A logical treatment of semi-free word order and bounded discontinuous constituency

Mike Reape
Centre for Cognitive Science, University of Edinburgh
2 Buccleuch Place, Edinburgh EH8 9LW
Scotland, UK

Abstract

In this paper we present a logical treatment of semi-free word order and *bounded* discontinuous constituency. We extend standard feature value logics to treat word order in a single formalism with a rigorous semantics without phrase structure rules. The elimination of phrase structure rules allows a natural generalisation of the approach to nonconfigurational word order and bounded discontinuous continuency via *sequence union*. Sequence union formalises the notions of *clause union* and *scrambling* by providing a mechanism for describing word order domains larger than the local tree. The formalism incorporates the distinction between *bounded* and *unbounded* forms of discontinuous constituency. Grammars are organised as algebraic theories. This means that linguistic generalisations are stated as axioms about the structure of *signs*. This permits a natural interpretation of implicational universals in terms of theories, subtheories and implicational axioms. The accompanying linguistic analysis is eclectic, borrowing insights from many current linguistic theories.

1. Introduction

In this paper we present a logical treatment of semi-free word order and *bounded* discontinuous constituency. By a logical treatment, we mean that the grammar is an axiomatic algebraic theory, i.e., a set of axioms formalised in a logic. By bounded discontinuous constituency, we refer to phenomena such as Dutch cross-serial dependencies, German Mittelfeld word order and clause-bounded extraposition in contrast to unbounded forms of discontinuous constituency such as cross-serial multiple extractions in Swedish relative clauses.

There is no scope within this paper to provide the linguistic argumentation sufficient to justify the approach described below. We shall have to limit ourselves to describing the key linguistic insight that we wish to formalise. That is that semi-free word order and nonconfigurationality are local phenomenon (i.e., bounded) and that word order domains are larger than the local trees of context-free based accounts of syntax. (This includes nearly all well-known unification-based grammar formalisms such as GPSG, LFG, HPSG and CUG.) This is simply a restatement of the notion of *clause union* or *scrambling* familiar from transformational analyses.

Our proposal is to provide a feature-value logic with a rigorous semantics with sufficient expressive power to allow the encoding of even syntactic structure within the single formalism. This means that the work of encoding syntactic structure is carried by the feature-value logic and not by formal language theoretic devices (i.e., phrase structure rules). Sequences of linguistic categories, or *signs* (following Saussure, HPSG and UCG), do the work of PSRs in our logic. The **phon** attribute of signs is functionally dependent on the **phon** attributes of the signs in sequences encoding local order domains. This allows us to trivially introduce word order domains larger than the local tree by introducing a *sequence union* operation. GPSG-style linear precedence (LP) statements express partial ordering constraints on elements of sequences.

The grammars we use consist of three types of elements: (1) descriptions of lexical signs, (2) descriptions of nonlexical signs and (3) axioms which specify the redundant structure of signs. This organisation is similar to that of HPSG (Pollard and Sag, 1987) from which we borrow many ideas. Subcategorisation is expressed in terms of sets of arguments. This borrows ideas from all of HPSG, LFG (Bresnan, 1982) and categorial grammar (CG). However, like HPSG and unlike LFG, our set descriptions are collapsible. We also share with CG the notions that linguistic structure is based on functor-argument structure and that lexical functors partially order their arguments.

All word order facts are captured in the way that lexical functors combine the ordering domains (**dtrs** sequences) of their arguments. Functors can combine order domains in one of two ways. They can take the *sequence union* of two sequences or *concatenate* one with the other. Discontinuity is achieved via sequence union. Continuity is achieved via concatenation. Since functors partially order sequences by LP statements, order amongst both continuous and discontinuous constituents is treated in the same way. This solves the problem often noted in the past of specifying the appropriate

constituents as sisters so that LP statements can apply correctly while satisfying the subcategorisation requirements of lexical heads and coindexing constituents correctly with subcategorised arguments. Furthermore, order is "inherited" from the "bottom" since sequence union preserves the relative order of the elements of its operands. The empirically falsifiable linguistic hypothesis made is that the whole range of *local* word order phenomena is treatable in this way.

In §2 we present the syntax and semantics of the feature-value logic In §3 we develop a methodology for organising grammars as algebraic theories. In §4 we present a toy analysis of Dutch subordinate clauses which illustrates the basic ideas underlying this paper. We very briefly discuss an interpretation of parametric variation in terms of theories and subtheories in §5 and possible implementation strategies for the logic in §6 .

2. The Syntax and Semantics of the Feature-Value Logic

This logic is a quantifier free first order language with both set and sequence descriptions. Intuitively, the underlying set theory is ZF − FA − EXT + AFA (where EXT is the axiom of extensionality, FA is the foundation axiom and AFA is Aczel's anti-foundation axiom). To cast this in more familiar terminology, two *type* identical elements of the domain need not be *token* identical. *Token identity* is indicated in the language via conjoining of the same variable to two or more descriptions. This is a generalisation of the notions of *type identity* and *token identity* familiar from conventional feature value logic semantics to set theory in general. Furthermore, we allow *nonwellfounded* structures. That is, nothing in the definition of the semantics prevents circular structures, i.e., structures which contain themselves. Otherwise, the set theory has the properties of classical set theory. However, in this paper, we will reconstruct the properties of the set theory we intend within standard set theory while observing that there is no difficulty in extending this treatment to either extensional or intensional nonwellfounded set theory.

