# COORDINATION IN UNIFICATION-BASED GRAMMARS

Richard P. Cooper
Department of Psychology
University College London
London WC1E 6BT, U.K.
JANET: ucjtrrc@ucl.ac.uk

## ABSTRACT

Within unification-based grammar formalisms, providing a treatment of cross-categorial coordination is problematic, and most current solutions either over-generate or under-generate. In this paper we consider an approach to coordination involving "composite" feature structures, which describe coordinate phrases, and present the augmentation to the logic of feature structures required to admit such feature structures. This augmentation involves the addition of two connectives, composite conjunction and composite disjunction, which interact to allow cross-categorial coordination data to be captured exactly. The connectives are initially considered to function only in the domain of atomic values, before their domain of application is extended to cover complex feature structures. Satisfiability conditions for the connectives in terms of deterministic finite state automata are given, both for the atomic case and for the more complex case. Finally, the Prolog implementation of the connectives is discussed, and it is illustrated how, in the atomic case, and with the use of the `freeze/2` predicate of second generation Prologs, the connectives may be implemented.

## The Problem

Given a modern unification-based grammar, such as HPSG, or PATR/FUG-style grammars, where feature structure descriptions are associated with the constituents of the grammar, and unification is used to build the descriptions of constituents from those of their subconstituents, providing a treatment of coordination, especially cross-categorial coordination, is problematic. It is well known that coordination is not restricted to like categories (see (1)), so it is too restrictive to require that the syntactic category of a coordinate phrase be just the unification of the syntactic categories of the conjuncts. Indeed, the data suggest that the syntactic categories of the conjuncts need not unify.

(1) a. Tigger became famous and a complete snob.
b. Tigger is a large bouncy kitten and proud of it.

Furthermore, it is only possible to coordinate certain phrases within certain syntactic contexts. Whilst the examples in (1) are grammatical, those in (2) are not, although the same constituents are coordinated in each case.

(2) a. *Famous and a complete snob chased Fido.
b. *A large bouncy kitten and proud of it likes Tom.

The difference between the examples in (1) and (2) is the syntactic context in which the coordinated phrase appears. The relevant generalisation, made by Sag *et al.* (1985) with respect to GPSG, is that constituents may coordinate if and only if the description of each constituent unifies with the relevant description in the grammar rule which licenses the phrase containing the coordinate structure. Example (1a) is grammatical because the phrase structure rule which licenses the constituent *became famous and a complete snob* requires that *famous and a complete snob* unify with the partial description of the object subcategorised for by *became*, and the descriptions of each of the conjuncts, *famous* and *a complete snob*, actually do unify with that partial description: *became* requires that its object be "either an NP or an AP", and each of *famous* and *a complete snob* is "either an NP or an AP". (1b) is grammatical for analogous reasons, though *is* is less fussy about its object, also allowing PPs and predicative VPs to fill the position. (2a) is ungrammatical as *chased* requires that its subject be a noun phrase. Whilst this is true of *a com-*

*plete snob*, it is not true of *famous*, so the description of *famous* does not unify with the description which *chase* requires of its subject. (2b) is ungrammatical for similar reasons.

## Two Approaches to a Solution

Two approaches to this problem are immediate. Firstly, we may try to capture the intuition that each conjunct must unify with the requirements of the appropriate grammar rule by generalising all grammar rules to allow for coordinated phrases in all positions. This general approach follows that of Shieber (1989), and involves the use of semi-unification. Note that this does not involve a grammar rule licensing coordinate constituents such as $\alpha$ *and* $\beta$: following this approach $\alpha$ *and* $\beta$ can never be a constituent in its own right.

An alternate approach is to preserve the original grammar rules, but generalise the notion of syntactic category to license composite categories — categories built from other categories — and introduce a rule licensing coordinate structures which have such composite syntactic categories. That is, we introduce a grammar rule such that if $\alpha$ and $\beta$ are constituents, then $\alpha$ *and* $\beta$ is also a constituent, and the syntactic category of this constituent is a composite of the syntactic categories of $\alpha$ and $\beta$.

