A Logic for Category Definition

Gerald Gazdar and Geoffrey K. Pullum

January 1987

Cognitive Science Research Paper

Serial No. CSRP 072

Cognitive Studies Programme,
The University of Sussex,
Brighton BN1 9QN

A LOGIC FOR CATEGORY DEFINITION

Gerald Gazdar and Geoffrey K. Pullum

Keywords: logic, unification, categories, syntax, computational linguistics

Abstract: This paper outlines a formal language for the statement of constraints on syntactic categories in order to represent the claims made by diverse grammatical frameworks in a uniform way, independent of the notations and substantive claims of any framework. We define a class of objects called "category structures[1]', each such object providing a constructive definition for a space of syntactic categories. Definitions for extension, unification, and identity are provided arbitrary syntactic categories, and a formal language for the statement of constraints on categories is provided. The language is basically a modal logic, one which includes K1.1 (also known as S4Grz), which is characterized by the class of all finite partial orderings. Some interesting special cases are exhibited. By combining a category structure with a set of constraints, one can define the category systems of many well-known grammatical frameworks. Checking a category for conformity to constraints can be done in linear time. This work aims at providing a unitary class of data structures for the representation of syntactic categories in a range of diverse grammatical frameworks, and thus making it possible for various pseudo-issues in natural language processing research to be avoided.

072

1. Introduction*

Syntactic categories are data structures containing information about grammatical constituents (words, phrases, clauses, sentences). In recent computational linguistics, categories are centrally important, and may have considerable internal complexity, often obscured by idiosyncratic notations and implicit abbreviations. Our work aims to provide a precise metatheoretical definition of 'syntactic category'. This exercise is complementary to that of Pereira and Shieber (1984) and to recent work of Rounds and others on the development of a logic for the description of elaborate grammatical representation structures (see Kasper and Rounds 1986, Moshier and Rounds 1986, Rounds and Kasper 1986).

2. Category structures

First we define the notion CATEGORY STRUCTURE. We will write '2' for the set $\{0, 1\}$, A^B for the set of total functions from B into A, $A^{(B)}$ for the set of partial functions from B into A, $\wp(A)$ for the power set of A, and $\Delta(f)$ for the domain of a (partial) function f (if f is a partial function then $\Delta(f)$ is the set of items which f assigns a value to).

A category structure is basically a choice of primitives (features and ranges of possible values for them). Formally, a category structure Σ is a quintuple $\langle K, F, A, \tau, \rho\rangle$, where K is the set of categories induced by F, A, τ and ρ, F is a finite set of features, A is a finite set of atoms, τ is a function in 2^F, and ρ is a function from $\{f|\tau(f)=0\}$ into $\wp(A)$. The function τ partitions F into two sets: the set of type 0 features $F^0 = \{f|\tau(f)=0\}$, and the set of type 1 features $F^1 = \{f|\tau(f)=1\}$. Type 0 features take atomic values; type 1 features take categories. The function ρ assigns a range of atomic values to each type 0 feature. K is recursively defined in terms of $\langle F, A, \tau, \rho\rangle$. We first define the set of pure type 0 categories of Σ (those containing only type 0 features), K^0, as in (1):

(1) $K^0 = A^{(F^0)} \cap \wp(\{\langle f, a\rangle | a \in \rho(f)\})$

Then we build up K as follows:

(2) a. $K_0 = \{\varnothing\}$

b. $K_{n+1} = \{C_0 \cup C_1 | C_0 \in K^0 \wedge C_1 \in K_n^{(F^1)}\}$

c. $K = \bigcup_{n \in N} K_n$ (where N is the set of natural numbers)

Given the way K is built up, the induction step in (2b) being restricted to union of finite partial functions, it should be clear that K is a recursive set.

We define certain relations and operations on the space K of potential categories. The most important is the relation **is extended by**, symbolized by '$\sqsubseteq$'. The definition is as follows.

