# Compositionality in the Inheritance Lexicon: English nouns

**Dafydd Gibbon**

**23 February 1997**

**(printed March 3, 1997)**

## Contents

# 1 Lexical signs and the Inheritance Lexicon

## 1.1 Lexical signs

What are signs, in linguistic terms? Do signs consist of other signs, in the way that sentences like *Let's listen to Charlie Byrd!* have constituents, or compound words like *mousetrap repair shop owner* are made up of other words? Or is the quality of being a sign rather a holistic one which only attaches to utterances or even dialogues in context, from 'Hi' to the entire proceedings of a business meeting? The present approach to the theory of word formation (the ILEX approach) encompasses the following assumptions about signs:[1]

1. All signs are pairs of some observable form and a meaning.
2. All signs are compositional in principle, down to their smallest phonological constituents.
3. Every language user is familiar with an inventory of more–or–less fixed signs, a *lexicon*, as well as with non–lexical, freely constructed signs.
4. Lexical signs are assigned to a scale of well–defined ranks, corresponding to linguistic levels of description from phoneme–size through morphemes, simple, derived and compound words, phrases and proverbs to ritualised exchanges, in an *idiomaticity hierarchy*; the *word* is a *basic rank*.
5. At each rank, linguistically significant generalisations are formulated in terms of *inheritance* relations for sets of inventorised lexical items at this rank: phonology (better, prosody) is the set of generalisations about speech sounds, morphology the set of generalisations about form and meaning of words, syntax the set of generalisations about form and meaning of phrasal idioms, and so on.
6. At any given rank, a sign has, in principle, four properties: its *surface* (physical appearance, e.g. the forms represented by the transcription /ɹˈæ.tl.sneɪk/, or the spelling *rattlesnake*), its *meaning* (its relation to the situation of use, including objects it refers to, speaker and addressee), its *category* (its co–occurrence with other signs in linguistic structures), and its *parts* (its internal structure or 'child' constituents, which are in general weighted in terms of *head* and *modifier* constituents).
7. The surface and the meaning of a sign are its *interpretative* properties, and the category and parts are its *compositional* properties.

There are interesting special cases. For example, the traditional phoneme is an inventorised item with no parts, no semantic interpretation and purely structural 'meaning'; the morph *cran* in *cranberry* has no parts at the same rank (morphology), and no semantic interpretation (except in Norfolk, where it means 'a basket of the type freshly caught herring are kept in'). Leprechaun items such as 'zero morphemes' and 'traces', for those who believe in them, have no parts and no phonetic interpretation, but a category and a semantic interpretation.

Recent work in syntax, notably within the paradigm of Head–driven Phrase Structure Grammar, HPSG (cf. [Pollard & Sag 1987], [Pollard & Sag 1994]), has revived a similar structuralist notion of sign to that outlined here, and formalised it as an attribute–value matrix (AVM). In this approach, a taxonomy (type hierarchy) of sign types is defined, from the most general type *sign* to the most specific types, individual words; each sign type is characterised by a set of appropriate attributes and appropriate values, and generalisations over more specific sign types are expressed by *inheriting* the properties of more general sign types along the branches of the sign taxonomy.

[1] A number of variants of the template outlined here have been known since the early nineties as the *ILEX* (*Inheritance LEXicon* or *Integrated LEXicon*) model; lexica based on the model have generally been formulated as DATR theories. Many published and unpublished ILEX/DATR 'microlexica' have been implemented on the basis of this model.

It is not yet clear how to integrate lexical problem areas into the word and sentence oriented (albeit lexicalistic) HPSG approach. The HPSG model contains three relevant kinds of entity: a base inventory of words, lexical rules of inflection, word–formation, subcategorisation and semantic selection which define an extended inventory of words, and principles of composition linking the 'head–daughter' (head part) and the 'complement–daughters' (modifier parts) of a phrase by concatenation and unification or other appropriate operation. Problem areas for this model currently still include the following:

1. *Idioms*, which are clearly lexical signs, but not elementary ones.
2. *Sentence prosody* and *word prosody*, which are involved in compositionality, but by complex varieties of *prosodic association* and not just by *concatenation*.
3. *Compositional principles* for the morphology of inflection, derivation and compounding, including compositionality in *morphophonology* and *morphographemics*.
4. *Degrees of irregularity* in the lexicon.
5. *Degrees of compositionality* in syntax and morphotactics.
6. *Markedness relations* based on neutralisation or familiarity.
7. *Compositional lexical semantics* (hard, if lexical items have no parts).

The present study addresses these problems and proposes an integrated, sign–based solution to lexical explanations. In the following sections, an HPSG–related theoretical framework and an operational DATR model for this theory are used to describe English compounds: linguistic concepts closely related to HPSG are described and implemented with representation techniques from DATR. After a summary of the main directions in Inheritance Lexicon Theory, modelling conventions for the inheritance lexicon are characterised, a summary of lexical properties of the main types of English noun, in particular noun compounds, is given, followed by an account of the DATR lexical knowledge representation formalism. An operational DATR model for English nouns is discussed, and a sample analysis is presented. The main results and conclusions are outlined in the final section.

### 1.2 Inheritance Lexicon Theory

A number of approaches to lexical theory are emerging in computational linguistics which address these problems and attempt to integrate descriptions in the known lexical problem areas. A central role is played by the *inheritance lexicon paradigm*, initiated by Flickinger [Flickinger 1987]. Inheritance Lexicon Theory (ILT) has been developed in three main directions, each with slightly different linguistic assumptions and conventions for lexical representation.

**HPSG:** In the HPSG lexicon, lexical signs are represented by AVM structures and classified into types, with more specific types inheriting generalisable properties from more general types. Lexical rules project a base lexicon on to a much larger (perhaps infinite) lexicon; the rules cover lexicon extension in morphology (inflections, derivations, compounds), syntax (complex subcategories such as *passive*), semantics (selectional conditions for disambiguating polysemy).

**OOL:** In the Object–Oriented Lexicon (OOL) lexical items are represented as *objects* (*classes* and *instances*) in an *object hierarchy*, in which objects communicate by *message–passing*, and *methods* for handling the messages are defined for each object. More specific objects inherit general methods from more general objects, and therefore methods do not necessarily have to be fully specified for any given object. Object–oriented representations originated as a means of representing one type of semantic network in Artificial Intelligence, generally implemented as functions in LISP, but have resulted in well–known class–oriented

programming languages such as SmallTalk, C++ and Java. The OOL concept was introduced by Daelemans [Daelemans 1987].

**DATR:** DATR is a lexical knowledge representation language developed by Evans and Gazdar (summarised in [Evans & Gazdar 1996]). In DATR, the basic unit is the *node* (roughly comparable with the *type* in the HPSG approach and the *object* in the OOL approach) organised into a *default inheritance hierarchy*. Each node in the hierarchy is characterised by a set of *attribute–value equations* (more precisely, equations pairing attribute paths and values), in which any *path* may only occur once, and each *value* evaluates to a sequence of atomic constituent values (possibly null). The constituent values directly specified for particular nodes may be atomic, or inherited from more general nodes. Since a node may therefore inherit values from several other, more general nodes, but only if constrained by a unique attribute, DATR is said to have *orthogonal multiple inheritance*. Default inheritance means that a value of a given attribute may be specified more than once in an inheritance path, in which case the values at lower (more specific) nodes in the hierarchy override values.[2]

# 2 Modelling conventions for the Inheritance Lexicon

## 2.1 Basic modelling conventions

Traditionally, the linguistic structure of signs is characterised in terms of three basic notions: *level of representation (abstraction, description)*; *syntagmatic relation*; *paradigmatic relation*. The *level of representation (abstraction, description etc.)*, includes compositional levels of morphology, syntax, text, and interpretative levels of semantics, phonetics. At each level, structure is further defined by *syntagmatic relations*, including concepts of *dependency*, *valency* and *headedness*, and by *paradigmatic relations*, including concepts of *markedness*. Syntagmatic relations are *part–whole* and *part–part* relations and paradigmatic relations are *similarity* relations which define classes of linguistic units and oppositions between sub–classes. These notions will be characterised in more detail below.

*Level of representation (abstraction, description etc.):* A coherent set of descriptive categories together with methodological criteria and formal representation devices for these categories. Levels are assigned to a scale of well–defined ranks corresponding to linguistic levels of description from phoneme–like units through morpheme–like units, simple, derived and compound words, phrases, sentences (including idioms and proverbs) to ritualised exchanges. At each rank a distinction between *lexical* and *nonce (ad hoc)* items is defined, and the rank scale of lexical items constitutes an *idiomaticity hierarchy*. The *word* is a *basic rank* in the sense of Rosch's notion of basic category [Rosch 1978].

A distinction is made at each rank between signs and their *co–interpretation* in terms of phonetic and orthographic *surface form* and *meaning*. The duality of co–interpretation, shared by many linguistic theories, explicates the traditional semiotic triangle in terms of a sign for which there exists on the one hand a model of surface form (sound or writing, gesture, scent etc.), and on the other hand a model of situational meaning. Whether the sign and its two types of interpretation are assigned cognitive (conceptual, mentalistic) interpretations in addition to

[2]Representative studies using DATR models have been carried out by Gibbon on Arabic paradigms and Kikuyu tone ([Gibbon 1990]), Reinhard & Gibbon on Arabic and Kikuyu ([Reinhard & Gibbon 1991]), Gibbon on German compounds ([Gibbon 1992]), Cahill on morphophonology in the lexicon ([Cahill 1993]), Bleiching on German morphology and lexical prosody ([Bleiching1992], [Bleiching 1994]), Corbett & Fraser on Russian inflection ([Corbett & Fraser 1995]), Bleiching, Drexel & Gibbon on German inflection ([Bleiching, Drexel & Gibbon 1996]), Gibbon, Tseng & Folikpo on Ewegbe tone ([Gibbon, Tseng & Folikpo 1997]).

the behavioural and observational criteria for surface (and, in part, semantic) interpretations is more a question of a linguist's epistemological stance than of direct empirical consequence.