Within a unification-based approach, this generalisation of syntactic category requires a generalisation of the logic of feature structures, with an associated generalisation of unification. This is the approach which we adopt in this paper. One of the consequences of this approach is that for (almost) any constituents $\alpha$ and $\beta$, the grammar should also license the string $\alpha$ *and* $\beta$ as a constituent, irrespective of whether there are any contexts in which this constituent may occur. Thus our grammar might admit *in the garden and chases Fido* as a constituent, though there may be no contexts which license such a constituent.

Our approach differs from other approaches to cross-categorial coordination (such as those employing generalisation, or that of Proudian & Goddeau (1987)) which have been suggested in the unification grammar literature in that it involves a real augmentation of the logic of feature structures. Other approaches which do not involve this augmentation tend to overgenerate (the approaches employing generalisation) or under-generate (the approach of Proudian & Goddeau).

Generalisation over-generates because in generalisation conflicting values are ignored. In the case of *became*, assuming that we analyse *became* as requiring an object whose description unifies with [CATEGORY NP ∨ AP], generalisation would license (1a), as well as both of the examples in (3).

(3) a. *Tigger became famous and in the garden.
    b. *Tigger became a complete snob and in the garden.

This is because the generalisation of the descriptions of the two conjuncts ([CATEGORY AP] and [CATEGORY PP] in the case of (3a) and [CATEGORY NP] and [CATEGORY PP] in the case of (3b)) is in each case [CATEGORY ⊥], which unifies with the [CATEGORY NP ∨ AP] requirement of *became*.

It is not clear how the approach of Proudian & Goddeau could be applied to the *became* example: the disjunctive subcategorisation requirements of *became* cannot be treated within their approach. For further details see Cooper (1990).

## Composite Atomic Values

Following Kasper & Rounds (1990), and earlier work by the same authors (Rounds & Kasper (1986) and Kasper & Rounds (1986)), we adopt a logical approach to feature structures via an equational logic. The domain of well-formed formulae is defined inductively in terms of a set $A$ of *atomic values* and a set $L$ of *labels* or *attributes*. These formulae are interpreted as descriptions of deterministic finite state automata.

In the formulation of Kasper & Rounds, these automata have atomic values assigned to (some of) their terminal states. A simplified reading of the coordination data suggests that these values need not be atomic, and that there is structure on this domain of "atomic" values. To model this structure we introduce an operator "∧", which we term composite conjunction, such that if $\alpha$ and $\beta$ are atomic values, then $\alpha \wedge \beta$ is also an atomic value. Informally, if *a large bouncy kitten* is described by the pair [CATEGORY NP] and *proud of it* is described by the pair [CATEGORY AP], then any coordination of those constituents, such as *neither a large bouncy kitten nor proud of it* will be described by the pair [CATEGORY NP ∧ AP].

Before discussing satisfiability, we consider some of the properties of ∧:

- $\curlywedge$ is symmetric: a noun phrase coordinated with an adjectival phrase is of the same category as an adjectival phrase coordinated with a noun phrase. Thus for all atomic values $\alpha$ and $\beta$, we require

$$\alpha \curlywedge \beta = \beta \curlywedge \alpha$$

- $\curlywedge$ is associative: in constructions involving more than two conjuncts the category of the coordinate phrase is independent of the bracketing. Hence for all atomic values $\alpha$, $\beta$ and $\gamma$, we require

$$(\alpha \curlywedge \beta) \curlywedge \gamma = \alpha \curlywedge (\beta \curlywedge \gamma)$$

- $\curlywedge$ is idempotent: the conjunction of two (or more) constituents of category $x$ is still of category $x$: Hence for all atomic values $\alpha$, we require

$$\alpha \curlywedge \alpha = \alpha$$

These properties exactly correspond to the properties required of an operator on finite sets. For full generality we thus take $\curlywedge$ to be an operator on finite subsets of atomic values rather than a binary operator satisfying the above conditions, but for simplicity use the usual infix notation for the binary case.

