

# Lexicalization of Context (HPSG-97 draft)

Graham Wilcock\*

Centre for Computational Linguistics  
University of Manchester Institute of Science and Technology  
PO Box 88, Manchester M60 1QD, United Kingdom  
graham@ccl.umist.ac.uk

## Abstract

We propose lexical rather than phrasal amalgamation of contextual features. On a theoretical level, the lexicalization of context naturally follows other recent revisions concerned with lexicalization of set-valued features (nonlocal features, quantifier storage). On a computational level, it offers advantages in using HPSG with head-driven generation algorithms. On a linguistic level, we sketch a lexical context approach to register variation in English relative clauses.

## 1 Introduction

We propose the lexicalization of contextual features. Instead of *phrasal* amalgamation of context information from a phrase's daughters, as specified in the Contextual Consistency Principle (Pollard and Sag, 1994), we propose *lexical* amalgamation of context information from a word's arguments, specified as a lexical constraint. The Contextual Consistency Principle is replaced by a new Contextual Head Inheritance Principle (CHIP): a phrase's CONTEXT is token-identical to that of its contextual-semantic head daughter. The paper gives theoretical, computational and linguistic motivations.

In Section 2, we show that this revision naturally follows other recent revisions in HPSG theory concerned with set-valued features, and brings the handling of context features into line on a theoretical level with the lexicalization of nonlocal features and of quantifier storage and retrieval. In Section 3 we argue that the lexicalization of both context and quantifier storage is advantageous on a computational level in using HPSG with *head-driven generation algorithms* (bottom-up generation or semantic head-driven generation). In Section 4 we briefly look at the lexicalization of context in linguistic analysis, sketching an approach to *register variation* within

Sag's analysis of English relative clauses without empty categories.

## 2 Set-valued features

### 2.1 Phrasal amalgamation

Three HPSG principles - the Nonlocal Feature Principle, the Quantifier Inheritance Principle, and the Contextual Consistency Principle - specify constraints on certain set-valued features of a phrase and the equivalent set-valued features of the daughters. In (Pollard and Sag, 1994), these constraints are specified on *phrases*, that is, the values of the relevant features of all the daughters of a phrase are amalgamated by set union (possibly with subtraction of certain elements) to give the value of the relevant feature of the phrase. We therefore refer to this phrase-based amalgamation of set-valued features as *phrasal amalgamation*.

Specifically, the Nonlocal Feature Principle requires each of the NONLOCAL|INHERITED features QUE, REL and SLASH of a phrase to be the set union of the equivalent feature of all the daughters, minus any elements of the equivalent NONLOCAL|TO-BIND sets of the head daughter. The Quantifier Inheritance Principle requires the QSTORE feature of a phrase to be the set union of the QSTOREs of all the daughters, minus any quantifiers in the phrase's RETRIEVED list. The Contextual Consistency Principle simply specifies the CONTEXT|BACKGROUND feature of a phrase as the set union of the BACKGROUND sets of all the daughters.

### 2.2 Lexical amalgamation

Recent revisions of HPSG theory divide phrasal amalgamation into two distinct parts: the amalgamation part and the inheritance part. While the inheritance part is still specified by constraints on phrases, the set-valued feature amalgamation part is now specified by constraints on *words*.

Following the proposals of (Manning and Sag, 1995), a word's arguments are lexically specified in its ARGUMENT-STRUCTURE (ARG-ST) list. A

---

\*Visiting researcher of Information Systems Product Development Laboratories, Sharp Corporation.

word's set-valued features are defined in terms of the amalgamation of the equivalent set-valued features of its arguments. This form of amalgamation, specified by lexical constraints, is therefore referred to as *lexical amalgamation*.

The lexicalization of nonlocal features is described in (Sag, to appear). The lexical amalgamation of SLASH values is stated as lexical constraint (1), where  $\uplus$  designates disjoint set union.

- (1) Lexical Amalgamation of SLASH:

$$\left[ \begin{array}{l} \text{ARG-ST} \left\langle [\text{SLASH } \square], \dots, [\text{SLASH } \square] \right\rangle \\ \text{SLASH} \quad \square \uplus \dots \uplus \square \end{array} \right]$$

This allows a simplification of the mechanism for inheritance of SLASH values. A new SLASH Inheritance Principle (SLIP) is stated as phrasal constraint (2), where '/' indicates a default value.