The pair of interpretation functions co–interprets items at different ranks such as the *phoneme*, the *morpheme*, the *word*, the *sentence*, the *turn* or *dialogue contribution*, the *dialogue*. Mapping functions between ranks and rank–specific interpretative models define the overall architecture of a linguistic theory.

*Syntagmatic relation:* A compositional relation, definable as

1. a part–whole (dominance) relation between parent categories and child categories (constituents), for example *head–of*, *modifier–of*, or
2. a part–part relation between sibling categories, e.g. dependency or valency relations, *affix–to*, *initial*, or
3. a transitive generalisation of these simple relations to more indirect relations (e.g. *head feature projection* as a generalisation of the part–whole relation, or *SVO* surface order as a generalisation of the simple part–part relation).

A fundamental distinction between (possibly universal) *immediate dominance (ID)* or part–whole relations and (partly language specific) *linear precedence (LP)* or temporally and spatially interpretable part–part relations is made in most computational grammars. For example, the ID structure of compound words in English and French is similar, but English is 'right–headed' whereas French is 'left–headed' and uses interfixed prepositions: *peau–rouge* 'redskin', *épingle à cheveux* 'hairpin', *pain d'épice* 'gingerbread'.

In the ILEX approach, the core type of syntagmatic relation is the ID relation, and the LP relation is generalised to the *quasi-linear precedence (QLP)* relation in order to include prosodic association for suprasegmentals in speech, highlights and layout in writing. The QLP relation plays a similar role in surface form interpretation to *logical form (LF)* in semantic interpretation. A distinction is therefore made between *compositional syntagmatic relations* and *interpretative syntagmatic relations*; it is the latter which generally features in traditional descriptions. In current theories of syntax, syntagmatic relations are formalised as operations of compositionality, e.g. the *slash* and *position* operations in categorial grammar, *rewrite* and *concatenation* operations in phrase structure grammar, and the ID and LP relations of unification grammar.

A straightforward definition of a syntagmatic relation is as follows:

$$\forall x, y, z \; SynRel(x, y, z) \equiv Part(x, z) \wedge Part(y, z) \wedge f(FS(x), FS(y)) = FS(z)$$

where at most one of $x$ or $y$ or $z$ may remain uninstantiated, $SynRel$ is a syntagmatic relation, and $FS$ is a feature structure (i.e. AVM). For example,

$$Spelling(jellyfish) = f_{spell}(Spelling(jelly), Spelling(fish))$$
$$Pronunciation(jellyfish) = f_{pron}(Pronunciation(jelly), Pronunciation(fish))$$
$$Meaning(jellyfish) = f_{mean\_jf}(Meaning(jelly), Meaning(fish))$$

This formula expresses *Frege's Principle (FP)* (cf. [Cresswell 1973]) of compositionality, i.e. the principle that a property of the whole is a function $f$ of this property of the parts, whereby $f$ may be concatenation, unification, slash cancellation, etc., depending on the formalism used and the empirical combinatory principle to be modelled. FP is generally applied only to semantic interpretation; in the present approach it is also applied to surface form interpretation. The function $f_{mean\_jf}$ is less general in this case than the surface interpretation functions, and needs components to account for metaphor and ellipsis.

*Paradigmatic relation:* A generalisation relation, characterising similarity between signs in terms of one or more sign properties, defining *sets* or *classes*, *elements of sets*, and *set–subset inclusion*,

with the usual set theoretic operations of union, intersection, and the formation of set theoretic relations as tuples. Sign properties are defined in terms of feature structures, and similarity is defined in terms of the subsumption ($\sqsubseteq$) operation[3]. Traditionally, paradigmatic relations define semantic fields, syntactic categories, phonological natural classes, and distributional classes of all kinds. Leaving aside some technical details, the terms used may be defined straightforwardly as follows, with feature structures representing complex lexical properties of quantifiable lexical objects, $FS_i$ (the *subsumer*) and $FS_j$ (the *subsumed*) are feature structures consisting of attribute–value (AV) pairs, and '$\rightarrow$' and '$\equiv$' represent conditional and biconditional propositional functions respectively:

Subsumption: $\forall\ x\ FS_i \sqsubseteq FS_j \equiv FS_j(x) \rightarrow FS_i(x)$

Paradigmatic relation: $\forall\ i,j\ ParaRel(i,j) \equiv \exists\ k\ FS_k \sqsubseteq FS_i \wedge FS_k \sqsubseteq FS_j$

Paradigmatic generalisations are expressed as inheritance relations between subclasses and classes, and among the subclasses of a given class. This concept is explained in the following sections.

## 2.2 Subsumption hierarchies, taxonomies and generalisation

The subsumption relation can be understood as a relation of *implication* which relates more specific to more general concepts in conceptual taxonomies. In formal terms, subsumption defines a lattice, a kind of partial ordering, which may be represented as a directed acyclic graph. The hierarchical graphs defined by subsumption need not be trees, but can be more general kinds of graph in which child nodes are *re–entrant*, i.e. a child node may have more than one parent node. However, commonly a subsumption lattice has a core tree structure, with superimposition of more than one tree, or of other cross–classifiying structures. The subsumption relation may be seen as a *generalisation* relation, in that the subsumer expresses a generalisation over the subsumed.

Examples of lexical subsumption are shown in Figure 1, which illustrates some of the following points:

1. The semantic properties of *horse* subsume the semantic properties of *stallion*.
2. The semantic properties {*male, animal*} subsume the semantic properties of *stallion*.
3. The semantic properties of *horse* subsume the semantic properties of *mare*.
4. The phonological properties of *lamp* subsume the phonological properties of *streetlamp*.
5. (Some) properties of heads subsume the corresponding properties of constructions whose heads they are.
6. $\left[\text{MANNER obstruent}\right] \sqsubseteq \left[\begin{array}{l}\text{MANNER obstruent}\\ \text{VOICING unvoiced}\end{array}\right]$
7. Archiphonemes subsume their phoneme members.

## 2.3 Generalisation hierarchies and inheritance

If $FS_i \sqsubseteq FS_j$, as in any of the cases illustrated above, then the subsumed $FS_j$ is redundant if all its AV pairs are completely specified. Consequently, the information in subsumer $FS_i$ may be subtracted from $FS_j$, leaving a non–redundant set of AV specifications, and a redundancy rule can be formulated which will allow the 'missing' features to be inferred or 'added in'. This is standard procedure in the rule notation of generative phonology and morphology:

[3]In some sources, the symbol is reversed by analogy with the subset relation over the extensions of the feature structures.

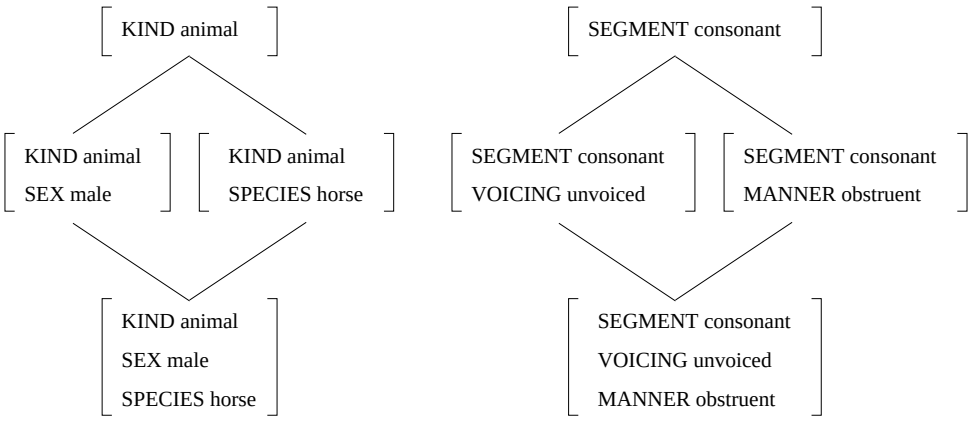


Figure 1: Reentrant subsumption graphs

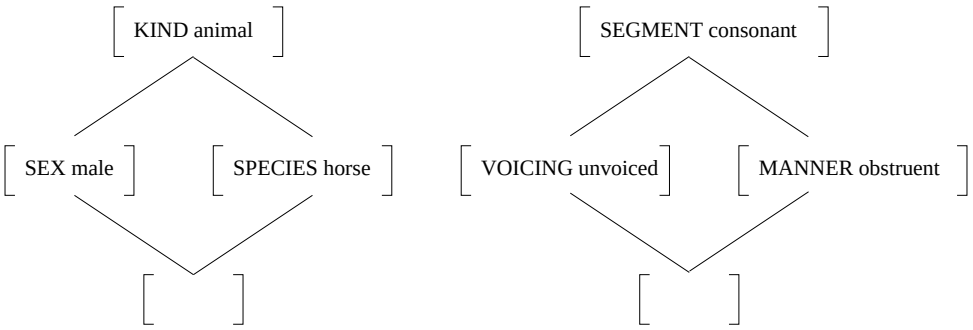


Figure 2: Reentrant inheritance graphs

The phonological redundancy rule: $\left[\text{MANNER obstruent}\right] \rightarrow \left[\text{VOICING unvoiced}\right] / \_ \#$

expands conventionally to: $\left[\begin{array}{l}\text{MANNER obstruent}\\ \text{VOICING [ ]}\end{array}\right] \# \rightarrow \left[\begin{array}{l}\text{MANNER obstruent}\\ \text{VOICING unvoiced}\end{array}\right] \#$

or, in terms of subsumption: $\left[\begin{array}{l}\textit{archi–segment}_i\\ \text{MANNER obstruent}\end{array}\right] \# \sqsubseteq \left[\begin{array}{l}\textit{archi-segment}_j\\ \text{MANNER obstruent}\\ \text{VOICING unvoiced}\end{array}\right] \#$