---

* This work originates in joint research with Robert Carpenter, Thomas E. Hukari, Ewan H. Klein, and Robert D. Levine, and some of their ideas, particularly Carpenter's idea of adding a modal operator to the language of feature cooccurrence restrictions in Gazdar et al. (1985), are crucial. Gazdar et al. (1986) presents a fuller application of this work in linguistic theory. We are grateful to Joseph Halpern, David J. Israel, Ronald M. Kaplan, William A. Ladusaw, Richard E. Otte, Fernando Pereira, P. Stanley Peters, Carl J. Pollard, Stuart M. Shieber, and Manfred Warmuth For very helpful conversations. They are not to be associated with any errors that this paper may contain. We also thank Calvin J. Pullum, who did the diagrams. The research was supported by grants from the (U.K.) SERC and ESRC (Gazdar), NSF grants BNS-85 11687 and BNS-85 19708 (Pullum), and the UCSC Syntax Research Center.

3) Definition: **extension**

If a and b are atoms, then a is extended by b ($a \sqsubseteq b$) if and only if $a = b$.

If α and β are categories, then α is extended by β ($\alpha \sqsubseteq \beta$) if and only if:

(i) $\forall f \in \Delta(\alpha) \cap F^0\ [\alpha(f) = \beta(f)]$ and

(ii) $\forall f \in \Delta(\alpha) \cap F^1\ [\alpha(f) \sqsubseteq \beta(f)]$.

This yields an obvious definition for identity of two categories:

4) Definition: **identity**

If α and β are categories, $\alpha = \beta$ if and only if $\alpha \sqsubseteq \beta$ and $\beta \sqsubseteq \alpha$.

Sets of categories can be unified in a sense familiar from recent logic and grammatical theory. A constructive definition of the **unification** operation (symbolized $\sqcup$) is given in (5).

5) Definition: **unification**

If a and b are atoms, then $a \sqcup b = a$ if and only if $a = b$.

If α and β are categories, then

(i) if $\langle f, v\rangle \in \alpha$ but $\beta(f)$ is undefined, then $\langle f, v\rangle \in \alpha \sqcup \beta$;

(ii) if $\langle f, v\rangle \in \beta$ but $\alpha(f)$ is undefined, then $\langle f, v\rangle \in \alpha \sqcup \beta$;

(iii) if $\langle f, v_i\rangle \in \alpha$ and $\langle f, v_j\rangle \in \beta$, then $\langle f, v_i \sqcup v_j\rangle \in \alpha \sqcup \beta$ if $v_i \sqcup v_j$ is defined and $\alpha \sqcup \beta$ is undefined otherwise;

(iv) nothing else is in $\alpha \sqcup \beta$.

The relation $\sqsubseteq$ induces a meet semilattice on K. The greatest lower bound for $\sqsubseteq$ in a set $S \subseteq K$ is the largest category that is extended by all the members of S, i.e. the intersection of the members of S. This is always defined, even if it is empty; and the empty set is of course a category in K. However, $\sqsubseteq$ does not induce a lattice on K. For arbitrary sets $S \subseteq K$, there is not necessarily any category that all the members of S are extended by, and hence not every pair of categories in K has a join.

We now introduce a language for imposing constraints on K in order to separate out proper subsets that respect specific grammatical frameworks. Note that there is no guarantee that the categories used by some specified grammatical framework will yield a semilattice under $\sqsubseteq$, for constraints may make certain intersections of categories illegal, creating sets of categories having no intersection that is itself a category.

3. The constraint language L_C

We formalize constraints as statements that can be true or false of a category. A constraint delimits a subspace within the set K induced by a given category structure Σ, namely, the subspace of categories that satisfy the constraint. Different varieties of grammar use different subspaces.

Our goals in formulating our constraint language, L_C, are rather different from those of Rounds et al. The language L_C is a language for characterizing sets of legal categories, not a language whose expressions are intended for use in place of categories. Crudely put, our language is for category definition whereas Rounds' is for category manipulation.