- (2) SLASH Inheritance Principle (SLIP):

$$hd-nexus-ph \Rightarrow \left[ \begin{array}{l} \text{SLASH} \quad / \quad \square \\ \text{HD-DTR} \quad [\text{SLASH } \square] \end{array} \right]$$

The combination of (1) and (2) means that a phrase inherits the SLASH values of its daughters indirectly, via the head daughter. Similarly, lexical amalgamation of QUE and REL features is used to introduce a WH-Inheritance Principle (WHIP) in which QUE and REL are inherited via a phrase's head daughter. The combination of SLIP and WHIP in (Sag, to appear) replaces the Nonlocal Feature Principle of (Pollard and Sag, 1994) and avoids the need for separate NONLOCAL|INHERITED and NONLOCAL|TO-BIND features.

The lexicalization of quantifier scoping is very similar. Following the proposals of (Pollard and Yoo, 1995), QSTORE is a LOCAL feature which can be included in the features subcategorized for by a lexical head, and can therefore be lexically amalgamated in that head. These proposals, extended to include lexicalization of quantifier retrieval, are stated by (Manning et al., to appear) as the lexical constraint (3), where  $\square$  is the set of retrieved quantifiers and '-'! designates contained set difference.

- (3) Lexical Amalgamation of QSTORE:

$$\left[ \begin{array}{l} \text{ARG-ST} \left\langle [\text{QSTORE } \square], \dots, [\text{QSTORE } \square] \right\rangle \\ \text{QSTORE} \quad (\square \uplus \dots \uplus \square) \text{ -! } \square \\ \text{CONT} \quad [\text{QUANTS} \quad \text{order}(\square)] \end{array} \right]$$

Unscoped quantifiers are inherited not from all daughters but only from the semantic head daughter. This can be stated as a revised Quantifier Inheritance Principle (QUIP) in (4).

- (4) Quantifier Inheritance Principle (QUIP):

$$hd-nexus-ph \Rightarrow \left[ \begin{array}{l} \text{QSTORE} \quad / \quad \square \\ \text{HD-DTR} \quad [\text{QSTORE } \square] \end{array} \right]$$

$$hd-adjunct-ph \Rightarrow \left[ \begin{array}{l} \text{QSTORE} \quad / \quad \square \\ \text{ADJ-DTR} \quad [\text{QSTORE } \square] \end{array} \right]$$

### 2.3 Lexicalization of CONTEXT

We now propose the lexicalization of CONTEXT, following the same approach as the lexicalization of nonlocal features and the lexicalization of quantifier scoping. CONTEXT includes two features, BACKGROUND (BACKGR) and CONTEXTUAL-INDICES (C-INDS). As BACKGR is a set-valued feature, we introduce the lexical constraint (5), in which a word's BACKGR set is the disjoint set union of the BACKGR sets of its arguments.

- (5) Lexical Amalgamation of BACKGR:

$$\left[ \begin{array}{l} \text{ARG-ST} \left\langle [\text{BACKGR } \square], \dots, [\text{BACKGR } \square] \right\rangle \\ \text{BACKGR} \quad \square \uplus \dots \uplus \square \end{array} \right]$$

As discussed by Pollard and Sag (1994), a phrase's C-INDS can be taken for simplicity as the unification of the C-INDS of its daughters. This is not a linguistic principle but is typical of discourse situations. We can specify an equivalent lexical constraint (6), in which a word's C-INDS feature is the unification of the C-INDS features of its arguments.

- (6) Lexical Amalgamation of C-INDS:

$$\left[ \begin{array}{l} \text{ARG-ST} \left\langle [\text{C-INDS } \square], \dots, [\text{C-INDS } \square] \right\rangle \\ \text{C-INDS} \quad \square \end{array} \right]$$

We introduce a new Contextual Head Inheritance Principle (CHIP), in which the CONTEXT feature of a phrase is by default token-identical to the CONTEXT value of its *contextual head* daughter. We define contextual head in the same way as semantic head, i.e. in a head-adjunct-phrase the adjunct daughter is the contextual head, and in a head-nexus-phrase the syntactic head is the contextual head. The principle is stated in (7).

- (7) Contextual Head Inheritance Principle (CHIP):

$$hd-nexus-ph \Rightarrow \left[ \begin{array}{l} \text{CONTEXT} \quad / \quad \square \\ \text{HD-DTR} \quad [\text{CONTEXT } \square] \end{array} \right]$$

$$hd-adjunct-ph \Rightarrow \left[ \begin{array}{l} \text{CONTEXT} \quad / \quad \square \\ \text{ADJ-DTR} \quad [\text{CONTEXT } \square] \end{array} \right]$$

The combination of (5) and (7), replacing the Contextual Consistency Principle of (Pollard and

Sag, 1994), means that a phrase inherits the BACKGR values of its daughters indirectly, via the contextual head daughter.