2.1. The Domain of Interpretation

Every element, $\mathcal{U}^i$, of the *universe* or *domain of interpretation*, is a pair $\langle i, \mathcal{U}\rangle$ where $i \in \mathbb{N}$ is the *index* and $\mathcal{U}$ is a *structure* which is one of the *basic types*. There are four basic types. They are *constants, feature structures, sets* and *sequences*. We will call a pair $\langle i, \mathcal{U}\rangle$ an *i-constant, i-feature structure, i-set* or *i-sequence* according to the type of $\mathcal{U}$. The *i-* is an abbreviation for *intensional*. So, an i-set is an *intensional set*. Although we will carefully distinguish between i-types and basic types in this section, we may occasionally refer to basic types in what follows when we really mean i-types.

We will use the following notational conventions. Script capitals denote the class of objects of basic types. +-superscripted script capitals denote the class of objects of the corresponding i-types. Bold script capitals denote elements of the types. Bold script capitals with superscript i denote elements of the i-types with index i. Capital Greek letters denote the class of descriptions of the i-types and lowercase Greek letters denote descriptions of elements of the i-types. I.e., $\mathcal{A}$ is the class of constants, $\mathcal{A}^+$ is the class of i-constants, $\mathcal{A}$ ($\in \mathcal{A}$) is a constant, $\mathcal{A}^i$ ($\in \mathcal{A}^+$) = $\langle i, \mathcal{A}\rangle$ is an i-constant, A is the class of i-constant descriptions and α ($\in$ A) is a description of an i-constant. We will also use +-superscripted bold script capitals to denote elements of an i-type when we don't need to mention the index. I.e., $\mathcal{A}^+ \in \mathcal{A}^+$ is an i-constant, etc. $\mathcal{F}$ is the class of feature structures, $\mathcal{K}$ the class of sets and $\mathcal{S}$ the class of sequences. $\mathcal{U} = \mathcal{A} \cup \mathcal{F} \cup \mathcal{K} \cup \mathcal{S}$ is the class of basic types. $\mathcal{U}^+ = \mathcal{A}^+ \cup \mathcal{F}^+ \cup \mathcal{K}^+ \cup \mathcal{S}^+$ is the class of basic i-types, i.e., the domain of interpretation. Sets and sequences may be heterogenous and are not limited to members of one particular type. A feature structure $\mathcal{F} \in \mathcal{F}$ is a partial function $\mathcal{F}: \mathcal{A} \rightarrow \mathcal{U}^+$. We will follow these conventions below in the presentation of the syntax and semantics of the language.

2.2. Syntax

2.2.1 Notational Conventions

Below, we present an inductive definition of the syntax of the language. A is the set of i-constant descriptions, N is the set of (object language) variables, Φ is the set of i-feature structure descriptions, K is the set of i-set descriptions, Σ is the set of i-sequence descriptions and $\Psi = A \cup N \cup \Phi \cup K \cup \Sigma$ is the set of descriptions of i-structures (formulas) of the entire language. Object language variables are uppercase-initial atoms. (I.e., they follow the Prolog convention.) Lowercase Greek letters are metavariables over descriptions of structures of the corresponding intensional type (E.g., $\alpha \in A$ is an i-constant description, $\phi \in \Phi$ is an i-feature structure description, $\kappa \in K$ is an i-set description and $\sigma \in \Sigma$ is an i-sequence description $\nu \in N$ may denote a structure of any i-type.)

2.2.2. Definition

Given the notational conventions, Ψ is inductively defined as follows:

(a) $\alpha \in A$

(b) $\nu \in N$

(c) $\phi \in \Phi ::= \alpha:\psi$

(d) $\kappa \in K ::= \nu \mid \{\} \mid \{\psi_1, \ldots, \psi_n\} \mid \kappa_1 \cup \kappa_2 \mid \kappa_1 \oplus \kappa_2 \mid [\sigma]$

(e) $\sigma \in \Sigma ::= \nu \mid \langle\rangle \mid \sigma_1 \bullet \sigma_2 \mid \langle\psi_1, \ldots, \psi_n\rangle \mid \sigma_1 \cup_{\leq} \sigma_2 \mid \{\psi_1, \ldots, \psi_n\}_{\leq} \mid \psi_1 \leq \psi_2 \mid \mathfrak{q} \otimes \mathfrak{Q}$

(f) $\psi \in \Psi ::= \alpha \mid \nu \mid \phi \mid \psi_1 \wedge \psi_2 \mid \psi_1 \vee \psi_2 \mid \sim\psi$

2.2.3. Notes on the syntax

We define $\psi_1 \rightarrow \psi_2$ to be $\sim\psi_1 \vee \psi_2$ and $\psi_1 \leftrightarrow \psi_2$ to be $(\sim\psi_1 \vee \psi_2) \wedge (\sim\psi_2 \vee \psi_1)$ in the usual way.

Set descriptions ($\{\psi_1, \ldots, \psi_n\}$) are multisets of formulas. Set descriptions describe i-sets of i-structures. A *set union* description ($\kappa_1 \cup \kappa_2$) describes the union of two i-sets. The union of two i-sets is an i-set whose second component is the union of the second components of the two operand i-sets. (Note that this definition means that the indices of the two subsets do not contribute to the union.)

