Computer Science Journal of Moldova, vol.**8**, no.2(23), 2000

A set-theoretic approach to linguistic feature structures and unification algorithms (I)

N. Curteanu P.-G. Holban

Abstract

The paper proposes formal inductive definitions for linguistic feature structures (FSs) taking values within a class of value types or sorts: single, disjunctive, (ordered) lists, multisets (or bags), **po**-multisets (multisets embedded into a partially ordered set), and indexed (re-entrance) values. The linguistic realization (semantics) of the considered sorts is proposed. The FSs having these multi-sort values are organized as (rooted) directed acyclic graphs. The concrete model of the FSs we had in mind for our set-theoretic definitions are the FSs used within the well-known HPSG linguistic theory. Set-theoretic general definitions for the proposed multi-sort FSs are defined. These constructive definitions start from atomic values and build recurrently multi-sorted values and structures, providing naturally a fixed-point semantics of the obtained FSs as a counterpart to the large class of logical semantics models on FSs. The linguistic unification algorithm based on tableau-subsumption is outlined. The Prolog code of the unification algorithm is provided and results of running it on some of the main multi-sort FSs is enclosed in the appendices. We consider the proposed formal approach to FS definitions and unification as necessary steps to set-theoretical implementations of natural language processing systems.

1 Introduction

The notion of linguistic feature represents a central concept in many contemporary linguistic theories or linguistic tools. We simply remind of GPSG (*Generalized Phrase Structure Grammar*) model developed by G. Gazdar et al. [17], [18], [19], LFG (*Lexical-Functional Grammars*) introduced by J. Bresnan and R. Kaplan [2], [23], *logic grammars* developed by A. Colmerauer [8], [9], V. Dahl and P. Saint-Dizier [15], [16], [30], *functional unification grammars* introduced by M. Kay [25], [26], HPSG (*Head-driven Phrase Structure Grammar*) theory developed by C. Pollard and I. Sag [27], [28], etc. *Feature structures* (FSs), conceived as data types of linguistic features, are the essential declarative object for all these theories when they are viewed as unification-based formalisms (despite appearances, even in Chomsky's syntactic theories [4], [5], [6], [7]). S. Shieber's PATR-II [32] was actually the first unification-based environment using explicitly linguistic data types based on FSs and their unification algorithm(s) to solve the parsing process. FS-based extensions to other linguistic theories, formalisms or strategies are currently common phenomena: e.g., N. Chomsky's GB (*Government and Binding*) and MinP (*Minimalist Program*) [4], [7], C. Pollard & I. Sag's HPSG [27], [28], A. Joshi's TAG (*Tree Adjoining Grammar*) [1], etc., to which we could add the N. Curteanu's S-C-D (*Segmentation-Cohesion-Dependency*) linguistic strategy [11], [12], relying mainly on Augmented X-bar Schemes as basic (constructive) syntactic structures, and on functional (relational) marker classes along with their hierarchies.

As we have shown in [14], the semantics of the parsing process is described in the literature mainly by three important approaches: parsing as *model-theoretic interpretation,* parsing as *automated deduction,* and parsing as operational semantics of *abstract machines.* E.g., in parsing as model-theoretic (dynamic) interpretation [3], FSs are integrated in a set-theoretically oriented formal language, called GEL (*Generalized Ensemble Language*), where grammatical or phrase structure rules become propositions in a dynamic semantics of GEL. Parsing a natural language (NL) phrase or sentence means evaluating recursively the corresponding GEL propositions to the minimal changes in the models making them true.

Inspired by [3] and our previous results [10], the goals of the present paper are: **(1)** to take into account the usual linguistic data types (e.g., those in HPSG) as sorts and to (re)define their set-theoretically oriented linguistic realization (semantics); **(2)** to define FSs in a set-theoretic manner, such that particular cases of the general definitions to fall on several important linguistic data types and theories (such as FSs in HPSG); **(3)** to become transparent that behind logical operations applied to NL categories, both at the surface and deep levels of NL representation structures, can be inserted a layer of set-theoretic operations on FSs, very efficient computationally; and **(4)** to obtain general and set-theoretical shapes of the algorithms involving linguistic operations on FSs such as subsumption, unification, generalization etc. which are essential to the NL parsing process.

2 Sorts and feature structures: approaches and examples

Before coming to formal definitions, let us first consider some intuitive examples that suggest the linguistic relevance of the following primitive concepts: *linguistic feature* (LF), *linguistic category* (LC), *feature structure* (FS), linguistic data types or *sorts,* etc. The following *simple* LF belong to the finite set of features: N (noun), V (verb), Case, Bar (the projection level in X-bar schemes), Gend (gender), Plu (plural), CAT (category), Pers (person), Agr (agreement), etc. Simple LF can take *values* from the specified sets:

$$\begin{gathered}\mathrm{Plu} = \{+,-\};\ \mathrm{Bar} = \{0,1,2\};\ \mathrm{Case} = \{\mathrm{nom}, \mathrm{acc}, \mathrm{gen}, \dots\};\\ \mathrm{Pform} = \{\mathrm{by}, \mathrm{to}, \mathrm{for}, \dots\};\ \mathrm{Gend} = \{\mathrm{masc}, \mathrm{fem}\};\\ \mathrm{V} = \{+,-\};\ \mathrm{N} = +,-;\ \mathrm{Pers} = \{1,2,3\};\ \text{etc.}\end{gathered} \tag{1}$$

A LC is (e.g., in GPSG) a *partial function* defined on the set F of LFs, taking values in the set V of possible feature values. Thus, a LC is a finite set, or bundle, of pairs:

$$\mathrm{D} = \{(f_1, v_1), (f_2, v_2), \dots, (f_n, v_n)\}. \tag{2}$$

FS is a concept similar to that of the above LC, but a LF in FS can take as value an *atomic one* or *any other* FS. In the definition of a FS there is important to emphasize that a FS is considered as a *mathematical function,* i.e. a feature attribute takes a unique value, no matter which type this value should be: single (i.e. atomic or another FS as) type value, disjunction-type value, list-type value, multiset-type value, **po**-multiset type value, indexed (sharing-type) value, etc. Thus the value assigned to a feature attribute must be *unique* and must *belong* to the corresponding set of feature attribute values. This is what we called *well-defined* property of an attribute and, by extension, of a whole FS.

A FS, as considered in (2), may be written as matrices of the form:

$$\mathrm{D} = \begin{bmatrix} f_1 : v_1 \\ f_2 : v_2 \\ . \quad . \\ . \quad . \\ . \quad . \\ f_n : v_n \end{bmatrix} = [f_i : v_i],\ i \in \{1 \div n\}. \tag{3}$$

Let us consider the following examples of matrix representations of FSs:

$$\begin{bmatrix} agreement : & \begin{bmatrix} number : singular \\ person : \ 3rd \end{bmatrix} \\ subject : & \begin{bmatrix} agreement : \begin{bmatrix} number : singular \\ person : \ 3rd \end{bmatrix} \end{bmatrix} \end{bmatrix}. \tag{4}$$

As it can be observed in (4), for each element of the matrix there exist a pair (f, v). A natural extension of FS (3) is to consider various sorted-type values for the features of the structure, as in (5).

$$\begin{bmatrix} agreement : & \begin{bmatrix} number : singular \\ person : \ 1st \end{bmatrix} \vee (1) \vee [number : plural] \\ subject : & \begin{bmatrix} agreement : (1) \begin{bmatrix} number : singular \\ person : \ 2nd \end{bmatrix} \end{bmatrix} \end{bmatrix}. \tag{5}$$

General FSs, i.e., whose values can be other FSs, and their unification algorithms from logic and linguistic points of view are discussed in [26], [27], [28], [29], etc. A set-theoretic approach that formally defines the FSs and their unification algorithms may be found in [10].

Since the HPSG theory [27], [28] is a classical example for a well typed FSs, HPSG will be considered here as a typical reference of the FSs we intend to define.

For the FS D $= [f_i : v_i]$, $i \in 1 \div n$, D is the name of the whole FS, f_i are its *attributes* (names), and v_i are the corresponding values. D is well-defined iff (if and only if) D is a finite function defined as $\mathrm{D}(f_i) = v_i$, $i = 1 \div n$, with the values v_i belonging to the value set of the corresponding f_i, no matter which kind of sorts are v_i: single, list, disjunctive (called $\vee$-set), multiset, **po**-multiset etc.

The value v_i within a FS may be optionally labeled with an index (n), written as a natural number (or another expression) in parentheses. Thus the general form of a FS *value* is $(n)v_i$, a non-indexed value of a FS being the case when the index is missing (or has a special value with the same meaning).

Often, the particularly FS indexed value $(m)[\,]$ ($[\,]$ denoting the empty FS) is reduced simply to the occurrence of the index itself, viz. the expression of the FS value is just (m). In a general FS, for a certain index (n), there exists at most one value $(n)v_i$, with $v_i \neq [\,]$. However, it is also possible that, inside a FS, one or several indices to be used just in the form $(n)[\,]$, or simply (n). The role of such a *bare index* within a FS is that of a global variable, whose type (sort) and unique instance depends on the context which that FS is processed in. Now, the FSs met in the HPSG *theory* may be defined informally as follows:

- D can be the empty FS, denoted by $[\,]$.
- In the FS D $= [f : v]$, the value v may be an *atomic* one a, with a being a constant, or another FS. Such a v is called single-sorted value.
- A powerful constructor of FSs is the conjunction operator on FSs, i.e. if $[f_i : v_i]$, $i = 1, 2, \ldots, n$, are FSs then D $= [f_1 : v_1] \wedge [f_2 :$

$v_2] \wedge \ldots \wedge [f_n : v_n]$ is a FS, written as the bundle of FSs in the form $\mathrm{D} = \begin{bmatrix} f_1 : v_1 \\ f_2 : v_2 \\ \cdot \quad \cdot \\ \cdot \quad \cdot \\ \cdot \quad \cdot \\ f_n : v_n \end{bmatrix}$, similarly to (3), provided that $f_i = f_j \Rightarrow i = j$.

Any permutation of the contained pairs $(f_i : v_i)$ in D gives an "equivalent" or isomorphic FS for any linguistic operation on D.

- The logical connector of *disjunction* is another powerful constructor of new values, called *disjunctive* FSs. The FS language in HPSG (and in many other linguistic theory languages, based or not on FSs) utilizes *disjunctive* FS values that we shall call $\vee$-*sets*, e.g., (5), (8). We shall denote the disjunctive value $v_1 \vee v_2 \vee \ldots \vee v_n$ with the specific $\vee$-set notation ${}^{\vee}\{v_1, v_2, \ldots, v_n\}^{\vee}$.