We define two types of constraint: BASIC and COMPLEX. If f is an element of F, and a is an element of A, then there are just two distinct types of well-formed basic constraint: f and $(f\!: a)$, where $\tau(f) = 0$. The following are well-formed complex contraints: $\neg\phi$, $\Box\phi$, $\Diamond\phi$, $(\phi \vee \psi)$, $(\phi \wedge \psi)$, $(\phi \rightarrow \psi)$, $(\phi \leftrightarrow \psi)$, and $(f\!: \phi)$, where $f \in F^1$ and ϕ and ψ are well-formed basic or complex constraints. (Here and from now on, we will omit parentheses in the obvious way whenever they are not needed to prevent ambiguity in the statement of constraints.)

All well-formed expressions of L_C have the same kind of denotation—they denote truth values (i. e. members of 2) relative to category structure Σ and a category α in Σ. If ϕ is a well-formed expression of L_C, then we use $[\![\phi]\!]_{\Sigma,\alpha}$ to stand for the denotation of ϕ with respect to the category structure Σ and category α. If $[\![\phi]\!]_{\Sigma,\alpha} = 1$ then we shall say that α SATISFIES ϕ. Our semantic rules are the following, where a, f, ϕ, and ψ are as above.

(6) a. []/Q ^a = 1 iff a(/) is defined.

b. D/:aQ2;a-liffa(/)-a.

c. $[\![f:\phi]\!]_{\Sigma,\alpha} = 1$ iff $[\![\phi]\!]_{\Sigma,\alpha(f)} = 1$.

d. $[\![\neg\phi]\!]_{\Sigma,\alpha} = 1$ iff $[\![\phi]\!]_{\Sigma,\alpha} = 0$.

e. $[\![\phi \vee \psi]\!]_{\Sigma,\alpha} = 1$ iff $[\![\phi]\!]_{\Sigma,\alpha} = 1$ or $[\![\psi]\!]_{\Sigma,\alpha} = 1$.

f. $[\![\phi \wedge \psi]\!]_{\Sigma,\alpha} = 1$ iff $[\![\phi]\!]_{\Sigma,\alpha} = 1$ and $[\![\psi]\!]_{\Sigma,\alpha} = 1$.

8- $[\![\phi \rightarrow \psi]\!]_{\Sigma,\alpha} = 1$ iff $[\![\phi]\!]_{\Sigma,\alpha} = 0$ or $[\![\psi]\!]_{\Sigma,\alpha} = 1$.

h. $[\![\phi \leftrightarrow \psi]\!]_{\Sigma,\alpha} = 1$ iff $[\![\phi]\!]_{\Sigma,\alpha} = [\![\psi]\!]_{\Sigma,\alpha}$.

i. QD<))O^a= 1 iff Q(})Qia= land forall/inF1nA(a),DD(t)D^a(^ = 1.

j. $[\![\Diamond\phi]\!]_{\Sigma,\alpha} = 1$ iff $[\![\phi]\!]_{\Sigma,\alpha} = 1$ or for some f in $F^1 \cap \Delta(\alpha)$, $[\![\Diamond\phi]\!]_{\Sigma,\alpha(f)} = 1$.

Informally, (6a) says that a category satisfies/just in case it contains some specification for the feature /; (6b) says that a category satisfying *(f:a)* has as one of its elements the pair *(Jya)*\ and (6c) says of a category satisfying (/: (J)) that the type 1 feature / is defined and that the category which is the value of / satisfies <j). Clauses (6d) through (6h) are self-explanatory.

The modal operators introduced in (6i) and (6j) allow for recursive constraints to be imposed on successively embedded layers of category values. A category a satisfies D<|> provided that, firstly, a satisfies <|>, and secondly, whenever a assigns a category (3 to a type 1 feature/, p satisfies D<|>. The possibility operator in (6j) is, as usual, the dual of the necessity operator: 0(|) says of a category a satisfying it that either *a* satisfies <|>, or there exists a category-value P assigned to a type 1 feature/by a such that p satisfies 0$.

