Formalising Realizational Morphology in Typed Feature Structures

Tomaž Erjavec*

Jožef Stefan Institute
Jamova 39,
61111 Ljubljana, Slovenia
tomaz.erjavec@ijs.si

Abstract

This paper presents a formalisation of realizational approaches to inflectional morphology, exemplified by Anderson's Extended Word and Paradigm theory and Stump's Paradigm Function Morphology. The three levels of morphological organisation, namely the morpholexical rules, disjunctive blocks and conjunctive sequences, are modelled in typed feature structures as used in the context of Head-driven Phrase Structure Grammar.

1 Introduction

Morphological theories can be broadly divided into two families; namely, phrase structure and realizational approaches. The phrase structure view of morphology (c.f. (Spencer, 1991)), essentially brings syntactic mechanisms to bear onto the domain of word formation. Morphemes are taken to be lexical items, and phrase structure rules, sometimes together with notions of headedness and X-bar theory, are responsible for constructing fully inflected words. Variants of this approach have also been by far the most common in computational treatments of morphology.[1]

A radically different approach is adopted by the *realizational approaches*, (a term we borrow from (Stump, 1992b)), which are exemplified by the work of (Matthews, 1972; Zwicky, 1985; Anderson, 1992; Stump, 1992a). The most striking difference presents the interpretation of what constitutes a morpheme. Morphemes do not even properly exist in the realizational approaches, as what would roughly correspond to morphemes are not lexical entries but (word formation) rules. Furthermore, these

*Thanks to Damjan Bojadžiev, Sandi Kodrič, Carl Vogel and the anonymous reviewers for their comments. Any errors remain my own.

[1]For HPSG such a "morphology as syntax" approach was proposed by (Krieger, 1993; Krieger and Nerbonne, 1991).

rules do not have to be concatenative in nature — their scope is the whole phonetic representation of the input which can be modified at will to constitute the output. In other words, it is argued that the minimal phonological unit over which morphological generalisations can be made consistently is the whole word or at least its stem, and that concatenation is only a special case of phonetic realization. This difference is of course exactly the difference between the Item-and-Arrangement and Item-and-Process views (Hockett, 1954) on morphology. But in addition, word formation rules in realizational morphology are not taken to specify the feature content of the word under construction. Rather, the complete morphosyntactic representation of the word is taken to be determined by the syntactic module of the grammar. This morphosyntactic representation together with the specification of the lexical stem is the input to the morphological module. This input uniquely determines which rules are to be used in producing the fully inflected word and in what sequence. The task of the morphological module is then only to *phonologically realize* morphosyntactic representations on lexical stems.

The question we attempt to answer in this paper is how to describe the functionality of realizational approaches to inflection in a declarative, typed feature structure formalism. In particular, we aim to integrate this view of morphology with the formalism of constraint based Head-driven Phrase Structure Grammar — HPSG (Pollard and Sag, 1987; Pollard and Sag, 1994).

Although lexical rules have been used extensively in unification-based grammars, there have been few serious attempts to develop an account of how these rules are to be organised into a morphological theory. Perhaps the most comprehensive attempt is Paradigmatic Morphology (Calder, 1989; Calder, 1990), where the Word-and-Paradigm model (Matthews, 1972; Matthews, 1974) is recast in a general unification-based setting.

In contrast to Paradigmatic Morphology, this paper is in spirit closest to Anderson's Extended Word and Paradigm theory (Anderson, 1992) and Stump's Paradigm Function Morphology (Stump, 1992a; Stump, 1992b), where the notion of "position class" is given more weight. We will cast this view of morphology is HPSG, employing the formal underpinning of typed feature structures (TFS - (Carpenter, 1992)). We will be using the closed-world variant of the TFS logic with recursive type constraints where types can be restricted by arbitrary descriptions.