### 3 Head-driven Generation

On a computational level, lexicalization of context combined with lexicalization of quantifier scoping is advantageous in using HPSG with head-driven generation algorithms.

#### 3.1 Semantic heads

Head-driven generation algorithms assume that most grammar rules have a semantic head daughter whose logical form is identical to the logical form of the mother. The basic head-driven bottom-up generation (BUG) algorithm (van Noord, 1990) requires that every rule has such a head, except rules for lexical entries. The semantic head-driven (SHD) algorithm (Shieber et al., 1990) relaxes this requirement, dividing rules into chain rules with such a head which are processed bottom-up, and non-chain rules which are processed top-down. Head-driven bottom-up generation is efficient because it is geared both to the input logical form (head-driven) and also to the information available in the lexicon (bottom-up).

HPSG is ideally suited to head-driven bottom-up generation. Because it is highly lexicalist, the rich information available in the HPSG lexicon supports efficient bottom-up generation. Furthermore, HPSG has a clear notion of semantic head: in head-adjunct phrases, the adjunct daughter is the semantic head; in other headed phrases, the syntactic head daughter is the semantic head. In both cases, the HPSG Content Principle basically requires the content of the semantic head to be identical to the content of the mother. If logical form is equated with semantic content, it follows that apart from coordinate structures, all HPSG grammar rules are chain rules for BUG/SHD generation.

However, there is a complication in (Pollard and Sag, 1994) caused by the use of Cooper storage to handle quantifier scope ambiguities. Although scoped quantifiers are included in the QUANTS list within CONTENT, unscoped quantifiers are stored in the QSTORE set outside CONTENT. This means that semantic content cannot be simply equated with CONTENT.<sup>1</sup>

#### 3.2 Logical form

Logical form is not recognised as a separate linguistic level in HPSG theory, but is more or less equated with semantic content. However, natural language generation requires other information besides semantic content. Presuppositions and other pragmatic and discourse factors are all required for

---

<sup>1</sup>It would be unsatisfactory to require all quantifiers to be scoped before generation, as this would exclude applications such as ambiguity-preserving translation.

communicatively adequate realisation. In HPSG, such factors are included in CONTEXT.

Tactical generation is viewed computationally as mapping from logical form to strings. The problem is how to include the required context factors. One approach is to restrict logical form to semantic factors, and provide separate mechanisms for handling context factors during the processing of the logical form. Linguistic context could be made available outside the logical form but inside the overall computational context.

However, the usual approach is to include some context factors inside the logical form to be used for generation. For example, (Shieber et al., 1990) included mood operators in their examples. This inclusive approach to logical form requires no additional sources of information for tactical generation, but means that logical form cannot be equated with semantic content.

In the framework of HPSG, Minimal Recursion Semantics (Copestake et al., 1995) could be such an inclusive logical form, representing unscoped quantifiers and context factors as well as semantic content. However, the relationships between MRS structures and existing HPSG structures are not yet clear.

For the purposes of head-driven generation we can be more conservative. A “logical form” consisting of the existing structures CONTENT, QSTORE and CONTEXT is sufficient.<sup>2</sup>

#### 3.3 Contextual-semantic heads

The efficiency of head-driven generation derives from identities of logical form between mother and semantic head daughter. Since an “inclusive” logical form cannot be equated with semantic content, the HPSG Content Principle alone cannot ensure this identity. We therefore need the lexicalization of context, together with the lexicalization of quantifier scoping, in order to achieve the identity.

The Content Principle (which could perhaps be renamed Semantic Head Inheritance Principle - SHIP) ensures the identity of CONTENT between mother and semantic head. The Quantifier Inheritance Principle (QUIP) (4) ensures the identity of QSTORE between mother and semantic head. The Contextual Head Inheritance Principle (CHIP) (7) ensures the identity of CONTEXT between mother and contextual head. The contextual head and the semantic head are the same daughter. If logical form consists of CONTENT, QSTORE and CONTEXT, the combination of SHIP, QUIP and CHIP ensures that the logical forms of the mother and the contextual-semantic head are identical, as required for efficient head-driven generation.

---

<sup>2</sup>This could be specified as a sign-level attribute LF, with attributes CT, QS, CX token-identical to CONTENT, QSTORE and CONTEXT respectively.