Note that if a C_P and a satisfies <j) it does NOT follow in L_c that p satisfies <|) (compare Rounds and Kasper 1986, Theorem 6). For example, we have 0 C {(F,^)} and 0 satisfies —iF, but {{F,a)} does not Likewise, the fact that both a and p satisfy some constraint <|> does not entail that a U P will satisfy <|>, even if a U P is defined. Observations such as these lead Moshier and Rounds (1986) to reject a classical semantics for their feature description language in favor of an intuitionistic semantics that, in effect, quantifies over possible extensions. But a classical semantics fits the intended meaning of our category constraints exactly, so there has been no need for us to make a similar move.

We will write h <|> to mean that for every category structure £ and category a in Z, a satisfies (j). Given this, we can list some valid formulae and valid formula schemata of the logic of category constraints.

(7) a. h (/: *a)* -> / (for all *a* e p(/), / E *F°)*

This simply says that if a feature has an atomic value, then it has a value. We also have all the valid formula of the standard prepositional calculus, which we will not list here. Furthermore, we have the following familiar valid modal formulas.

(7) b. $\models \Box\phi \leftrightarrow \neg\Diamond\neg\phi$

c. l=n(<j>-><{>)

d. hn<i>—>§

g^ 1= (^ —y Q<J>

f. hQ(<!>A\|/)«-»(n<{> AD\|/)

g. $\models \Diamond(\phi \vee \psi) \leftrightarrow (\Diamond\phi \vee \Diamond\psi)$

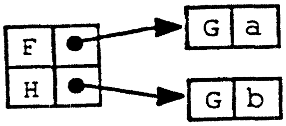
h. hn<>->nno

Here, (7h) shows us that our logic at least contains S4 (we follow the nomenclature of Hughes and Cresswell

968) throughout). But we do not have $\models \Diamond\phi \rightarrow \Box\Diamond\phi$, and so our logic does not contain S5. To see this, ›nsider the following category, assuming F is a category-valued feature: $\{\langle \text{F}, \varnothing\rangle\}$. This category satisfies $\Diamond$F but ›t $\Box\Diamond$F.

The category $\{\langle \text{F}, \{\langle \text{G}, a\rangle\}\rangle, \langle \text{H}, \{\langle \text{G}, b\rangle\}\rangle\}$ (graphically represented in (8), below) provides us with an ıalogous falsifying instance for $\models \Diamond\Box\phi \rightarrow \Box\Diamond\phi$ when we set $\phi = (\text{G}: a)$.

)



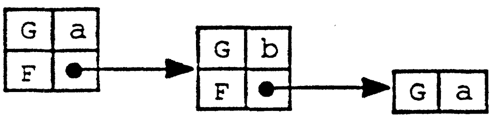
ıis shows that our logic does not contain S4.2. Interestingly, the converse of this constraint IS valid; hence:

) i. $\models \Box\Diamond\phi \rightarrow \Diamond\Box\phi$

ıis is easy to demonstrate: if α satisfies $\Box\Diamond\phi$ then $\Diamond\phi$ must hold in all the categories that terminate α, and if $\Diamond\phi$ ›lds in those categories, then ϕ and $\Box\phi$ hold in them as well. So $\Box\phi$ holds in at least one category in α, and thus must satisfy $\Diamond\Box\phi$. This shows that our logic at least contains **K1** and, as a consequence, is not contained by S5.

However, our logic cannot contain **K2**, since the latter contains **S4.2**. Nor does it contain **K1.2** since the tter's characteristic axiom, namely $\models \phi \rightarrow \Box(\Diamond\phi \rightarrow \phi)$ is shown to be invalid by the category $\langle \text{G}, a\rangle, \langle \text{F}, \{\langle \text{G}, b\rangle, \langle \text{F}, \{\langle \text{G}, a\rangle\}\rangle\}\rangle\}$ (shown in (9), below) when we set $\phi = (\text{G}: a)$.

