GENERALIZED CONJUNCTION AND TYPE AMBIGUITY

Barbara Partee and Mats Rooth
University of Massachusetts, Amherst

January, 1982

To appear in von Stechow, Arnim et al,, eds.: Meaning, Use, and Interpretation, de Gruyter.
(Papers from the 1981 Konstanz conference)

Generalized Conjunction and Type Ambiguity

Barbara Partee and Mats Rooth

University of Massachusetts, Amherst

I. Conjoinable categories.

It is well-known that in English and in many other languages, virtually every major category can be conjoined with *and* and *or*. The question we address here is whether we can give a single meaning for *and* and a single meaning for *or* that covers their uses across the full range of categories.

In order to make the question more precise and forestall a quick negative answer, we need to distinguish the central use of *and* from a number of special uses. Informally characterized, what we mean by the central use of *and* with conjuncts of any category is related to the sentential conjunction *and* with the meaning of ordinary logical conjunction (although because of interaction with other logical elements in the sentence, there may not always exist a natural paraphrase in terms of conjoined English sentences.) Examples of the central use of *and* with various categories are given in (1); some special uses which we will not treat are given in (2). (There do not appear to be any special uses of *or* which need to be excluded; if there are, we are not treating them.)

(1) (a) John and Mary are in Chicago.

(b) Bacon and eggs are (both) high in cholesterol.

(c) Susan will retire and buy a farm.

(d) She was wearing a new and expensive dress.

(2) (a) John and Mary are a happy couple.[1]

(b) Bacon and eggs is my favorite breakfast.

(c) Susan will try and sell her house.

(d) She was wearing a blue and white dress.

We assume a framework in which all kinds of conjoined constituents are directly generated syntactically, not derived via conjunction-reduction from conjoined sentences. In order to be able to say that it is nevertheless the same *and* and the same *or* that appear in all the constituent conjunction rules, we want to be able to give a single meaning for (normal) *and*, and a single meaning for *or*.

We begin by reviewing the type theory of Montague (1973) (PTQ), and his treatment of phrasal conjunction. Then in section III we present a generalized conjunction schema that is a natural generalization of that presented in Gazdar (1980). In section IV we show some repercussions that the conjunction schema has on Montague's type theory and propose changes (in particular, that extensional verbs be assigned simpler types than intensional ones, plus predictable homonyms of higher type.) In section V we compare this approach with that of Keenan and Faltz (1978), and in the final section we discuss some difficulties and summarize our conclusions

II. Montague's type theory.

We assume familiarity with PTQ, but review the type theory here for future reference. The basic types are *e* and *t*, the types of entities and truth-values (sentence extensions) respectively. There are two recursive rules:

(i) If a,b are types, then $\langle a,b \rangle$ is a type (the type of functions from a-type things to b-type things)

(ii) If a is a type, then $\langle s,a \rangle$ is a type (the type of functions from world-time pairs to a-type things)

In the rest of the paper, we will simplify the discussion by omitting all the s's from types that should contain them. This is purely for expository purposes and does not affect the substance of the discussion or the proposals; wherever it could make a difference, we provide the "correct" treatment in a footnote; intensional types are also discussed in Appendix A.

Omitting s's, then, PTQ has the following correspondence between syntactic categories and semantic types (not a complete list):

(1) t (sentence) – t

e (empty) – e

CN, IV – <e,t>

T (term, or NP) – <<e,t>, t>

TV – <type(T), type(IV)>

In particular, intransitive verbs (and verb phrases) are treated as denoting sets of entities, term phrases as denoting higher-order sets, and transitive verbs as denoting functions from term-phrase interpretations to sets of entities.[2]

Montague introduces and and or syncategorematically for three categories in PTQ: t, IV, and T.[3] The rules produce pieces of derivation trees of the following forms:

(2) (a) φ_1 and φ_2, t — φ_1, t φ_2, t

(b) δ_1 and δ_2, IV — δ_1, IV δ_2, IV

(c) α_1 and α_2, T — α_1, T α_2, T

His translation rules provide the following translations for the three cases (where φ_1' is the translation of φ_1, etc.)

(3) (a) $\varphi_1' \wedge \varphi_2'$

(b) $\lambda x[\delta_1'(x) \wedge \delta_2'(x)]$, x a variable of type e

(c) $\lambda P[\alpha_1'(P) \wedge \alpha_2'(P)]$, P a variable of type <e,t>

By linguists' usual standards, there is clearly a generalization being missed here. It is no accident that the same words and and or are

introduced in each rule, and the translations for IV and T conjunction are clearly predictable from the translation of corresponding sentential conjunction plus the type assigned to the category: provide each conjunct with the type of variables they need as arguments to make a sentence (identical variables for each conjunct), conjoin the resulting sentences, then lambda-abstract on those variables to get back to the original phrasal type. We could clearly do the same thing for higher types such as TV, just by adding on more variable arguments and then abstracting on them in the corresponding order. We express this more precisely in the next section.