Since we are at this stage of obtaining new sorts on the basis of logical connectives, let us specify that we shall not deal with *negation* here. This topic was and still is hard worked, with essential results, but we are interested here only in describing the "positive" sorts of FS values. Anyway, for the logical connectives considered, conjunction and disjunction (as well as for different kinds of negation), there exists clear and sound set-theoretical semantics.

- In the FS $\mathrm{D} = [f : v]$, the value v may be an usual, *ordered list*, denoted as HPSG [27], [28] $v = \langle v_1, v_2, \ldots, v_n \rangle$, that we shall call simply "list", made up of atomic values or not. List-valued FSs occur frequently in HPSG as values of the attributes SUBCAT, QUANTS etc., e.g. (8).

- In the FS $\mathrm{D} = [f : v]$, the value v may be an *unordered list* that in mathematics and/or computer science is called "multiset", or "bag". We shall use for a *multiset* (or *bag*) the notation $v =^{+} \{v_1, v_2, \ldots, v_n\}^{+}$. A multiset is a common mathematical set whose elements may (finitely times) be repeated. This sort is not properly met in HPSG [27], [28] but we need it from technical reasons to define the next more complex sort of *multiset with a partial ordering relation.*

- In the FS D = $[f : v]$, the value v may be a multiset whose elements belong to a (larger) partially ordered set, (E, $\geq_{\text{part}}$). This sort is denoted by $v =^{+\text{p}} \{v_1, v_2, \ldots, v_n\}^{+\text{p}}$ and called **po**-*multiset*. There is no relationship between the existing or non-existing of a linear precedence order among the elements of a list and the partial ordering relation $\geq_{\text{part}}$ to which the elements v_i $(i = 1 \div n)$ belong eventually. The **po**-multiset corresponds to what in HPSG is called "set description" and (unfortunately) denoted by "$\{\ldots\}$". The HPSG "set descriptions" or "sets" are, actually, **po**-*multisets* whose elements are FSs belonging to the set FS of all FSs, partially ordered by the *subsumption relation*. Thus, in HPSG, the partial order $\geq_{\text{part}}$ is identified with the *subsumption relation* on FSs. E.g., $^{+\text{p}}\{[\,],\ [\,]\}^{+\text{p}}$ is an HPSG FS, according to the relation (90), [27, pp. 48], but, of course, not a set in the mathematical sense, where the repetition of the same element is not allowed. An illustrative example of what we called a **po**-multiset (or "set description" in [27]) is the following ([27], p. 47, (88)):

$$^{+\text{p}}\left\{ \begin{bmatrix} \text{MAKE TOYOTA} \\ \text{YEAR 1984} \end{bmatrix}, \begin{bmatrix} \text{MAKE DATSUN} \\ \text{YEAR 1985} \end{bmatrix}, \begin{bmatrix} \text{COLOR RED} \end{bmatrix} \right\}^{+\text{p}}. \quad (6)$$

As Pollard and Sag [27] remark, "there is no prohibition against set (n.b.: our **po**-multiset) descriptions of the form $\{A_1, \ldots, A_n\}$ where two of the A_i turn out to be descriptions of the same object." The construction of a **po**-multiset is characterized by the following two properties ([27], pp. 47): (i) every object in a **po**-multiset is described by at least one of the enclosed descriptions; and (ii) each description in the **po**-multiset gives information about only one member of the set of objects being described. Thus (6) gives information that the lot of cars contains a Toyota, a Datsun, and a red car, i.e., *at least two* cars but *no more* *than three* cars. As usual, **po**-multiset valued attributes in HPSG FSs (represented in HPSG with the notation of mathematical sets "$\{\ldots\}$") we can mention RESTR ("sets", actually **po**-multisets of restrictions), INDICES ("sets", actually **po**-multisets of indices), or the **po**-multiset ADJ-DTRS of adjuncts involved by a predicative category, etc.

- With the above values defined, in the FS D $= [f : v]$ the value v may be made up of single, list, $\vee$-set, multiset, and **po**-multiset sort values. Thus the types of FS values are extended to the list and **po**-multiset sorts, namely the value v may be what in HPSG are called "lists" and "sets". E.g. , in $\begin{bmatrix} a: & {}^{+\mathrm{p}}\{v_1, [\,], v_1\}^{+\mathrm{p}} \\ b: & [c : \langle (v_2, [d:\ v_3], v_4 \rangle] \end{bmatrix}$, the value of the attribute c is the list $(v_2, [d:\ v_3], v_4)$, while the value of a is the **po**-multiset (or HPSG "set") ${}^{+\mathrm{p}}\{v_1, [\,], v_1\}^{+\mathrm{p}}$. This point of our informal description of sorts is the strongest extension of the expressive power of the FS set FS, involving the *inductive* definitions on FSs and the sort expressions we introduced until now.

In the HPSG theory occurs also various operations specific to the defined sorts introduced by *union* and *append* functions.

- Another substantial way of extending FSs is to consider structures with *sharing values* as in [27], [28], [32], etc. FS *sharing* assigns one and the same value to several features in the FS, and can be expressed by labelling (indexing) the value of a LF and referring it within two or several substructures of a FS. FS sharing (called also *re-entrance*) is important not only as a data type representation technique, but it introduces more specific linguistic properties (see FS subsumption examples 3.7, 3.10. Sharing FSs corresponds to *indexed* FSs [32], [10]:

$$\begin{bmatrix} agreement : (1) \begin{bmatrix} number : singular \\ person :\ 3rd \end{bmatrix} \\ subject : \quad [agreement :\ (1)] \end{bmatrix}. \tag{7}$$

A FS can be represented naturally as a *graph* assigning *nodes* to the FS attributes and *arcs* to the FS values. As it can be observed, in each node of the graph that represents the FS in (7) there exists at least a pair (f, v) (see Fig. 1). In (4) the values considered are single ones. However it is natural to consider any kind of sort values or their combination for the features situated into the nodes of the FSs. Introducing sharing to FSs is not only motivated by a more compact representation obtained but also, and especially, by the possibilities offered by FS sharing mechanism to define new restrictions and more precise descriptions of the involved linguistic objects.

In [13] we described the organization of a *linguistic knowledge base* having FSs as basic elements and designed for the automatic *analysis and generation* of Romanian. Here there is the FS corresponding to the (Romanian) noun "cal".

$$\begin{bmatrix} \text{PHON : cal} & \\ \text{SYN :} & \begin{bmatrix} \text{CAT :} & \begin{bmatrix} \text{CATM :} & \text{substantiv} \\ \text{TIP :} & \text{comun} \\ \text{PROPR : (1)} & \begin{bmatrix} \text{NR :} & \text{sing} \\ \text{GEN :} & \text{M} \\ \text{CAZ :} & \text{A} \vee \text{N} \\ \text{ART :} & - \end{bmatrix}_{propr} \end{bmatrix}_{scat} \\ \text{HEAD :} & \begin{bmatrix} \text{MAJ :} & \text{N} \\ \text{PRED :} & - \\ \text{ACORD :} & (1) \end{bmatrix}_{head} \\ \text{SUBCAT :} & \left\langle [\text{SYN}|\text{CAT}|\text{CATM : articol}] \vee \begin{bmatrix} \text{PHON : pe} \\ \text{SYN}|\text{CAT}|\text{CATM : prepoziţie} \end{bmatrix} \right\rangle \\ \text{LEX :} & + \\ \text{MARKED :} & - \end{bmatrix}_{sint} \\ \text{SEM :} & \begin{bmatrix} \text{VAR :} & x \\ \text{RESTR :} & \begin{bmatrix} \text{REIN : CAL} \\ \text{ARG :}\ x \end{bmatrix}_{rel} \end{bmatrix}_{sem} \end{bmatrix}_{semn} \quad (8)$$

Having in mind these concrete and intuitive linguistic FSs one can proceed to more formal definitions and algorithms. We shall analyse the two main directions on which a FS like (2) can be extended: the non-sharing but multi-sorted FS values, and sharing (indexed) multi-sorted FS values. The tableau-based subsumption will be specified for the multi-sort non-indexed FSs, and since the algebraic properties of these structures are the same, a general unification algorithm can be designed (for a thorough discussion and results on *logic unification* in associative and commutative theories see, e.g., [21]). In what follows we shall define the FS concept relying on the well-known notion of (rooted) *directed (acyclic) graph* (DAG).

3 Subsumption and unification on FSs as DAGs

This section is devoted to a closer and intuitive look at FSs, to their basic FS operations and properties, involving FS approaches which are different from the set-theoretical ones.

Definifion 3.1 *A (linguistic)* feature structure *(*FS*) is a labelled DAG:*

(a) The labels on arcs are feature names (whose set is denoted by F);

(b) The labels on nodes are feature values (whose set is denoted by V).

Definifion 3.2 *A* well-defined FS *is a* FS *for which the sets F and V are disjoint and any two distinct nodes have distinct labels.*

Example 3.3 *The feature matrix in (7) corresponds to the graphic representation in Fig. 1, where the sets F and V are:*

$F = \{agreement, number, person, subject\}$*;*

$V = \{\mathrm{e}_0, \mathrm{e}_1, \mathrm{e}_2, \mathrm{singular}, \mathrm{third}\}$.

Using the notations for FSs as in (2) we have:

$$dom(\mathrm{D}) = \{f_1, f_2, \ldots, f_n\}, \text{and } val(\mathrm{D}, f_i) = v_i. \tag{9}$$

Example 3.4 *The following matrices are not well-defined FSs:*

$$\begin{bmatrix} a : (1)[b : \ (1)] \\ c : (2)[d : \ (3)] \\ e : (3)[f : \ (2)] \end{bmatrix}, \tag{10}$$

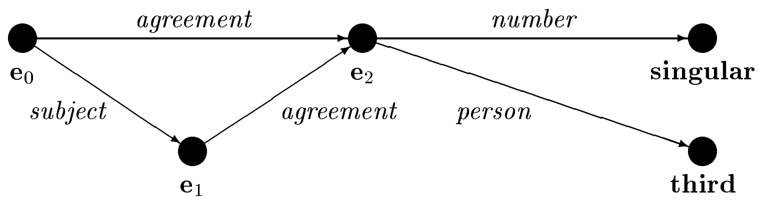

Figure 1.The representation of the FS in (7) as a DAG

since the matrix, or the corresponding graph contains cycles;

$$\begin{bmatrix} a : b \\ b : c \end{bmatrix}, \tag{11}$$

since the same symbol labels both a feature name and an atomic value; if b is a feature name then the first line is irrelevant, and if b is an atomic value then the second line is inconsistent.

Let D denote a FS of the type (3), and let $DAG(\mathrm{D})$ denote the DAG associated to the FS D.