A *sequence concatenation* description ($\sigma_1 \bullet \sigma_2$) describes the concatenation of two i-sequences. (Sometimes in grammars, we will be sloppy and write subformulas which denote arbitrary i-types. This should be understood as a shorthand for subformulas surrounded by sequence brackets). $\{\psi_1, \ldots, \psi_n\}_{\leq}$ describes an i-sequence of elements the order of which is unspecified. $\psi_1 \leq \psi_2$ describes an implicitly universally quantified ordering constraint over a sequence. The intuitive interpretation is: "$\psi_1 \leq \psi_2$ is satisfied by a sequence if every element of the sequence that satisfies ψ_1 precedes (or is equal to) every element of the sequence that satisfies ψ_2". This is essentially the same interpretation as that given to GPSG LP constraints (as modified for sequences).

2.2.4. Matrix notation and other abbreviatory conventions

We will use a variant of the familiar matrix notation below adapted to the extra expressive power that our logic provides. We will briefly outline here the translation from the matrix notation to the logic.

A conjunction of feature-value pairs $\alpha_1{:}\psi_1 \wedge \ldots \wedge \alpha_n{:}\psi_n$ is represented using the traditional matrix notation:

$$\begin{bmatrix} \alpha_1{:}\psi_1 \\ . \\ . \\ . \\ \alpha_n{:}\psi_n \end{bmatrix}$$

Any other type of conjunction is represented as specified above. The connectives $\rightarrow, \vee, \sim, \leftrightarrow$ are used in the normal way except that their arguments may be conjunctions written in matrix notation. For set (sequence) descriptions, "big" set (sequence) brackets are used where the elements of the set (sequence) may be in matrix notation. We will also often use boxed integers in the matrix notation to indicate identity instead of variables. The interpretation should be obvious.

We will also use a few abbreviatory syntactic conventions. They should be obvious and will be introduced as needed. For example, the following formulas are formally equivalent

$$\psi_1 \leq \psi_2 \leq \psi_3$$
$$\psi_1 \leq \psi_2 \wedge \psi_2 \leq \psi_3$$

In addition, we will occasionally write partial ordering statements in which the first (second) description in the ordering statement is a variable which denotes a sequence. In this case, the intent is that the elements of the denoted sequence all follow (precede) the elements satisfying the other description. For example, if VP denotes a sequence of feature structures then the description

cat: verb $\leq$ VP

stands for

(cat:verb $\leq$ Initial) $\wedge$
(NonVP $\cup_{\leq}$ (VP $\wedge$ ($\langle$Initial$\rangle$ $\bullet$ Tail)))

and all of the elements of the VP sequence must follow any verb. Similarly,

VP $\leq$ cat: verb

stands for

(Final $\leq$ cat: verb) $\wedge$
(NonVP $\cup_{\leq}$ (VP $\wedge$ (Front $\bullet$ $\langle$Final$\rangle$)))

and all of the elements of the VP sequence must precede any verb.

2.3. Semantics

An *i-structure*, $\mathcal{U}^i$, is an element of $\mathcal{U}^+$. A function g: N $\rightarrow \mathcal{U}^+$ is an *assignment to variables*. A *model* is a pair $\langle \mathcal{U}^i, g \rangle$

2.3.1. Constants

(a) $\langle \mathcal{A}^i, g \rangle \models \alpha$ iff $\mathcal{A}^i = \langle i, \alpha \rangle = \langle i, \mathcal{A} \rangle$ (i.e., $\alpha = \mathcal{A} \in \mathcal{A}$)

2.3.2. Variables

(b) $\langle \mathcal{U}^+, g \rangle \models \nu$ iff $g(\nu) = \mathcal{U}^+$ ($\nu \in$ N)

2.3.3. Feature-value pairs

(c) $\langle \mathcal{F}^+, g \rangle \models \alpha{:}\psi$ iff $\mathcal{F}\downarrow\alpha$ and $\langle \mathcal{F}(\alpha), g \rangle \models \psi$

2.3.4. Classical connectives

(d) $\langle \mathcal{U}^+, g \rangle \models \psi_1 \wedge \psi_2$ iff $\langle \mathcal{U}^+, g \rangle \models \psi_1$ and $\langle \mathcal{U}^+, g \rangle \models \psi_2$

(e) $\langle \mathcal{U}^+, g \rangle \models \psi_1 \vee \psi_2$ iff $\langle \mathcal{U}^+, g \rangle \models \psi_1$ or $\langle \mathcal{U}^+, g \rangle \models \psi_2$

(f) $\langle \mathcal{U}^+, g \rangle \models \sim\psi$ iff $\langle \mathcal{U}^+, g \rangle \not\models \psi$

2.3.5. Set descriptions

(g) $\langle \varnothing^+, g \rangle \models \{\}$

(h) $\langle \mathcal{K}^+, g \rangle \models \kappa$ where $\kappa = \{\psi_1, \ldots, \psi_n\}$ iff there exists a surjection π: n $\rightarrow \mathcal{K}$ s.t. $\forall i \in$ n: $\langle \pi(i), g \rangle \models \psi_i$

(i) $\langle \mathcal{K}^+, g \rangle \models \kappa_1 \cup \kappa_2$ iff $\exists \mathcal{K}^+_1 \mathcal{K}^+_2$: $\mathcal{K} = \mathcal{K}_1 \cup \mathcal{K}_2$ and $\langle \mathcal{K}^+_1, g \rangle \models \kappa_1$ and $\langle \mathcal{K}^+_2, g \rangle \models \kappa_2$

(j) $\langle \mathcal{K}^+, g \rangle \models \kappa_1 \oplus \kappa_2$ iff $\exists \mathcal{K}^+_1 \mathcal{K}^+_2$: $\mathcal{K} = \mathcal{K}_1 \cup \mathcal{K}_2$ and $\mathcal{K}_1 \cap \mathcal{K}_2 = \emptyset$ and $\langle \mathcal{K}^+_1, g \rangle \models \kappa_1$ and $\langle \mathcal{K}^+_2, g \rangle \models \kappa_2$