This paper is organised as follows: Section 2 deals with morpholexical rules. Section 3 describes their organisation into disjunctive blocks which obey the "Elsewhere Condition". Finally, the ordering of morphological formatives constituting a fully inflected word is considered in Section 4, followed by some concluding remarks in Section 5.

2 Morpholexical Rules

In process based morphology, the fundamental operations by which complex morphological expressions are constructed from more basic elements have been called word formation rules, lexical redundancy rules, or *morpholexical rules*. Each morpholexical rule phonologically realizes a specified set of morphosyntactic features ψ on its input. Such a rule takes as its input a (partially inflected) stem whose phonology is ϕ and whose morphosyntactic representation includes ψ and phonologically realizes ϕ as ϕ'. The format (after (Stump, 1992b)) of morpholexical rules is given in (1), while an example for the rule of English plural suffixation is given in (2).

(1) $\textsc{mlr}_{n,\psi}(\phi) = \phi'$

(2) $\textsc{mlr}_{1,[\textsc{num}:plu]}(X) = Xs$

The *slot number*, denoted by the subscript n, is similar to what has been traditionally known as a *position class* in morphology. Position classes have been extensively used to describe the morphology of highly agglutinative languages, in particular in the framework of template morphology. The basic idea behind position classes is that morphemes (or word formation rules) are labelled, and this label (position class) determines their placement in the word form. So, for instance, the lexical stem of a word-form could correspond to position class 0, the suffix closest to the stem to position class 1, the next one to position class 2, and so on. It should be stressed, however, that as the type of morphology considered here is not morpheme based, (i.e. need not be concatenative) the position classes should be thought of as abstract layers of inflectional morphology — as will be discussed in Section 4, these abstract position classes will determine the order of the rule application.

However, the above formulation is too weak. Most realizational approaches to inflection do, at least implicitly (see e.g. (Anderson, 1992, p.174)), allow for the transformation of the morphosyntactic representation as well. We will forego the question whether inflection may change features, and adopt the most general formulation in which arbitrary portions of the feature content can be changed. After (Calder, 1990), we will call the rules that non-monotonically change features *defeating* rules and rules that serve only to (further) instantiate features *non-defeating*.

Such rules can be represented in the notation as given in (3), where ψ and ψ' are not necessarily unifiable.

(3) $\textsc{mlr}_{n}(\phi : \psi) = \phi' : \psi'$

The standard HPSG approach to morphology (Pollard and Sag, 1987) has been in terms of such structure changing rules as well, albeit without position class information. The lexical redundancy rules of HPSG connect two TFS descriptions: (4) gives the equivalent of (3), while (5) is a simplified[2] example for the English 3rd singular rule, similar to the one in (Pollard and Sag, 1987, p.210).

(4) $$\begin{bmatrix} \text{PHON} & \phi \\ \text{SYNSEM} & \psi \end{bmatrix} \Rightarrow \begin{bmatrix} \text{PHON} & \phi' \\ \text{SYNSEM} & \psi' \end{bmatrix}$$

[2]In particular we ignore the SYNSEM|NONLOCAL, QSTORE, and RETRIEVED attributes.

(5)
$$\begin{bmatrix} \textit{base} & \\ \text{PHON} & \boxed{1} \\ \text{SYNSEM}|\text{LOC} & \begin{bmatrix} \text{CAT}|\text{SUBCAT} & \boxed{2} \\ \text{CONT} & \boxed{3} \end{bmatrix} \end{bmatrix} \Rightarrow \begin{bmatrix} \textit{3rdsng} & \\ \text{PHON} & f_{3RDSNG}(\boxed{1}) \\ \text{SYNSEM}|\text{LOC} & \begin{bmatrix} \text{CAT}|\text{SUBCAT} & \boxed{2} \\ \text{CONT} & \boxed{3} \end{bmatrix} \end{bmatrix}$$

As usual, the antecedent embodies the structural description which must be met by any input to the rule, and the consequent specifies the structural change that the rule performs. The output of the rule is generated by the type and feature-value specifications of the consequent description and by explicitly copying portions of the feature content from the input. The above lexical rules of is defeating: in (5), *base* and *3rdsng* represent incompatible types, and thus the SYNSEM values of the antecedent and consequent are non-unifiable. In fact, lexical rules in HPSG *must* be defeating as the base lexicon, which serves as the initial input to the lexical rules already contains valid word-forms. So in the example above, *base* is already a surface form of the verb.