Given one further requirement, that for any $\alpha$

$$\curlywedge \{\alpha\} = \alpha$$

(and hence that $\alpha \curlywedge \alpha = \curlywedge \{\alpha\}$) the use of an operator on sets directly reflects all of the above properties:

$$\alpha \curlywedge \beta = \curlywedge \{\alpha, \beta\} = \beta \curlywedge \alpha$$

$$(\alpha \curlywedge \beta) \curlywedge \gamma = \curlywedge \{\alpha, \beta, \gamma\} = \alpha \curlywedge (\beta \curlywedge \gamma)$$

$$\alpha \curlywedge \alpha = \curlywedge \{\alpha\}$$

Given this structure on the domain of atomic values, we restate the satisfiability requirements. We deal in terms of deterministic finite state automata (DFSAs) specified as six-tuples, $\langle Q, q_0, L, \delta, A, \pi \rangle$, where

- $Q$ is a set of atoms known as states,

- $q_0$ is a particular element of $Q$ known as the start state,

- $L$ is a set of atoms known as labels,

- $\delta$ is a partial function from $[Q \times L]$ to $Q$ known as the transition function,

- $A$ is a set of atoms, and

- $\pi$ is a function from final states (those states from which according to $\delta$ there are no transitions) to $A$.

To incorporate conjunctive composite values we introduce structure on $A$, requiring that for all finite subsets $X$ of $A$, $\curlywedge X$ is in $A$. Satisfiability of formulae involving composite conjunction is defined as follows:

- $\mathcal{A} \models \curlywedge \{\alpha_1, \ldots, \alpha_n\}$ iff $\mathcal{A} = \langle Q, q_0, L, \delta, A, \pi \rangle$ where $\delta(q_0, l)$ is undefined for each $l$ in $L$ and $\pi(q_0) = \curlywedge \{\alpha_1, \ldots, \alpha_n\}$.[1]

This is really just the same clause as for all atomic values:

- $\mathcal{A} \models \alpha$ iff $\mathcal{A} = \langle Q, q_0, L, \delta, A, \pi \rangle$ where $\delta(q_0, l)$ is undefined for each $l$ in $L$ and $\pi(q_0) = \alpha$.

As such nothing has really changed yet, though note that by an "atomic value" now we mean an element of the domain $A$. The structure which we have introduced on $A$ means that strictly speaking these values are not atomic. They are, however, "atomic" in the feature structure sense: they have no attributes.

The real trick in handling composite conjunctive formulae correctly, however, comes in the treatment of disjunction. We introduce to the syntax a further connective $\curlyvee$, composite disjunction. As the name suggests, this is the analogue of disjunction in the domain of composite values. Like standard disjunction $\curlyvee$ exists only in the syntax, and not in the semantics. For satisfiability we have:

- $\mathcal{A} \models (\alpha \curlyvee \beta)$ where $\alpha, \beta \in A$ iff $\mathcal{A} \models \alpha$ or $\mathcal{A} \models \beta$ or $\mathcal{A} \models (\alpha \curlywedge \beta)$.

More generally:

- $\mathcal{A} \models \curlyvee \Phi$ where $\Phi \subset A$ and $\Phi$ is finite iff $\mathcal{A} \models \curlywedge \Phi'$ for some subset $\Phi'$ of $\Phi$.

With this connective, disjunctive subcategorisation requirements may be replaced with composite disjunctive requirements. The intuition behind this modification stems from the fact that if a constituent has a disjunctive subcategorisation requirement, then that requirement can be met by any of the disjuncts, *or* the composite of those disjuncts. To illustrate this reconsider

[1] For simplicity we ignore connectivity of DFSAs. If connectivity is to be included in the definitions, then in this case $Q$ must be the singleton $\{q_0\}$.

the example in (1a). Originally the subcategorisation requirements of *became* might have been stated with the disjunctive specification [CATEGORY NP $\vee$ AP]. This could be satisfied by either an NP or an AP, but not by a conjunctive composite composed of an NP and an AP, i.e., not by the result of conjoining an NP and an AP. To allow this we respecify the requirements on the subcategorised for object as [CATEGORY NP$\veebar$AP]. This requirement may be legitimately met by either an NP or an AP *or* a conjunctive composite NP$\barwedge$AP.