(k) $\langle \mathcal{K}^+, g \rangle \models [\sigma]$ iff $\exists \mathcal{S}^+$: $\langle \mathcal{S}^+, g \rangle \models \sigma$ and $\mathcal{K} = [\mathcal{S}]$

2.3.6. Sequence descriptions

(l) $\langle \langle \rangle^+, g \rangle \models \Diamond$

(m) $\langle \mathcal{S}^+, g \rangle \models \sigma_1 \bullet \sigma_2$ iff $\exists \mathcal{S}^+_1 \mathcal{S}^+_2$: $\mathcal{S} = \mathcal{S}_1 \bullet \mathcal{S}_2$ and $\langle \mathcal{S}^+_1, g \rangle \models \sigma_1$ and $\langle \mathcal{S}^+_2, g \rangle \models \sigma_2$

(n) $\langle \mathcal{S}^+, g \rangle \models \langle \psi_1, \ldots, \psi_n \rangle$ iff $\exists \mathcal{U}^+_1, \ldots, \mathcal{U}^+_n$: $\mathcal{S} = \langle \mathcal{U}^+_1, \ldots, \mathcal{U}^+_n \rangle$ and $\langle \mathcal{U}^+_1, g \rangle \models \psi_1, \ldots, \langle \mathcal{U}^+_n, g \rangle \models \psi_n$

(o) $\langle \mathcal{S}^+, g \rangle \models \{\psi_1, \ldots, \psi_n\}_{\leq}$ iff $\exists \mathcal{K}^+$: $\mathcal{K} = [\mathcal{S}]$ and $\langle \mathcal{K}^+, g \rangle \models \{\psi_1, \ldots, \psi_n\}$

(p) $\langle \mathcal{S}^+, g \rangle \models \psi_1 \leq \psi_2$ iff $\mathcal{S} = \langle \mathcal{U}^+_1, \ldots, \mathcal{U}^+_n \rangle$ and $\forall ij \in n$ s.t. $\langle \mathcal{U}^+_i, g \rangle \models \psi_1$ and $\langle \mathcal{U}^+_j, g \rangle \models \psi_2$: $i \leq j$

(q) $\langle \mathcal{S}^+, g \rangle \models \sigma_1 \cup_{\leq} \sigma_2$ iff $\exists \mathcal{S}^+ \mathcal{S}'^+$: $\langle \mathcal{S}^+, g \rangle \models \sigma_1$ and $\langle \mathcal{S}'^+, g \rangle \models \sigma_2$ and $[\mathcal{S}] = [\mathcal{S}] \cup [\mathcal{S}']$ and n = length($\mathcal{S}$) and l = length($\mathcal{S}'$) and m = length($\mathcal{S}'$) and $\exists \pi' \pi''$ s.t. π': l $\rightarrow$ n and π'': m $\rightarrow$ n and range(π') $\cup$ range(π'') = n and $\forall i,j \in \pi'$: $i \leq j \rightarrow \pi'(i) \leq \pi'(j)$ and $\forall i,j \in \pi'$: $i \leq j \rightarrow \pi'(i) \leq \pi'(j)$

(r) $\langle \mathcal{S}^+, g \rangle \models \sigma_1 \otimes \sigma_2$ iff $\langle \mathcal{S}^+, g \rangle \models \sigma_1 \cup_{\leq} \sigma_2$ and $\exists \pi' \pi''$ as in (q) and range(π') $\cap$ range(π'') = $\emptyset$

2.3.7. Notes on the semantics

Note that the set of syntactic constants A and the set of semantic constants $\mathcal{A}$ are the same, i.e., $A = \mathcal{A}$ and $[\![\alpha]\!] = \alpha$. $\bullet$ is the *sequence concatenation* operator. It is a total function $\bullet: \mathcal{S} \times \mathcal{S} \rightarrow \mathcal{S}$. It is defined to be

$$\langle \mathcal{U}_1, \ldots, \mathcal{U}_i \rangle \bullet \langle \mathcal{U}_{i+1}, \ldots, \mathcal{U}_n \rangle = \langle \mathcal{U}_1, \ldots, \mathcal{U}_n \rangle.$$

$[\mathcal{S}]$ is the *underlying set* of the sequence $\mathcal{S}$, i.e., the set consisting of the elements of sequence $\mathcal{S}$.

2.3.8. The feature structure notation for models

Below we will use matrix notation for representing i-structures. Since i-structures are completely conjunctive, there is no indication of disjunction, negation or implication. Furthermore, the order of elements in i-sequences are totally specified so there are no partial ordering statements. I-structures are composed of only i-feature structures, i-sets, i-sequences and i-constants.

Obviously, there are no variables in structures. Rather than explicitly indicate all indices of intensional structures, identity of two structures is indicated with boxed integers.

2.4. A Partial Proof Theory

We use a partial Hilbert-style proof theory consisting of one rule of inference and many axioms and axiom schema. Space prevents us from presenting even this partial proof theory. We will note briefly that many of the axioms allow rather large disjunctions to be inferred. For example, if we have a formula

$$\langle 1,2 \rangle \wedge (S1 \bullet S2)$$

then we can infer

$$((S1 \wedge \langle 1,2 \rangle) \bullet (S2 \wedge \Diamond)) \vee ((S1 \wedge \langle 1 \rangle) \bullet (S2 \wedge \langle 2 \rangle)) \vee ((S1 \wedge \Diamond) \bullet (S2 \wedge \langle 1,2 \rangle)).$$

Similar axioms hold for most of the two place connectives in the language including *sequence union*.