The use of lexical redundancy rules has been criticised on various grounds (Krieger, 1993; Krieger and Nerbonne, 1991), the main one being that they have a completely different semantics from the rest of the (TFS) formalism. It is therefore not surprising that a number of alternatives have been proposed for expressing morphological dependencies in HPSG. Perhaps the most interesting redefinition of lexical rules (for the domain of derivational morphology) is the approach by (Riehemann, 1993). For inflection, (Bird and Klein, 1993) propose a similar treatment to hers, but in a constraint based setting. In a variant of their proposal, the lexical rule from (5) can be recast as in (6), where $\frown$ denotes the concatenation operation.

(6)
$$\begin{bmatrix} \textit{3rdsng} & \\ \text{PHON} & \boxed{1} \frown \langle \text{s} \rangle \\ \text{SYNSEM}|\text{LOC} & \begin{bmatrix} \text{CAT}|\text{SUBCAT} & \boxed{2} \\ \text{CONT} & \boxed{3} \end{bmatrix} \\ \text{MORPH} & \begin{bmatrix} \textit{base} & \\ \text{PHON} & \boxed{1} \\ \text{SYNSEM}|\text{LOC} & \begin{bmatrix} \text{CAT}|\text{SUBCAT} & \boxed{2} \\ \text{CONT} & \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

As can be seen, the gist of this proposal is to replace lexical rules with underspecified feature structures, enabling morphology to be expressed in the same formalism as the rest of HPSG.[3]

It should be noted that even though we do give exponency statements (i.e. the values of the PHON attribute) for illustrative purposes, the concern of this paper is

[3]Crucial for such an approach to work is that the semantics of the TFS logic be of a "closed world" type.

not in morpho-phonology but with morphotactics, and exponency will not be discussed further. The envisioned phonological module would, however, allow subsumption to be defined over the values of the PHON attribute. In other words, it would allow for underspecification of phonological material and could be integrated into the type hierarchy (c.f. (Bird, 1992; Bird and Klein, 1993)).

In (7) we turn to augmenting the HPSG type hierarchy to encompass morphological organisation. The definition of the HPSG *lexical-sign* is changed, so that it introduces the attribute MORPH: lexical signs (can) have internal morphological structure. One of the subtypes of *lexical-sign* is *stem*; it is partitioned by the slot number types, which model the position classes of the language. Our "morpholexical rules" will thus be descriptions of constraints on these types.

(7) *lexical-sign* $\Rightarrow$ MORPH : *morph-dtr*
morph-dtr $\Rightarrow$ *lexical-sign* $\vee$ *nil*
lexical-sign $\Rightarrow$ *stem* $\vee \ldots$
stem $\Rightarrow 0 \vee 1 \vee \ldots \vee n$ (parochial)

The attribute MORPH is typed to *morph-dtr*, which has two subtypes. For lexical signs with no internal structure, the value of MORPH is *nil*. Otherwise, *lexical-sign* has a recursive structure, where the value of MORPH is of type *lexical-sign*. The analogy of *morph-dtr* with the value of the DTRS attribute which encodes the constituent structure of phrasal signs is obvious. But whereas DTRS have a tree like structure, there is only one morphological "daughter" to a rule.

Morpholexical rules will describe constraints on the slot subtypes of *stem*. What has traditionally been called a morpheme or, in realizational approaches, a morpholexical rule is thus taken to be a (partially inflected) stem of the language. This duality follows from the object oriented nature of the TFS representations. Our inflectional "morphemes" embody their mode of combination with the stem; a morpholexical rule represents the structure of the stem together with the formative, i.e. it too is a stem.