## Composite Feature Structures

This use of an algebra of atomic values allows composites only to be formed at the atomic level. That is, whilst we may form $\alpha \barwedge \beta$ for $\alpha, \beta$ atomic, we may not form $\alpha \barwedge \beta$ where $\alpha, \beta$ are non-atomic feature structures. However, such composites do appear to be useful, if not necessary. In particular, in an HPSG-like theory, the appropriate thing to do in the case of coordinate structures seems to be to form the composite of the HEAD features of all conjuncts. The above approach to composite atoms does not immediately generalise to allow composite feature structures. In particular, whilst the intuitive behaviour of the connectives should remain as above, the semantic domain must be revised to allow a satisfactory rendering of satisfiability.

With regard to syntax we revert back to an unstructured domain $A$ of atoms but augment the system of Kasper & Rounds (1990) with two clauses licensing composite formulae:

- $\barwedge \Phi$ is a valid formula if $\Phi$ is a finite set, each element of which is a valid formula;
- $\veebar \Phi$ is a valid formula if $\Phi$ is a finite set, each element of which is a valid formula.

The generalisation of satisfiability holds for composite disjunction:

- $\mathcal{A} \models \veebar \Phi$ iff $\mathcal{A} \models \barwedge \Phi'$ for some subset $\Phi'$ of $\Phi$.

We must alter the semantic domain, the domain of deterministic finite state automata, however, to allow a sensible rendering of satisfaction of composite conjunctive formulae — we need something like composite states to replace the composite atomic values of the preceding section.

In giving a semantics for $\barwedge$ we take advantage of the equivalence of $\barwedge\{\alpha\}$ and $\alpha$. We begin by generalising the notion of a deterministic finite state automaton such that the transition function maps states to sets of states:

A *generalised deterministic finite state automaton* (GDFSA) is a tuple $\langle Q, q_0, L, \delta, A, \pi \rangle$, where

- $Q$ is a set of atoms known as states,
- $q_0 \in Pow(Q)$ is a distinguished set of states known as the start state set,
- $L$ is a set of atoms known as labels,
- $\delta$ is a partial function from $[Q \times L]$ to $Pow(Q)$,
- $A$ is a set of atoms, and
- $\pi$ is a partial assignment of atoms to final states.

Any DFSA $\mathcal{A} = \langle Q, q_0, L, \delta, A, \pi \rangle$ has a corresponding GDFSA $\mathcal{A}'$ given by $\langle Q, \{q_0\}, L, \delta', A, \pi \rangle$ where $\delta'(q, l) = \{\delta(q, l)\}$.

Given a GDFSA $\mathcal{A}$ we define satisfiability of conjunctive, disjunctive and atomic formulae as usual. There is a slight differences in satisfiability of path equations:

- $\mathcal{A} \models l : \phi$ iff $\mathcal{A}/l$ is defined and $\mathcal{A}/l \models \phi$, where if $\mathcal{A} = \langle Q, \{q\}, L, \delta, A, \pi \rangle$, then $\mathcal{A}/l = \langle Q, \delta(q, l), L, \delta, A, \pi \rangle$.

This clause has been altered to enforce the requirement that $q_0$ be a singleton, and that $\delta$ maps this single element to a set.[2]

The extensions for $\veebar$ and $\barwedge$ are:

- $\mathcal{A} \models \veebar \Phi$ iff $\mathcal{A} \models \barwedge \Phi'$ for some subset $\Phi'$ of $\Phi$ (as above).
- $\mathcal{A} \models \barwedge \Phi$ iff for each $\phi \in \Phi$, there exists a $q' \in q_0$ such that $\langle Q, \{q'\}, L, \delta, A, \pi \rangle \models \phi$.