The only rule of inference is modus ponens.

From α and $\alpha \rightarrow \beta$ infer β.

3. The organisation of the grammar

3.1. Basic organisation

$\Lambda = \{\delta_1, \ldots, \delta_m\}$ is the set of *lexical signs*. $P = \{\rho_1, \ldots, \rho_n\}$ is the set of *nonlexical signs*. The *sign axiom*, $\phi_\Sigma \in \mathcal{T}$, encodes the signs $\Delta \cup P$ where

$$\phi_\Sigma: (\text{cat: Cat}) \rightarrow (\delta_1 \vee \ldots \vee \delta_m \vee \rho_1 \vee \ldots \vee \rho_n).$$

A model $\mathcal{M}$ *satisfies* a formula ψ with respect to a theory $\mathcal{T} = \{t_1, \ldots, t_n\}$, written $\mathcal{M} \models_{\mathcal{T}} \psi$ iff

$$\mathcal{M} \models t_1 \wedge \ldots \wedge t_n \wedge \psi.$$

(We assume that the individual formulas in a theory have disjoint variables. When they don't, the assumption is that the variables in the entire theory are renamed such that this property holds.)

A sequence P is a category C iff

$$\exists \mathcal{M} \text{ s.t. } \mathcal{M} \models_{\mathcal{T}} \begin{bmatrix} \text{phon: P} \\ \text{cat: C} \end{bmatrix}.$$

The set of all sequences Σ of category C is

$$\Sigma = \left\{ \sigma \mid \exists \mathcal{M} \text{ s.t. } \mathcal{M} \models_{\mathcal{T}} \begin{bmatrix} \text{phon: } \sigma \\ \text{cat: C} \end{bmatrix} \right\}.$$

(This provides the *generates* relation for a grammar.)

3.2. Two Axioms

The following two axiom schema are included in every grammar which we consider.

The dtrs-phon axiom

((phon: Phon) ∧ dtrs:⟨phon: X1, ... ,phon: Xn⟩) ↔
phon: (X1 • ... • Xn)

This axiom states that the value of the **phon** feature is the concatenation of the **phon** features of the elements of the **dtrs** sequence in the same order as they occur in the **dtrs** sequence. This means that the **phon** sequence of any feature structure is completely *flat*. That is, there are no embedded levels of sequence structure corresponding to phrase structure.

The head-subcat-slash-dtrs axiom

(head: Head) ∧ (subcat: Subcat) ∧ (dtrs: Dtrs) ∧
(slash: Slash) →
subcat: ({dtrs: X1, ... , dtrs: Xn} ⊕ [NonUnionSubcat]
⊕ Slash) ∧
dtrs: ({Head} ⊗ (X1 $\cup_{\leq}$ ... $\cup_{\leq}$ Xn) ⊗
NonUnionSubcat)

This axiom says that in any headed sign, any element of the **subcat** set is either an element of the **slash** set, an element of the **dtrs** sequence or is "unioned into" the **dtrs** sequence and that there are no other elements of the **slash** set or **dtrs** sequence.

3.3. A simple example

Consider the following three element lexicon.

$$\phi 1 = \begin{bmatrix} \text{phon: Phon} \\ \text{cat: sentence} \\ \text{head: } \begin{bmatrix} \text{phon: } \langle \text{likes} \rangle \\ \text{cat: verb} \end{bmatrix} \\ \text{subcat: } \left\{ \begin{bmatrix} \text{phon: Subj} \\ \text{cat: np} \\ \text{case: nom} \end{bmatrix}, \begin{bmatrix} \text{phon: Obj} \\ \text{cat: np} \\ \text{case: acc} \end{bmatrix} \right\} \\ \text{dtrs: Dtrs} \\ \text{slash: Slash} \end{bmatrix}$$

$$\phi 2 = \begin{bmatrix} \text{phon: } \langle \text{he} \rangle \\ \text{cat: np} \\ \text{case: nom} \end{bmatrix}$$

$$\phi 3 = \begin{bmatrix} \text{phon: } \langle \text{her} \rangle \\ \text{cat: np} \\ \text{case: acc} \end{bmatrix}$$

Then the grammar $\mathcal{T}$ is the one axiom theory $\mathcal{T} = \{\phi\}$ where ϕ = cat: C → $\phi 1 \vee \phi 2 \vee \phi 3$.

That is, if a FS is defined for **cat** then it must satisfy one of $\phi 1$, $\phi 2$ or $\phi 3$. Given this grammar, the only sentence defined is "he likes her" and the only NP's defined are "he" and "her".