)



ı fact, our logic does not merely contain **K1**, it also contains **K1.1**, whose characteristic axiom is:

') j. $\models \Box(\Box(\phi \rightarrow \Box\phi) \rightarrow \phi) \rightarrow \phi)$

ughes and Cresswell note that **K1.1** 'is characterized by the class of all finite partial orderings, i.e. finite frames ı which *R* [the accessibility relation] is reflexive, transitive and antisymmetrical' (1984, 162). So it should be no ırprise, given the basis for our semantics, that our logic turns out to include **K1.1**. This logic, also known as 4Grz (after Grzegorczyk 1967), 'is decidable, for every nontheorem of **S4Grz** is invalid in some *finite* weak artial ordering' (Boolos 1979, 167).

Two further valid formula schemata of L_C have some interest, before we conclude the list of valid formulæ ı (7):

7) k. $\models \Diamond\neg f$ (for all $f \in F^1$)

l. $\models (f: \phi) \rightarrow \Diamond\phi$ (for all $f \in F^1$)

he first of these follows from the fact that categories are finite in size and thus ultimately grounded in categories ıat contain no category-valued features: f must be false of these terminating embedded categories, and hence $\Diamond\neg f$ must be true of the category as a whole. The second states that if a category is defined for a category-valued eature whose value satisfies ϕ, then the category as a whole satisfies $\Diamond\phi$.

7) m. $\models (f: \phi) \rightarrow f$ (for all $f \in F^1$)

n. $\models ((f: \phi) \wedge (f: \psi)) \leftrightarrow (f: \phi \wedge \psi)$ (for all $f \in F^1$)

o. $\models ((f: \phi) \vee (f: \psi)) \leftrightarrow (f: \phi \vee \psi)$ (for all $f \in F^1$)

It is worth considering at this point the valid formulæ one would get in certain restricted classes of category tructures. Suppose we consider category structures which contain only atom-valued features (i.e. $F = F^0$). In his case, as one would expect, the modal logic collapses into the propositional calculus and the relevant notion of 'alidity (call it $\models_0$) gives us the following:

(10) $h_0$4><→ □ϕ

The converse case, where we only permit category-valued features (i.e. $F = F^1)_9$ is uninteresting, since it is not distinct from the general case: we can always encode atom-valued features as (sets of) category-valued features and subject the latter to appropriate constraints, as follows. For every feature specification (/, *a)* such that/<E F° and *a* e p(/), we introduce a new type 1 feature *fa* and use the presence of *{fa,* 0) to encode the presence of (/, *a)* and likewise absence to encode absence. Then, for each pair of atoms *a* and *b* in p(/), we require the new features to satisfy •—\(*fa* A/Z>). And to constrain each new feature *fa* to have the empty set as its value, we stipulate •—\(*fa: g)* for every feature *g.*

However, consider validity in category structures containing at most one category-valued feature (call this kind of validity h^). With this restriction, the S4.2 axiom considered earlier becomes valid:

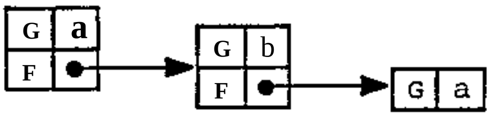
(11) }=!<>•<{>-»•<>({>

In addition, we get (12).

(12) h_1n(D(t)^nv)vn(n\}/->n<)))

This means that this restricted logic at least contains K3, but it cannot contain K4, since h_t $ —» (ODcj) —> D<}>) is falsified by the category « G , *a),* (F {(G, *b)*$_y$ <F, {(G, *a)})})}* when we set <}> = (G: *a).*

(13)



In fact it must also contain K3*l, in view of the validity of (7j) above, and this logic, also known as S4JGrz, is characterized by finite linear orderings (Hughes and Cresswell 1984, 162-163). This is the characterization we would expect given the character of the hj restriction on the form of permissible categories, since with only one categoiy-valued feature, there is at most one path through the structure of a category and so the partial order becomes a linear order. These observations concerning the logic induced by category structures where $\backslash F^1\backslash = 1$ are of some potential relevance to the study of indexed grammars (Aho 1968) whose categories can be construed as being restricted in just this way (see Gazdar et al. 1986).