Note that there is a transparent mapping between the morpholexical rule notation and its respective TFS encoding. The following AVM is taken to represent a constraint on the slot number type n:

(8) $$\textsc{mlr}_n(\phi : \psi) = (\phi' : \psi') \quad \sim \quad \begin{bmatrix} n & \\ \text{PHON} & \phi' \\ \text{SYNSEM} & \psi' \\ \text{MORPH} & \begin{bmatrix} \text{PHON} & \phi \\ \text{SYNSEM} & \psi \end{bmatrix} \end{bmatrix}$$

As was mentioned at the beginning of this section (and will be further discussed in Section 4), realizational morphology differs from the HPSG type morphology in that the base lexicon is not taken to contain surface word forms: a lexical item must "pass through" the morphological module in order to become a fully inflected word. The lexical items of the base lexicon will be therefore unspecified for morphosyntactic information, e.g. verb form, person and number. From this it follows that typical inflectional rules are not defeating: $\text{MLR}_n(\phi : \psi) = (\phi' : \psi)$. So, if English verbal endings are taken to belong to position class 1, then the morpholexical rule for the third person singular could be formulated as in (9).

$$(9) \quad \left[\begin{array}{ll} \textit{1} & \\ \text{PHON} & \boxed{1} \frown \langle \text{s} \rangle \\ \text{SYNSEM} & \boxed{2} \\ \text{MORPH} & \left[\begin{array}{ll} \textit{3rdsng} & \\ \text{PHON} & \boxed{1} \\ \text{SYNSEM} & \boxed{2} \end{array}\right] \end{array}\right]$$

In this section we have shown how the morpholexical rules, usually taken to be functions between two lexical items, can be reformulated as typed feature structures; the antecedent of the rule is given as the value of the MORPH attribute, while the consequent is modelled by the PHON and SYNSEM values of the outer feature structure.

3 Disjunctive blocks

The first level of organisation of morpholexical rules is that of *disjunctive ordering*. All the rules with the same slot number compete for the realization of the corresponding abstract position class. As at most one rule can "fill", i.e. phonologically realize a certain position class, these rules stand in a disjunctive relation to each other. It is commonly assumed that such competing rules are subject to the *Elsewhere Condition* (EC — e.g. (Kiparsky, 1973)): if two rules are both applicable at a certain point in the derivation (i.e. have antecedents which stand in a subsumption relation) then the rule with the more specific antecedent applies, while the more general rule is forbidden to apply.

Such a rule ordering introduces an element of non-monotonicity into the rule system, and incorporating non-monotonicity into linguistic processing and descriptions has long been a subject of controversy. In fact, the advantages of monotonic systems are so numerous that there is a general trend on all levels of linguistic descriptions to eschew defaults. This certainly holds for the syntactic description of HPSG and for the underlying feature logic, and for certain phonological models as well.[4]

Morphology, being to a large extent entangled with the lexicon (which was traditionally thought of as the list of exceptions) is the least likely to manage without defaults of some kind. This is well attested by the plethora of proposals (e.g. DATR (Evans and Gazdar, 1990), ELU (Russell *et al.*, 1992), ACQUILEX (Copestake, 1991)) which utilise defaults for morphological and lexical description. Our system is somewhat different from most other proposals, in that the question with the EC is not how

[4]c.f. (Bird, Forthcoming) for an overview of proposals that go under the rubric of "constraint based phonology".

to inherit information but rather which rule (i.e. constraint) from a disjunctive block applies in resolving a given feature structure.

As has been explained in the preceding section, a morpholexical rule in our system is a constraint on a slot type. If the proposed system were monotonic, i.e. order independent, then all the morpholexical rules of a position class would simply be disjunctive constraints on a given slot. If a slot type n has k morpholexical rules associated with it, then the constraints on this slot would be as illustrated in (10).