Consider the description

$$\begin{bmatrix} \text{phon: } \langle \text{X,likes,Y} \rangle \\ \text{cat: C} \end{bmatrix}.$$

Then the minimal FS which satisfies it is

$$\begin{bmatrix} \text{phon: } \langle \text{he,likes,her} \rangle \\ \text{cat: sentence} \\ \text{head: } \begin{bmatrix} \text{phon: } \langle \text{likes} \rangle \\ \text{cat: verb} \end{bmatrix} \boxed{2} \\ \text{subcat: } \left\{ \begin{bmatrix} \text{phon: } \langle \text{he} \rangle \\ \text{cat: np} \\ \text{case: nom} \end{bmatrix} \boxed{1}, \begin{bmatrix} \text{phon: } \langle \text{her} \rangle \\ \text{cat: np} \\ \text{case: acc} \end{bmatrix} \boxed{3} \right\} \\ \text{dtrs: } \{\boxed{1}, \boxed{2}, \boxed{3}\} \\ \text{slash: } \{\} \end{bmatrix}$$

4. An analysis of Dutch subordinate clauses

In this section, we will present a toy analysis of simple Dutch subordinate clauses. The example that we will look at is the clause *Jan Piet Marie zag helpen zwemmen* (minus the complementiser *omdat*). We require the following lexical entries.

$$\text{'Jan'}: \begin{bmatrix} \text{phon: } \langle \text{Jan} \rangle \\ \text{cat: np} \end{bmatrix}$$

$$\text{'Piet'}: \begin{bmatrix} \text{phon: } \langle \text{Piet} \rangle \\ \text{cat: np} \end{bmatrix}$$

$$\text{'Marie'}: \begin{bmatrix} \text{phon: } \langle \text{Marie} \rangle \\ \text{cat: np} \end{bmatrix}$$

$$\text{'zag'}: \begin{bmatrix} \text{phon: Phon} \\ \text{cat: sentence} \\ \text{vform: fin} \\ \text{head: } \begin{bmatrix} \text{phon: } \langle \text{zag} \rangle \\ \text{cat: verb} \\ \text{vform: fin} \end{bmatrix} \\ \text{subcat: } \{\phi 1, \phi 2, \phi 3\} \\ \text{dtrs: Dtrs} \\ \text{slash: Slash} \end{bmatrix}$$

where $\phi 1$, $\phi 2$, and $\phi 3$ are:

$$\phi 1 = \begin{bmatrix} \text{phon: Subj} \\ \text{cat: np} \\ \text{case: nom} \end{bmatrix}$$

$$\phi 2 = \begin{bmatrix} \text{phon: Obj} \\ \text{cat: np} \\ \text{case: acc} \end{bmatrix}$$

$$\phi 3 = \begin{bmatrix} \text{phon: VP} \\ \text{cat: vp} \\ \text{vform: inf} \end{bmatrix}$$

'helpen':

$$\begin{bmatrix} \text{phon: Phon} \\ \text{cat: vp} \\ \text{vform: inf} \\ \text{head: } \begin{bmatrix} \text{phon: } \langle \text{helpen} \rangle \\ \text{cat: verb} \\ \text{vform: inf} \end{bmatrix} \\ \text{subcat: } \left\{ \begin{bmatrix} \text{phon: NP} \\ \text{cat: np} \\ \text{case: acc} \end{bmatrix}, \begin{bmatrix} \text{phon: VP} \\ \text{cat: vp} \\ \text{vform: inf} \end{bmatrix} \right\} \\ \text{dtrs: Dtrs} \\ \text{slash: Slash} \end{bmatrix}$$

'zwemmen':

$$\begin{bmatrix} \text{phon: Phon} \\ \text{cat: vp} \\ \text{vform: inf} \\ \text{head: } \begin{bmatrix} \text{phon: } \langle \text{zwemmen} \rangle \\ \text{cat: verb} \\ \text{vform: inf} \end{bmatrix} \\ \text{subcat: } \{\} \\ \text{dtrs: Dtrs} \\ \text{slash: Slash} \end{bmatrix}$$

We also need the following axioms.

cat: (vp ∨ sentence) ∧ subcat: ({(cat: vp) ∧ (dtrs: Dtrs) ∧ VP} ∪ X) → ((extra: - ∧ dtrs: (Dtrs ∪≤ Y)) ∨ (-extra: Z ∧ slash: ({VP} ∪ W)))

dtrs: Dtrs → dtrs: (cat: np ≤ cat: verb ∧ case: nom ≤ case: acc)

((head: Head) ∧ (dtrs: Dtrs)) → dtrs: (Head ≤ cat: verb)

The first axiom simply states that VP complements are either extracted (i.e., members of the **slash** set) or are sequence unioned into the **dtrs** sequence. The second axiom says that NPs precede verbs and that nominative NPs precede accusative NPs. The third axiom says that a head precedes any other daughters in the **dtrs** sequence. This encodes the generalisation for Dutch subordinate clauses that governing verbs precede governed verbs.

We'll now present the analysis. (We will necessarily have to omit considerable detail due to considerations of space.) We start as indicated in §3 with the following description

phon: ⟨Jan,Piet,Marie,zag,helpen,zwemmen⟩ ∧ cat: C

The sign axiom will have the disjunction of the six lexical entries in its consequent. Since our formula is specified for **cat**, thus satisfying the antecedent of the sign axiom, we can apply the sign axiom. The disjunct that we will pursue will be the one for 'zag'. This means we infer the formula

$$\begin{bmatrix} \text{phon: } \langle \text{Jan,Piet,Marie,zag,helpen,zwemmen} \rangle \\ \text{cat: sentence} \\ \text{vform: fin} \\ \text{head: } \begin{bmatrix} \text{phon: } \langle \text{zag} \rangle \\ \text{cat: verb} \\ \text{vform: fin} \end{bmatrix} \\ \text{subcat: } \{\phi_1, \phi_2, \phi_3\} \\ \text{dtrs: Dtrs} \\ \text{slash: Slash} \end{bmatrix}$$