Definifion 3.5 *A* path *in* $\mathrm{D} = \mathrm{D}_1$ *is an ordered sequence* $p = (f_1, f_2, \ldots, f_{i-1}, f_i, \ldots, f_m, v)$ *where* $f_1 \in dom(\mathrm{D})$ *and* $\forall i \in \{2, 3, \ldots, m\}$ *there exists a FS* D_i *such that* $\mathrm{D}_i = val(\mathrm{D}_{i-1}, f_{i-1})$, $f_{i-1} \in dom(\mathrm{D}_{i-1})$ *and* m *is the* length *of the path* p.

We shall denote $v = val(\mathrm{D}, p)$ for every $p \in Path(\mathrm{D}) = \{p | p$ is a path in the FS D$\}$. The subsumption relationship between FS is recursively defined as follows:

Definifion 3.6 *A* FS D subsumes *the* FS D$'$ *iff:*

(i) $dom(\mathrm{D}) \subseteq dom(\mathrm{D}')$, *($\subseteq$ is the sign for the set inclusion);*

(ii) If p, $q \in Path(\mathrm{D})$ *and* $val(\mathrm{D}, p) \equiv val(\mathrm{D}, q)$, *then* $val(\mathrm{D}', p) \equiv val(\mathrm{D}', q)$*;* and

(iii) For any FS *name* $f \in dom(\mathrm{D})$, $val(\mathrm{D}, f)$ subsumes $val(\mathrm{D}', f)$.

D *subsumes* D$'$ is denoted by D$\geq_{\mathrm{sub}}$D$'$. Whether the condition (i) is replaced by "$dom(\mathrm{D}) \subset dom(\mathrm{D}')$", where "$\subset$" is the sign for the set "strict inclusion", then the *subsumption is strict*, denoted by D$>_{\mathrm{sub}}$D$'$. The notation "$\equiv$" refers to the possible equivalence between FSs generated by the commutativity property of the conjunction and/or disjunction constructors. According to the definition 3.6 we have:

Example 3.7

$$\begin{bmatrix} agreement: & \begin{bmatrix} number: singular \\ person:\ third \end{bmatrix} \\ subject: & [agreement: [\,]] \end{bmatrix} \geq_{\mathrm{sub}}$$

$$\geq_{\mathrm{sub}} \begin{bmatrix} subject: & \begin{bmatrix} agreement: \begin{bmatrix} number: singular \\ person:\ third \end{bmatrix} \end{bmatrix} \\ agreement: & \begin{bmatrix} number: singular \\ person:\ third \end{bmatrix} \end{bmatrix}, \quad (12)$$

$$\begin{bmatrix} agreement: & \begin{bmatrix} number: singular \\ person:\ third \end{bmatrix} \\ subject: & \begin{bmatrix} agreement: \begin{bmatrix} number: singular \\ person:\ third \end{bmatrix} \end{bmatrix} \end{bmatrix} >_{\mathrm{sub}}$$

$$>_{\mathrm{sub}} \begin{bmatrix} agreement: (1) \begin{bmatrix} number: singular \\ person:\ third \end{bmatrix} \\ subject: \quad [agreement: (1)] \end{bmatrix}. \quad (13)$$

Intuitively, D *subsumes* D$'$ if D is *more general* than D$'$ or, in other words, D contains *less linguistic information* than D$'$, or "D$'$ is *more informative* than D" [27]. Subsumption is often defined as the *dual relation* to *extension* between FSs, i.e., D$'$ *extends* D, written D$\leq_{\mathrm{sub}}$D$'$, if D is *at least as informative* as D$'$. The subsumption relation is a *partial ordering* on the set FS of all FSs, being reflexive and transitive. Its *maximum* (top) *element* to subsumption in the set of all FSs is the *empty* FS or [ ] (whose domain is empty), denoted by TOP. TOP was also denoted by [ ] because it suggest the FS with the empty content and, according to the above definition, [ ] *subsumes* for *any other* FS.

Another important observation is that two FSs may be *not comparable* at subsumption, written $\neq_{\mathrm{sub}}$. For instance, the following two

FSs are not comparable, i.e. does hold:

$$\left[agr : \begin{bmatrix} nr : & sg \\ pers : & third \end{bmatrix}\right] \neq_{\text{sub}} \left[agr : \begin{bmatrix} nr : & pl \\ pers : & third \end{bmatrix}\right]. \qquad (14)$$

In order to have a *minimum* (or BOTTOM) *element* to subsumption of all FSs there is introduced the FS denoted by $\perp$, such that the following inequalities (subsumptions) hold for any FS D:

$$\forall \mathrm{D} \in FS,\ \text{TOP (or } [\,]) \ \geq_{\text{sub}} \mathrm{D} \geq_{\text{sub}} \text{BOTTOM (or } \perp). \qquad (15)$$

We introduce now the following equivalence relation between two FSs, expressing the fact that they provide exactly the same linguistic information in the set FS of all FSs.

Definifion 3.8 *Let* D_1, $\mathrm{D}_2 \in FS$. *Then* $\mathrm{D}_1 \approx_{sub} \mathrm{D}_2$, *i.e., are* equivalent to subsumption, *iff* $\mathrm{D}_1 \geq_{sub} \mathrm{D}_2$ *and* $\mathrm{D}_2 \geq_{sub} \mathrm{D}_1$.

Convention 3.9 *For linguistic reasons, we consider to hold:* $v \approx_{\text{sub}}$ $\approx_{\text{sub}} {}^{\vee}\{v\}^{\vee} \approx_{\text{sub}} \langle v\rangle \approx_{\text{sub}} {}^{+}\{v\}^{+} \approx_{\text{sub}} {}^{+p}\{v\}^{+p}$. *This is important to avoid the meaningless multiplication of embedded parentheses in multi-sort construction when describing the* same *linguistic objects.*

Example 3.10 *The following* FS*s are equivalent:*

$$\begin{bmatrix} agreement : (1) \begin{bmatrix} number : singular \\ person : \ third \end{bmatrix} \\ subject : \qquad [agreement : (1)] \end{bmatrix} \approx_{\text{sub}}$$

$$\approx_{\text{sub}} \begin{bmatrix} subject : (1)\ [agreement : (2)] \\ agreement : (2) \begin{bmatrix} person : \ third \\ number : singular \end{bmatrix} \end{bmatrix}.$$

The *subsumption* relation between the FSs D_1 and D_2 represented as DAGs may also be seen as the checking operation that $DAG(\mathrm{D}_1)$ is a subgraph of $DAG(\mathrm{D}_2)$. The *equivalence* relation $\approx_{\text{sub}}$ can now be rewritten as the isomorphism between the graphs $DAG(\mathrm{D}_1)$ and $DAG(\mathrm{D}_2)$.

Definifion 3.11 *Let* D, D_1, $\mathrm{D}_2 \in FS$. D *is an* unifier *for* D_1 *and* D_2 *iff* $\mathrm{D}_1 \geq_{\mathrm{sub}} \mathrm{D}$ *and* $\mathrm{D}_2 \geq_{\mathrm{sub}} \mathrm{D}$*, denoted by* $\mathrm{D} \in uni(\mathrm{D}_1, \mathrm{D}_2)$*, the* unifier set *of* D_1 *and* D_2.

The definition says that D is a *lower bound* for D_1 and D_2 at the *subsumption* ordering.

Definifion 3.12 D^* *is* the most general unifier *(mgu) for* D_1 *and* D_2 *iff:*

(i.1) $\mathrm{D}^* \in uni(\mathrm{D}_1, \mathrm{D}_2)$, and

(i.2) $\forall \mathrm{D}$ *such that* $\mathrm{D} \in uni(\mathrm{D}_1, \mathrm{D}_2)$*, then* $\mathrm{D}^* \geq_{\mathrm{sub}} \mathrm{D}$*, i.e.,* D^* *is the* maximal *element among the (possibly) existing unifiers at subsumption of* D_1 *and* D_2.

We denote by $\mathrm{D}^* = mgu(\mathrm{D}_1, \mathrm{D}_2)$ the *most general unifier* D of D_1 and D_2. The definition says that *mgu* of D_1 and D_2 represents their *greatest lower bound* (or glb_{sub}) at the partial ordering $\geq_{\mathrm{sub}}$. Thus:

$$mgu(\mathrm{D}_1, \mathrm{D}_2) = glb_{\mathrm{sub}}(\mathrm{D}_1, \mathrm{D}_2). \tag{16}$$

As [27] notices, $mgu(\mathrm{D}_1, \mathrm{D}_2)$ is “the least informative FS which is at least as informative as D_1 and at least as informative as D_2, ... but nothing more”.

Remark 3.13 *It is obvious that* $mgu(\mathrm{D}, [\,]) = mgu([\,], \mathrm{D}) = \mathrm{D}$*, showing that the TOP* element *(or the empty FS* $[\,]$*) is also the* neutral element *for the semi-lattice FS of all* FS*s at the mgu operation.*

Remark 3.14 *It is possible that two* FS*s are not comparable to each other, thus, an unifier* D does not exist *in the definition 3.11 for the* FS*s* D_1 *and* D_2*. In this case,* D_1 *and* D_2 *are called* non-unifiable*, e.g., the* FS*s in (14). The* BOTTOM element $\perp$ *is also used to denote the “unifier” of two* FS*s whose unification fails and represents the* inconsistent information *“resulted” from two non-unifiable FSs.* $\perp$ *is the same* minimum element *at the subsumption partial order on the set FS of all* FS*s.*

4 Unification algorithms for multi-sort valued FSs

4.1 Sort semantics and constructive hierarchies

We shall begin by specifying the sorts, their semantics (linguistic realization) and their hierarchy in defining the FSs (also corresponding to HPSG theory). These sorts represent the types of values that FSs can take. Subsequently, the usual operations on these sorts are introduced. The main six sorts involved within the present approach are the following:

(S1) The sort of *single* values, *single*-sort. The elements (values) of this sort are atoms or other FS denoted by (single) identifiers: SUBCAT, CAZ, prepozitie, *sint*, *sem*, etc.

(S2) The sort that defines *disjunctive values*, $\vee$-*set*-sort, corresponding to ordinary mathematical *sets*, and denoted by $^{\vee}\{v_1, v_2, \ldots, v_n\}^{\vee}$. The meaning of the $\vee$-set $^{\vee}\{v_1, v_2, \ldots, v_n\}^{\vee}$ is the *disjunction* $v_1 \vee v_2 \vee \ldots \vee v_n$, the same as in HPSG.

(S3) The sort of usual (ordered) *lists*, *list*-sort. The *lists* are denoted just as in HPSG, namely by $\langle v_1, v_2, \ldots, v_n \rangle$, and have exactly the same meaning.

