
      }
      else
      {
        index_mrk = index_fvg2;
        insert_boundary(index_mrk);
      }
    }
  }
  index_fvg1 = index_fvg2;
  nr_vb ++;
}//end_while
```

7.2 Running Marcu's and SCD Algorithms

When working with SCD marker tags, *i.e.* Mpq identifiers ($p = 1 \div 4$, $q = 1 \div 5$) or X-p marker labels in Fig. 2.1-2.2 (Part I), it is necessary to transform a morphologic (or POS) *automatic tagging* into SCD *tagging.* SCD *annotation* can be performed with a small computational price, the only problem arising is the assignment of the *predicational* feature values PROC or EXIST to those major lexical categories N, V, or A that bear it (if it was not already assigned at the *lexicon level*). The *TexTag* C++ environment was developed for both the (manual) control of morphologic (POS) and SCD tagging, as well as for the automatic transformation of POS tagging into SCD annotation.

For segmentation / parsing tasks, we developed two main programs: the *ClauSEGM environment*, written in Visual C++ 6.0 [14], [17] and used to implement the Marcu's segmentation algorithm, and the *SCD-Segmentation environment*, written in Java and used to implement the SCD segmentation and parsing algorithms for Romanian sentence. Here it is the result of running an example with both programs.

Ex.7.2.1.Marcu_SEG. Marcu's (unstructured) segmentation-at-discourse algorithm within *ClauSEGM* environment (see Fig. 7.2.1)

[În toamna aceea n-a nins decât foarte târziu.]1 [Locuiam într-un *chalet* din lemn, aflat într-o pădure de pini de pe coasta unui munte şi noaptea totul îngheţa, încât dimineaţa cele două căni cu apă de pe bufet aveau o pojghiţă de gheaţă pe deasupra.]2 [Dimineaţa, devreme, Mrs. Guttingen intra în cameră]3 [ca să închidă ferestrele şi făcea focul în soba cea mare de folosinţă.]4 [Surcelele de brad pârâiau şi scoteau scântei şi focul începea]5 [să duduie în sobă.]6 [A doua oară, Mrs. Guttingen venea cu nişte butuci groşi de lemn pentru foc şi o cană cu apă fierbinte.]7

Ex.7.2.2.SCD_SEG. *SCD*-2004 (unstructured) segmentation-at-clause algorithm within *SCDSegmentation* environment

[În toamna aceea n-a nins decât foarte târziu.]1 [Locuiam într-un *chalet* din lemn, aflat într-o pădure de pini de pe coasta unui munte]2 [şi noaptea totul îngheţa,]3 [încât dimineaţa cele două căni cu apă de pe bufet aveau o pojghiţă de gheaţă pe deasupra.]4 [Dimineaţa, devreme, Mrs. Guttingen intra în cameră]5 [ca să închidă ferestrele]6 [şi făcea focul în soba cea mare de folosinţă.]7 [Surcelele de brad pârâiau]8 [şi scoteau scântei]9 [şi focul începea]10 [să duduie în sobă.]11 [A doua oară, Mrs. Guttingen venea cu nişte butuci groşi de lemn pentru foc şi o cană cu apă fierbinte.]12

Example 7.2.3. The SCD analyses, as tagged code, are as follows:

POS Tagged Input:

<TOK ID=”**TOK111**” root=”**Nu**” pv=”**Particle**” Type=”**negation**”>**Nu**</TOK> <TOK ID=”**TOK112**” root=”**avea**” pv=”**Verb**” Type=”**main**” Mood=”**indic.**” Tense=”**imperfect**” Person=”**third**” Number=”**singular**”>**avea**</TOK> <COMP ID=”**COMP4**” pv= ”**Determiner**” Person=”**third**” Gender=”**masculine**” Number=”**singular**” Quantification=”**existential**”>**nici un**</COMP> <TOK ID= ”**TOK113**” root=”**rost**” pv=”**Noun**” Type=”**common**” Gender= ”**masculine**” Number=”**singular**” Definiteness=”**no**”>**rost**</TOK> <TOK ID=”**TOK114**” root=”**sã**” pv=”**Particle**” Type=”**subjunctive**”>**sã**</TOK> <TOK ID=”**TOK115**” root=”**încerca**” pv= ”**Verb**” Type=”**main**” Mood=”**subj.**” Tense=”**present**” Person= ”**third**”>**încerce**</TOK> <TOK ID=”**TOK116**” root=”**la**” pv=

”**Adposition**” Type=”**preposition**” Formation=”**simple**”> **la** </TOK> <TOK ID=”**TOK117**” root=”**lift**” pv=”**Noun**” Type= ”**common**” Gender=”**masculine**” Number=”**singular**” Definiteness= ”**no**”>**lift**</TOK> <PTERM_P ID=”**PTERM_P4**” type=”**PERIOD**”>**.**</PTERM_P>

SCD Tagging output:

<TOK ID=”**TOK111**” root=”**Nu**” Type=”**negation**” pv=”**Particle**” mark=”**M13**”>**Nu**</TOK> <FVGIN ID=”**FVGIN_9**” mark= ”**M24**”> <TOK Type=”**main**” Mood=”**indic.**” mark=”**M24**” Number=”**singular**” ID=”**TOK112**” Person=”**third**” root=”**avea**” Tense =”**imperfect**” pv=”**Verb**”>**avea**</TOK> </FVGIN> <TOK ID= ”**COMP4**” Gender=”**masculine**” Person=”**third**” Number=”**singular**” pv=”**Determiner**” Quantification=”**existential**” mark=”**M14**”> **nici un**</TOK> <TOK ID=”**TOK113**” root=”**rost**” Gender=”**masculine**” Type=”**common**” Number=”**singular**” pv=”**Noun**” Definiteness=”**no**” mark=”**M11N**”>**rost**</TOK> <FVGIN ID= ”**FVGIN_10**” mark=”**M24**”> <TOK ID=”**TOK114**” root=”**sã**” Type=”**subjunctive**” pv=”**Particle**” mark=”**M24**”>**sã**</TOK> <TOK ID=”**TOK115**” root=”**încerca**” Person=”**third**” Type= ”**main**” pv=”**Verb**” Mood=”**subj.**” Tense=”**present**” mark= ”**M24**”>**încerce**</TOK> </FVGIN> <TOK ID=”**TOK116**” root= ”**la**” Type=”**preposition**” Formation=”**simple**” pv=”**Adposition**” mark=”**M14**”>**la**</TOK> <TOK ID=”**TOK117**” root=”**lift**” Gender=”**masculine**” Type=”**common**” Number=”**singular**” pv= ”**Noun**” Definiteness=”**no**” mark=”**M11N**”>**lift**</TOK> <MRK ID=”**MRK24**” mark=”**M34**”> <TOK ID=”**PTERM_P4**” type= ”**PERIOD**” mark=”**M34**”>**.**</TOK> </MRK>

SCD Segmentation output:

<clauza id=”**9**”> <TOK ID=”**TOK111**” root=”**Nu**” Type=”**negation**” pv=”**Particle**” mark=”**M13**”>**Nu**</TOK> <FVGIN ID= ”**FVGIN_9**” mark=”**M24**”> <TOK Type=”**main**” Mood=”**indic.**” mark=”**M24**” Number=”**singular**” ID=”**TOK112**” Person=”**third**” root=”**avea**” Tense=”**imperfect**” pv=”**Verb**”>**avea**</TOK> </FVGIN> <TOK ID=”**COMP4**” Gender=”**masculine**” Person= ”**third**” Number=”**singular**” pv=”**Determiner**” Quantification=”**ex-**