The subtraction operation between a subsumed $\text{AVM}_1$ and a subsumer $\text{AVM}_2$ yields a non–redundant $\text{AVM}_3$ in an *inheritance* relation with $\text{AVM}_2$. The inheritance relation whereby $\text{AVM}_3$ inherits the features of $\text{AVM}_2$, and thereby reconstitutes $\text{AVM}_1$, s the inverse of the subtraction operation, and is expressed as a special case of unification: $\text{AVM}_1 = \text{AVM}_2 \sqcup \text{AVM}_3$, where $\text{AVM}_3 \sqcap \text{AVM}_2 = \emptyset$. The generalisation (feature intersection) operator '$\sqcap$' is defined as the set of features shared by $\text{AVM}_3$ and $\text{AVM}_2$ and the specialisation (unification) operator '$\sqcup$' is defined recursively for compatible AVMs: two attribute–value pairs unify either if the values are identical atoms, or if an attribute in one AVM is not specified in the other, or if the values of identical attributes in the AVMs unify. Under the type inheritance operation expressed by unification, the AVMs in Figure 2 and the AVMs in Figure 1 are equivalent. The elementary case of non–recursive unification has been familiar in linguistics since the introduction of the *lexical insertion* operation by Chomsky [Chomsky 1965]; Shieber [Shieber 1986] summarises the more general unification operation used in unification grammars.

In the DATR formalism, a form of *default inheritance* is defined, in which the subsumption relation and the unification operation do not hold. Instead, there is a *default–override* relation between paths in AVMs, and based on this an elementary form of *default unification* operation involving the inheritance of values of paths. In the default–override relation, a value for a given

attribute may be specified more than once in the same inheritance path, and the specification of the lower (more specific) class overrides the specification of the higher (more general) class. In a famous illustration, Tweety, *qua penguin* cannot fly, but Tweety, *qua bird* can fly. Clearly, the penguin specification is more specific than the bird specification, therefore the dispositional predicate 'cannot fly' overrides the dispositional predicate 'can fly'.

In the ILEX version of Inheritance Lexicon Theory, default inheritance is used in order to explain exceptions and subregularities of this kind.

## 2.4 Signs, archi–signs, and generalisation over signs

The four main properties of a sign have complex values whose structure is summarised in the following nested attribute value template (with illustrative values inserted), which will be referred to as the ILEX template:

$$\left[\begin{array}{ll} \text{LEMMA} & \textit{pussy–willow} \\ \text{STRUC} & \left[\begin{array}{ll} \text{CAT} & \textit{compound\_noun} \\ \text{PARTS} & \left[\begin{array}{ll} \text{HEAD} & \textit{willow} \\ \text{MODI} & \textit{pussy} \end{array}\right] \end{array}\right] \\ \text{INT} & \left[\begin{array}{ll} \text{MEAN} & \left[\begin{array}{ll} \text{EVENT} & \textit{state} \\ \text{QUALIA} & \left[\text{RELN RESEMBLE}(\textit{willow},\textit{pussy})\right] \\ \text{TECH} & \text{SALIX CAPREA PENDULA} \\ \text{INDEX} & j \end{array}\right] \\ \text{SURF} & \left[\begin{array}{ll} \text{PHON} & /\text{pUsI\#w'IlәU}/ \\ \text{ORTH} & \text{“pussy–willow”} \end{array}\right] \end{array}\right] \end{array}\right]$$

The attributes have the following interpretations (abbreviations in parentheses):

LEMMA: Name of the lexical entry; the lowest type in the inheritance hierarchy; it can be compared with types in HPSG (except that the ILEX approach uses default inheritance lattices, while HPSG uses type subsumption lattices).
STRUCTURE (STRUC): The syntagmatic properties of the sign.
CATEGORY (CAT): The relation of a head sign to its parent and siblings (cf. HPSG 'HEAD' and 'SUBCAT' attributes).
PARTS: The constituents of a sign (cf. HPSG 'DTRS').
HEAD: The head constituent (cf. Zwicky [Zwicky 1993]).
MODIFIER (MODI): The non–head constituents of a sign (cf. HPSG 'COMP'); for noun compounds, generally a single item.
INTERPRETATION (INT): The basic semiotic properties of a sign.
MEANING (MEAN): The semantic interpretation attribute.
EVENT: Taken from Generative Lexicon Theory (cf. Pustejovsky [Pustejovsky 1995]).
QUALIA: Taken from Generative Lexicon Theory.
RELATION (RELN): Taken from HPSG–flavoured semantic role structure.
TECHNICAL (TECH): Indicates a technical meaning from a special sublanguage.
INDEX: Taken from HPSG–flavoured situation semantics.
SURFACE (SURF): The phonetic/orthographic interpretation attribute.
PHON: Phonetic interpretation (with prosodic association and concatenation, when represented in full detail).
ORTH: Orthographic interpretation.

Values which are shared by a class of signs (i.e. values defining paradigmatic similarity relations) may be generalised by applying the operator '$\sqcap$' to the AVMs of the signs. In this case, the values are *inherited* from the 'archi–sign' representing this class, and need not be represented explicitly for each member of the class. Inheritance therefore expresses implication, the paradigmatic relation which constitutes taxonomies. For example, *serenity* inherits certain phonological

Table 1: AVM inheritance operations.

| Symbol: | Type of inheritance: |
|---|---|
| $\rightarrow$ | Paradigmatic inheritance from an archi–sign |
| $\leftarrow$ | Orthogonal multiple inheritance from an archi–sign |
| $\Downarrow$ | Syntagmatic inheritance from a PART |
| $\Uparrow$ | Lexical insertion of a property of a PART into an interpretation template (or an evaluable path) |

properties from the archi–sign representing the class of English words affected by tri–syllabic shortening; *bake* inherits the details of its inflections from the archi–sign representing the class of all weak verbs; *chair* inherits certain general semantic properties from the archi–sign representing all items of furniture; *surfboard* inherits compositional properties from the archi–sign representing the class containing *skateboard* and *blackboard*, and in particular it inherits 'head features' such as CAT from its HEAD PART *board*.

The inheritance of properties from (or by) a PART is commonly referred to as *feature percolation*, and defines the notion of compositionality in attribute–value terms.

The four main kinds of inheritance, which are closely related to mechanisms in the DATR lexical knowledge representation language, are listed in Table 1.

Orthogonal multiple inheritance simply means that the values of several different specified attributes may be inherited from different archi–signs or types, rather than from a single archi–sign for the CAT attribute. In general, any attribute which is not explicitly specified inherits its value from the archi–sign; the notation given here permits explicit expression of this relation.

## 2.5 Surface compositionality and semantic compositionality

The concepts of *lexical compositionality* and *partial lexical compositionality* can now be illustrated in terms of a generalisation of the ILEX template togather with inheritance relation (see Table 2). Immediate Dominance compositionality is represented by the STRUC attribute, and compositional interpretation is indicated by parentheses which represent the application of a semantic or phonetic operator (the first element in the enclosed list) to its operands (the remaining list elements). The notions SEMANTICALLY_LINK and PROSODICALLY_LINK are defined in terms of default unification. The operation of hyphenation is straightforward concatenation of the parts with an intervening hyphen, with the concatenation operation interpreted as a spatial precedence relation. compositionality is defined in general terms for all interpretative features, but each type of interpretation specifies its own operators.

An interesting feature is the operation of *lexical insertion*, licensed by constraints specified by the '$\Uparrow$' inheritance type and expressed as *attribute paths*, i.e. nested AVMs with only one attribute specified per recursion,

In the illustration, the LEMMA *pussy–willow* paradigmatically inherits properties by default inheritance from the archi–sign *compound_noun*, and syntagmatically inherits from the head *willow* 'salix' and the modifier *pussy* 'felis'. Those ILEX template properties for *pussy–willow* which are not specified are completed by unification via inheritance: either percolated up from the head *willow* or inherited from the archi–sign *compound_noun*. The LEMMA *pussy–willow* is seen to be partially rather than fully compositional in that the value for the attribute path INT|MEAN|QUALIA|RELN is specified idiosyncratically. At a higher level in the inheritance path, the value for INT|MEAN|QUALIA|RELN may be specified differently, e.g. as IS_A; the more specific value overrides the more general value.

A lexical sign which inherits *all* its INT properties from the properties of its PARTS, and its general compositional properties from its CAT attribute (such as function application, concatenation, association), and is not otherwise idiosyncratically specified for INT (i.e. has no

Table 2: Paradigmatic and syntagmatic inheritance for *pussy–willow*.