(10) $$n \Rightarrow \bigvee_{i=1}^{k} (\textsc{phon} : \phi_i' \wedge \textsc{synsem} : \psi_i' \wedge \textsc{morph} : \ (\textsc{phon} : \phi_i \wedge \textsc{synsem} : \psi_i)\ )$$

In order to make use of the EC ordering in the description language while still preserving the monotonicity of the feature logic, a translation step is needed. First we introduce an "ordered disjunction" operator ($\vec{\vee}$) into the description language. The intuition behind this operator is simply that its operands are ordered, and that left operand has precedence over the right one. The translation into a feature logic with negation is as given in (11).

(11) $$\alpha \vec{\vee} \beta \ \equiv \ \alpha \vee (\neg\alpha \wedge \beta)$$

As was discussed, the EC ordering must respect the relative specificity of the antecedents of the rules. With our formulation of morpholexical rules this means that the constraints must be ordered according to the subsumption ordering on the values of the MORPH attribute. It is of course possible that two rules belonging to the same slot have non-unifiable antecedents. If the descriptions of the morpholexical rules are normalised in such a way that all the constraints on the antecedent of a rule (i.e. the value of MORPH) are collected together into α and the constraints on the consequent (i.e. PHON and SYNSEM) are collected into α' then desired mapping of a disjunctive block into the typed feature logic is achieved by the formula in (12).

(12) $$\textsc{mlr}_n^1(\alpha_1) = (\alpha_1'),\ \textsc{mlr}_n^2(\alpha_2) = (\alpha_2'),\ \ldots,\ \textsc{mlr}_n^k(\alpha_k) = (\alpha_k') \sim$$
$$n \Rightarrow \vec{\bigvee}_{i=1}^{k} (\alpha_i' \wedge \textsc{morph} : \alpha_i) \text{ where for each } i, j, i < j\text{: } \alpha_j \sqsubset \alpha_i \text{ or } \alpha_i \sqcup \alpha_j = \top$$

In addition to the EC there is one other condition on disjunctive blocks. Each disjunctive block always encompasses a most general rule (see e.g. (Stump, 1992a)), which imposes no constraints in the antecedent and simply maps the input onto the output via the *identity function*. In effect this means that morpholexical rules cannot have negative force, i.e. they cannot prevent a certain combination of morphosyntactic features from being realized. It is assumed that the syntactic component of the grammar allows only well formed combinations of morphosyntactic features to appear on stems.

The identity function default has repercussions on the way realizational morphological grammars are written. Namely, it is useful for describing certain types of zero morphs. Phonetically null morphemes, which crop up in phrase structure approaches have long been considered an undesired consequence of such concatenative theories. In realizational approaches they can be dispensed with quite easily: there is simply no rule for realizing e.g. the singular form of English nouns. The fact that the fully inflected form is identical (has a "null ending") to the lexical stem follows from the identity function default.

It is straightforward to incorporate the identity function default into the above proposal. For each position class, we simply postulate a morpholexical rule, which specifies token identity between the antecedent and consequent (i.e. between PHON and MORPH|PHON and between SYNSEM and MORPH|SYNSEM) and imposes no constraints on the antecedent. The fact that this is the most general, and thus the most "default" description, then simply follows from the definition given in (12).

This section has explained the notion of disjunctive blocks and the Elsewhere Condition that governs the application of rules inside a disjunctive block. We followed this by presenting a method of translating the functionality of disjunctive blocks into typed feature structures.

4 Morphotactics

With the correct realization of a particular slot in place, there is still the linear ordering of the rules to consider. In the case of affixal morphology, the ordering of morpholexical rule application corresponds to the ordering of affixes; the affixes closest to the root are added first, and the final layer of inflection is realized by the last disjunctive block in the sequence. The input to such a conjunctive, ordered sequence of disjunctive blocks is the *root* of the word, and the set of all the possible well-formed outputs on a root is the *paradigm* of the word.