istential” mark=”**M14**”>**nici un**</TOK> <TOK ID=”**TOK113**” root=”**rost**” Gender=”**masculine**” Type=”**common**” Number=”**singular**” pv=”**Noun**” Definiteness=”**no**” mark=”**M11N**”>**rost** </TOK> </clauza> <clauza id=”**10**”> <FVGIN ID=”**FVGIN_10**” mark=”**M24**”> <TOK ID=”**TOK114**” root=”**sã**” Type=”**subjunctive**” pv=”**Particle**” mark=”**M24**”>**sã**</TOK> <TOK ID= ”**TOK115**” root=”**încerca**” Person=”**third**” Type=”**main**” pv= ”**Verb**” Mood=”**subj.**” Tense=”**present**” mark=”**M24**”>**încerce** </TOK> </FVGIN> <TOK ID=”**TOK116**” root=”**la**” Type=”**preposition**” Formation=”**simple**” pv=”**Adposition**” mark=”**M14**”>**la** </TOK> ¡TOK ID=”**TOK117**” root=”**lift**” Gender=”**masculine**” Type=”**common**” Number=”**singular**” pv=”**Noun**” Definiteness= ”**no**” mark=”**M11N**”>**lift**</TOK> <MRK ID=”**MRK24**” mark= ”**M34**”> <TOK ID=”**PTERM_P4**” type=”**PERIOD**” mark= ”**M34**”>**.**</TOK>

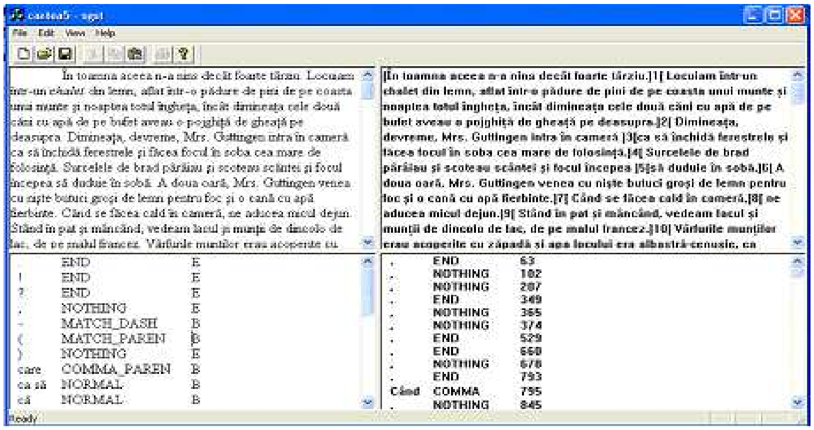
</MRK> </clauza>

SCD Parsing output:

<clauza id=”**9**”> <FVG id=”**9**”> <TOK ID=”**TOK111**” root= ”**Nu**” Type=”**negation**” pv=”**Particle**” mark=”**M13**”>**Nu**</TOK> <FVGIN ID=”**FVGIN_9**” mark=”**M24**”> <TOK Type=”**main**” Mood=”**indic.**” mark=”**M24**” Number=”**singular**” ID=”**TOK112**” Person=”**third**” root=”**avea**” Tense=”**imperfect**” pv=”**Verb**”>**avea** </TOK> </FVGIN> </FVG> <NG id=”**39**”> <TOK ID= ”**COMP4**” Gender=”**masculine**” Person=”**third**” Number=”**singular**” pv=”**Determiner**” Quantification=”**existential**” mark=”**M14**”> **nici un**</TOK> <TOK ID=”**TOK113**” root=”**rost**” Gender=”**masculine**” Type=”**common**” Number=”**singular**” pv=”**Noun**” Definiteness=”**no**” mark=”**M11N**”>**rost**</TOK> </NG> </clauza> <clauza id=”**10**”> <FVG id=”**10**”> <FVGIN ID=”**FVGIN_10**” mark=”**M24**”> <TOK ID=”**TOK114**” root=”**sã**” Type=”**subjunctive**” pv=”**Particle**” mark=”**M24**”>**sã**</TOK> <TOK ID= ”**TOK115**” root=”**încerca**” Person=”**third**” Type=”**main**” pv= ”**Verb**” Mood=”**subj.**” Tense=”**present**” mark=”**M24**”>**încerce** </TOK> </FVGIN> </FVG> <NG id=”**40**”> <TOK ID= ”**TOK116**” root=”**la**” Type=”**preposition**” Formation=”**simple**”

pv="**Adposition**" mark="**M14**">**la**</TOK> <TOK ID="**TOK117**" root="**lift**" Gender="**masculine**" Type="**common**" Number="**singular**" pv="**Noun**" Definiteness="**no**" mark="**M11N**">**lift**</TOK> </NG> <MRK ID="**MRK24**" mark="**M34**"> <TOK ID="**PTERM_P4**" type="**PERIOD**" mark="**M34**">**.**</TOK> </MRK> </clauza>

Example 7.2.4. Here there are *two snapshots* of running the *environments: ClauSEGM* and *SCDSegmentation*.



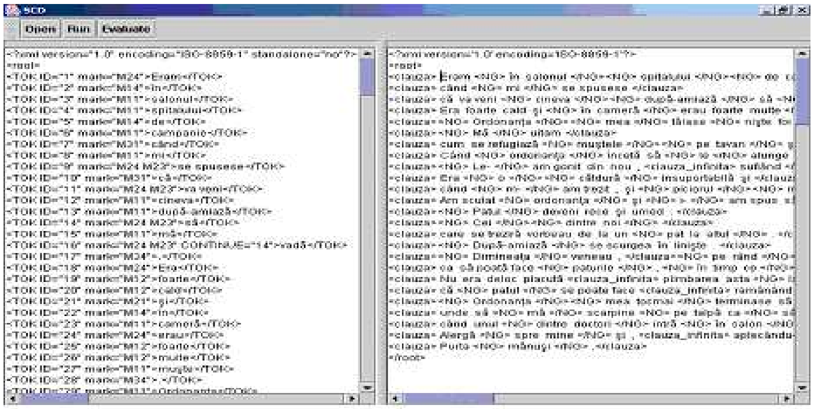
```
În toamna aceea n-a nins decât foarte târziu. Locuiam
într-un chalet din lemn, aflat într-o pădure de pini de pe coasta
unui munte şi noaptea totul îngheţa, încât dimineaţa cele două
căni cu apă de pe bufet aveau o pojghiţă de gheaţă pe
deasupra. Dimineaţa, devreme, Mrs. Guttingen intra în cameră
ca să închidă ferestrele şi făcea focul în soba cea mare de
folosinţă. Surcelele de brad pârâiau şi scoteau scântei şi focul
începea să duduie în sobă. A doua oară, Mrs. Guttingen venea
cu nişte butuci groşi de lemn pentru foc şi o cană cu apă
fierbinte. Când se făcea cald în cameră, ne aducea micul dejun.