(S4) The sort corresponding to the notion of *multiset*, or *bag*, or set with repetition. The *multiset*-sort is denoted by $^{+}\{v_1, v_2, \ldots, v_n\}^{+}$ and corresponds to a common mathematical set whose elements may (finitely times) be repeated (see section 2). An example of multiset: $^{+}\{a, b, c, a\}^{+}$, which is the same object as $^{+}\{a, a, b, c\}^{+}$ or $^{+}\{a, b, a, c\}^{+}$, etc., but distinct of $^{+}\{a, b, c\}^{+}$. This sort may be not effectively relevant to HPSG but is useful to express the linguistic realization of other sorts.

(S5) The sort of *multiset with a partial ordering relation*, or **po**-*multiset* sort, denoted by $^{+p}\{v_1, v_2, \ldots, v_n\}^{+p}$. The elements v_i, $(i = 1 \div n)$, of the **po**-multiset belong to a set $(E, \geq_{\text{part}})$, where

$\geq_{\text{part}}$ is a partial ordering relation. The **po**-multiset corresponds to what in HPSG is called "set description". The HPSG "set descriptions" or "sets" are, actually, **po**-multisets whose elements are FSs belonging to the set FS of all FSs, partially ordered by the subsumption relation. Thus, in HPSG, the partial order $\geq_{\text{part}}$ is identified by the *subsumption relation* on FSs in FS. (6) is a typical **po**-multiset ([27], pp. 47), with $\geq_{\text{part}} \equiv \geq_{\text{sub}}$.

(S6) The last sort considered is that of *indexed* (*shared*, or *re-entrance*) values, *indexed*-sort, where any (value of) FS may be labelled with an expression called *index*. This is usually denoted by $(k)\mathrm{D}_{k(m)}$, where the *index* (k) is a natural number or a letter, $m = 1 \div p$, p is a natural number, and the indexed FS $(k)\mathrm{D}_{k(m)}$ is repeatedly met within a larger FS D, provided that $\mathrm{D}_{k(m)} \neq [\,]$ for at most one value of m in the set $\{1 \div p\}$. Whether $\mathrm{D}_{k(m)} = [\,]$ $\forall m \in \{1 \div p\}$, then the indexed structures $(k)[\,]$ play the role of an *uninstantiated variable* within the larger FS D. It is important to remark the particular situation when $\mathrm{D}_k = [\,]$, in which case the role of the index (k) is that of a specific variable.

Examples of FSs containing lists, **po**-multisets (HPSG "sets"), $\vee$-sets (HPSG disjunctions), or *indexed* values are (5), (7), (8). As operations between the considered sorts may be taken *append* as the *concatenation* of the lists and **po**-multisets, and the usual set union for $\vee$-sets:

If $\langle v_1, v_2, \ldots, v_m\rangle$ and $\langle w_1, w_2, \ldots, w_n\rangle$ are lists, then:

$$\begin{aligned}&append(\langle v_1, v_2, \ldots, v_m\rangle, \langle w_1, w_2, \ldots, w_n\rangle) =\\ &= \langle v_1, v_2, \ldots, v_m, w_1, w_2, \ldots, w_n\rangle.\end{aligned}$$

Similarly, if ${}^+\{v_1, v_2, \ldots, v_m\}^+$ and ${}^+\{w_1, w_2, \ldots, w_n\}^+$ are multisets then:

$$\begin{aligned}&append({}^+\{v_1, v_2, \ldots, v_m\}^+, {}^+\{w_1, w_2, \ldots, w_n\}^+) =\\ &{}^+\{v_1, v_2, \ldots, v_m, w_1, w_2, \ldots, w_n\}^+.\end{aligned}$$

In particular, the same is true when multisets become **po**-multisets.

If ${}^{\vee}\{v_1, v_2, \ldots, v_m\}^{\vee}$ and ${}^{\vee}\{w_1, w_2, \ldots, w_n\}^{\vee}$ are $\vee$-sets then:

$$\begin{aligned} & \mathit{append}({}^{\vee}\{v_1, v_2, \ldots, v_m\}^{\vee},\ {}^{\vee}\{w_1, w_2, \ldots, w_n\}^{\vee}) = \\ & \quad = {}^{\vee}\{v_1, v_2, \ldots, v_m\}^{\vee} \cup {}^{\vee}\{w_1, w_2, \ldots, w_n\}^{\vee} = \\ & \quad = {}^{\vee}\{v_1, \ldots, w_n\}^{\vee}, \end{aligned}$$

where the sign "$\cup$" denotes the *union operation* between usual, mathematical sets. Thus the *append* operation between $\vee$-sets is not just a concatenation but it also reduces the (eventual) multiple occurrences of elements to single ones (as in common sets).

The problem of modelling the *semantics* of the considered FS sorts is of special interest for our construction. Solving this problem in an appropriate way reduces it to establishing a sound relationship between *words and phrases* of a NL (including, among them, the FS value names) on one side, and the *semantic individuals* they intend to describe, on the other side. The semantic individuals can be, in particular, linguistic notions, concrete forms of a morphologic paradigm, etc.

Let WT be the set of the word tokens of a NL, WT^* the closure of WT to concatenation. A semantic individual, or object, (whose domain we avoid to specify) is described by a certain phrase, or phrase set from WT^*. Thus we consider that a semantic individual described by a FS corresponds to an element of the set $\mathit{Pow\text{-}fin}(WT^*)$, denoting the set of all finite subsets in WT^*. This somewhat hidden relationship between FSs and the semantic individuals they intend to describe can be encoded by the so-called *linguistic realization* of a FS, i.e. a mapping $\mathcal{R}_L$ defined inductively as follows:

$$\mathcal{R}_L : FS \rightarrow \mathit{Pow\text{-}fin}(WT^*), \tag{17}$$

holding:

$$\begin{aligned} \mathcal{R}_L(\mathrm{D}) = \{ & x \in WT^* \ | x \models \mathrm{D}(f) = v, \\ & \text{for } \mathrm{D} = [f : v] \in \mathit{Pow\text{-}fin}(WT^*). \end{aligned} \tag{18}$$

In other words, the semantic individual(s) described by the FS $[f : v]$ is encoded by the set of those phrases (in particular, words) for which the value v, or the path beginning with $f(v)$ when v is a new FS, holds ("$\models$" representing the truth relation).

Definifion 4.1 *Now $\mathcal{R}_L$ can be defined recursively on the considered* FS *sorts.*

(i) $\mathcal{R}_L(a) = \{x \in WT* \mid x \models a\}$ *if a is a (*single-*sorted) atom.*

(ii) When $\mathrm{D} = \begin{bmatrix} f_1 & v_1 \\ f_2 & v_2 \\ \cdots & \cdots \\ f_n & v_n \end{bmatrix}$ *is a (*single-*sorted)* FS, *then*

$$\begin{aligned}\mathcal{R}_L(\mathrm{D}) = \mathcal{R}_L(\bigwedge_{i=1\div n}[f_i : v_i]) &= \\ &= \{x \in WT^* \mid x \models \bigwedge_{i=1\div n} \mathrm{D}(f_i) = v_i\} = \\ &= \{x \in WT^* \mid \bigwedge_{i=1\div n}(x \models v_i)\} = \\ &= \bigcap_{i=1\div n} \mathcal{R}_L([f_i : v_i]),\end{aligned}$$

for the (default) conjunction *constructor of* FS*s, corresponding actually to the* FS unification *process. The relation $\mathcal{R}_L$ extends naturally from single-sort to the other more complex sorts.*

(iii) For the (general) disjunction-*sort:*

$$\begin{aligned}\mathcal{R}_L(^{\vee}\{\mathrm{D}_1, \mathrm{D}_2, \ldots, \mathrm{D}_n\}^{\vee}) = \mathcal{R}_L(\bigvee_{i=1\div n}\mathrm{D}_i) = \bigcup_{i=1\div n}\mathcal{R}_L(\mathrm{D}_i) = \\ =^{\vee} \{\mathcal{R}_L(\mathrm{D}_1), \mathcal{R}_L(\mathrm{D}_2), \ldots, \mathcal{R}_L(\mathrm{D}_n)\}^{\vee} \in \mathit{Pow\text{-}fin}(WT^*).\end{aligned}$$

The FS *disjunction corresponds to what in computational linguistics is known to be the* FS generalization. *In particular, for the* sort-value disjunction *we have:*

(iiia) $\mathcal{R}_L([f : v_1 \vee v_2 \ldots \vee v_n]) = \mathcal{R}_L([f : {}^{\vee}\{v_1, v_2, \ldots, v_n\}^{\vee}]) = = \bigcup_{i=1\div n} \mathcal{R}_L(\mathrm{D}(f) = v_i))$.

(iv) $\mathcal{R}_L(\langle \mathrm{D}_1, \mathrm{D}_2, \ldots, \mathrm{D}_n\rangle) =$
$= \langle \mathcal{R}_L(\mathrm{D}_1), \mathcal{R}_L(\mathrm{D}_2), \ldots, \mathcal{R}_L(\mathrm{D}_n)\rangle \in (\mathit{Pow\text{-}fin}(WT^*))^n$, *for the* list*-sort values. Furthermore, from linguistic considerations, we impose* $\mathcal{R}_L(\langle \mathrm{D}\rangle) = \mathcal{R}_L(\mathrm{D})$.

(v) $\mathcal{R}_L({}^+\{D_1, \mathrm{D}_2, \ldots, \mathrm{D}_n\}^+) = \mathcal{R}_L(\bigvee_{p\in\Pi_n} \langle \mathrm{D}_{i(1)}, \ldots, \mathrm{D}_{i(n)}\rangle) =$
$= \bigcup_{p\in\Pi_n} (\mathcal{R}_L\langle \mathrm{D}_{i(1)}, \ldots, \mathrm{D}_{i(n)}\rangle)$, *where* $p = (i(1), \ldots, i(n))$ *is a permutation taking all the values over the* permutation set Π_n *of* n *elements.*

(vi) $\mathcal{R}_L({}^{+p}\{\mathrm{D}_1, \mathrm{D}_2, \ldots, \mathrm{D}_n\}^{+p}) =$
$\mathcal{R}_L(\bigvee_{P_1,\ldots,P_k} {}^+\{glb(P_1), glb(P_2), \ldots, glb(P_k)\}^+) =$
$\bigcup_{P_1,\ldots,P_k} \mathcal{R}_L({}^+\{glb(P_1), glb(P_2), \ldots, glb(P_k)\}^+)$, *where* $P_1, P_2, \ldots, P_k$ *is a* partition *of the multiset* ${}^+\{\mathrm{D}_1, \mathrm{D}_2, \ldots, \mathrm{D}_n\}^+$, *for which* there exists $glb(P_i), i \in \{1 \div k\}$. *The glb (*greatest lower bound*) is the conjunction (thus* FS *unification) of those elements* D_j, *with* $j \in \{1 \div n\}$, *in the above multiset that belong to the partition* P_i, $i \in \{1 \div k\}$.