The question of rule ordering has not arisen yet in the literature on HPSG lexical rules or, for that matter, in most of the literature on computational morphology. In the context of lexical redundancy rules, it is commonly assumed that the complete lexicon is defined as the closure of the base lexicon under derivability via the set of lexical rules, possibly leading to an infinite lexicon. While this might be advantageous for derivation, it is not desirable for inflection.

Typically it is presupposed that the morphosyntactic representation of the lexical item is modified so as to make the item a valid input for the next generalised position class of rules and block it as the input to the rules of the same position class. These restrictions are, however, often not enough to specify the correct linear order of the morphemes for languages with complex inflection morphology. Such a solution to rule ordering is also problematic for a realizational approach, where all the relevant morphosyntactic features are already in place. There is another problem with such an approach, namely that the output of any rule is taken to be a valid lexical item, i.e. a word. This certainly need not be the case in languages with complex inflectional morphology, as only the application of all relevant rules in the conjunctive sequence realizes the root as a word.

The most interesting contribution of Paradigm Function Morphology (Stump, 1992a; Stump, 1992b) is its treatment of morphotactics. The ordering of the disjunctive blocks is performed by a special class of *paradigm functions*. Several paradigm functions can be defined for a language, thus making paradigm functions a generalisation of Anderson's approach (Anderson, 1992, p.123–128), where a uniform ordering relation is imposed over all the disjunctive blocks of the language. As argued by Stump, this is a motivated enrichment of the expressive power of the theory, as paradigm functions allow for an elegant explanation of bracketing paradoxes (Stump, 1991), portmanteau morphs (Stump, 1992b) and other morphological phenomena. Paradigm functions are defined as *compositions* of morpholexical functions. Morpholexical functions are taken to describe the morphology of a particular position class, i.e. they are equivalent to disjunctive blocks, modelled by slot types and their constraints. The form of paradigm functions is as given in (13).

(13) $\text{PF}_\sigma(x) = y$
where:

- σ is the complete and fully specified set of morphosyntactic features associated with y,
- x is the root of the paradigm,
- y is a member of x's paradigm.

For a concrete example which illustrates the definition of a PF as the composition of morpholexical functions (MLF), we reproduce the paradigm function schemas for Breton finite verbs (14) and inflected prepositions (15) from (Stump, 1992a, p.227).

(14) $\text{PF}_\sigma([_v x]) =_{def} \text{MLF}_{3,\sigma}(\text{MLF}_{2,\sigma}(\text{MLF}_{1,\sigma}([_v x])))$, where σ is any well-formed set of feature specifications of the type $[\text{TENSE} : \alpha, \text{MOOD} : \beta, \text{AGR} : \gamma]$

(15) $\text{PF}_\sigma([_p x]) =_{def} \text{MLF}_{3,\sigma}([_p x])$, where σ is the unit set of any well-formed set of feature specifications of the type $[\text{AGR} : \alpha]$

The TFS version of paradigm functions is defined in much the same manner and follows rather directly from our definition of disjunctive blocks. Following (Matthews, 1972) (c.f. also (Stump, 1992b)) we distinguish three types of objects relevant to inflection, namely roots, stems and words.

The type *root* is the basic form of the lexeme upon which the words in that lexeme's paradigm are built; it is a lexical sign and its subtypes will contain the base lexicon of the language. A further constraint on *root* is that its MORPH value must be *nil*. The type *stem* refers to an intermediate form in the development of a word from a root with, as we have seen in Section 2, its subtypes corresponding to the disjunctive blocks, i.e. position classes of the language. Finally, the type *word* is taken to be a fully inflected stem on which lexical insertion takes place. In other words, it is the type *word* which describes (via its subtypes) the paradigm functions of the language.

As paradigm functions make reference to the root of the paradigm, so will our *word* type; in particular, the attribute ROOT is taken to be appropriate for *word*. The upper reaches of the *lexical-sign* hierarchy are given in (16).