Stând în pat şi mâncând, vedeam lacul şi munţii de dincolo de
lac, de pe malul francez. Vârfurile munţilor erau acoperite cu
```

```
.       END            E
!       END            E
?       END            E
,       NOTHING        E
-       MATCH_DASH     B
(       MATCH_PAREN    B
)       NOTHING        E
care    COMMA_PAREN    B
ca să   NORMAL         B
că      NORMAL         B
```

```
[În toamna aceea n-a nins decât foarte târziu.]1[ Locuiam într-un
chalet din lemn, aflat într-o pădure de pini de pe coasta unui munte şi
noaptea totul îngheţa, încât dimineaţa cele două căni cu apă de pe
bufet aveau o pojghiţă de gheaţă pe deasupra.]2[ Dimineaţa,
devreme, Mrs. Guttingen intra în cameră ]3[ca să închidă ferestrele şi
făcea focul în soba cea mare de folosinţă.]4[ Surcelele de brad
pârâiau şi scoteau scântei şi focul începea ]5[să duduie în sobă.]6[ A
doua oară, Mrs. Guttingen venea cu nişte butuci groşi de lemn pentru
foc şi o cană cu apă fierbinte.]7[ Când se făcea cald în cameră,]8[ ne
aducea micul dejun.]9[ Stând în pat şi mâncând, vedeam lacul şi
munţii de dincolo de lac, de pe malul francez.]10[ Vârfurile munţilor
erau acoperite cu zăpadă şi apa lacului era albastră-cenuşie, ca
```

```
.       END        63
.       NOTHING    102
.       NOTHING    207
.       END        349
.       NOTHING    365
.       NOTHING    374
.       END        529
.       END        660
.       NOTHING    678
.       END        793
Când    COMMA      795
.       NOTHING    845
```

Figure 7.2.1. *ClauSEGM* run with Marcu's algorithm



```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<root>
<TOK ID="1" mark="M24">Eram</TOK>
<TOK ID="2" mark="M14">în</TOK>
<TOK ID="3" mark="M11">salonul</TOK>
<TOK ID="4" mark="M11">spitalului</TOK>
<TOK ID="5" mark="M14">de</TOK>
<TOK ID="6" mark="M11">campanie</TOK>
<TOK ID="7" mark="M31">când</TOK>
<TOK ID="8" mark="M11">mi</TOK>
<TOK ID="9" mark="M24 M22">se spusese</TOK>
<TOK ID="10" mark="M31">că</TOK>
<TOK ID="11" mark="M24 M23">va veni</TOK>
<TOK ID="12" mark="M11">cineva</TOK>
<TOK ID="13" mark="M11">după-amiază</TOK>
<TOK ID="14" mark="M24 M22">să</TOK>
<TOK ID="15" mark="M11">mă</TOK>
<TOK ID="16" mark="M24 M23" CONTINUE="14">vadă</TOK>
<TOK ID="17" mark="M34">.</TOK>
<TOK ID="18" mark="M24">Era</TOK>
<TOK ID="19" mark="M12">foarte</TOK>
<TOK ID="20" mark="M12">cald</TOK>
<TOK ID="21" mark="M21">şi</TOK>
<TOK ID="22" mark="M14">în</TOK>
<TOK ID="23" mark="M11">cameră</TOK>
<TOK ID="24" mark="M24">erau</TOK>
<TOK ID="25" mark="M12">foarte</TOK>
<TOK ID="26" mark="M12">multe</TOK>
<TOK ID="27" mark="M11">muşte</TOK>
<TOK ID="28" mark="M34">.</TOK>
<TOK ID="29" mark="M11">Ordonanţa</TOK>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<root>
<clauza> Eram <NG> în salonul </NG><NG> spitalului </NG><NG> de c
<clauza> când <NG> mi </NG> se spusese </clauza>
<clauza> că va veni <NG> cineva </NG><NG> după-amiază </NG> să <N
<clauza> Era foarte cald şi <NG> în cameră </NG> erau foarte multe <
<clauza><NG> Ordonanţa </NG><NG> mea </NG> lăsase <NG> nişte fo
<clauza><NG> Mă </NG> uitam </clauza>
<clauza> cum se refugiază <NG> muştele </NG><NG> pe tavan </NG> ş
<clauza> Când <NG> ordonanţa </NG> încetă să <NG> le </NG> alunge
<clauza><NG> Le- </NG> am gonit din nou , <clauza_infinita> suflând <
<clauza> Era <NG> o </NG><NG> căldură </NG> insuportabilă şi </clauz
<clauza> când <NG> m- </NG> am trezit , şi <NG> piciorul </NG><NG> m
<clauza> Am sculat <NG> ordonanţa </NG> şi <NG> > </NG> am spus s
<clauza><NG> Patul </NG> deveni rece şi umed . </clauza>
<clauza><NG> Cei </NG><NG> dintre noi </NG> </clauza>
<clauza> care se treziră vorbeau de la un <NG> pat la altul </NG> . </c
<clauza><NG> După-amiază </NG> se scurgea în linişte . </clauza>
<clauza><NG> Dimineaţa </NG> veneau , </clauza><NG> pe rând </NG>
<clauza> ca să poată face <NG> paturile </NG> , <NG> în timp ce </NG>
<clauza> Nu era deloc plăcută <clauza_infinita> plimbarea asta <NG> î
<clauza> că <NG> patul </NG> se poate face <clauza_infinita> rămânând
<clauza><NG> Ordonanţa </NG><NG> mea </NG> terminase să
<clauza> unde să <NG> mă </NG> scarpine <NG> pe talpă ca </NG> să
<clauza> când unul <NG> dintre doctori </NG> intră <NG> în salon </NG
<clauza> Alergă <NG> spre mine </NG> şi , <clauza_infinita> aplecându-
<clauza> Purta <NG> mănuşi </NG> . </clauza>
</root>
```

Figure 7.2.2. *SCDSegmentation* run with *SCD*-2004 algorithm

8 SCD Global Parsing Algorithms

8.1 SCD Clause-Level Global Parsing

The following step *after* the SCD *segmentation* is to use the resulted clauses for establishing the dependencies between them. In order to do this, there are two working possibilities: **(a)** dependency determination using the same marker classes as in the segmentation algorithm; **(b)** dependency resolving using lexical markers.