The main dependencies between the introduced sorts as they result from our definitions of the linguistic realization $\mathcal{R}_L$ are given in Fig. 2.

The relationship between *S-Sort* (Source sort) and *T-Sort* (Target sort) values has the following meaning: "*any value of* T-Sort *can be effectively constructed, using the* T-Sort *specific operator applied on the* S-Sort *values, but not the reverse*".

Though we informally presented above the indexed FSs, the formal recurrent approach of FSs defined on directed acyclic graphs (DAGs) will be the purpose of the forthcoming (Part II) paper. In the next sections we restrict to FSs organized as *directed trees* (DTs), called *tree* FSs (TFSs), and their natural extension to *multi-sort* valued TFSs (MS-TFSs).

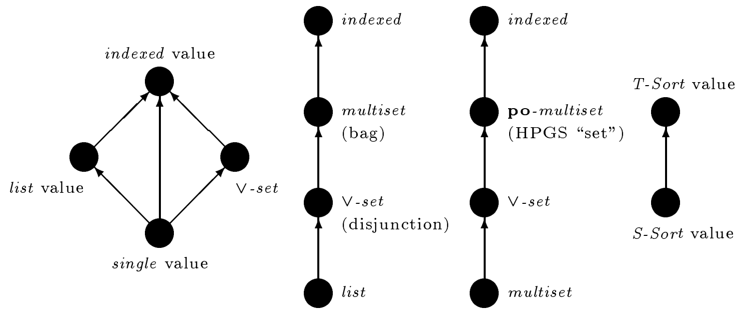

Figure 2.Constructive dependencies between sorts

4.2 A set-theoretic approach to MS-TFSs

Let V be a non-empty set and $Pow(V)$ its *power set*, i.e. the set of all subsets of V or, equivalent, the set of characteristic functions defined on V or, equivalent, the *set of all unit functions partially defined on* V. The following definition is important for the set-theoretic approach we proposed here.

Definifion 4.2 *Let* $T \subseteq Pow(V)$ *and* F *is a set of labels, being in a one-to-one correspondence with* T*, written in what follows by the notation* $T \leftrightarrow F$*.* V *is called the* set of FS values *and* F *is called the* set of FS attributes *defined on* V*.*

Definifion 4.3 *The one-to-one (bijective) mapping* $T \leftrightarrow F$ *will be denoted by* $ValAt : T \to F$*, its (unique) inverse function being denoted with* $AtVal : F \to T$*, thus* $ValAt^{-1} = AtVal$*.*

The bijective function $ValAt$ maps a set $A \in T \subseteq Pow(V)$ to a unique label f, called the *attribute* assigned to the *value* A, $ValAt(A) = = f$. The reverse is also true, $AtVal(f) = A$.

Definifion 4.4 *Let* V *be a non-empty* finite *set of* FS *values, and* F *a non-empty set of* FS *attributes defined on* V*. Then a* FS *defined on* V *can be generally introduced as being a subset* $\mathrm{D} \subset F \times V$*.*

Definifion 4.5 *A* FS $\mathrm{D} \subset F \times V$ *is* well-defined on V *iff the following two conditions hold:*

(4.5S) soundness*:* $(f, v) \in \mathrm{D}$*, then* $v \in_{\mathrm{sort}} ValAt(f)$*, where "*$\in_{\mathrm{sort}}$*" is the specific sort* membership relation*.*

(4.5U) uniqueness*:* $(f, v), (f, w) \in \mathrm{D}$*, then* $v =_{\mathrm{sort}} w$*, where "*$=_{\mathrm{sort}}$*" is the sort-depending* equality relation*.*

For the beginning we consider only FSs whose support are TFSs, and whose values are *single*-sorted; this means that aFS attribute can receive as its value only an atom or a (single) FS. For the *sort* of *single* values, TFSs can be introduced set-theoretically as follows:

Definifion 4.6 *Let* V_0 *be a non-empty finite set of* atomic *single* values*,* $Pow(V_0)$ *the set of* all possible values *for the* 0-level attributes*. Then a non-empty set* $T_0 \subseteq Pow(V_0)$ *is called the set of* effective values for the 0-level attributes*, while* $F_0 \leftrightarrow T_0 \subseteq Pow(V_0)$ *(as in definition 4.2) is said to be the set of single*-valued effective attributes defined on V_0.

(I_0*) An* atomic *single*-valued TFS *(or* TFS of level 0*) is any set* TFS_0 *well-defined on* V_0*, i.e.* $TFS_0 \subset F_0 \times V_0$ *with (4.5S) and (4.5U). Let* $s(TFS_0)$ *be the set of* all possible, well-defined TFS*s* of level 0*, i.e.,* $s(TFS_0) = \{\mathrm{D} | \mathrm{D} \subset F_0 \times V_0$*, and* D *is well-defined on* $V_0\}$.

(I_k*) A* single-*valued* TFS of level $(k+1)$*,* $k \geq 0$*, or a* $(k+1)$-level TFS*, is any well-defined* TFS_k *on* V_k*,* $TFS_k \subset F_k \times V_k$ *with:*

(v_k*)* $V_k = V_{k-1} \cup s(TFS_{k-1})$ *is the set of* all possible values *for* TFS_k*, i.e. the set of* all possible values until level $(k+1)$, *including it.*

$Pow(V_k)$ *represents the set of* all possible values *for the* k-level attributes*. Let* $T_k \subseteq Pow(V_k)$ *be the set of* TFS effective values until level k*, including it, such that* $T_k \backslash T_{k-1} \neq \emptyset$*. Then:*

(f_k) F_k *with* $F_{k-1} \subseteq F_k \leftrightarrow T_k \subseteq Pow(V_k)$ *is the set of* effective attributes until level $(k+1)$, *including it, defined on* V_k *(as in definitions 4.2–4.3). The mechanism of extending the attribute set* F_k *from the set* F_{k-1} *is the following:*

Let B $\in T_k \backslash T_{k-1}$, $AtVal(\mathrm{B}) = f \in F_k$. *Then there exists two situations:*

(i) $f \in F_k \backslash F_{k-1}$, *i.e.,* f *is a new attribute in* F_k, *assigned to a new* FS *value set; or*

(ii) $f \in F_k \cap F_{k-1}$ *; i.e., an already existing attribute in* F_{k-1} *receives in* F_k *a new, extended value set, on the level* k. *Namely,* B *is characterized by:* $\exists \mathrm{A} \in T_{k-1}$ *such that* $AtVal(\mathrm{A}) = f$ *and* $\mathrm{B} \supset \mathrm{A}$.

(tfs_k) $s(TFS_k) = \{\mathrm{D} | \mathrm{D} \subset F_k \times V_k, V_k = V_{k-1} \cup s(TFS_{k-1}),$ $s(TFS_{-1}) = F_{-1} = \emptyset$, *and* D *is well-defined on* V_k $\}, \forall k \in N$, *is called the set of* all possible, well-defined TFSs until level k.

Definifion 4.7 *Now the set of* all *single*-valued TFS*s is defined as:*

$$\mathrm{TFS}(V_0) = \bigcup_{k \geq 0} s(TFS_k) \tag{19}$$

represents the set of all *single*-valued TFS*s.*

The natural extension of the definition 4.6 is to introduce *multi-sort valued* TFSs (MS-TFSs) on the skeleton of a *directed tree* (DT). The definitions are similar to the above ones. Examples of MS-TFSs are in (5), (6), (8).

Convention 4.8 *When an integer variable (subscript or superscript)* $i \in N$ *takes all the values* $1, 2, \ldots, n$, *this is written by one of the following notations:* $i = 1 \div n$ *or* $\forall i \in \{1 \div n\}$. *The existential corresponding counterpart of this abbreviation is, of course,* $\exists i \in \{1 \div n\}$.

Definifion 4.9 *Let us consider the following notations:*

$$\begin{aligned}
&Sort^1 = V;\\
&Sort^2 = \{^{\vee}\{v_1, v_2, \ldots, v_n\}^{\vee} | v_i \in V, i = 1 \div n\};\\
&Sort^3 = \{\langle v_1, v_2, \ldots, v_n\rangle | v_i \in V, i = 1 \div n\};\\
&Sort^4 = \{^{+}\{v_1, v_2, \ldots, v_n\}^{+} | v_i \in V, i = 1 \div n\};\\
&Sort^5 = \{^{+\mathrm{p}}\{v_1, v_2, \ldots, v_n\}^{+\mathrm{p}} | v_i \in V, i = 1 \div n\}.
\end{aligned} \tag{20}$$

Definifion 4.10 *The set W of multi*-sort expressions *defined on V for the sorts $Sort^i$, denoted $W = ms(V)$, is introduced as being the* closure to sort composition *for the sorts considered in (20), i.e., $V \subset W$ and $\forall w_1, w_2, \ldots, w_n \in W$, then $^{\vee}\{w_1, w_2, \ldots, w_n\}^{\vee} \in W$, $\langle w_1, w_2, \ldots, w_n\rangle \in$ $\in W$, $^{+}\{w_1, w_2, \ldots, w_n\}^{+} \in W$, and $^{+\mathrm{p}}\{w_1, w_2, \ldots, w_n\}^{+\mathrm{p}} \in W$.*

The elements in W are sort-*expressions* of any depth on V. The set $W = ms(V)$ of multi-sort expressions defined on V can also be seen as the following union:

$$W = MSort^1 \cup MSort^2 \cup MSort^3 \cup MSort^4 \cup MSort^5, \tag{21}$$

where:

$$\begin{aligned}
&MSort^1 = V;\\
&MSort^2 = \{^{\vee}\{v_1, v_2, \ldots, v_n\}^{\vee} | v_i \in ms(V), i = 1 \div n\};\\
&MSort^3 = \{\langle v_1, v_2, \ldots, v_n\rangle | v_i \in ms(V), i = 1 \div n\};\\
&MSort^4 = \{^{+}\{v_1, v_2, \ldots, v_n\}^{+} | v_i \in ms(V), i = 1 \div n\};\\
&MSort^5 = \{^{+\mathrm{p}}\{v_1, v_2, \ldots, v_n\}^{+\mathrm{p}} | v_i \in ms(V), i = 1 \div n\}.
\end{aligned} \tag{22}$$

It is important to notice that by the present (or additional, if necessary) convenient notations, $MSort^i$, $i = 1 \div 5$, are *pairwise disjoint*.