(where ϕ_1, ϕ_2, ϕ_3 are as in the lexical entry for 'zag') From the **head-subcat-slash-dtrs** axiom we ca infer a large disjunction one of whose disjuncts is

$$\begin{bmatrix} \text{phon: } \langle \text{Jan,Piet,Marie,zag,helpen,zwemmen} \rangle \\ \text{cat: sentence} \\ \text{vform: fin} \\ \text{head: } \begin{bmatrix} \text{phon: } \langle \text{zag} \rangle \\ \text{cat: verb} \\ \text{vform: fin} \end{bmatrix} \wedge \text{D4} \\ \text{subcat: } \{\text{D1} \wedge \phi_1', \text{D2} \wedge \phi_2', \phi_3'\} \\ \text{dtrs: } \langle \text{D1,D2,D3,D4,D5,D6} \rangle \\ \text{slash: } \{\} \end{bmatrix}$$

where ϕ_1', ϕ_2' and ϕ_3' are:

$$\phi_1' = \begin{bmatrix} \text{phon: } \langle \text{Jan} \rangle \\ \text{cat: np} \\ \text{case: nom} \end{bmatrix}$$

$$\phi_2' = \begin{bmatrix} \text{phon: } \langle \text{Piet} \rangle \\ \text{cat: np} \\ \text{case: acc} \end{bmatrix}$$

$$\phi_3' = \begin{bmatrix} \text{phon: } \langle \text{Marie,helpen,zwemmen} \rangle \\ \text{cat: vp} \\ \text{vform: inf} \\ \text{dtrs: } \langle \text{D3,D5,D6} \rangle \end{bmatrix}$$

Again, we can apply the sign axiom to each of thes embedded formulas. ϕ_1' and ϕ_2' will be consister with the lexical entries for 'Jan' and 'Pie respectively and can be rewritten no further. ϕ_3' wil be consistent with the lexical entry for 'helpen' so w will be able to infer

$$\begin{bmatrix} \text{phon: } \langle \text{Marie,helpen,zwemmen} \rangle \\ \text{cat: vp} \\ \text{vform: inf} \\ \text{head: } \begin{bmatrix} \text{phon: } \langle \text{helpen} \rangle \\ \text{cat: verb} \\ \text{vform: inf} \end{bmatrix} \\ \text{subcat: } \left\{ \begin{bmatrix} \text{phon: NP} \\ \text{cat: np} \\ \text{case: acc} \end{bmatrix}, \begin{bmatrix} \text{phon: VP} \\ \text{cat: vp} \\ \text{vform: inf} \end{bmatrix} \right\} \\ \text{dtrs: } \langle \text{D3,D5,D6} \rangle \\ \text{slash: Slash} \end{bmatrix}$$

Again, from the **head-subcat-slash-dtrs** axiom we can infer a large disjunction one of whose disjuncts is

$$\begin{bmatrix} \text{phon: } \langle \text{Marie,helpen,zwemmen} \rangle \\ \text{cat: vp} \\ \text{vform: inf} \\ \text{head: } \begin{bmatrix} \text{phon: } \langle \text{helpen} \rangle \\ \text{cat: verb} \\ \text{vform: inf} \end{bmatrix} \wedge \text{D5} \\ \text{subcat: } \{\phi_4' \wedge \text{D3}, \phi_5' \wedge \text{D6}\} \\ \text{dtrs: } \langle \text{D3,D5,D6} \rangle \\ \text{slash: } \{\} \end{bmatrix}$$

where ϕ_4' and ϕ_5' are

$$\phi_4' = \begin{bmatrix} \text{phon: } \langle \text{Marie} \rangle \\ \text{cat: np} \\ \text{case: acc} \end{bmatrix}$$

$$\phi_5' = \begin{bmatrix} \text{phon: } \langle \text{zwemmen} \rangle \\ \text{cat: vp} \\ \text{vform: inf} \end{bmatrix}.$$

Again the sign axiom can be applied to the subcategorised accusative NP and VP. The NP is consistent with the sign for 'Marie' and no further rewriting is possible. The VP is consistent with the sign for 'zwemmen' and so we can infer

$$\begin{bmatrix} \text{phon: } \langle \text{zwemmen} \rangle \\ \text{cat: vp} \\ \text{vform: inf} \\ \text{head: } \begin{bmatrix} \text{phon: } \langle \text{zwemmen} \rangle \\ \text{cat: verb} \\ \text{vform: inf} \end{bmatrix} \\ \text{subcat: } \{\} \\ \text{dtrs: Dtrs} \\ \text{slash: Slash} \end{bmatrix}$$

Again, the **head-subcat-slash-dtrs** axiom can be applied leaving only one possibility in this case, namely, that both **dtrs** and **slash** has value {}. No further rewriting is possible. Under the assumption that the proof theory axioms that we have used are sound, we have determined that the original clause is in fact a finite sentence of the theory.

There are two other points to make about the analysis. First, the first axiom we gave above guaranteed that VP complements which are specified **extra:** - are sequence unioned into the surrounding sign while NPs are not. We simply chose the **extra:** - option for every complement VP. Second, although we freely guessed at the values of **dtrs** sequences (within the limits allowed by the **head-subcat-slash-dtrs** axiom) a quick glance will establish that every **dtrs** sequence obeys the ordering constraints expressed in the second and third axioms.