Both cases are using the information from a database containing, for each marker, the class to which it belongs, the type of the introduced relation (subordination, coordination, super-ordination), the name of the relation (Relative, Conditional etc.), the succession of the clauses in that relation, the place and distance to the clause of the related current marker, as well as information about the markers that correlate to the current marker (position and correlation distance).

For each sentence in the text, divided into clause, the marker sequence is being processed in several steps.

In the *first step* we establish a partial tree of the dependency relations using the information in the database described above.

In the *next step*, we use a set of rules in order to correct and complete the tree obtained in the previous step, by processing the whole marker sequence, corresponding to the current sentence.

In order to determine the dependencies between clauses, we must take under consideration some general cases of resolving, using the M3 level lexical markers.

The same operations will be used for determination of the rhetorical relations, in a subsequent step, the difference being determined only by the nature of the discourse units and the structure of the marker database.

The general case considers a clausal unit (M*i*, C*i*) (Marker, Clause) which contains a marker that can be found, with the information necessary to establish its super-ordinate, in the database.

i) Marker Correlation Processing

In the case of *marker correlation*, in order to ascertain the dependencies, the problem can be solved by structuring the information from

Table 8.1. SCD M3 Lexical/Class Marker Database for Clause-level

Mar-ker	**Class**	**Type**	**Rela-tion**	**Succ**	**Wh2-Lnk**	**Dist**	**Corre-late**	**Dist-2Corr**	**Wh-2C**	**Disc role**
Care	M25	Sub	Atr	R S	b(efore)	1	-			no
Dacă	M31	Sub Sub	Cond	R S	b	1	atunci altfel	2 1	b a	yes
Altfel	M33	Sub Coord	Cond	R R	b	2	dacă atunci	1 2	b a	yes
Şi	M21	Coord		R R	b	1	şi	1	b	no
a căror	M25	Sub	Atr	R S	b	1	-			no
aşa încât	M31 M12V	Sub	Cons	R S	b	1				yes
Aşa cum	M31 M12V	Sub	Mod	S R	a	1	tot aşa	2	b	yes
Acolo unde	M31	Sub	Loc	R S	b	1				yes
Cât	M31 M12V	Sub	Nspec		a	1				yes
Fie	M33	Coord	Coord	R R			fie	1	a	yes
Nici	M33	Coord	Coord	R R			nici	1	a	yes

correlated markers as in the database described above.

Thus, besides the information necessary to establish the dependencies in the general case, (M*i*, C*i*) (M*j* C*j*), the database contains several fields referring to the markers that correlate to the current one, namely the *lexical* marker(s), the correlation *distance* (number of marker units to correlate over), the correlation *direction* (before or after the current unit).

For instance, the necessary facts referring to correlation in case of "*dacă*" (*if*) marker are: the markers that *dacă*" is correlated with (*atunci*, *altfel*) (*then*, *else*), the correlation distance (2, 1), and the correlation direction (Before, After) respectively, for each of the markers that correlate with "*dacă*" (*if*).

dacă	Sub	atunci	2	b	yes
	Sub	altfel	1	a	

In case at some parsing point one encounters the "*dacă*" (*if*) marker, the unit that contains it (Unit i) will be correlated in the *Sub*(*ordination*) relation to the $(i-2)$ Unit, which contains the "*atunci*" (*then*) marker, and to $(i+1)$ Unit, which contains the "*altfel*" (*else*) marker, as well in the *Sub* relation. In the same manner is solved the coordination, which represents a particular case of correlation.