### 3.4 Nondeterminism

Amalgamation of sets, although convenient for parsing, is intrinsically difficult for generation. Given the set values from the daughters during bottom-up parsing, the set values of a phrase are directly specified by set union (minus certain elements) according to the relevant principles. For parsing, phrasal amalgamation can be implemented by procedural attachments, as set union is deterministic and efficient. However, given the set values of a phrase during generation, the set values of individual daughters are not specified by principles based on phrasal amalgamation, which only impose constraints across all the daughters. In this case, procedural attachments are highly nondeterministic.

The change to lexical amalgamation helps to reduce this nondeterminism. During head-driven generation, the relevant set values from the input logical form are passed directly downwards from a phrase to its contextual-semantic head, quickly reaching lexical entries. The lexical argument-structure and valency information may then constrain the distribution of set elements among the siblings (arguments) of the contextual-semantic head, reducing the degree of nondeterminism.

## 4 Register Variation

We now briefly sketch a simple approach to register variation, to introduce the lexicalization of context in linguistic analysis, here combined with the lexicalization of nonlocal features.

To keep a representation for register variation as simple as possible, we assume that CONTEXT has an additional attribute REGISTER (REGSTR), with value of sort *register*, which has only two subsorts, *formal* and *informal*. To maintain the lexicalization of context, we need lexical amalgamation of REGSTR as stated in (8), which has a defeasible status similar to (6).

(8) Lexical Amalgamation of REGSTR:

$$\left[ \begin{array}{l} \text{ARG-ST} \left\langle \left[ \text{REGSTR } \square \right], \dots, \left[ \text{REGSTR } \square \right] \right\rangle \\ \text{REGSTR } \square \end{array} \right]$$

### 4.1 Relative pronouns

A revised analysis of English relative clauses without empty relativizers is presented by Sag (to appear), who argues for treating relative *that* as a pronominal, rather than a complementizer, as the only real obstacle is that it disallows pied piping (9a), and this property is shared with relative *who* (9b) in many varieties of English.

- (9) a. \*The person [with that we were talking] ...  
 b. \*The person [with who we were talking] ...  
 c. The person [with whom we were talking] ...

Noting that in such varieties the only pied-piped relative pronouns are *whose*, *which* and *whom* (9c), Sag comments that “the constraints on this variation have to do with case assignment, register restrictions, or both”.

While *whom* is always accusative and formal (10), we note systematic covariation<sup>3</sup> of case assignment and register restrictions in *who* and *that*. Relative *who* is either nominative and unrestricted (REGSTR value *register*) (11) or accusative and informal (12). The same covariation is seen in relative *that* (and also in interrogative *who*).

(10) whom (accusative):  $\left[ \begin{array}{ll} \text{PHON} & \langle \textit{whom} \rangle \\ \text{CAT} & \text{NP}[\textit{acc}] \\ \text{REGSTR} & \textit{formal} \end{array} \right]$

(11) who (nominative):  $\left[ \begin{array}{ll} \text{PHON} & \langle \textit{who} \rangle \\ \text{CAT} & \text{NP}[\textit{nom}] \\ \text{REGSTR} & \textit{register} \end{array} \right]$

(12) who (accusative):  $\left[ \begin{array}{ll} \text{PHON} & \langle \textit{who} \rangle \\ \text{CAT} & \text{NP}[\textit{acc}] \\ \text{REGSTR} & \textit{informal} \end{array} \right]$

### 4.2 Construction types

We can put register restrictions on construction types. Sag (to appear) treats nonsubject relative clauses like *whose bagels I like* and *from whom I bought these bagels* in terms of a single type *fin-wh-fill-rel-cl*. He hints that the constraints on this type have more work to do than the simplified formulation shown in (13), which states only that the filler daughter must be an NP or a PP.

(13) *fin-wh-fill-rel-cl*  $\Rightarrow$   
 $\left[ \text{FILLER-DTR} \left[ \text{HEAD } \textit{noun} \vee \textit{prep} \right] \right]$

Since the construction with a PP filler is restricted to formal register, we could split this type into two distinct subtypes *fin-wh-np-fill-rel-cl* (14) and *fin-wh-pp-fill-rel-cl* (15). The formal register restriction in (15) would block examples (9a) and (9b) but allow (9c) as required.