A few words are in order about how we can accomodate "canonical" German and Swiss-German subordinate clause order. In either case, the first axiom is maintained as is. For German we need to either eliminate the strict ordering condition concerning case of NPs in the second axiom or add disjunctive ordering constraints for NPs as Uszkoreit suggests. The ordering constraints for Swiss-German are essentially the same. The first half of the consequent of the second axiom must be maintained for German. For Swiss-German, however, this constraint must be eliminated. It seems that the correct generalisation for at least the Zürich dialect (Züritüüsch) is that NP complements need only precede the verb that they depend on but not all verbs. (Cf. Cooper 1988.) Therefore, for Züritüüsch we must add an axiom something like

subcat: ({cat: np ∧ NP} ∪ X) ∧ head: (cat: verb ∧ Verb) → dtrs: (NP ≤ Verb).

(This condition is actually more general than the first half of the consequent of the original second axiom. I.e., it is a logical consequent of the second axiom.)

For German, the third axiom is simply the one for Dutch with the order of **Head** and **cat: verb** reversed. This encodes the generalisation for German subordinate clauses that governed verbs precede governing verbs. For Züritüüsch, the third axiom is simply eliminated since verbs are unordered with respect to each other.

This analysis has been oversimplified in every respect and has ignored a considerable amount of data which violates one or more of the axioms given. It is intended to be strictly illustrative. It should, however, indicate that for "canonical" subordinate clauses, the differences which account for the variation in Dutch, German and Züritüüsch word order are fairly small and related in straightforward ways. It is this aspect which we briefly address next.

5. Parametric Variation

If T_1 and T_2 are theories and $T_1 \subseteq T_2$, then T_2 is a *subtheory* of T_1. This means that T_2 axiomatises a smaller class of algebraic structures than T_1. Typically, T_1 (and T_2) contain many implicational axioms. The implicational axioms of T_1 actually limit the class of structures which T_2 axiomatises. A theory of universal grammar has a natural interpretation in terms of algebraic theories, subtheories and implicational axioms which potentially allows a richer account of parametric variation than the naive parameter setting interpretation. The approach is entirely analogous to the relation of the theories of Brouwerian and Boolean lattices to the general theory of lattices.

6. Implementation

There has been no work done yet on the implementation of the logic. There are at least three obvious implementation strategies. First, as implied in §3, parsing of a sequence P as a category C can be reduced to testing satisfiability of the formula **phon:** P $\wedge$ **cat:** C. This means that we should be able to use a general purpose proof environment (such as Edinburgh LF) to implement the logic and test various proof theories for it. Second, there is an interpretation in terms of *head-driven parsing* (Proudian and Pollard 1985). Third, we might try to take advantage of the simple structure of the grammars (i.e., the dependency of **phon** on **dtrs** sequences) and implement a parser augmented with *sequence union*. We hope to investigate these possibilities in the future.

7. Conclusion

There are several comments to make here. First, the specific logic presented here is not important in itself. There are undoubtedly much better ways of formalising the same ideas. In particular, the semantics of the logic is unduly complicated compared to the simple intuitions about linguistic structure whose expression it is designed to allow. Specifically, a logic which uses *partially ordered intensional sets* instead of sequences is simpler and intuitively more desirable. However, this approach also has its drawbacks. What is significant is the illustration that syntactic structure and a treatment of nonconfigurational word order can be treated within a single logical framework.

Second, the semantics is complicated a great deal by the reconstruction of intensional structures within classical set theory. A typed language which simply distinguishes atomic *tokens* from *types* and the use of intensional nonwellfounded set theory would give a far cleaner semantics.

Third, the programme outlined here is obviously unsatisfactory without a sound and complete proof theory. The entire point is to have a completely logical characterisation of grammar. A complete axiomatisation is still in work. This is largely due to the complexity of the semantics of set and sequence descriptions and the belief that there should be an adequate logic with a simpler (algebraic) semantics and consequently a simpler proof theory. We simply note here that we believe that a Henkin style completeness proof can be given for the logic (or an equivalent one).

8. Acknowledgements

I would first like to thank Jerry Seligman. If this paper makes any sense technically, it is due to his great generosity and patience in discussing the logic with me. I would also like to thank Inge Bethke for detailed comments on the semantics of the logic and Jo Calder and Ewan Klein for continuing discussion. Any errors in this paper are solely the author's responsibility.

9. References

Aczel, P. (1988) *Non-Well-Founded Sets.* CSLI Lecture Notes No. 14. Stanford.

Bresnan, J. (Ed.) (1982) *The Mental Representation of Grammatical Relations.* Cambridge, Mass.: MIT Press.

Cooper, K. (1988) Word Order in Bare Infinitival Complement Constructions in Swiss German. Master's Thesis, Centre for Cognitive Science, University of Edinburgh, Edinburgh.

Gazdar, G., E. Klein, G. K. Pullum and I.A. Sag. (1985) *Generalised Phrase Structure Grammar.* Cambridge: Blackwell, and Cambridge, Mass.: Harvard University Press.

Kasper, R. and W. Rounds. (1986) A Logical Semantics for Feature Structures. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics,* Columbia University, New York, 10-13 June, 1986, 235-242.

Johnson, M. (1987) Attribute-Value Logic and the Theory of Grammar. Ph.D. Thesis, Department of Linguistics, Stanford University, Stanford.

Pollard, C. and I. Sag. (1987) *Information-Based Syntax and Semantics.* CSLI Lecture Notes No. 13. Stanford.

Proudian, D. and C. Pollard. (1985) Parsing Head-Driven Phrase Structure Grammar. In Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics, University of Chicago, Chicago, 8-12 July, 1985, 167-171.

Smolka, G. (1988) A Feature Logic with Subsorts. Lilog-Report 33. May, 1988, IBM Deutschland, Stuttgart.