III. Generalized conjunction.

Gazdar (1980), von Stechow (1974), and Keenan and Faltz (1978) (in their 'Lifting Theorem') describe how conjunction in $D_{<a,b>}$ (the possible denotations for a phrase with type $\langle a,b \rangle$) can be defined in terms of conjunction in D_b. This allows the recursive extension of operations defined in D_t, the truth values, to D_a, where a is any conjoinable type.

(4) Definition: Conjoinable Type

(i) t is a conjoinable type

(ii) if b is a conjoinable type, then for all a, $\langle a,b \rangle$ is a conjoinable type.

The basic idea is this. Elements f,g of $D_{<a,b>}$, which are functions from D_a to D_b, are viewed as sequences of elements of the set D_b indexed by the set D_a. f and g are combined by performing an operation defined in D_b index by index, in the manner that vectors are added.[4] (5) defines the operators $\sqcap$ and $\sqcup$ ('meet' and 'join') corresponding to and and or. The symbols $\wedge$ and $\vee$ are reserved for $D_t = \{0,1\}$; they are defined by their finite truth-tables.[5]

(5) Pointwise definition of $\sqcap$ and $\sqcup$

$X \sqcap Y$ = $X \wedge Y$ if X and Y are truth values

= $\{\langle z, x \sqcap y\rangle : \langle z,x\rangle \in X$ and $\langle z,y\rangle \in Y\}$ if X and Y are functions (which are represented as sets of ordered pairs)

$X \sqcup Y$ = $X \vee Y$ if X and Y are truth values

= $\{\langle z, x \sqcup y\rangle : \langle z,x\rangle \in X$ and $\langle z,y\rangle \in Y\}$ if X and Y are functions.

The definition (5) is given in terms of functions, which are objects in the model. It is convenient to have some rules for computing with meet and join in intensional logic (IL).

(6) <u>Facts</u>: Where ϕ and ψ are a (single) functional type, and z is a variable of appropriate type not occuring free in ϕ or ψ.

a. $\phi \sqcap \psi = \lambda z[\phi(z) \sqcap \psi(z)]$

$\phi \sqcup \psi = \lambda z[\phi(z) \sqcup \psi(z)]$

This is an immediate consequence of (5).

b. $[\phi \sqcap \psi](\alpha) = \phi(\alpha) \sqcap \psi(\alpha)$; $[\phi \sqcup \psi](\alpha) = \phi(\alpha) \sqcup \psi(\alpha)$

<u>Verification</u>: $[\phi \sqcap \psi](\alpha) = [\lambda z[\phi(z) \sqcap \psi(z)]](\alpha)$ (by (6a))

$= \phi(\alpha) \sqcap \psi(\alpha)$ (by lambda conversion)

c. $\lambda v\phi \sqcap \lambda v\psi = \lambda v[\phi \sqcap \psi]$

$\lambda v\phi \sqcup \lambda v\psi = \lambda v[\phi \sqcup \psi]$

<u>Verification</u>: by fact a, $[\lambda v\, \phi] \sqcap [\lambda v\, \psi]$

$= \lambda u[[\lambda v\, \phi](u) \sqcap [\lambda v\, \psi](u)]$ where u is a variable not occuring free in ϕ or ψ

$= \lambda u\, [\phi^{u/v} \sqcap \psi^{u/v}]$ by lambda conversion ($\phi^{u/v}$ is ϕ with u substituted for all free occurences of v)

$= \lambda v[\phi \sqcap \psi]$ by a change of variables

Because Montague's IL substitutes the operators ^ and ˇ for explicit reference to possible worlds, a special statement is needed for them. Corresponding to (6) we have (7).

(7) a. $\phi \sqcap \psi = {}^\wedge[{}^\vee\phi \sqcap {}^\vee\psi]$ if ϕ, ψ have type <s,b> for some b,

$\phi \sqcup \psi = {}^\wedge[{}^\vee\phi \sqcup {}^\vee\psi]$ and ϕ, ψ are modally closed.[6]

b. ${}^\vee[\phi \sqcap \psi] = {}^\vee\phi \sqcap {}^\vee\psi$

${}^\vee[\phi \sqcup \psi] = {}^\vee\phi \sqcup {}^\vee\psi$

c. ${}^\wedge\phi \sqcap {}^\wedge\psi = {}^\wedge[\phi \sqcap \psi]$

${}^\wedge\phi \sqcup {}^\wedge\psi = {}^\wedge[\phi \sqcup \psi]$

These identities allow us to relate the definition (5) to Montague's PTQ definitions which were discussed in section II. Take for instance his T13, governing terms:

(8) If α, β P_T translate into α', β' respectively, then α <u>or</u> β translates into $\lambda P(\alpha'(P) \vee \beta'(P))$.