(16) $\quad$ *lexical-sign* $\Rightarrow$ MORPH : *morph-dtr*
lexical-sign $\Rightarrow$ *root* $\vee$ *stem* $\vee$ *word*
root $\Rightarrow$ MORPH : *nil*
word $\Rightarrow$ ROOT : *root*

A paradigm function for a certain category is a subtype of *word*. A particular paradigm function specifies (1) the categorial information on the words which it models, (2) constraints on the morphological structure of these words, i.e. the ordering of position classes and (3) the structure sharing of ROOT with the innermost level of inflection, which will, for convenience, be factored out into *word*. In a comparison with (14), the example in (17) illustrates the type and its constraints for the PF of Breton verbs.[5]

(17)
$$\begin{bmatrix} \textit{v-word} & \\ \text{SYNSEM}|\text{CAT} & \textit{verb} \\ \text{ROOT} & \boxed{1} \\ \text{MORPH} & \begin{bmatrix} \textit{3} & \\ \text{MORPH} & \begin{bmatrix} \textit{2} & \\ \text{MORPH} & \begin{bmatrix} \textit{1} & \\ \text{MORPH} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

To make the specification of paradigm functions somewhat less cumbersome and to expose their templatic and list-like nature, some notational convenience is in order. We informally define the following mapping:

(18)
$$\text{ATT} :\ll \textit{type}_n, \textit{type}_{n-1}, \cdots, \textit{type}_1 \gg \;\equiv\; \text{ATT} \begin{bmatrix} \textit{type}_n & \\ \text{ATT} & \begin{bmatrix} \textit{type}_{n-1} & \\ \text{ATT} \cdots & \begin{bmatrix} \textit{type}_1 & \\ \text{ATT} & \textit{nil} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Armed with this notation, we define, for *word*, in (19) the structure sharing between the innermost level of inflection and the value of the root attribute.

(19) $\quad$ *word* $\Rightarrow$ ROOT : $\boxed{1}$ $\wedge$ MORPH :$\ll \ldots \boxed{1} \gg$

The final specification for Breton finite verbs and inflected prepositions is then given in (20).

(20) $\quad$ *word* $\Rightarrow$ *v-word* $\vee$ *p-word* $\vee \ldots$
v-word $\Rightarrow$ SYNSEM:CAT : *verb* $\wedge$ MORPH :$\ll \textit{3}, \textit{2}, \textit{1}, \textit{root} \gg$
p-word $\Rightarrow$ SYNSEM:CAT : *prep* $\wedge$ MORPH :$\ll \textit{3}, \textit{root} \gg$

[5](Oliva, 1992) proposes a somewhat similar treatment in terms of typed lists to deal with word order in German.

In this section we have shown how the disjunctive blocks are ordered via paradigm functions and how the notion of a paradigm function can be modelled in typed feature structures.

5 Conclusions

This paper presented a formalisation of certain realizational approaches to morphological theory in the framework of the constraint-based theory of HPSG. Given the monotonic, object oriented nature of TFS descriptions, morphological operations are seen as partially instantiated objects. A parallel is drawn between the type *root* and lexical roots, the type *stem* and the generalised position classes and between the type *word* and paradigm functions.

Such a formalisation of a proper morphological theory opens up the possibility of bringing the explanatory power of the theory to describing morphological phenomena in a model that is common to other levels of linguistic description. This makes for an integrated representation, which has a better chance of dealing with problems that arise on the boundary between phonology and morphology on one hand and between morphology and syntax on the other. Its advantage over other formal or computational morphological proposals is thus primarily of descriptive adequacy, while questions of computational adequacy are the subject of further research.

References

Stephen R. Anderson. *A-Morphous Morphology.* Cambridge University Press, Cambridge, England, 1992.

Steven Bird and Ewan Klein. Enriching HPSG phonology. Research Paper EUCCS/RP-56, Centre for Cognitive Science, University of Edinburgh, 1993.