Examples of correlation: *dacă...atunci...altfel*; *deşi...totuşi*;

Examples of coordination: *Fie...fie*; *Nici ...nici*; *Sau...sau*; *Ori ...ori*; *Nici ...ci*; *ci...şi...ci...* ; *ori...însă...ori*; *şi...dar ...şi.*

ii) Marker Sequence Processing

In order to establish the dependencies at the inter-clausal level, the marker sequences are treated as follows: any sequence of two or more markers is divided in other two subsequences, until one obtains sequences of two markers, treated analogously to the general case.

Therefore, for sequences like (Mi Ci) (Mj Cj), if the super-ordinate (regent) clause of Ci is established based on the information from database about the Mi marker, then the regent of Cj is established based on the information about the Mj marker.

For sequences like (Mi (Mj Cj) Ci), the regent clause of Ci is established according to the data about Mi, and Cj will have as regent the unit that includes it, that is Ci, rule specified in the database, at the Mj information.

If marker classes are utilized, the type of the relation (subordination, coordination, super-ordination) can be determined as well, but with a smaller precision, without being able to specify exactly the name of the relation, like using lexical markers. In this case, some important advantages would provide the increased processing speed and the reduced dimension of the database and dependency establishing rules.

After building the tree from the local relations (that can be found in the database), turn back to the obtained tree to correct the existing relations and to add some new ones, resulted as supplementary information from processing the whole sequence of markers ascribed to the entire sentence (e.g. verification of the last XG in the previous clause could give additional information about the type of the currently processed clause). Also, a set of rules is applied to the firstly resulted dependency tree.

Examples of rules:

Rule 1: "*dacă*" marker can correlate to a comma in the absence of "*atunci*".

Rule 2: If after a "*dacă*" follows a "*să*", "*care*", "*ca*" marker, the clause that contains "*dacă*" correlates to one (usually the first) after those clauses that contain the respective markers.

Rule 3: If there is an "*atunci*", but not preceded by "*dacă*", then one can correlate the "*atunci*" marker with "*când*", if any.

Rule 4: Repetition of the same marker (preceded or not by comma or "*şi*") represents a correlation.

8.2 SCD Discourse-Level Global Parsing

A type of parsing that presents a great deal of interest is *discourse parsing*. As noticed in Part I of this paper, discourse parsing can be done in various ways. The method proposed by the SCD strategy includes building the discourse tree using M4-level lexical markers, while discourse segments being obtained by clause parsing.

Using the results of the SCD clausal parsing and a database which contains information about the discourse markers, one can obtain the discourse structure of a text. The result is represented as a *discourse tree* whose terminal nodes are (almost always) clauses, having specified on the arcs the name of the involved rhetorical relations.

Discourse parsing can be done at sentence level [38], as well as between larger spans of text, like between sentences, paragraphs or sections of a text. For each type of text unit, separate sets of markers and rules are determined and applied.

Table 8.2. SCD M4 Lexical Marker Database for Discourse

Marker	**Rhet-Rel**	**Succ**	**Wh2L**	**Dist**	**Correlation**	**DisTo-Corr**	**Wh2C**
dacă	Cond	N S	B	1	atunci altfel	2 1	B A
altfel	Cond	N N	B	1	dac atunci	2 1	b b
în primul rând	Enum	N N	A	1	în al doilea rând	1	a
în al doilea rând	Enum Enum	N N N N	a b	1 1	în al treilea rând în primul rând	1 1	a b
în al treilea rând	Enum Enum	N N N N	a b	1 1	în al patru-lea rând în al doilea rând	1 1	a b

9 Evaluations and Conclusions

Part II of the paper outlines SCD variants of the segmentation/parsing algorithms, considering *criteria* of lexical and class levels of phrase markers, in the context of a general hierarchy of marker classes proposed for SCD (Fig. 2.3, Part I). The *newly defined* notions of strong and weak *functional generative capacity* are introduced for the lexical-level and class-level phrase markers, respectively.

In [14] and [17], SCD segmentation and parsing at the clause level are compared to several clause or clause-like segmentation and parsing algorithms such as those exposed in [25], [36], [31], [6] dealing with Romanian or English text segmentation / parsing. E.g., Marcu's clause-like (actually, discourse) segmentation is proved to be "embedded" into SCD (finite) clause segmentation, except the special situation of *subclausal discourse segments* [14]. Applied on several hundreds of (both Romanian and English) sentences, the implementation of [25] clause-like segmentation algorithm has a precision of 73% and a recall of 69% for Romanian, and a precision and recall of approx. 95% for English (Hemingway's "Farewell to Arms"). [36] and [31] provide similar segmentation methods for Romanian, respectively English, using lexical word-patterns around the clause boundary, obtaining 89.03% precision and 88.51% recall for Romanian, similar results for English, and increasing the precision / recall rate to approx. 95% when applied machine learning to the segmentation process.