Definifion 4.11 *Similarly to the definition 4.2, let G be a set of* MS-TFS attributes defined on W, $W = ms(V)$ *as in definition 4.10, i.e., there exists $T \subseteq Pow(W)$ and $T \leftrightarrow G$. Following the definition 4.3, we have the one-to-one functions $AtVal : G \to T$, and $ValAt : G \to T$, with $ValAt = AtVal^{-1}$.*

With the notations in (21), (22) and definition 4.11, the notion of *well-definedness* of MS-TFSs with values in the considered sorts is as follows:

Definifion 4.12 *A* MS-TFS $\mathrm{D} \subset G \times W$, $W = ms(V)$, *is* well-defined on V *iff:*

(4.12S) soundness*:* $(f, v) \in \mathrm{D}$*, then* $v \in MSort^i \Rightarrow v \in^i_{sort} ValAt(f)$, $i = 1 \div 5$*, with the* membership relation $\in^i_{sort}$ *specific to the sorts* $MSort^i$ *in (22). In detail:*

(sound$_1$) $v \in MSort^1 \Rightarrow v \in ValAt(f)$;

(sound$_2$) $v \in MSort^2 \Rightarrow v \subseteq ValAt(f)$;

(sound$_3$) $v \in MSort^3 \Rightarrow v \in ValAt(f)$;

(sound$_4$) $v \in MSort^4 \Rightarrow v \in ValAt(f)$;

(sound$_5$) $v \in MSort^5 \Rightarrow v \in ValAt(f)$;

(4.12U) uniqueness*:* $(f, v), (f, w) \in \mathrm{D}$*, then* $v =^i_{sort} w$*,* $i = 1 \div 5$*, with the* equality relation *specific to the sorts* $MSort^i$ *in (22). More precisely:*

(unique$_1$) if $v \in MSort^1$ *then* $w \in MSort^1$*, and* $v = w$ *(equality as identity relation);*

(unique$_2$) if $v \in MSort^2$ *then* $w \in MSort^2$*, and* $v = w$ *(as set equality);*

(unique$_3$) if $v \in MSort^3$ *then* $w \in MSort^3$*, and* $v = w$ *(as list equality);*

(unique$_4$) if $v \in MSort^4$ *then* $w \in MSort^4$*, and* $v = w$ *(as multiset equality);*

(unique$_5$) if $v \in MSort^5$ *then* $w \in MSort^5$*, and* $v = w$ *(as* **po***-multiset equality).*

Similarly to the definition 4.6, the MS-TFSs can be introduced as follows:

Definifion 4.13 *Let V_0 be a non-empty set of* atomic *single* values*, and $W_0 = ms(V_0)$ the set of all possible* atomic *multi*-sort values *on V_0, constructed similarly to the definitions 4.9–4.10 and relations (20)–(22). Let $T_0 \subseteq Pow(W_0)$ be a non-empty set called the set of* effective values for the 0-level attributes*, while $G_0 \leftrightarrow T_0 \subseteq Pow(W_0)$ (as in definition 4.11) is said to be the set of multi*-sort valued effective attributes defined on W_0. *Then:*

(I_0) An atomic *multi*-sort (valued) TFS *(*MS-TFS*), or* MS-TFS of level 0*, is any $MS-TFS_0$ well-defined on V_0, i.e., $MS-TFS_0 \subset G_0 \times W_0$ satisfying (4.12S) and (4.12U). We denote by $s(MS-TFS_0)$ the set of* all possible, well-defined MS-TFS*s* of level 0*, i.e., $s(MS-TFS_0) = \{\mathrm{D} | \mathrm{D} \subset G_0 \times W_0$, and D is well-defined on $V_0\}$.*

(I_k) A multi-sort valued MS-TFS of level k*, $k \geq 0$, or a* k-level MS-TFS*, is any well-defined $MS-TFS_k$ on V_k, $MS-TFS_k \subset G_k \times W_k$ with:*

(v_k) $V_k = V_{k-1} \cup s(MS-TFS_k - 1)$ *the set of* all possible values for $MS-TFS_k$.

Let W_k be defined similarly to the definition 4.10 be the set of all possible *multi*-sort values defined on V_k until level k*, including it.*

$Pow(W_k)$ represents the set of all possible multi-sort values *for the k*-level attributes*. Let $T_k \subseteq Pow(W_k)$ be the set of* MS-TFS effective values until level k*, including it, such that $T_k \backslash T_{k-1} \neq \emptyset$. Then:*

(g_k) G_k *with $G_{k-1} \subseteq G_k \leftrightarrow T_k \subseteq Pow(W_k)$ is the set of* effective attributes until level k*, including it,* defined on V_k *(as in definition 4.11). The mechanism of extending the attribute set G_k from the set G_{k-1} is the following:*

Let B $\in T_k \backslash T_{k-1}$, $AtVal(\mathrm{B}) = g \in G_k$. Then there exists two situations:

(i) $g \in G_k \backslash G_{k-1}$, i.e., g is a new attribute in G_k, assigned to a new MS-TFS *value set;* or

(ii) $g \in G_k \cap G_{k-1}$ *; i.e., an already existing attribute in* G_{k-1} *receives in* G_k *a new, extended* MS-TFS *value set, on the level* k*. Namely,* B *is characterized by:* $\exists \mathrm{A} \in T_{k-1}$ *such that* $AtVal(\mathrm{A}) = g$ *and* $\mathrm{B} \supset \mathrm{A}$*.*

(ms-tfs$_k$) $s(MS-TFS_k) = \{\mathrm{D}|\mathrm{D} \subset G_k \times W_k,\ W_k = ms(V_k),\ V_k = V_{k-1} \cup s(MS-TFS_k - 1),\ s(MS-TFS-1) = G-1 = \emptyset$*, and* D *is well-defined on* $V_k\}$, $\forall k \in \mathrm{N}$*, is called the set of* all possible, well-defined MS-TFS*s* until level k*.*

Now the set of all MS-TFSs is defined as:

$$\text{MS-TFS}(V_0) = \bigcup_{k \geq 0} s(MS - TFS_k). \tag{23}$$

4.3 Subsumption on MS-TFSs

In order to introduce multi-*sort subsumption* there are necessary some preliminary definitions.

Definifion 4.14 *If* $\mathrm{D} \in \text{MS-TFS}(V)$*, then:*

(i) $dom(\mathrm{D}) = \{f | \exists (f, v) \in \mathrm{D}\}$ *is the* domain *of* D*.*

(ii) $val(\mathrm{D}, f) = \begin{cases} v, & \text{if } (f, v) \in \mathrm{D} \\ [\,] & \text{otherwise} \end{cases}$ *represents the* value *of* f *in the* D*, where* $[\,]$ *represents a special* FS*, called the* empty FS*, such that it subsumes for any other* FS *from the considered sorts.*

If "$\geq_{\text{sub}}$" denotes the subsumption relation between two FSs, then we have:

Definifion 4.15 *For any* $\mathrm{D} \in \text{MS-TFS}(V)$*, hold:*

$$[\,] \geq_{\text{sub}} \mathrm{D}. \tag{24}$$

Let us denote with "$\geq_{\text{part}}$" the partial order involved with the elements of the **po**-multiset.

Definifion 4.16 *The* rank *of a multiset* $v = {}^{+\mathrm{p}}\{v_1, v_2, \ldots, v_n\}^{+\mathrm{p}}$, *whose elements* v_i *belong to a set partially ordered by the relation* $\geq_{\mathrm{part}}$, *is given by the following number:*

$$rank(v) = \min\{k \mid \exists P_1, \ldots, P_k, \text{a partition of } v, \text{such that } \forall i \in \{1 \div k\},\ \exists glb_{\mathrm{part}}(P_i)\}. \tag{25}$$

Of course, $glb_{\mathrm{part}}(\mathrm{P})$ means the *greatest lower bound* from the elements of the set P, computed for the partial ordering $\geq_{\mathrm{part}}$. It is also important to mention that the *partition of a multiset* $v = {}^{+\mathrm{p}}\{v_1, v_2, \ldots, v_n\}^{+\mathrm{p}}$ is a set of disjoint multisets that "cover" v, i.e., whose multiset union is exactly v. The meaning of $rank(v)$ for a multiset $v = {}^{+\mathrm{p}}\{v_1, v_2, \ldots, v_n\}^{+\mathrm{p}}$ is that it represents the *minimal number* of elements v_i, $i = 1 \div n$, taken from the **po**-multiset v, such that they are $\geq_{\mathrm{part}}$-*independent* or, in other words, $\geq_{\mathrm{part}}$-*incomparable*.

Let us now reconsider the *subsumption* definition for MS-TFSs. In this definition, the *rank of a* **po**-*multiset* is depending on the partial ordering represented by subsumption itself, thus $\geq_{\mathrm{part}} \equiv \geq_{\mathrm{sub}}$.

Table 1.The tableau-based subsumption definition

$v \geq_{\mathrm{sub}} w$	$w \in MSort^1$		$w \in MSort^2$	$w \in MSort^3$	$w \in \bot$
	$w \in V$	$w \in$ $\in$ MS-TFS(V) $w = [g_j : w_j]$ $j \in \{1 \div n\}$	$w =$ $= {}^{\vee}\{w_1, \ldots, w_n\}^{\vee}$	$w =$ $= \langle\{w_1, \ldots, w_n\}\rangle$	True
$v \in MSort^1 \cap$ $\cap V$	$v = w$	False	$\forall j \in \{1 \div n\}$ $v = w_j$	$n = 1$ $v = w_1$	True
$v \in MSort^1 \cap$ $\cap$MS-TFS(V) $v = [f_i : v_i]$ $i \in \{1 \div m\}$	$v = [\,]$	$\forall i \in \{1 \div m\}$ $\exists j \in \{1 \div n\}$ $f_i = g_j$ $v_i \geq_{\mathrm{sub}} w_j$	$v = [\,]$ **or** $\forall j \in \{1 \div n\}$, $v \geq w_j$	$v = [\,]$ **or** $n = 1$, $v \geq_{\mathrm{sub}} w_1$	True
$v \in MSort^2$ $v =$ $= {}^{\vee}\{v_1, \ldots, v_m\}^{\vee}$	$\exists i \in \{1 \div m\}$ $v_i = w$	$\exists i \in \{1 \div m\}$ $v_i \geq_{\mathrm{sub}} w$	$\forall j \in \{1 \div n\}$ $\exists i \in \{1 \div m\}$ $v_i \geq_{\mathrm{sub}} w_j$	$\exists i \in \{1 \div m\}$ $v_i \geq w$	True
$v \in MSort^3$ $v =$ $= \langle v_1, \ldots, v_m\rangle$	$m = 1$, $v_1 = w$	$m = 1$, $v_1 \geq_{\mathrm{sub}} w$	$\forall j \in \{1 \div n\}$, $v \geq w_j$	$m = n$ $\forall j \in \{1 \div n\}$, $v \geq w_j$	True
$v = \bot$	False	False	False	False	True

Definifion 4.17 *Let* $\mathrm{D}_1, \mathrm{D}_2 \in$ MS-TFS(V). *Then* D_1 subsumes D_2, *written* $\mathrm{D}_1 \geq_{\mathrm{sub}} \mathrm{D}_2$ *if:*

(i.1) $dom(\mathrm{D}_1) \subseteq dom(\mathrm{D}_2)$, *and*

(i.2) $\forall (f, v) \in \mathrm{D}_1$, *and* $w = val(\mathrm{D}_2, f)$, v *subsumes* w, *written* $v \geq_{\mathrm{sub}} w$ *if only one of the conditions in Table 1 hold.*

4.4 The subsumption tableau (SubTab) on MS-TFSs

The FS (multi-sort expression) v *subsumes* the FS (multi-sort expression) w, denoted $v \geq_{\mathrm{sub}} w$, if one (or several, in a recursive manner) of the conditions in Table 1 hold.