Steven Bird. Finite-state phonology in HPSG. In *Proceedings of the 14th International Conference on Computational Linguistics*, pages 74–80, Kyoto, 1992.

Steven Bird. *An Introduction to Constraint Based Phonology.* den Hague: Kluwer, Forthcoming.

Jonathan Calder. Paradigmatic morphology. In *Proceedings of the 4nd Conference of the European Chapter of the Association for Computational Linguistics*, pages 58–65, Manchester, 1989.

Jonathan Calder. *An Interpretation of Paradigmatic Morphology.* PhD thesis, Centre for Cognitive Science, University of Edinburgh, 1990.

Bob Carpenter. *The Logic of Typed Feature Structures.* Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, 1992.

Anne Copestake. Defaults in the LRL. In Ted Briscoe, Ann Copestake, and Valeria de Paiva, editors, *Proceedings of the ACQUILEX Workshop on Default Inheritance in the Lexicon*, Cambridge, 1991. University of Cambridge, Computer Laboratory.

Roger Evans and Gerald Gazdar. The DATR papers. Cognitive Science Research Paper CSRP-139, University of Sussex, Brighton, 1990.

Charles F. Hockett. Two models of grammatical description. *Word*, 10:210–311, 1954.

Paul Kiparsky. "Elsewhere" in phonology. In Steven R. Anderson, editor, *Festschrift for Morris Halle*. HOLT, 1973.

Hans-Ulrich Krieger and John Nerbonne. Feature-based inheritance networks for computational lexicons. In *Proceedings of the ACQUILEX Workshop on Default Inheritance in the Lexicon*, Cambridge, 1991. University of Cambridge, Computer Laboratory.

Hans-Ulrich Krieger. Derivation without lexical rules. In M. Rosner, C.J. Rupp, and R. Johnson, editors, *Constraint propagation, Linguistic Description and Computation*. Academic Press, 1993.

Peter H. Matthews. *Inflectional Morphology: A Theoretical Study Based Aspects of Latin Verb Conjugation*. Cambridge University Press, Cambridge, England, 1972.

Peter H. Matthews. *Morphology*. Cambridge University Press, Cambridge, England, 1974.

Karel Oliva. The proper treatment of word order in HPSG. In *Proceedings of the 14th International Conference on Computational Linguistics*, pages 184–190, Kyoto, 1992.

Carl Pollard and Ivan A. Sag. *Information-Based Syntax and Semantics*, volume 1. CSLI/Chicago University Press, Stanford/Chicago, 1987.

Carl Pollard and Ivan A. Sag. *Head-Driven Phrase Structure Grammar*. University of Chicago Press and Stanford: CSLI Publications, Chicago, 1994.

Sussane Riehemann. Word formation in lexical type hierarchies: a case study of bar-adjectives in German. Master's thesis, Dept. of Linguistics, University of Tübingen, 1993.

Graham Russell, Afzal Ballim, John Carroll, and Susan Warwick-Armstrong. A practical approach to multiple default inheritance for unification-based lexicons. *Computational Linguistics*, 18(2), 1992.

Andrew Spencer. *Morphological Theory: An Introduction to Word Structure in Generative Grammar*. Blackwell, 1991.

Gregory T. Stump. A paradigm-based theory of morphosemantic mismatches. *Language*, 67:675–725, 1991.

Gregory T. Stump. On the theoretical status of position class restrictions on inflectional affixes. In G. Booji and J. van Marle, editors, *Yearbook of Morphology*, pages 221–241. Kluwer Academic Press, 1992.

Gregory T. Stump. Position class and morphological theory. In G. Booji and J. van Marle, editors, *Yearbook of Morphology 1992*, pages 129–180. Kluwer Academic Press, 1992.

Arnold M. Zwicky. How to describe inflection. In *11th Annual Meeting of the Berkeley Linguistics Society*, pages 372–86, 1985.