Note that in the case of $\Phi$ a singleton, this last clause reduces to $\mathcal{A} \models \barwedge\{\phi\}$ iff $\mathcal{A} \models \phi$.

The reason why the satisfiability clauses for these connectives are so simple resides principally in the equivalence of $\barwedge\{\alpha\}$ and $\alpha$. We cannot follow this approach in giving a semantics for standard set valued attributes because in the case of sets we want $\{\alpha\}$ and $\alpha$ to be distinct.

[2] Again we are ignoring connectivity.

## Properties of Composites

The properties of composite feature structures and the interaction of $\wedge$ and $\vee$ may be briefly summarised as follows:

- Disjunctive composite feature structures are a syntactic construction. Like disjunctive feature structures they exist in the language but have no direct correlation with objects in the world being modelled.
- Conjunctive composite feature structures describe composite objects which do exist in the world being modelled.
- A disjunctive composite feature structure describes an object just in case one of the disjuncts describes the object, or it describes a composite object.
- A disjunctive composite feature structure describes a composite object just in case each object in the composite is described by one of the disjuncts.
- A conjunctive composite feature structure describes an object just in case that object is a composite object consisting of objects which are described by each of the descriptions making up the conjunctive composite feature structure.

The crucial point here is that conjunctive composite objects exist in the described world whereas disjunctive composite objects do not.

## An Example

To illustrate in detail the operation of composites we return to the example of (1a). In an HPSG-like formalism (see Pollard & Sag (1987)) employing composites, the object subcategorised for by *became* would be required to satisfy:

$$\begin{bmatrix} \textsc{syn}|\textsc{loc} & \begin{bmatrix} \textsc{head} & \begin{bmatrix}\textsc{maj} & \mathit{Adj}\end{bmatrix} \vee \begin{bmatrix}\textsc{maj} & \mathit{Noun}\end{bmatrix} \\ \textsc{subcat} & \langle\,\rangle \end{bmatrix} \end{bmatrix}$$

According to our satisfiability clauses above, this may be satisfied by:

- an AP such as *famous*, having description

$$\begin{bmatrix} \textsc{phon} & \mathit{famous} \\ \textsc{syn}|\textsc{loc} & \begin{bmatrix} \textsc{head}|\textsc{maj} & \mathit{Adj} \\ \textsc{subcat} & \langle\,\rangle \end{bmatrix} \end{bmatrix}$$

- an NP such as *a complete snob*, having description

$$\begin{bmatrix} \textsc{phon} & \mathit{a\ complete\ snob} \\ \textsc{syn}|\textsc{loc} & \begin{bmatrix} \textsc{head}|\textsc{maj} & \mathit{Noun} \\ \textsc{subcat} & \langle\,\rangle \end{bmatrix} \end{bmatrix}$$

- or an AP $\wedge$ NP such as *famous and a complete snob*, having description[3]

$$\begin{bmatrix} \textsc{phon} & \mathit{famous\ and\ a\ complete\ snob} \\ \textsc{syn}|\textsc{loc} & \begin{bmatrix} \textsc{head} & \begin{bmatrix}\textsc{maj} & \mathit{Adj}\end{bmatrix} \wedge \begin{bmatrix}\textsc{maj} & \mathit{Noun}\end{bmatrix} \\ \textsc{subcat} & \langle\,\rangle \end{bmatrix} \end{bmatrix}$$

The subcategorisation requirements may not, however, by satisfied by, for example, a PP, or any conjunctive composite containing a PP. Hence the examples in (3) are not admitted.