For the current Java implementation of the SCD segmentation / parsing algorithm [17], tests have been effectuated from the whole "1984" (George Orwell) corpus, representing approximately 6.500 sentences and 15.000 clauses. The number of correctly recognized clauses rises up to 14.500, which means a precision of 96,6% and a recall of 95%. Comparing to the previous version of the parser, [17] obtained an improvement of approx. 10%, the previous test being done on 1500 sentences from the whole corpus.

In taking the tests on the SCD segmentation algorithm, several problems were identified, which led to an incorrect parsing of a number of approx. 500 sentences from the 15.000. The 3.4% error rate is due to some problems of lexical nature, to the lack of proper punctuation marks. Also, parsing errors can be found in the case of imbricate clauses, where there are no key words to indicate the continuation of a clause previously opened.

As a continuation of current topics, a *class of parsing algorithms* (at clausal and discourse level) is proposed, which can be described within the SCD strategy by refining the SCD marker classes (and hierarchy) towards the *lexical level* of the contained markers. A *generalization* of Marcu's *lexical marker database* was defined, both at the M3 and M4 classes of SCD marker hierarchy (Tables 8.1 and 8.2), which can be used in (inter-clausal and discourse rhetorical) parsing algorithms.

References

[1] Barbu, Ana-Maria; Emil Ionescu (1996): *Contemporary grammatical theories*: *grammars of the phrasal head.* in Limba româna, no. 45, (1-6) pp. 31–55 (in Romanian).

[2] Barbu, Ana-Maria (1999): *The Verbal Complex.* Studii si Cercetari Lingvistice, L, no.1, Bucureşti, pp. 39–84 (In Romanian).

[3] Chomsky, Noam (1981): *Lectures on Government and Binding.* Foris, Dordrecht.

[4] Chomsky, Noam (1986): *Barriers.* The MIT Press, Cambridge.

[5] Chomsky, Noam (1995): *The Minimalist Program.* The MIT Press, Cambridge, Massachusetts.

[6] Cristea, D., O. Postolache, I. Pistol (2005): *Summarisation through Discourse Structure.* In Proceedings of CiCling 2005, Springer LNSC, vol. 3406.

[7] Curteanu, Neculai (1988): *Augmented X-bar Schemes.* COLING'88 Proceedings, Budapest, pp. 130–132.

[8] Curteanu, Neculai (1990): *A Marker-Hierarchy-based Approach Supporting the SCD Parsing Strategy.* Research Report no. 18, Institute of Technical Cybernetics, Bratislava, Slovak Republik.

[9] Curteanu, Neculai (1994): *From Morphology to Discourse Through Marker Structures in the SCD Parsing Strategy. A Marker-Hierarchy Based Approach.* Language and Cybernetics, Akademia Libroservo, Prague, pp. 61–73.

[10] Curteanu, Neculai; G. Holban (1996): *SCD Linguistic Strategy Applied to the Analysis and Generation of Romanian.* In (Dan Tufi, Ed.) Language and Technology, Romanian Academy, Bucharest, pp. 169–176 (in Romanian).

[11] Curteanu, Neculai (2002): *Elements of a Functional X-bar Theory Within the SCD Linguistic Strategy*, ECIT2002 Conference, Iaşi, România.

[12] Curteanu, Neculai (2003): *Towards a Functional X-bar Theory.* In the volume "The Romanian Language in the Informational Society", Dan Tufiş, F. Filip (Eds.), Edited by the Romanian Academy, Research Institute for Artificial Intelligence, Bucharest, pp. 51–86 (in Romanian).

[13] Curteanu, Neculai; D. Gâlea; C. Linteş (2003): *Segmentation Algorithms for Clause-Type Textual Units.* In the volume "The Romanian Language in the Informational Society", Dan Tufiş, F. Filip (Eds.), Edited by the Romanian Academy, Research Institute for Artificial Intelligence, Bucharest, pp. 165–190 (in Romanian).

[14] Curteanu, Neculai; D. Gâlea; C. Butnariu; C. Bolea (2004): *Marcu's Clause-like Discourse Segmentation Algorithm and SCD Clause Segmentation-based Parsing*, In the volume "Intelligent Systems" (Ed. Horia-Nicolai Teodorescu). Selected Papers from ECIT2004 Conference, Iaşi, România, pp. 59–86.

[15] Curteanu, Neculai (2003-2004): *Contrastive Meanings of the Terms* "Predicative" *and* "Predicational" *in Various Linguistic Theories* (I, II). Computer Science Journal of Moldova (R. Moldova), Vol. 11, No. 4, 2003 (I); Vol. 12, No. 1, 2004 (II).

[16] Curteanu, Neculai (2005): *Functional FXbar Theory Extended to Discourse (Rhetorical) Structures.* In 'Intelligent Systems' Conference Volume, H.-N. Teodorescu *et al.* (Editors), Performantica Press, Iaşi (Romania), pp. 169–182.