(1) Lexical sign:

$$\left[\begin{array}{ll} \text{LEMMA} & \textit{pussy–willow} \\ \text{STRUC} & \left[\begin{array}{ll} \text{CAT} & \rightarrow \textit{compound\_noun} \\ \text{PARTS} & \left[\begin{array}{l} \text{HEAD} \Downarrow \textit{willow} \\ \text{MODI} \Downarrow \textit{pussy} \end{array}\right] \end{array}\right] \\ \text{INT} & \left[\text{MEAN} \left[\begin{array}{ll} \text{QUALIA} & \left[\text{RELN RESEMBLE}\right] \\ \text{TECH} & \text{SALIX CAPREA PENDULA} \end{array}\right]\right] \end{array}\right]$$

(2) Lexical archi–sign:

$$\left[\begin{array}{ll} \text{LEMMA} & \textit{compound\_noun} \\ \text{STRUC} & \left[\text{CAT} \rightarrow \textit{noun}\right] \\ \text{INT} & \left[\begin{array}{ll} \text{MEAN} & \left(\begin{array}{l} \text{SEMANTICALLY\_LINK}, \\ \Uparrow\text{INT}|\text{MEAN} \\ \Uparrow\text{STRUC}|\text{PARTS}|\text{MODI}|\text{INT}|\text{MEAN} \\ \Uparrow\text{STRUC}|\text{PARTS}|\text{HEAD}|\text{INT}|\text{MEAN} \end{array}\right) \\ \text{SURF} & \left[\begin{array}{ll} \text{PHON} & \left(\begin{array}{l} \text{PROSODICALLY\_LINK}, \\ \Uparrow\text{STRUC}|\text{PARTS}|\text{MODI}|\text{INT}|\text{SURF}|\text{PHON}, \\ \Uparrow\text{STRUC}|\text{PARTS}|\text{HEAD}|\text{INT}|\text{SURF}|\text{PHON} \end{array}\right) \\ \text{ORTH} & \left(\begin{array}{l} \textit{hyphenate}, \\ \Uparrow\text{STRUC}|\text{PARTS}|\text{MODI}|\text{INT}|\text{SURF}|\text{ORTH}, \\ \Uparrow\text{STRUC}|\text{PARTS}|\text{HEAD}|\text{INT}|\text{SURF}|\text{ORTH} \end{array}\right) \end{array}\right] \end{array}\right] \end{array}\right]$$

default–overrides), is *totally compositional*.

A lexical sign which inherits *none* of its INT properties from properties of PARTS, all of these properties being specified idiosyncratically, is *totally noncompositional*. An extreme example of a sign which is totally non–compositional is a hesitation particle interjection such as 'er', i.e. /əː/; however, even this is debatable because the /ə/ is associated with a flat stylised intonation and together with this intonation has a 'phatic' channel–sustaining function.

A lexical sign which inherits *some* of its INT properties from properties of its PARTS, others being specified idiosyncratically, or which does not inherit compositional properties from the most general subsumer in the inheritance graph, is *partially compositional*.

The totally compositional and totally non–compositional or idiosyncratic cases are 'ideal types' corresponding to absolute or zero adherence to Frege's Principle. Lexical signs, in the general case, exhibit varying degrees of partial compositionality (or, conversely, *exceptionality* or *irregularity*), measurable by their depth in the type inheritance hierarchy. The concept of a scale of compositionality applies not just to semantics, but also to surface form.

For example, orthography is partially compositional: in *ladies' fingers* 'okra', the ORTH of the plural *fingers* is a function of the ORTH of the PARTS *finger* and *s*, but the ORTH of the genitive plural *ladies'* is a more specific function of the PARTS *lady* and *s*.

The PHON property is also only partly compositional. The plural /fɪŋgəz/ appears at first sight to be a general compositional function of the PARTS /fɪŋgə/ and /z/, namely concatenation (interpreted as temporal immediate precedence: /fɪŋgə/ $\prec^{\circ}$ /z/). However, the compositional function is in fact a more complex morphophonological function which is sensitive to the MANNER and VOICING specifications of the stem–final segment. Morphophonology therefore defines a scale of partial phonetic compositionality.

Perhaps the most interesting cases are the MEAN–SURF parallels in partial compositionality which characterise diachronically lexicalised compounds. For example, the ORTH of *dustman* is

perfectly compositional. The MEAN (in informal terms) is, however, only partially compositional (e.g. 'municipally employed professional refuse collector'), whereby

1. the collective noun 'refuse' (rubbish, garbage) has a very general semantic paradigmatic relation to 'dust',
2. the deverbal derivation 'collector' characteristically denotes a male agent,
3. further details are elliptical, a typical feature of compounds.

But the PHON property is also only partially compositional: /dʌsmən/, and not /dʌstmæn/, i.e. the final consonant of /dʌst/ is elided and the vowel of /mæn/ is weakened. Partial compositionality of this kind has to be specified idiosyncratically for each lexical item concerned; this is the kind of partial compositionality which, on the diachronic dimension, has led in time to the total non–compositionality of PHON and ORTH with words like *woman* = $f_{diachron}$(*wife*,*man*) or *husband* = $f_{diachron}$(*house*,*bond*).

## 2.6 Lexical items as structural semiotic types

The notion *lexical item* is used to cover any lexical sign type but also other inventorisable items such as affixes and phonemes, whose lexical status in linguistics is controversial. Some examples of structural semiotic characterisations of these items are given below.
*Phoneme:* A minimal sign with no MEAN specification and no PARTS (*pace* proponents of distinctive features and autosegmental lattices; sub–morphemic morphological composition is not at issue here).
*Morpheme:* A sign with elementary MEAN specification, the PHON of whose PARTS is specified for a concatenation of *phonemes*.
*Lexical morpheme, lexical base, root:* A *simple stem*; a *grammatical morpheme* is an *affix*.
*Cranberry morph:* A *morpheme* with no specification for MEAN.
*Word:* A *word* (in English) is specified recursively for all four structural semiotic properties:

1. an uninflectable root, or
2. an inflectable root with an inflection, or
3. a derivation terminated by an inflected suffix, or
4. a compound terminated by a word.

*Stem:* A lexical root, or an item to which an affix is attached to form a derivation or an inflection, or to which a word or another stem is attached to form a compound word.
*Derivation:* A complex *stem* consisting of a single *root* attached to an *affix*; the type *affix* covers prefixes, suffixes, infixes, interfixes, introfixes (intercalations), superfixes, and 'attached to' covers the relevant compositional part–part operations.
*Compound:* A complex *stem* consisting of more than one *root*, each of which may be the centre of a *derivation* and may be inflected; a compound *word* must terminate in an inflected *root* or an inflected *derivational suffix*.
*Phrasal idiom:* A lexical sign licensed by the principles and rules of sentence structure, with some PARTS unspecified according to the *frozenness hierarchy* of idiomaticity.
*Lexical prosody:* A *superfix* item with semiotic properties like those of *phonemes* or *morphemes*, but which is not concatenated but prosodically associated with other phonemes or morphemes. Prosodic association is interpreted as *temporal overlap* ($X \circ Y$) of phonetic events, while concatenation is interpreted as *immediate precedence* ($X \prec^{\circ} Y$) of temporal events [Carson–Berndsen 1993]. A more general relation of *precedence* ($X \prec Y$) is often used.
*Nonce word:* A sign licensed by the *word* constraints, but not inventarised as a lexical sign.
*Phrase, sentence:* A sign licensed by the *phrasal idiom* constraints, but not inventarised as a lexical sign.

In the view represented by the ILEX model, all sign types are *grounded in lexical signs* of the corresponding ranks. Morphology is thus seen as the discipline dealing with generalisations over

lexicalised words, syntax in the traditional sense of the term is seen as the discipline dealing with generalisations over phrasal idioms, and so on.

## 3 A selection of English noun compound types

Four of the main kinds of compound noun in English (*tatpurusa*, *bahuvrihi*, *dvandva*, *synthetic*) will suffice to demonstrate the ILEX approach.

*Tatpurusa (endocentric) compounds:* In endocentric or tatpurusa compounds, the MEAN of the whole is subsumed by the MEAN of the HEAD of the PARTS. A milk–bottle is a bottle, a mouse–trap is a trap: MEAN(*bottle*) $\sqsubseteq$ MEAN(*milk–bottle*), MEAN(*trap*) $\sqsubseteq$ MEAN(*mouse–trap*).

There are metaphorical variants: a pineapple is not an apple, but functionally similar or jocularly relatable to an apple (maybe when seen from a considerable distance or eaten blindfolded after a hot curry). The MEAN of *apple* still subsumes the MEAN of *pineapple*; the MEAN of both is subsumed by the MEAN of *fruit*. The inheritance structure of 'pineapple' is very similar to that of 'pussy–willow', illustrated above, but but with a metapor relation RESEMBLE which applies both to the head and the modifier ('something like an apple which grows on something like a pine').

*Bahuvrihi (exocentric) compounds:* In bahuvrihi compounds, the MEAN of the whole is not subsumed by the MEAN of the HEAD of the PARTS, but by an elliptical 'understood' semantic category.

The simplest kinds of exocentric compound are items such as 'redskin' or 'longlegs', paraphasable informally as 'SOMEONE who will typically HAVE *skin* which is kinda *red*' and 'SOMEONE who will typically HAVE *legs* which are kinda *long*', with a 'has property' relation. Capitalisation indicates elliptical terms, parentheses indicate elliptical relations which are characteristic of the kind of compound concerned, italics indicate overt components. Capitalised and bracketed items are the largest factors in the partial compositionality of exocentric compounds.

A more complex type is *pickpocket*, i.e. 'SOMEONE who will typically professionally surreptiously *pick*[=extract] VALUABLES from someone else's *pocket*'. Exocentric compounds are modelled with more deeply nested inheritance structures than endocentric compounds.