The *Remarks* 4.18 that follow are consistent with the *Table* 1, represent the *extension of the Table 1*, and complete the tableau-based subsumption definition for the sorts $MSort^4$ and $MSort^5$ (Remarks 4-8) accordingly to their linguistic realization semantics (see definition 4.1).

Remarks 4.18 *1. The subsumption relation* $\geq_{\mathrm{sub}}$ *generates on the set* MS-TFS(V) *of multi-sorted* FS *expressions a lattice algebraic structure.*

2. $[\,]$ *(the empty* FS*) represents the maximal element* TOP *to* $\geq_{\mathrm{sub}}$ *in the* MS-TFS(V) *lattice.*

3. $\perp$ *(the universal inconsistent* FS*) represents the minimal element* $BOTTOM$ *to* $\geq_{\mathrm{sub}}$ *in the* MS-TFS(V) *lattice.*

4. *If* $x =^{+} \{x_1, \ldots, x_n\}^{+} \in MSort^4$ *then the* multiset x *is rewritten as a disjunction of (ordered) lists of the form* $\langle x_{k(1)}, \ldots, x_{k(n)} \rangle$ *from* $MSort^3$*, for all the permutations of the multiset* x *elements.*

5. *If* $x =^{+p} \{x_1, \ldots, x_n\}^{+p} \in MSort^5$ *then the* **po**-multiset x *is rewritten as a disjunction of multisets of the form* ${}^{+}\{glb(\mathrm{P}_1), \ldots, glb(\mathrm{P}_k)\}^{+}$ *from* $MSort^4$*, for every partition* $\mathrm{P}_1, \ldots, \mathrm{P}_k$ *of the elements of* x *for which there exists* $glb(\mathrm{P}_i)$, $i \in \{1 \div k\}$.

6. *The glb, i.e.,* greatest lower bound*, over a set of elements is computed by the recursive definition of subsumption and the decomposition of the more complex sorts* $MSort^4$ *and* $MSort^5$ *in simpler ones.*

7. $v, w \in MSort^4$ *are developed as disjunctions of lists (according to Remark 4), then the corresponding rules in the* **SubTab** *Table 1 are applied.*

8. $v, w \in MSort^5$ *are developed as disjunctions of multisets (bags) (according to Remark 5), then the Remark 6 is applied.*

Similarly to the definitions 3.8, 3.11, 3.12, on the class of sort-expressions $v, w \in V_0 \cup \text{MS-TFS}(V_0)$ one can introduce the following subsumption-based equivalence relation:

Definifion 4.19 $v \approx_{\text{sub}} w$ *iff* $v \geq_{\text{sub}} w$ *and* $w \geq_{\text{sub}} v$.

Definifion 4.20 *Let* $u, v, w \in V_0 \cup \text{MS-TFS}(V_0)$*. Then* u *is the* most general unifier *for* v *and* w *(whether it exists), denoted* $u = mgu(v, w)$*, iff* $u = glb_{\text{sub}}(v, w)$.

There exists a straight relationship between the *linguistic realization* of a **po**-multiset and its *rank*. This can be illustrated on the **po**-multiset in (6), (relation (88) in [27, pp. 47]).

$$\text{CARs} = {}^{+\text{P}}\left\{ \begin{bmatrix} \text{MAKE TOYOTA} \\ \text{YEAR 1984} \end{bmatrix}, \begin{bmatrix} \text{MAKE DATSUN} \\ \text{YEAR 1985} \end{bmatrix}, \begin{bmatrix} \text{COLOR RED} \end{bmatrix} \right\}^{+\text{P}}. \qquad (26)$$

Let us compute the *linguistic realization* of the **po**-multiset CARs in (26), identic to (6), with the elements of CARs belonging to the set FS of all FSs, partially ordered by the subsumption relation (i.e., $\geq_{\text{part}} = \geq_{\text{sub}}$). According to the rules (i)–(vi) in the definition 4.1, we obtain:

$$
\begin{aligned}
&\mathcal{R}_L(\mathrm{CARs}) = \mathcal{R}_L\left({}^{+\mathrm{p}}\left\{\begin{bmatrix}\text{MAKE TOYOTA}\\ \text{YEAR } 1984\end{bmatrix}, \begin{bmatrix}\text{MAKE DATSUN}\\ \text{YEAR } 1985\end{bmatrix}, \begin{bmatrix}\text{COLOR RED}\end{bmatrix}\right\}^{+\mathrm{p}}\right) = \\
&= \mathcal{R}_L\left({}^{+}\left\{glb(\begin{bmatrix}\text{MAKE TOYOTA}\\ \text{YEAR } 1984\end{bmatrix}), glb(\begin{bmatrix}\text{MAKE DATSUN}\\ \text{YEAR } 1985\end{bmatrix}), \ldots, glb(\begin{bmatrix}\text{COLOR RED}\end{bmatrix})\right\}^{+}\right) \vee \\
&\vee \mathcal{R}_L\left({}^{+}\left\{glb(\begin{bmatrix}\text{MAKE TOYOTA}\\ \text{YEAR } 1984\end{bmatrix}, \begin{bmatrix}\text{COLOR RED}\end{bmatrix}), glb(\begin{bmatrix}\text{MAKE DATSUN}\\ \text{YEAR } 1985\end{bmatrix})\right\}^{+}\right) \vee \\
&\vee \mathcal{R}_L\left({}^{+}\left\{glb(\begin{bmatrix}\text{MAKE DATSUN}\\ \text{YEAR } 1985\end{bmatrix}, \begin{bmatrix}\text{COLOR RED}\end{bmatrix}), glb(\begin{bmatrix}\text{MAKE TOYOTA}\\ \text{YEAR } 1984\end{bmatrix})\right\}^{+}\right) \vee \\
&\vee \mathcal{R}_L\left({}^{+}\left\{glb(\begin{bmatrix}\text{MAKE TOYOTA}\\ \text{YEAR } 1984\end{bmatrix}, \begin{bmatrix}\text{MAKE DATSUN}\\ \text{YEAR } 1985\end{bmatrix}, \begin{bmatrix}\text{COLOR RED}\end{bmatrix})\right\}^{+}\right) = \\
&= \mathcal{R}_L\left({}^{+}\left\{glb(\begin{bmatrix}\text{MAKE TOYOTA}\\ \text{YEAR } 1984\end{bmatrix}), glb(\begin{bmatrix}\text{MAKE DATSUN}\\ \text{YEAR } 1985\end{bmatrix}), \ldots, glb(\begin{bmatrix}\text{COLOR RED}\end{bmatrix})\right\}^{+}\right) \vee \\
&\vee \mathcal{R}_L\left({}^{+}\left\{glb(\begin{bmatrix}\text{MAKE} & \text{TOYOTA}\\ \text{YEAR} & 1984\\ \text{COLOR} & \text{RED}\end{bmatrix}), glb(\begin{bmatrix}\text{MAKE DATSUN}\\ \text{YEAR } 1985\end{bmatrix})\right\}^{+}\right) \vee \qquad (27) \\
&\vee \mathcal{R}_L\left({}^{+}\left\{glb(\begin{bmatrix}\text{MAKE} & \text{DATSUN}\\ \text{YEAR} & 1985\\ \text{COLOR} & \text{RED}\end{bmatrix}), glb(\begin{bmatrix}\text{MAKE TOYOTA}\\ \text{YEAR } 1984\end{bmatrix})\right\}^{+}\right) = \\
&= \Bigg\{ {}^{+}\left\{glb(\begin{bmatrix}\text{MAKE TOYOTA}\\ \text{YEAR } 1984\end{bmatrix}), glb(\begin{bmatrix}\text{MAKE DATSUN}\\ \text{YEAR } 1985\end{bmatrix}), \ldots, glb(\begin{bmatrix}\text{COLOR RED}\end{bmatrix})\right\}^{+}, \\
&\qquad {}^{+}\left\{glb(\begin{bmatrix}\text{MAKE} & \text{TOYOTA}\\ \text{YEAR} & 1984\\ \text{COLOR} & \text{RED}\end{bmatrix}), glb(\begin{bmatrix}\text{MAKE DATSUN}\\ \text{YEAR } 1985\end{bmatrix})\right\}^{+}, \\
&\qquad {}^{+}\left\{glb(\begin{bmatrix}\text{MAKE} & \text{DATSUN}\\ \text{YEAR} & 1985\\ \text{COLOR} & \text{RED}\end{bmatrix}), glb(\begin{bmatrix}\text{MAKE TOYOTA}\\ \text{YEAR } 1984\end{bmatrix})\right\}^{+} \Bigg\}.
\end{aligned}
$$

The relationship between the linguistic realization of the **po**-multiset in (26) and its *rank* is the following: *rank*(CARs) represents the minimal cardinality of the (multi)sets describing $\mathcal{R}_L$(CARs), thus $rank(\mathrm{CARs}) = \min\{cardinality(\mathrm{A})|\mathrm{A} \in \mathcal{R}_L(\mathrm{CARs})\} = 2$.