(14) *fin-wh-np-fill-rel-cl*  $\Rightarrow$   
 $\left[ \text{FILLER-DTR} \left[ \text{HEAD } \textit{noun} \right] \right]$

(15) *fin-wh-pp-fill-rel-cl*  $\Rightarrow$   
 $\left[ \text{FILLER-DTR} \left[ \begin{array}{l} \text{HEAD } \textit{prep} \\ \text{REGSTR } \textit{formal} \end{array} \right] \right]$

<sup>3</sup>Computational treatments of covariation in lexical entries are discussed by (Sanfilippo et al., 1994) and (Meurers and Minnen, 1995).

### 4.3 Lexical types

We would probably prefer to block the PP *with who* in (9b) without building the rest of the clause, as a *wh*-PP is formal, but accusative *who* is informal. This could be done by putting register restrictions on PP construction types instead of on the relative clause type. However, we explore the lexicalization of context to show that it can also be done by constraints on lexical types as stated in (16) – (18).

$$(16) \quad \textit{rel-pp-prep} \Rightarrow \left[ \begin{array}{ll} \text{HEAD} & \textit{prep} \\ \text{QUE} & \{\} \\ \text{REL} & \{\square\} \\ \text{SLASH} & \{\} \\ \text{REGSTR} & \textit{formal} \end{array} \right]$$

$$(17) \quad \textit{que-pp-prep} \Rightarrow \left[ \begin{array}{ll} \text{HEAD} & \textit{prep} \\ \text{QUE} & \{\square\} \\ \text{REL} & \{\} \\ \text{SLASH} & \{\} \\ \text{REGSTR} & \textit{formal} \end{array} \right]$$

$$(18) \quad \textit{slash-pp-prep} \Rightarrow \left[ \begin{array}{ll} \text{HEAD} & \textit{prep} \\ \text{QUE} & \{\} \\ \text{REL} & \{\} \\ \text{SLASH} & \{\square\} \\ \text{REGSTR} & \textit{informal} \end{array} \right]$$

Lexical type constraint (16) requires a *rel-pp-prep* preposition to take as argument a relative pronoun which is, or can be, formal. The non-empty REL requires the argument to be a relative pronoun, as its REL value is acquired by lexical amalgamation of nonlocal features. The *formal* register requires the argument also to be formal, as its REGSTR value is acquired by lexical amalgamation of context.

This combination of lexical types, lexicalization of nonlocal features and lexicalization of context provides an alternative way to block the PP *with who*, and to block examples (9a) and (9b) but allow (9c), as required.

## 5 Conclusion

The lexicalization of context naturally follows other recent revisions in HPSG theory concerned with set-valued features. It seems useful on a computational level, when HPSG is combined with a head-driven generation algorithm. Its value in linguistic analysis is not yet established, but at least it appears to be worth further exploration.

## References

- Ann Copestake, Dan Flickinger, Rob Malouf, Susanne Riehemann, and Ivan Sag. 1995. Translation using Minimal Recursion Semantics. In *Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation*. Association for Computational Linguistics.
- Christopher Manning and Ivan Sag. 1995. Dissociations between argument structure and grammatical relations. Paper presented at the Tübingen HPSG workshop.
- Christopher Manning, Ivan Sag, and Masayo Iida. to appear. The lexical integrity of Japanese causatives. In G. Green and R. Levine, editors, *Readings in HPSG*. Cambridge University Press.
- W. Detmar Meurers and Guido Minnen. 1995. A computational treatment of HPSG lexical rules as covariation in lexical entries. In *Proceedings of the Fifth International Workshop on Natural Language Understanding and Logic Programming*, Lisbon, Portugal.
- Carl Pollard and Ivan Sag. 1994. *Head-driven Phrase Structure Grammar*. University Press, Chicago.
- Carl Pollard and Eun Jung Yoo. 1995. Quantifiers, *wh*-phrases and a theory of argument selection. Paper presented at the Tübingen HPSG workshop.
- Ivan Sag. to appear. English relative clause constructions. *Journal of Linguistics*.
- Antonio Sanfilippo, Kerima Benkerimi, and Dagmar Dwehus. 1994. Virtual polysemy. In *15th International Conference on Computational Linguistics*, pages 696–700. Association for Computational Linguistics.
- Stuart M. Shieber, Gertjan van Noord, Fernando C.N. Pereira, and Robert C. Moore. 1990. Semantic head-driven generation. *Computational Linguistics*, 16(1):30–42.
- Gertjan van Noord. 1990. An overview of head-driven bottom-up generation. In Robert Dale, Chris Mellish, and Michael Zock, editors, *Current Research in Natural Language Generation*, pages 141–165. Academic Press.