A word is in order about tthe time complexity of the decision procedure for L_c. We define the checking problem for categories as the problem of determining for an arbitrary category a and a fixed formula <j) of *LQ* whether a satisfies ()). For L_c, it it is easy to show that the checking problem for categories is solvable in linear time (see Gazdar et al. 1986). Of somewhat less interest for practical purposes than the checking problem is the universal checking problem, that of determining for arbitrary inputs <J>a, <}> a formula and a a category, whether a satisfies <}>. Here <}> is not held constant, so its size contributes an additional factor to the complexity of the problem. Nonetheless, the universal checking problem is solvable in at worst quadratic time. For some special cases, both the checking problem and the universal checking problem are of course much easier. For example, if only type 0 features are permitted, checking is decidable in real time by a simple inspection of the finite number of (/, *a)* pairs, regardless of whether <}> is part of the input or not.

The much harder satisfiability problem, that of determining for an arbitrary formula <}> whether there exists a category a that satisfies it, is of even less interest in the present context When a grammatical framework intended for practical use is devised, the constraints on its category system are formulated to delimit a particular set of categories already well understood and exemplified. There is no practical interest in questions about arbitrary formulae of L_c for which no one has ever considered what a satisfying category would be like. We would expect the satisfiability problem for L_c to be PSPACE-complete, like the satisfiability problem for most modal logics.

4. Two example applications

The first of our two example applications is the category system employed in categorial grammar, which originates with work by Lesniewski and Adjukiewicz in the 1940s, and which has attracted renewed interest in the 1980s (see Bach, Oehrle and Wheeler 1986; van Benthem 1986; van Benthem, Buszkowski and Marciszewski

The set of categories used in categorial grammar is infinite. It is often defined as the smallest set containing ome set of basic categories $\{a_1, ..., a_n\}$, and closed under the operation of forming from two categories α and β a ew category $\alpha|\beta$.

To reconstruct the category system for categorial grammar, we define Σ as shown in (14).

(14) a. F = {LABEL, DOMAIN, RANGE}

b. $A = \{a_1, ..., a_n\}$

c. τ = {⟨LABEL, 0⟩, ⟨DOMAIN, 1⟩, ⟨RANGE, 1⟩}

d. ρ = {⟨LABEL, A⟩}

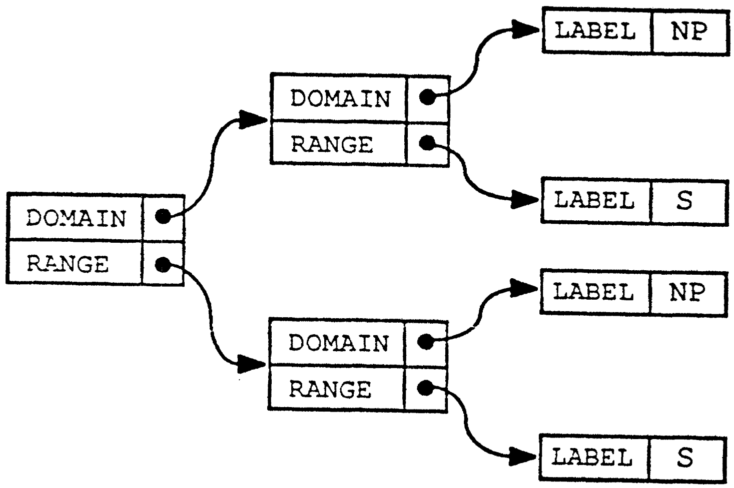
Ve then add the following:

(15) a. □(DOMAIN ↔ ¬ LABEL)

b. □(DOMAIN ↔ RANGE)