## Implementation Issues

The problems of implementing a system involving composites really stem from their requirement for a proper implementation of disjunction. Implementation may be approached by adopting a strict division between the objects of the language and the objects of the described world. According to this approach, and in Prolog, Prolog terms are taken to correspond to the objects in the semantic domain, with Prolog clauses being interpreted much as in the syntax of an equational logic, as constraints on those terms. Conjunctive constraints correspond to unification. The formation of conjunctive composites is also no problem: such objects exist in the semantic domain, so structured terms may be constructed whose subterms are the elements of the composite. Thus if we implement the composite connectives as binary operators, `*` for $\wedge$ and `+` for $\vee$, we may form Prolog terms `(A * B)` corresponding to conjunctive composites. Disjunction, and the use of disjunctive composites, cannot, however, be implemented in the same way. The problem with disjunction is that we cannot normally be sure which disjunct is appropriate, and a term of the form `(A + B)` will not unify with the term `A`, as is required by either form of disjunction. The `freeze/2` predicate of many second generation Prologs provides some help here. For standard

[3] We assume that the rule licensing coordinate structures unifies all corresponding values (such as the values for each SUBCAT attribute) except for the values of the HEAD attributes. The value of the HEAD attribute of the coordinate structure is the composite of the values of the HEAD attribute of each conjunct.

disjunction, we might augment feature structure unification clauses (using <=> to represent the unification operator and \/ to represent standard disjunction) with special clauses such as:

```
    A <=> (A1 \/ A2) :-
            freeze(A, ((A <=> A1);
                       (A <=> A2)))
```

Similarly for composite disjunction, we might augment the unification clauses with:

```
 A <=> (A1 + A2) :-
         freeze(A, ((A <=> A1);
                    (A <=> A2);
                    (A <=> (A1 * A2))))
```

The idea is that the **freeze/2** predicate delays the evaluation of disjunctive constraints until the relevant structure is sufficiently instantiated. Unfortunately, "sufficiently instantiated" here means that it is **nonvar**. Only in the case of atoms is this normally sufficient. Thus the above approach is suitable for the implementation of composites at the level of atoms, but not suitable in the wider domain of composite feature structures.

## Concluding Remarks

In giving a treatment of coordination, and in particular cross-categorial coordination, within a unification-based grammar formalism we have introduced composite feature structures which describe composite objects. A sharp distinction is drawn between syntax and semantics: in the semantic domain there is only one variety of composite object, but in the syntactic domain there are two forms of composite description, a conjunctive composite description and a disjunctive composite description. Satisfiability conditions are given for the connectives in terms of a generalised notion of deterministic finite state automaton. Some issues which arise in the Prolog implementation of the connectives are also discussed.

## ACKNOWLEDGEMENTS

This research was carried out at the Centre for Cognitive Science, Edinburgh, under Commonwealth Scholarship and Fellowship Plan AU0027. I am grateful to Robin Cooper, William Rounds and Jerry Seligman for discussions concerning this work, as well as to two anonymous referees for their comments on an earlier version of this paper. All errors remain, of course, my own.

## REFERENCES

Cooper, Richard. *Classification-Based Phrase Structure Grammar: an Extended Revised Version of HPSG.* Ph.D. Thesis, University of Edinburgh. 1990.

Kasper, Robert & William Rounds. A Logical Semantics for Feature Structures. In *Proceedings of the 24th ACL*, 1986, 257-265.

Kasper, Robert & William Rounds. The Logic of Unification in Grammar. *Linguistics and Philosophy*, **13**, 1990, 35-58.

Pollard, Carl & Ivan Sag. *Information-Based Syntax and Semantics, Volume 1: Fundamentals.* 1987, CSLI, Stanford.

Proudian, Derek & David Goddeau. *Constituent Coordination in HPSG.* CSLI Report #CSLI-87-97, 1987.

Rounds, William & Robert Kasper. A Complete Logical Calculus for Record Structures Representing Linguistic Information. In *Proceedings of the 1st IEEE Symposium on Logic in Computer Science*, 1986, 38-43.

Sag, Ivan, Gerald Gazdar, Thomas Wasow and Steven Weisler. *Coordination and How to Distinguish Categories.* Natural Language and Linguistic Theory, **3**, 1985, 117-171.

Shieber, Stuart. *Parsing and Type Inference for Natural and Computer Languages.* Ph.D. Thesis, Stanford University, 1989.