[17] Curteanu, Neculai; E. Zlavog; C. Bolea (2005): *Sentence-Based and Discourse Segmentation / Parsing with SCD Linguistic Strategy.* In 'Intelligent Systems' Conference Volume, H.-N. Teodorescu *et al.* (Editors), Performantica Press, Iaşi (Romania), pp. 153–168.

[18] Curteanu, Neculai; Diana Trandabăţ (2006): *Functional (F)X-bar Projections for Local and Global Text Structures. The Anatomy of Predication.* Revue Roumaine de Linguistique, Bucharest (to appear).

[19] Dobrovie-Sorin, Carmen (1994): *The syntax of Romanian. Comparative Studies.* Berlin: Mouton de Gruyter.

[20] Irimia, Dumitru (1997): *The Morphosyntax of the Romanian Verb.* The Editorial House of the "Al. I. Cuza" Iaşi University (in Romanian).

[21] Joshi, Aravind K. and Ives Schabes (1997): *Tree Adjoining Grammars.* In "Handbook of Formal Languages and Automata" (A. Salomaa *et al.*, Eds.), Vol. 3, Heidelberg, Springer-Verlag.

[22] Kornai, András, Geoffrey Pullum (1990): *The X-bar Theory of Phrase Structure*, Language, Vol. 66, No. 1, pp. 24–50.

[23] Legendre, Géraldine (1999): *Optimal Romamian clitics: a cross-linguistic perspective.* In: V. Motapanyane (Ed.) Comparative Studies in Romanian Syntax. HAG, The Hague.

[24] Mann, William, Sandra Thompson (1988): *Rhetorical Structure Theory: A Theory of Text Organization.* Research Report RS-87-190, Information Sciences Institute, University of Southern California, Marina del Rey, California, 80 p.

[25] Marcu, Daniel (1997): *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*, Ph.D. Thesis, Univ. of Toronto, Canada, pp. 331.

[26] Marcu, Daniel (2000): *The Theory and Practice of Discourse Parsing and Summarization.* The MIT Press.

[27] Miller, Philip (1999): *Strong Generative Capacity. The Semantics of Linguistic Formalism.* CSLI Publications, Stanford, California, 1999.

[28] Monachesi, Paola (1998): *The Morphosyntax of Romanian Cliticization.* In: P.-A. Coppen *et al.* (Eds.), Proceedings of Computational Linguistics in The Netherlands 1997, pp. 99–118, Amsterdam-Atlanta: Rodopi.

[29] Monachesi, Paola (2000): *Clitic placement in the Romanian verbal complex.* In: B. Gerlach and J. Grijzenhout (eds.) Clitics in phonology, morphology, and syntax. Linguistik Aktuell. John Benjamins. Amsterdam.

[30] Monachesi, Paola (2005): *The Verbal Complex in Romance. A Case Study in Grammatical Interfaces.* Oxford University Press, Oxford Studies in Theoretical Linguistics.

[31] Orasan C.(2000): A hybrid method for clause splitting in unrestricted English texts Available at: http://www.wlv.ac.uk/sles/compling/papers/orasan-00.pdf

[32] Passonneau, Rebecca; Diane Litman (1997): *Intention-based segmentation: human reliability and correlation with linguistic cues*, in Proc. 31th Annual Meeting of ACL, Ohio, pp. 148–155.

[33] Gheorghe Păun: *Marcus Contextual Grammars.* Kluwer Academic Publishers, Dordrecht, 1997.

[34] Pollard, Carl; Ivan Sag (1994): *Head-Driven Phrase Structure Grammar.* The University of Chicago Press, Chicago & London.

[35] Popârda, O.; N. Curteanu (2002): *L'évolution du discours juridique français analysé par la stratégie linguistique SCD.* In the volume "Representations du Sens Linguistique", Lagorgette, P. Larrivée (Eds.), LINCOM Europa, series Studies in Theoretical Linguistics, München, Germany, pp. 487–502.

[36] Puşcaşu, G. (2003): *Elementary discourse unit segmentation.* Dissertation thesis. "Al.I.Cuza" University of Iaşi.

[37] Sgall, Petr; E. Hajičova, J. Panevova (1986): *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects.* Kluwer Academic Publishers, Dordrecht.

[38] Soricut, R. and Daniel Marcu (2003): Sentence Level Discourse Parsing using Syntactic and Lexical Information. *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, May 27-June 1, Edmonton, Canada.

N. Curteanu,

Received March 6, 2006

Institute for Computer Science,
Romanian Academy, Iaşi Branch
B-dul Carol I, nr. 22A, 6600 IAŞI, ROMÂNIA
E–mail: *ncurteanu@yahoo.com* and *curteanu@iit.tuiasi.ro*