ι simple structural induction argument suffices to show that we can obtain a bijection between the categorial rammar categories and the admissible categories induced by F, A, and the constraints defined above (see Gazdar t al. 1986). Here, for example, is how the adverbial category ((S | NP) | (S | NP)) would appear:

(16)



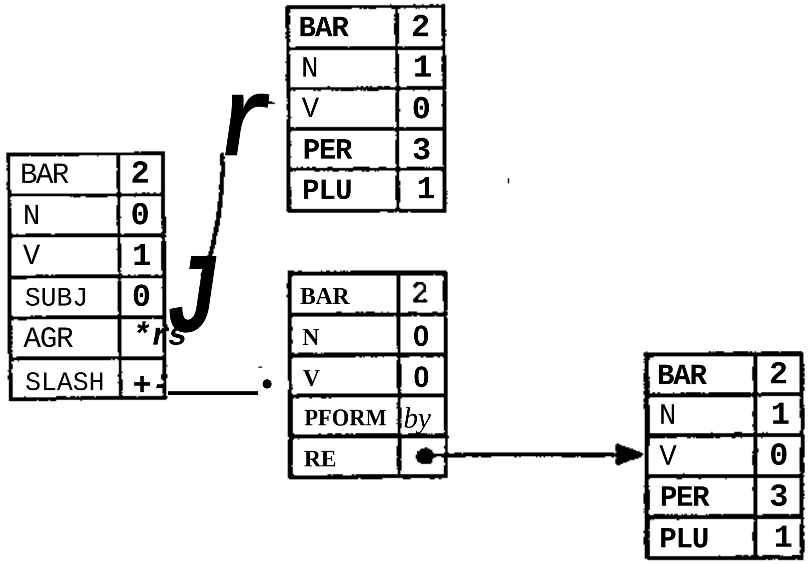
The categories defined thus far are non-directional, in the sense that a complex category can combine with an ırgument either to its left or its right. However, most definitions assume directional categories (cf. Bach 1984). This further specification can be easily incorporated by introducing a new feature name DIRECTION which takes values in 2. We then add a constraint that categories taking values for DOMAIN also take a value for DIRECTION, thus letermining the directionality of the category.

(17) □(DOMAIN ↔ DIRECTION)

Clearly we could employ an analogous move to subsume the α/β vs. $\alpha//\beta$ category distinction employed in Montague (1973).

Turning now to our second sample application, the generalized phrase structure grammar framework (GPSG), as set out in Gazdar, Klein, Pullum, and Sag (1985; henceforth *GKPS*), makes extensive use of features that are permitted to have categories as their values, as the following example indicates:

(18)



We show here how the set of categories for the *GKPS* version of GPSG would be reconstructed in the framework presented here (see *GKPS* pp. 245-6, for the complete lists where we abbreviate with '...').

(19) a. *F* = {SUBJ, N, V, PLU, PFORM, PER, BAR, ..., RE, AGR, SLASH}

b. *A* = {0, 1,2,3,..., fry, to,...}

c. T - {(SUBJ, 0), <N, 0>, (V, 0), (PLU, 0>, (PFORM, 0>, (PER, 0>, (BAR, 0), ..., <RE, 1>, <AGR, 1>, (SLASH, 1»

d. p = {(SUBJ, 2), (N, 2), (v, 2>, (PLU, 2>, (PFORM, *{by, to, ...}*>,..., <PER, {1,2,3}), (BAR, {0,1,2})}

We add to this, for each feature/e F[1], the following constraint:

(20) •-.(/:<>/)

This prevents a category-valued feature / from being specified anywhere within the value of an occurrence of /. This gives us exactly the set of *GKPS* categories, which is, of course, finite in virtue of (20).