*Dvandva (coordinate) compounds:* The parts of coordinate compounds occur in a fixed order, and are morphologically headed, but semantically have no head–modifier structure. The functor is, basically, conjunction. Examples of this relatively simple type are 'fighter–bomber', which is both a fighter and a bomber.

*Synthetic compounds:* The second element of a synthetic compound a derived noun whose ending enters into the same semantic construction as its stem and the preceding noun. Examples of this type are *busdriver*, *screwdriver*. The ORTH derivational structure of *busdriver* is bracketed as

$$\begin{aligned}&\text{ORTH}(\textit{busdriver}) = \\ &\text{CONCAT}_{orth}(\text{ORTH}(\textit{bus}),\text{ORTH}(\textit{driver})) = \\ &\text{CONCAT}_{orth}(\text{ORTH}(\textit{bus}),(\text{CONCAT}_{orth}(\text{ORTH}(\textit{drive}),\text{ORTH}(\textit{er})))).\end{aligned}$$

However, the MEAN structure is bracketed differently (omitting some details):

$$\begin{aligned}&\text{MEAN}(\textit{busdriver}) = \\ &\lambda x\ (\text{SEMANTICALLY\_LINK}(\text{MEAN}(\textit{drive}),(\text{MEAN}(x),\text{MEAN}(\textit{bus}))))\end{aligned}$$

Some apparent synthetic compounds involve so–called *bracketing paradoxes*, which can be explained as different compositional structures defined for SURF and MEAN attributes. One classical case has the semantic bracketing ((*transformation al grammar*) *ian*), i.e.

$\lambda x$(SEMANTICALLY_LINK(PROFESSIONALLY_PRODUCE,
MEAN('x'),
SEMANTICALLY_LINK(MEAN('al'),
MEAN('transformation')),
MEAN('grammar'))))

versus the morphological bracketing ((*transformation al*) (*grammar ian*)).

# 4 The DATR formalism

## 4.1 Theories and models

A *theory* such as the AVM–based account of English compounds sketched above, may simultaneously describe any number of *models*. A model may be formal, such as a set–theoretic representation of an empirical domain, or more informal, as is generally the case in descriptive linguistics, formulated in plain text enriched with symbols and line drawings. A theory is simply a subset of sentences in a formalism for which a model exists in terms of which the sentences can be interpreted.

One kind of formal model for a theory is an 'implementation', i.e. an interpretation of the theory in terms of an operational knowledge representation language or programming language. This is actually a special case of a more general kind of formal interpretation; interpretations for AVMs have been given, for example, in terms of finite state automata (see [Kasper & Rounds 1986]). An interpretation of a theory in terms of a different but perhaps more well–known formalism permits conclusions to be drawn about whether the theory is complete (describes all it is supposed to describe) and sound (does not describe anything it is not supposed to describe). If the interpretation function is bijective, then in principle the model could be regarded as the theory and the theory as the model; this is then just a question of perspective.

In this sense, the lexical representation formalism DATR will be used to provide an operational model for the theory which permits quick consistency checking of complex theories by the automatic deduction of hypotheses. Descriptions in DATR are, however, generally referred to as 'theories'.

DATR 'theories', used here as 'operational models' for AVM theories, are sets of DATR sentences. DATR sentences are pairs of a *node* and a set of equations, each of which is a pair of an attribute path and a value.

In the ILEX approach, therefore, a lexicon is an AVM theory which describes an operational model formulated in DATR; this model can itself be seen as an empirical theory which is interpreted (like the AVM theory) by an empirical model with observationally identifiable categories.

## 4.2 DATR syntax

The syntax of DATR expresses three kinds of hierarchical structure:

1. Syntagmatic:
   (a) Nested attribute value structures (here used to represent ID relations between and property assignment to signs),
   (b) Hierarchies of sequences, with property percolation through the hierarchy expressed by 'local inheritance', and lexical insertion expressed by 'global inheritance',
2. Paradigmatic: class inclusion (or implication) hierarchies expressed by local inheritance.

In DATR, nested AVMs are represented as nodes paired with conjunctions of equations. The left–hand side of each equation is an attribute path with attributes represented as atoms[4]:

[4]DATR nodes are character strings starting with an upper case character, or declared character strings; DATR atoms are either character strings starting with a lower case character, or character strings enclosed in single right quotes, or declared character strings.

```
<struc parts modi int surf>
```

The right–hand side is a sequence of value expressions which may be either atoms or inheritance descriptors. There are two main kinds of inheritance descriptor, those which denote *local inheritance* and those which denote *global inheritance*, and in each case there are three subtypes of descriptor which constrain inheritance from different positions in the inheritance hierarchy: by specification of a *node–path pair*, a *node* alone, or a *path* alone. For each of these seven cases, i.e. atomic value expressions and the three types each of local and global inheritance, there are seven inference rules.

An important feature of DATR is that paths on the right–hand side are *evaluable*, that is, they have exactly the same formal structure as an entire right–hand side sequence, and may thus contain any value expressions, not just atoms. In particular including other paths, which may in turn include nested value expressions, and so on.

A selective version of the initial example *pussy_willow*, incorporating local (paradigmatic) and global (syntagmatic) inheritance, can be rendered in DATR as follows, with the IPA transcription characters rendered in a slightly modified version of the SAMPA ASCII coding of Wells (cf. [Wells 1989]), in which '/' is used to denote lexical stress:

```
% Query definitions (node-path pairs):
% All nodes except those declared under 'hide',
% combined with all paths declared under 'show':
# hide Noun Compound_noun .
# show <int mean> <int surf> .
% Lexical entry ranks (simplex and compound nouns):
Willow:
  <>                        == Noun
  <int mean qualia reln>    == salix
  <int surf phon>           == 'w/Il@U'
  <int surf orth>           == willow.
Pussy:
  <>                        == Noun
  <int mean qualia reln>    == felis
  <int surf phon>           == 'pUsI'
  <int surf orth>           == pussy.
Pussy_willow:
  <>                        == Compound_noun
  <struc parts head>        == "Willow:<>"
  <struc parts modi>        == "Pussy:<>"
  <int mean qualia reln>    == ' RESEMBLE '
  <int surf reln orth>      == '-'.
% Paradigmatic inheritance hierarchy (<int surf reln> has default null value):
Compound_noun:
  <>                    == Noun
  <int surf reln > ==
  <int mean>            == "<int mean qualia reln>" '('
                           "<struc parts head int mean qualia reln>" ,
                           "<struc parts modi int mean qualia reln>" ')'
  <int surf>            == "<struc parts modi int surf>"
                           "<int surf reln>"
                           "<struc parts head int surf>".
Noun:
  <>                    ==
  <int mean>            == "<int mean qualia reln>".
```

The empty path, which appears as a left–hand–side under each node, is the path with no attributes specified. This is the most general path, and indicates the inheritance path to the next more general node or class. Any values which are explicitly specified in an equation associated with the current class override values of the same attributes specified at a higher (more general) node; in this case, the INT values are exhaustively specified, so only information about the category itself is locally inherited.

Information about the parts is globally inherited from each part lemma, the head *Willow* and the modifier *Pussy*. In HPSG terms, the HEAD features are inherited from the head or HEAD-DTR, and the COMP features are inherited from the modifier or COMP-DTRS.

Global inheritance means that the parts concerned are treated quite independently of each other and of the larger unit, ensuring compositionality (which can be modified if necessary for descriptive reasons).

Among the DATR equations that can be derived from the theory are the following:

```
Pussy:< int mean > = felis .
Pussy:< int surf phon > = pUsI .
Pussy:< int surf orth > = pussy .
Willow:< int mean > = salix .
Willow:< int surf phon > = w/Il@U .
Willow:< int surf orth > = willow .
Pussy_willow:< int mean > =  RESEMBLE  ( salix , felis ) .
Pussy_willow:< int surf phon > = pUsI w/Il@U .
Pussy_willow:< int surf orth > = pussy - willow .
```

Table 3: Inheritance operations.

| DATR operation | DATR notation | AVM notation |
|---|---|---|
| Local node:path inheritance | A:<b c d> | $\rightarrow$ |
| Local node inheritance | A | A special case of $\rightarrow$ |
| Local path inheritance | <b c d> | $\leftarrow$ (also a special case of $\rightarrow$) |
| Global node:path inheritance | "A:<b c d>" | $\Downarrow$ |
| Global node inheritance | "A" | Rarely used. |
| Global path inheritance | "<b c d>" | $\Uparrow$ |

The DATR inheritance rules were the starting point for the definition of the paradigmatic and syntagmatic inheritance relations used in the AVM–based theory introduced in the preceding sections. For this reason, there is a simple mapping between the inheritance and compositionality operators used in the AVM theory, and the six inheritance operations defined for DATR, though not all the DATR possibilities are exhausted in the AVM theory (see Table 3). Atomic values are basically the same in each formalism.

### 4.3 DATR rules of deduction

The DATR rules of deduction will be explained here in procedural terms (though declarative explanations may be given, see [Evans & Gazdar 1996], and [Langer 1992] for an account in terms of default unification). The inference rules are of four types: an initialisation rule, a query connection (matching) rule, a path extension rule, and finally an inference rule for each of the seven value expression types.