4.5 Unification on MS-TFSs

The unification algorithm *for FSs in* MS-TFS(V) *reduces to mgu (thus glb) computing (definition 4.20) and follows directly from the* tableau-based *(Table 1)* subsumption *definition 4.17.*

If $\mathrm{D}_1, \mathrm{D}_2 \in ms(V_0 \cup \text{MS-TFS}(V_0))$, then $mgu(\mathrm{D}_1, \mathrm{D}_2) = mgu(\mathrm{D}_2, D_1)$ and it can be effectively calculated by the (eventually recursive) application of the rules (U1–U9):

U1. If $\mathrm{D}_1 = [\,]$, then $mgu(\mathrm{D}_1, \mathrm{D}_2) = \mathrm{D}_2$;

U2. If $\mathrm{D}_1 = \bot$, then $mgu(\mathrm{D}_1, \mathrm{D}_2) = \bot$;

U3. If $\mathrm{D}_1 \in V_0$ then $mgu(\mathrm{D}_1, \mathrm{D}_2) = \begin{cases} \mathrm{D}_1, & \text{if } \mathrm{D}_1 = \mathrm{D}_2 \\ \bot, & \text{otherwise} \end{cases}$;

U4. If $\mathrm{D}_1 \in V_0$, and $\mathrm{D}_2 \in ms(\text{MS-TFS}(V_0))$, then $mgu(\mathrm{D}_1, \mathrm{D}_2) = \bot$;

U5. If $\mathrm{D}_1, \mathrm{D}_2 \in ms(\text{MS-TFS}(V_0)$, then

$$\begin{aligned} mgu(\mathrm{D}_1, \mathrm{D}_2) = \{(f, v) \in \mathrm{D}_1 | f \in dom(\mathrm{D}_1) \backslash dom(\mathrm{D}_2)\} \cup \\ \cup \{(g, w) \in \mathrm{D}_2 | g \in dom(\mathrm{D}_2) \backslash dom(\mathrm{D}_1)\} \cup \\ \cup \{(f, mgu(v, w)) | (f, v) \in \mathrm{D}_1 \text{ and } (f, w) \in \mathrm{D}_2\}. \end{aligned}$$

U6. If $\mathrm{D}_1, \mathrm{D}_2 \in MSort^2, \mathrm{D}_1 =^{\vee} \{v_1, \ldots, v_m\}^{\vee}, D_2 =^{\vee} \{w_1, \ldots, w_n\}^{\vee}$, then

$$\begin{aligned} & mgu(\mathrm{D}_1, \mathrm{D}_2) = \\ & \quad =^{\vee} \{mgu(v_1, w_1), \ldots, mgu(v_1, w_n), \ldots, mgu(v_m, w_n)\}^{\vee}. \end{aligned}$$

U7. If $\mathrm{D}_1, \mathrm{D}_2 \in MSort^3$, $\mathrm{D}_1 = \langle v_1, \ldots, v_m \rangle$, $\mathrm{D}_2 = \langle w_1, \ldots, w_m \rangle$, then $mgu(\mathrm{D}_1, \mathrm{D}_2) = \langle mgu(v_1, w_1), \ldots, mgu(v_m, w_m) \rangle$.

U8. In all other cases, $mgu(\mathrm{D}_1, \mathrm{D}_2)$ is computed: *either* (a) by complex sort decomposition ($MSort^4$, $MSort^5$) in disjunctions ($MSort^2$) of lists ($MSort^3$), *or* (b) using the Convention 3.9: by default, one considers $v \approx^{\vee}_{\text{sub}} \{v\}^{\vee} \approx_{\text{sub}} \langle v \rangle \approx^{+}_{\text{sub}} \{v\}^{+} \approx^{+\text{p}}_{\text{sub}} \{v\}^{+\text{p}}$, *or* (c) $mgu(\mathrm{D}_1, \mathrm{D}_2) = \bot$.

U9. *Simplification rule*, applicable to any step of the $mgu(\mathrm{D}_1, D_2)$ computation:

U9.1. $^{\vee}\{v, w\}^{\vee} = {}^{\vee}\{v\}^{\vee}$ if $v \geq_{\text{sub}} w$;

U9.2. $\langle v_1, w_n \rangle = \bot$ if $\exists i \in \{1 \div n\}$ such that $v_i = \bot$;

U9.3. If $(f, \bot) \in \mathrm{D}$, then $\mathrm{D} = \bot$ for any (sub)structure occurring in $mgu(\mathrm{D}_1, \mathrm{D}_2)$.

If, after the application of these rules, one obtains $mgu(\mathrm{D}_1, \mathrm{D}_2) = \perp$, then the FSs D_1 and D_2 are considered *not to have an unifier* (thus they are contradictory). Though $\perp$ is considered to play the role of a special symbol, actually $\mathrm{D}_1, \mathrm{D}_2 \in \{\perp\} \cup ms(V_0 \cup \text{MS-TFS}(V_0))$.

References

[1] Anne Abeillé, Y. Schabes, A. Joshi: *Using Lexicalized TAGs for Machine Translation*, Proceedings of COLING'90, Helsinki, Vol. 3, pp. 1–6, 1990.

[2] Joan Bresnan (Ed.): *The Mental Representation of Grammatical Relations*, MIT Press, Cambridge, Mass, 1980.

[3] H. Bunt, Ko van der Sloot: *Parsing as Dynamic Interpretation of Feature Structures*, in: *Recent Advances of Parsing Technologies* (Bunt & Tomita, Eds.), Kluwer Academic Publishers, Dordrecht, pp. 91.

[4] Noam Chomsky: *Lectures on Government and Binding.* The Pisa Lectures, Foris Publications, Dordrecht, Holland, 1982.

[5] Noam Chomsky: *Barriers*, The MIT Press, Cambridge, Massachusetts, 1986.

[6] Noam Chomsky: *Knowledge of Language. Its Nature, Origin, and Use.* Holt, Rinehart & Winston, New York, 1984.

[7] Noam Chomsky: *The Minimalist Program*, The MIT Press, Cambridge, Massachusetts, 1996.

[8] Alain Colmerauer: *Metamorphosis Grammars*, in: *Natural Language Communication with Computers*, L. Bolc (Ed.), pp. 133–189, Springer-Verlag, Berlin, Heidelberg, 1978.

[9] Alain Colmerauer: *Prolog and Infinite Trees*, in: *Logic Programming*, K.L. Clark, S.-A. Tarnlund (Eds.), pp. 45–66, Academic Press, New York, London, 1982.

[10] N. Curteanu, G. Holban, A. Cărăuşu: *On the Linguistic Feature Structure Unification*, Proceedings of the 2nd National Colloquium on Languages, Logic, and Mathematical Linguistics, Braşov (Romania), june 1988, pp. 93–110 (in Romanian).

[11] N. Curteanu: *A Marker Hierarchy-based Approach Supporting the S–C–D Parsing Strategy*, Research Report, Institute of Technical Cybernetics, Bratislava, 30 pp, 1990.

[12] N. Curteanu: *From Morphology to Discourse Through Marker Structures in the S–C–D Parsing Strategy. A Marker Hierarchy-based Approach*, Language and Cybernetics, Akademia Libroservo, Prague, pp. 61.

[13] N. Curteanu et al.: *A Linguistic Knowledge Base for Romanian*, Language and Technology (D. Tufiş, Ed.), The Editing House of Romanian Academy, pp. 101–108, 1996 (in Romanian).

[14] N. Curteanu, A. Todiraşcu: *Syntactic and Semantic Theories of Natural Language. Denotational Semantics*. Research Report, Res. Inst. Comp. Sci., Romanian Academy, Iaşi Branch, 96 pp., Dec. 1998 (in Romanian).

[15] Verónica Dahl, Patrick Saint-Dizier (Eds.): *Natural Language Understanding and Logic Programming*, North-Holland, Amsterdam & New York, XI + 243 pp., 1985.

[16] Verónica Dahl, Patrick Saint-Dizier: *Constrained Discontinuous Grammars – A Linguistically Motivated Tool for Processing Language*, INRIA, France, Research Report No. 573.

[17] Gerald Gazdar: *Phrase Structure Grammars and Natural Languages*, Proceedings of the Intern. Joint Conf. on Artif. Intell., A. Bundy (Ed.), IJCAI Public., pp. 556–565, 1983.

[18] Gerald Gazdar, G.K. Pullum: *Generalized Phrase Structure Grammar: A Theoretical Synopsis*, Cognitive Science Res. Paper CSRP 007, University of Sussex, 51 pp., 1982.

[19] G. Gazdar, E. Klein, G. Pullum, I. Sag: *Generalized Phrase Structure Grammar*, Harvard University Press, Cambridge, Massachusetts, 1985.

[20] Stephen J. Hegner: *A Family of Decidable Feature Logics which Support HPSG-Style Set and List Constructions*, in: (C. Retore; Ed.) *Logical Aspects of Computational Linguistics*, LNCS 1328, Springer.

[21] Alexander Herold, Jörg Siekmann: *Unification in Abelian Semigroups*, Journal of Automatic Reasoning Vol. 3, pp. 247–283, 1987.

[22] Mark Johnson: *Computing with Features as Formulae*, Computational Linguistics, Vol. 20, No. 1, March 1994.

[23] Ronald Kaplan, Joan Bresnan: *Lexical-Functional Grammar: A Formal System for Grammatical Representation*, in: [2], 1982.

[24] R. Kasper, W. Rounds: *A Logical Semantics for Feature Structures*, Reprint from the Proceedings of the 24th ACL Meeting, Columbia University, New York, 10 pp., June 1986.

[25] Martin Kay: *Unification Grammar*, Xerox Palo Alto Res. Center, Palo Alto, California, 1983.

[26] Martin Kay: *Parsing in Functional Unification Grammar*. In (D. Dowty et al.; Eds) it Natural Language Parsing, Cambridge Univ. Press, Cambridge, England, 1985.

[27] C. Pollard, I. Sag: *Information-based Syntax and Semantics*, CSLI, Stanford, California, 1987.

[28] C. Pollard, I. Sag: *Head-Driven Phrase Structure Grammar*, The University of Chicago Press, Chicago & London, 1994.

[29] W.C. Rounds, R. Kasper: *A Complete Logical Calculus for Record Structures Representing Linguistic Information*, Res. Report, University of Michigan, Ann Arbor, 13 pp., March 1986.

[30] Patrick Saint-Dizier: *Context-Dependent Determiners in Logic Programming: Semantic Representation and Properties*, IRISA, Rapport No. 2977, Rennes, France, 1986.

[31] Peter Sells: *Lectures on Contemporary Syntactic Theories*, CSLI Lecture Notes No. 3, Stanford University, California, 1985.

[32] Stuart Shieber: *An Introduction to Unification-Based Approaches to Grammar*, CLSI Lecture Notes No. 4, Stanford University, California, 1986.

[33] Stuart Shieber: *A Uniform Architecture for Parsing and Generation*, Proceedings of COLING'88, Budapest, Vol. 2, pp. 614–619, 1988.

Neculai Curteanu, Paul-Gabriel Holban,
Research Institute of Computer Science,
Romanian Academy, Iasi Branch
B-dul Carol I, nr. 22A, 6600, IASI, ROMANIA
e-mail: *curteanu@iit.tuiasi.ro*

Received April 15, 2000