Gazdar et al. (1986) give a number of other examples of category systems used by a variety of well-known grammatical frameworks. The unitary form of representation given to the objects used in these diverse systems is potentially of assistance in the exploration and comparison of grammatical formalisms. Questions concerning whether particular rule types and operations on categories that are familiar from one approach to grammar can be carried over unproblematically to another approach, and questions concerning the implementation difficulties that arise when a given formalism is adopted can in many cases be settled in a straightforward and familiar way, namely by reducing them to previously encountered types of question.

References

Aho, Alfred V. 1968. Indexed grammars. *Journal of the Association for Computing Machinery* 15, 647-671.

Bach, Emmon. 1984. Some generalizations of categorial grammar. In Fred Landman and Frank Veltman (eds.) *Varieties of Formal Semantics: Proceedings of the fourth Amsterdam Colloquium, September 1982*, Groningen-Amsterdam Studies in Semantics, 1-23. Dordrecht: Foris Publications.

Bach, Emmon, Richard T. Oehrle and Deirdre W. Wheeler, eds. 1986. *Proceedings of the Conference on Categorial Grammar, Tucson 1985.* Dordrecht: Reidel.

van Benthem, Johan. 1986. Categorial grammar. In Johan van Benthem. 1986. *Essays in Logical Semantics.* Dordrecht: Reidel, 123-150.

van Benthem, Johan, W. Buszkowski, and W. Marciszewski, eds. 1986. *Categorial Grammar.* Amsterdam: John Benjamin.

Boolos, George. 1979. *The Unprovability of Consistency.* Cambridge: Cambridge University Press.

Gazdar, Gerald; Ewan Klein; Geoffrey K. Pullum; and Ivan A. Sag. 1985. *Generalized Phrase Structure*

Grammar. Cambridge, Massachusetts: Harvard University Press.

ǰazdar, Gerald; Geoffrey K. Pullum; Robert Carpenter; Ewan Klein; Thomas Hukari; and Robert D. Levine. 1986. Category structures. Submitted for publication,

ǰrzegorczyk, Andrzej. 1967. Some relational systems and the associated topological spaces. *Fundamentae Mathematicae* **60, 223-231.**

lughes, G. E., and Max J. Cresswell. 1968. *An Introduction to Modal Logic.* London: Methuen.

iughes, G. E., and Max J. Cresswell. 1984. *A Companion to Modal Logic.* London: Methuen.

iasper, Robert T., and William C. Rounds. 1986. A logical semantics for feature structures. *24th Annual Meeting of the association for Computational Linguistics: Proceedings of the Conference,* **257-266.** Morristown, New Jersey: Association for Computational Linguistics.

L^vy, Leon S., and Aravind K. Joshi. 1978. Skeletal structural descriptions. *Information and Control* 39.192-211.

Montague, Richard. 1973. The proper treatment of quantification in ordinary English. In J. Hintikka, **J. Moravcsik, and P. Suppes, eds.** ***Approaches to Natural Language: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics.*** **Dordrecht: Reidel, 221-242. Reprinted in his** ***Formal Philosophy*** (Richmond H. Thomason, ed.), 1974. New Haven: Yale University Press, 247-270.

Moshier, M. Drew, and William C Rounds. 1986. A logic for partially specified data structures. Unpublished paper, Department of Electrical Engineering and Computer Science, University of Michigan, May 5.

Pereira, Fernando C. N., and Stuart M. Shieber. 1984. The semantics of grammar formalisms seen as computer **languages. In** ***Proceedings of the 10th International Conference on Computational Linguistics and the 22nd Annual Meeting of the Association for Computational Linguistics,*** **Stanford University, California, 2-** 6 July, 1984, 123-129.

Rounds, William C, and Robert T. Kasper. 1986. A complete logical calculus for record structures representing **linguistic information.** ***Proceedings of the 15th Annual Symposium on Logic in Computer Science,*** Cambridge, Massachusetts.

;Gazdar)
Cognitive Studies Programme
University of Sussex
^almer
Brighton BN1 9QN
United Kingdom

[Pullum)
Lowell College
University of California, Santa Cruz
Santa Cruz
California 95064
USA