*Environments, initialisation and modification:* The DATR rules of deduction refer to a *local environment* and a *global environment*. Each environment consists of a pair of variables, one for evaluation of the local node–path pair, node, and path descriptors, and the other for evaluation of global node–path pair, node and path descriptors. The global environment is initialised to the value of the query node–path pair, and re–defined by the global inheritance descriptors. When the global environment is initialised and whenever it is changed, the variables in the local environment are copied into the local environment. Environment changes are encapsulated for the inheritance descriptor concerned, whether local or global, and do not affect sibling descriptors in the same sequence. However, the same local and global environments are valid for all paths at all depths of recursion in the descriptor concerned.

*Matching:* The matching of a query attribute path with the paths on the left–hand side of a DATR equation is based on two operations over the local environment and the theory, *connection* and *extension*.

*Connection:* The local environment connects with a NODE:PATH==SEQUENCE equation defined in a theory iff

1. NODE is identical to the node in the local environment,
2. PATH is a prefix of the path in the local environment, e.g.: the local environment path
   `<int mean qualia reln>`
   matches the following prefixes (whereby the identical path and the zero path both count as prefixes):
   `<int mean qualia reln>`
   `<int mean qualia>`
   `<int mean>`
   `<int>`
   `<>`
3. PATH is the longest path under NODE which is also a prefix of the path in the local environment.
   For example, given the local environment path `<int mean qualia reln>`, and two competing paths under NODE which are prefixes of this path,
   `<int mean qualia>`
   `<int mean>`
   the match is with `<int mean qualia>`: 'the longest path wins'. This principle defines *default inheritance* in DATR.

*Extension:* The path in a connected local environment consists of a matching prefix and an extension suffix (possibly zero); in the preceding example, `<int mean qualia>` is the matching prefix and `<reln>` is the extension suffix; the matched local environment can be represented by `<int mean qualia || reln>`. Extension is the concatenation of all paths in an equation (however deeply embedded, in both local and global inheritance descriptors) with the extension suffix, for example, with the local environment and matching equation

`<int mean qualia reln>`
`<int mean qualia> == Semantics:<qualia>`

The extension of the equation is

`<int mean qualia reln> == Semantics:<qualia reln>`.

The following notation will sometimes be used for clarity:

`<int mean qualia || reln> == Semantics:<qualia || reln>`.

This mechanism expresses a form of constraint propagation for orthogogonal inheritance through the inheritance network.

*Inheritance:* The right–hand side of a connected and extended equation is evaluated according to seven rules of inference or inheritance rules, one for atoms and three each for inheritance descriptors in the local and global environments. The inheritance rules define how the value expressions on the right–hand side of DATR equations are to be evaluated. Evaluation consists of finding a value for a DATR query, i.e. a node–path pair, by recursive application of the seven inference rules to the elements of sequences and evaluable paths.

*Inference rules:*

1. DATR sequences and DATR atoms:
   DATR sequences evaluate to a concatenation of the values of their parts, i.e. sequences of atoms.
   Rule I: DATR atoms evaluate to themselves.
2. DATR local inheritance:

   Rule II: Local NODE:PATH descriptor. Substitute NODE for the node and PATH (after evaluation and extension) for the path in the local environment, and connect the local environment with the theory.

Rule III: Local NODE descriptor. Substitute NODE for the node in the local environment, and connect the local environment with the theory.

Rule IV: Local PATH descriptor. Substitute PATH (after evaluation and extension) for the local environment path, and connect the local environment with the theory.

3. DATR global inheritance:

   Rule V: Global NODE:PATH descriptor. Substitute NODE for the node in the global environments, and PATH (after evaluation and extension) for the global environment path; copy the global environment to the local environment and connect the local environment with the theory.

   Rule VI: Global NODE descriptor. Substitute NODE for the node in the global environment; copy the global environment to the local environment and connect the local environment to the theory.

   Rule VII: Global PATH descriptor. Substitute PATH (after evaluation and extension) for the global environment path; copy the global environment to the local environment and connect the local environment to the theory.

The following is an example[5] of the inference steps involved in deriving the DATR sentence `Pussy_willow:< int mean > = RESEMBLE(salix,felis).`

```
=0,0,0> LOCAL Pussy_willow:< || int mean > == Compound_noun
        GLOBAL Pussy_willow:< int mean >
RULE III.(NODE)
=1,0,0> LOCAL Compound_noun:< int mean > == "< int mean qualia reln >"
              ( "< struc parts head int mean qualia reln >" ,
              "< struc parts modi int mean qualia reln >" )
        GLOBAL Pussy_willow:< int mean >
RULE VII.(GPATH)
=2,0,0> LOCAL Pussy_willow:< int mean qualia reln > ==  RESEMBLE
        GLOBAL Pussy_willow:< int mean qualia reln >
RULE I.(ATOM)
 RESEMBLE
RULE I.(ATOM)
(
RULE VII.(GPATH)
=2,0,2> LOCAL Pussy_willow:< struc parts head || int mean qualia reln > == "Willow: <  > "
        GLOBAL Pussy_willow:< struc parts head int mean qualia reln >
RULE V.(GNODE:GPATH)
=3,0,0> LOCAL Willow:< int mean qualia reln > == salix
        GLOBAL Willow:< int mean qualia reln >
RULE I.(ATOM)
salix
RULE I.(ATOM)
,
RULE VII.(GPATH)
=2,0,4> LOCAL Pussy_willow:< struc parts modi || int mean qualia reln > == "Pussy: <  > "
        GLOBAL Pussy_willow:< struc parts modi int mean qualia reln >
RULE V.(GNODE:GPATH)
=3,0,0> LOCAL Pussy:< int mean qualia reln > == felis
        GLOBAL Pussy:< int mean qualia reln >
RULE I.(ATOM)
felis
RULE I.(ATOM)
)
[Query 4 (12 Inferences)] Pussy_willow:< int mean > =  RESEMBLE (salix,felis).
```

# 5 An operational DATR model for English compounds

## 5.1 Descriptive scope of the model

The model described in the following pages is constructed on the lines outlined in the preceding sections, with a few minor modifications; for example, the AVMs operationalised in the model

[5] The derivation was produced with the ZDATR interpreter, [Schillo 1996]. The numbers indicate depth of local inheritance, path inheritance, and position in the right–hand–side sequence; the '‖' sequence separates the matched prefix of the local environment from the remaining suffix, and the RULE number refers to the DATR inference rule which applies to the current value expression under evaluation.

are flatter, and the descriptive scope of the model is much broader, but the model contains additional relatively informal attribute specifications. There are also many possible 'style options' for modelling in DATR, which will not be discussed here.

The descriptive scope of the model includes the following:

1. simplexes and inflection;
2. compound types tatpurusa, dvandva and bahuvrihi;
3. informal compositional semantic interpretation;
4. phonetic interpretation (pre- and postmorphophonemic representations);
5. orthographic interpretation (pre- and postmorphographemic representations);
6. compositionality generalised for meaning and surface interpretation at all ranks;
7. morphophonological finite state transducer;
8. morphographemic finite state transducer.

Table 4: Terms used in the DATR model.

| Term: | Description: |
|---|---|
| cat | category, cf. 'CAT' in AVM |
| compound | morphological category specification |
| graph | morphographemic interpretation |
| head | cf. 'HEAD' in AVM |
| mass | value for mass noun |
| modi | cf. 'MODI' in AVM |
| morph | morphological attribute |
| operator | compositionality operator, cf. 'RELN' in AVM |
| orth | orthographic (post-morphographemic) interpretation |
| phon | phonetic interpretation (including stress marks) |
| plur | plural inflection |
| mean | semantic interpretation, cf. AVM 'MEAN' |
| sing | singular inflection (default value) |
| stem | morphological category specification |
| stress | lexical stress |
| surf | surface interpretation (default is morphophonemic) |
| plain | inflectional status of modifier |

The morphophonological and morphographemic finite state transducers demonstrate how one formalism can be used to operationalise different theories, in this case not as an AVM based theory modelled with directed *acyclic* graphs, but as automata of the kind used in two-level morphology [Koskenniemi 1983], modelled with directed *cyclic* graphs. The terms used are listed in Table 4.

Not all aspects of the model can be discussed in the present context, but some of the lexical specifications which can be inferred by application of the inheritance rules are illustrated here with the synthetic compound *busdriver*:

```
Busdriver:<surf graph>           = bus-drive+er.
Busdriver:<surf graph orth>      = bus-driver.
Busdriver:<surf>                 = bVs#draiv+@.
Busdriver:<surf phon>            = //bVs/draiv@.
Busdriver:<mean>                 = {{{one_OF_{agent|instrument}}
                                     _CAN_{{action_OF_{move_vehicle}}}}
                                           _AFFECT_{{one_OF_{public_road_vehicle}}}}.
Busdriver:<plur surf graph>      = bus-drive+er#+s.
Busdriver:<plur surf graph orth> = bus-drivers.
Busdriver:<plur surf>            = bVs#draiv+@#+/Z.
Busdriver:<plur surf phon>       = //bVs/draiv@z.
Busdriver:<plur mean>            = {{{more_than_one_OF_{agent|instrument}}
                                     _CAN_{{action_OF_{move_vehicle}}}}
                                           _AFFECT_{{one_OF_{public_road_vehicle}}}.
```

## 5.2 DATR model: lexicon extract

*Simplexes:*

```
Pale:
  <>                      == Adjective
  <modi mean>             == rather_white
  <modi surf graph>       == p a l e
  <modi surf>             == p e I l.
Face:
  <>                      == Noun
  <modi mean>             == front_of_head
  <modi surf graph>       == f a c e
  <modi surf>             == f e I s.
```

*Derivational suffix:*

```
Er:
  <>                      == Noun_suffix
  <modi mean>             == agent|instrument
  <modi surf graph>       == e r
  <modi surf>             == @.
```

*Derivations:*

```
Bomber:
  <>                      == Noun_derivation
  <modi>                  == "Bomb:<plain>"
  <head>                  == "Er:<>"
  <operator mean>         == CAN.
Driver:
  <>                      == Noun_derivation
  <modi>                  == "Drive:<plain>"
  <head>                  == "Er:<>"
  <operator mean>         == CAN.
```

*Standard tatpurusa representation:*

```
Mousetrap:
  <>                      == Noun_compound
  <operator mean>         == FOR
  <modi>                  == "Mouse:<plain>"
  <head>                  == "Trap:<>".
Mousetrapcheese:
  <>                      == Noun_compound
  <operator mean>         == FOR
  <operator surf graph>   == _
  <modi>                  == "Mousetrap:<plain>"
  <head>                  == "Cheese:<>".
```

*Two–stage bahuvrihi representation:*

```
Paleface2:
  <>                      == Noun_compound:<>
  <operator mean>         == HASPROP
  <modi>                  == "Paleface:<plain>"
  <head mean>             == someone.
Paleface:
  <>                      == Noun_compound:<>
  <operator mean>         == IS
  <modi>                  == "Pale:<plain>"
  <head>                  == "Face:<>".
```

*Dvandva representation:*

```
Fighterbomber:
  <>                      == Noun_compound
  <operator mean>         == AS-WELL-AS
  <operator surf graph>   == -
  <modi>                  == "Fighter:<plain>"
  <head>                  == "Bomber:<>".
```

## 5.3 Noun inheritance hierarchy

The top–level node, *Sign*, is completely unspecified and has the null value. At the *Word* node, information about inflectional neutralisation and constraints on the interpretation mapping is specified. For reasons which cannot be argued here, the default interpretation for HEAD SURF is the null value. The extrinsic inflection category 'plur' is apparently modelled with an attribute; this hybrid construction combines DATR inter–level transducer modelling with DATR AVM modelling, and cannot be explained further here.

```
Sign:
<>                        == .
Word:
<>                        == Sign
<plur surf phon>          == Morphophon:<Interpretation>
<surf phon>               == Morphophon:<Interpretation>
<plur surf graph orth>    == Morphograph:<Interpretation>
<surf graph orth>         == Morphograph:<Interpretation>
<surf>                    == Interpretation
<plur surf>               == Interpretation
<mean>                    == Interpretation
<plur mean>               == Interpretation
<plain>                   == Interpretation
<plain plur mean>         == <plain mean>
<operator sing>           == "<operator>"
<operator plur>           == "<operator>"
<operator plain mean>     == "<operator mean>"
<operator mean>           == OF
<sing>                    == <>
<modi sing>               == "<modi>"
<modi plur>               == "<modi>"
<head plur>               == "<head>"
<head surf>               ==
<head mean>               == <indiv "<mean indiv>">
<indiv exists>            == APPLICATION
<indiv mass>              == some
<indiv>                   == one
<head plur mean>          == more_than_one
Noun:
<>                        == Word
<cat surf>                == noun
<plain sing>              == <plain>
<plain plur>              == <plain sing>
<head plur surf graph>    == #+ s
<head plur surf>          == #+ /Z .
Noun_compound:
<>                        == Noun
<cat morph>               == compound
<plain operator>          == "<operator>"
<operator surf>           == #.
Noun_derivation:
<>                        == Noun
<cat morph>               == derivation
<operator surf>           == +.
```

## 5.4 Co–interpretation for semantics and surface form

Semantic and phonetic interpretation are, in principle, treated identically, with a number of specific constraints concerned with the assignment of recursive brackets to MEAN and the entirely analogous recursive assignment of lexical stress to SURF PHON, depending on the morphological category.

```
Interpretation:
<>                        == First Operator Second
<plain>                   == <>.
First:
<>                        == StressOp "<modi>"
<mean>                    == { "<head mean>" _
<plur mean>               == { "<head plur mean>" _ .
Second:
<>                        == "<head>"
<mean>                    == _ { "<modi mean>" } }
<plur mean>               == _ { "<modi plur mean>" } }.
Operator:
<>                        == "<operator>".
StressOp:
<>                        ==
<surf phon>               == <stress "<cat morph>">
<plur surf phon>          == <surf phon>
<stress stem>             == /
<stress compound>         == /
<stress>                  == .
```

## 5.5 Surface interpretation: morphophonemic and morphographemic mapping

The morphographemic transducer maps characters from the lexical level to the post–lexical level taking into account specific restrictions on character mapping at inflectional boundaries.

In the general (default) case (the 'elsewhere condition'), a DATR variable '$char' defines the identity mapping. Boundary diacritics are deleted. Theoretically this traditional use of boundary diacritics is not optimal, but a more adequate treatment would go beyond the scope of the paper.

```
Morphograph:
  <>       ==
  <+>      == <>
  <#+>     == <>
  <#>      == <>
  <##>     == <>
  <$char>  == $char <>
  <e + e>  == e <>
  <e #+ e> == e <>
  <y + s>  == i e s <>
  <y #+ s> == i e s <>
  <s #+ s> == s e s <>.
```

Very much like the morphographemic mapping, in the morphophonemic mapping, the plural morphophoneme '/Z' is realised dependent on its left context as one of /s, z, ɪz/. Other segments, another case of the 'elsewhere condition', are realised unchanged using a DATR variable '$phon'. Phonemes and (as with spelling) boundary diacritics are not the theoretically optimal choice for phonetic interpretation, but a full feature lattice treatment is not possible in the present context.

```
Morphophon:
  <>          ==
  <$phon>     == $phon
  <+>         == <>
  <#+>        == <>
  <#>         == <>
  <##>        == <>
  <p #+ /Z>   == p s <>
  <t #+ /Z>   == t s <>
  <k #+ /Z>   == k s <>
  <f #+ /Z>   == f s <>
  <T #+ /Z>   == T s <>
  <s #+ /Z>   == s I z <>
  <S #+ /Z>   == S I z <>
  <z #+ /Z>   == z I z <>
  <Z #+ /Z>   == Z I z <>
  </Z>        == z <>.
```

## 6 A sample analysis

The complexity of the theory is shown by derivations generated by the operational DATR model. In order to derive the post-lexical phonetic representation of the synthetic compound *busdriver*, 173 DATR inferences (rule applications) are required, in order to derive the simplex plural form *buses*, 44 inferences are needed. It will be sufficient to illustrate the process using a simplex plural, *buses*, as the general definition of head–modifier based on interpretative compositionality covers all morphological ranks.

*Initial local inheritance:*

```
=0,0,0> LOCAL Bus:< || plur surf phon > == Noun
        GLOBAL Bus:< plur surf phon >
RULE III.(NODE)
=1,0,0> LOCAL Noun:< || plur surf phon > == Word
        GLOBAL Bus:< plur surf phon >
RULE III.(NODE)
=2,0,0> LOCAL Word:< plur surf phon > == Morphophon:< Interpretation >
        GLOBAL Bus:< plur surf phon >
RULE II.(NODE/PATH)
RULE III.(NODE)
```

*Assignment of linear precedence and lexical stress to inflected word:*

```
=3,1,0> LOCAL Interpretation:< || plur surf phon > == First Operator Second
        GLOBAL Bus:< plur surf phon >
RULE III.(NODE)
=4,1,0> LOCAL First:< || plur surf phon > == StressOp "< modi >"
        GLOBAL Bus:< plur surf phon >
RULE III.(NODE)
=5,1,0> LOCAL StressOp:< plur surf phon > == < surf phon >
```

```
        GLOBAL Bus:< plur surf phon >
RULE IV.(PATH)
=6,1,0> LOCAL StressOp:< surf phon > == < stress "< cat morph >" >
        GLOBAL Bus:< plur surf phon >
RULE IV.(PATH)
RULE VII.(GPATH)
=7,2,0> LOCAL Bus:< || cat morph > == Noun
        GLOBAL Bus:< cat morph >
RULE III.(NODE)
=8,2,0> LOCAL Noun:< || cat morph > == Word
        GLOBAL Bus:< cat morph >
RULE III.(NODE)
=9,2,0> LOCAL Word:< || cat morph > == Sign
        GLOBAL Bus:< cat morph >
RULE III.(NODE)
=10,2,0> LOCAL Sign:< || cat morph > ==
        GLOBAL Bus:< cat morph >
RULE I.(ATOM)
=7,1,0> LOCAL StressOp:< stress > ==
        GLOBAL Bus:< plur surf phon >
RULE I.(ATOM)
RULE VII.(GPATH)
```

*Assignment of morphophonemic (lexical) representation:*

```
=5,1,1> LOCAL Bus:< || modi plur surf phon > == Noun
        GLOBAL Bus:< modi plur surf phon >
RULE III.(NODE)
=6,1,0> LOCAL Noun:< || modi plur surf phon > == Word
        GLOBAL Bus:< modi plur surf phon >
RULE III.(NODE)
=7,1,0> LOCAL Word:< modi plur || surf phon > == "< modi >"
        GLOBAL Bus:< modi plur surf phon >
RULE VII.(GPATH)
=8,1,0> LOCAL Bus:< modi surf || phon > == b V s
        GLOBAL Bus:< modi surf phon >
RULE I.(ATOM)
b
RULE I.(ATOM)
V
RULE I.(ATOM)
s
RULE III.(NODE)
```

*Interpretation of (null) inflection operator:*

```
=4,1,1> LOCAL Operator:< || plur surf phon > == "< operator >"
        GLOBAL Bus:< plur surf phon >
RULE VII.(GPATH)
=5,1,0> LOCAL Bus:< || operator plur surf phon > == Noun
        GLOBAL Bus:< operator plur surf phon >
RULE III.(NODE)
=6,1,0> LOCAL Noun:< || operator plur surf phon > == Word
        GLOBAL Bus:< operator plur surf phon >
RULE III.(NODE)
=7,1,0> LOCAL Word:< operator plur || surf phon > == "< operator >"
        GLOBAL Bus:< operator plur surf phon >
RULE VII.(GPATH)
=8,1,0> LOCAL Bus:< || operator surf phon > == Noun
        GLOBAL Bus:< operator surf phon >
RULE III.(NODE)
=9,1,0> LOCAL Noun:< || operator surf phon > == Word
        GLOBAL Bus:< operator surf phon >
RULE III.(NODE)
=10,1,0> LOCAL Word:< || operator surf phon > == Sign
        GLOBAL Bus:< operator surf phon >
RULE III.(NODE)
=11,1,0> LOCAL Sign:< || operator surf phon > ==
        GLOBAL Bus:< operator surf phon >
RULE I.(ATOM)
RULE III.(NODE)
=4,1,2> LOCAL Second:< || plur surf phon > == "< head >"
        GLOBAL Bus:< plur surf phon >
RULE VII.(GPATH)
```

*Assignment of plural morphophoneme:*

```
=5,1,0> LOCAL Bus:< || head plur surf phon > == Noun
        GLOBAL Bus:< head plur surf phon >
```

```
RULE III.(NODE)
=6,1,0> LOCAL Noun:< head plur surf || phon > == #+ /Z
        GLOBAL Bus:< head plur surf phon >
RULE I.(ATOM)
#+
RULE I.(ATOM)
/Z
```

*Morphophonemic mapping:*

```
=3,0,0> LOCAL Morphophon:< b || V s #+ /Z > == b <  >
        GLOBAL Bus:< plur surf phon >
RULE I.(ATOM)
b
RULE IV.(PATH)
=4,0,1> LOCAL Morphophon:< V || s #+ /Z > == V <  >
        GLOBAL Bus:< plur surf phon >
RULE I.(ATOM)
V
RULE IV.(PATH)
=5,0,1> LOCAL Morphophon:< s #+ /Z > == s I z <  >
        GLOBAL Bus:< plur surf phon >
RULE I.(ATOM)
s
RULE I.(ATOM)
I
RULE I.(ATOM)
z
RULE IV.(PATH)
=6,0,3> LOCAL Morphophon:< > ==
        GLOBAL Bus:< plur surf phon >
RULE I.(ATOM)
[Query 49 (44 Inferences)] Bus:< plur surf phon > = bVsIz.
```

# 7 Discussion and prospects

The goal of this contribution to Inheritance Lexicon Theory is to take a step towards a solution of problems such as the integration of morphology, idioms, and lexical prosody, to introduce a general notion of compositional sign and compositional co–interpretation for surface and semantic interpretation at all structural ranks.

In pursuing this goal, the concept of inheritance was introduced and used to account for both paradigmatic and syntagmatic generalisations, including ID and LP relations and morphographemic and morphophonemic mappings. Starting with a theory based on attribute–value matrices, a formal description of English compounds was outlined. As a heuristic device for investigating the complex implications of the theory, a technique for developing an operational DATR model for the theory was outlined, and an operational DATR model was presented in some detail. An explicit mapping from the theory to the model was not defined. Many key aspects of lexicalisation and compositionality remain to be discussed, for example the question of whether embedded complex stems in compounds are lexicalised (e.g. the instrumental *driver* in *screwdriver* as opposed to the agentive *driver* in *busdriver*).

But the results demonstrate the flexibility of the ILEX methodology, and provide a vivid illustration both of the complexity of natural language, in terms of the length and depth of the derivation of interpretative representations. But the results also demonstrate the elegance and simplicity of natural language lexical items, in terms of highly underspecified lemma entries. The operational model demonstrates for the first time that it is possible to integrate a variety of different facts about compositionality in the lexicon in a homogeneous, theoretically well–founded and computationally tractable fashion, without sacrificing linguistic perspicuity.

As well as adding a dimension of compositionality to the basic structuralist concept of a sign, the multiply linked lattice structures of inheritance lexicon methodology contribute towards a new interpretation of another basic structuralist position in respect of the structure of language: *un système où tout se tient.*

## References


[Bleiching1992] Bleiching, D. 1992. Prosodisches Wissen im Lexikon. In: G. Görz, ed., *KONVENS 92.* Berlin: Springer–Verlag. 59–68.

[Bleiching 1994] Bleiching, D. 1994. Integration von Morphophonologie und Prosodie in ein hierarchisches Lexikon. In: H. Trost, ed., *KONVENS 94, Wien.* Berlin: Springer–Verlag. 32–41.

[Bleiching, Drexel & Gibbon 1996] Bleiching, D., G. Drexel & D. Gibbon 1996. Ein Synkretismusmodell für die deutsche Morphologie. In: D. Gibbon, ed., *Natural Language Processing and Speech Technology: Results of the 3rd KONVENS Conference, Bielefeld 1996.* Berlin: Mouton de Gruyter. 237–248.

[Cahill 1993] Cahill, L. 1993. Morphonology in the lexicon. In: *Sixth Conference of the European Chapter of the Association for Computational Linguistics, Utrecht.* 87–96.

[Carson–Berndsen 1993] Carson–Berndsen, J. 1993. *Time Map Phonology and the Projection Problem in Spoken Language Recognition.* Ph.D. thesis, U. Bielefeld.

[Chomsky 1965] Chomsky1965. *Aspects of the Theory of Syntax.* Cambridge: MIT Press.

[Corbett & Fraser 1995] Corbett, G. G. & N. M. Fraser 1995. Network morphology: a DATR account of Russian nominal inflection. *Journal of Linguistics* 29, 113–142.

[Cresswell 1973] Cresswell, M. 1973. *Logics and Languages.* London: Methuen.

[Daelemans 1987] Daelemans, W. 1987. *Studies in Language Technology: An Object–Oriented Computer Model of Morphophonological Aspects of Dutch.* Ph.D. thesis, U Leuven.

[Evans & Gazdar 1996] Evans, R. & G. Gazdar 1996. DATR: A language for lexical knowledge representation. In: *Computational Linguistics* 22:2, 167–216.

[Flickinger 1987] Flickinger, D. 1987. *Lexical Rules in the Hierarchical Lexicon.* Ph.D. thesis, Stanford University.

[Gibbon 1990] Gibbon, D. 1990. Prosodic association by template inheritance. In: W. Daelemans & G. Gazdar, eds., *Proceedings of the Workshop on Inheritance in Natural Language Processing.* Tilburg: Institute for Language Technology. 65–81.

[Gibbon 1992] Gibbon, D. 1992. ILEX: A linguistic approach to computational lexicology. In: U. Klenk, ed., *Computatio Linguae*, Beiheft zur *Zeitschrift für Dialektologie und Linguistik.* Stuttgart: Steiner. 32–53.

[Gibbon, Tseng & Folikpo 1997] Gibbon, D. S.–c. Tseng & K. Folikpo 1997. Prosodic Inheritance and Phonetic Interpretation: lexical tone. To appear.

[Kasper & Rounds 1986] Kasper, R. & W. Rounds 1986. A logical semantics for feature structures. In: *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics.* Morristown: ACL.

[Koskenniemi 1983] Koskenniemi, Kimmo 1983. *Two–level Morphology.* Ph.D. thesis, U Helsinki.

[Langer 1992] Langer, H. 1992. *DELASOUL: Eine constraintbasierte Beschreibungssprache f"ur lexikalische Repr"asentationen.* Technical Report ASL–TR–26–92/UBI. University of Bielefeld.

[Pollard & Sag 1987] Pollard, C. & I. A. Sag 1987. *Information–based Syntax and Semantics. Volume I: Fundamentals.*. Stanford: CSLI.

[Pollard & Sag 1994] Pollard, C. & I. A. Sag 1994. *Head–Driven Phrase Structure Grammar.*. Chicago: U Chicago Press.

[Pustejovsky 1995] Pustejovsky, J. 1995. *The Generative Lexicon.* Cambridge: MIT Press.

[Reinhard & Gibbon 1991] Reinhard, S. & D. Gibbon 1991. Prosodic association and template inheritance. In: *Fifth Conference of the European Chapter of the Association for Computational Linguistics, Berlin*, 131–136.

[Rosch 1978] Rosch, E. H. 1978. Principles of categorization. In: E. Rosch & B. Lloyd, eds., *Cognition and Categorization.* Hillsdale, N.J.: Erlbaum Associates. 27-48.

[Schillo 1996] Schillo, C. 1996. *A DATR Implementation in C: ZDATR Manual Version 1.0.* Technical Report, University of Bielefeld.

[Shieber 1986] Shieber, S. 1986. *An Introduction to Unification–Based Approaches to Grammar.* Stanford: CSLI.

[Wells 1989] Wells, J. C. 1989. Computer–coded phonemic notation of individual languages of the European Community. In: *Journal of the IPA* 19:1, 31–54.

[Zwicky 1993] Zwicky, A. 1993. Heads, bases, and functors. In: G.G. Corbett, N. Fraser, & S. McGlashan, eds., *Heads in grammatical theory.* Cambridge: Cambridge U Press. 292-315.