This follows from the identities: $\alpha' \sqcup \beta' = \lambda P(\alpha'(P) \sqcup \beta'(P))$ (by (6a)), which is $\lambda P(\alpha'(P) \vee \beta'(P))$ since $\alpha'(P)$ and $\beta'(P)$ have type t.

IV. Repercussions on the type theory.

In this section we show how the generalized conjunction schema makes some correct predictions and some wrong ones, and propose some alterations in the types assigned to lexical items, alterations that we believe are desirable from the psycholinguistic perspectives of sentence processing and language acquisition as well. Similar suggestions can also be found in Dowty (forthcoming) and Cooper (ms.).

A. Transitive verbs. If as in PTQ, the type of all transitive verbs is <type (T), type (IV) >, then the generalized conjunction schema predicts (by two applications of fact (6a) of section III) that the interpretation of [TVP_1 and TVP_2] should be

(9) $\lambda\mathcal{P}\,\lambda x[TVP'_1(\mathcal{P})(x) \wedge TVP'_2(\mathcal{P})(x)]$

with $\mathcal{P}$ a variable over term-phrase interpretations and x an individual variable. In other words, a sentence of the form (10) should have a paraphrase of the form (11).

(10) John TV_1 and TV_2 T(erm phrase)

(11) John TV_1 T and John TV_2 T

When we conjoin two extensional transitive verbs, this gives the wrong result, as seen in (12) and (13).

(12) John caught and ate a fish.

(13) John hugged and kissed three women.

Unless the sentences are given a very marked intonation or the context is heavily loaded, we must interpret (12) as involving just one fish, and (13) as saying that the same three women were hugged and kissed.[7]

On the other hand, when we conjoin two intensional verbs as in (14) or an intensional and an extensional verb as in (15),[8] the reading predicted by the generalized conjunction schema is indeed the primary

reading of the sentence.

(14) John wants and needs two secretaries.

(15) John needed and bought a new coat.

(A "quantified-in" reading, "there are two secretaries such that...", may also be available in these cases, with about as much ease or difficulty as with a single intensional verb; the mechanism that accounts for quantified-in readings, whether generative or interpretive, would presumably operate in the same way whether the TV was complex or simple.)

Now suppose the type of TV were <e, type(IV)>; then the generalized conjunction schema would predict that [TVP_1 and TVP_2] would be interpreted as (16):

(16) $\lambda y \lambda x[TVP'_1(y)(x) \wedge TVP'_2(y)(x)]$

In that case, sentences of the form (10) above should have (stilted) paraphrases of the form (17):

(17) NP is/are such that John TV_1 it/them and John TV_2 it/them.

In this case the matching of prediction to judgments on sentences (12) - (15) above is exactly reversed: now we get the right result for (12) and (13), and the wrong result for (14) and (15).[9]

Note that the cases where the simpler type assignment makes the right predictions are just those cases where both verbs are subject to Montague's meaning postulate 4, which says that such verbs do indeed have counterparts of type <e,type(IV)>, $catch'_*$, eat'_*, etc.

These results suggest that we should depart from Montague's strategy of assigning to all members of a given syntactic category the "highest" type needed for any of them. Montague had reason to let intensional verbs like seek denote functions taking term-phrase intensions as arguments, and because he wanted a uniform category-type correspondence he assigned the same type to the "simpler" extensional transitive verbs, and then provided meaning postulates to guarantee that the extensional verbs would behave semantically as if they were of the simpler type <e, type(IV)>. Suppose we give up this uniformity and enter each verb lexically in its minimal type, where the minimal type for verbs like seek is Montague's TV-type but the minimal type for the extensional verbs is <e,<e,t>>.

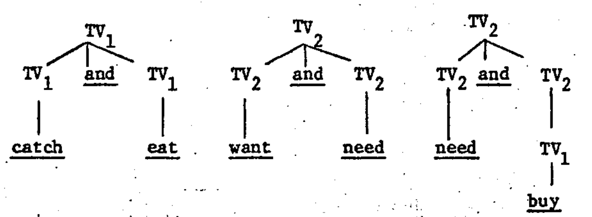
Then in order to account for the conjoinability of intensional and extensional verbs and the fact that the result patterns with the intensional verbs, we add a "redundancy rule" to insure that each "low-type" verb has predictable homonyms of higher type. For example, given $\underline{buy_1}$ of type <e,<e,t>>, we predict the availability of $\underline{buy_2}$ of type <type(T), <e,t>>, where:[10]

(18) $\underline{buy'_2} = \lambda\mathcal{P}\,\lambda x[\mathcal{P}(\lambda \underline{y}[\underline{buy_1}(y)(x)])]$

A fuller specification of these redundancy rules is given in Appendix A.

As a third part of our proposal, we suggest as a processing strategy that all expressions be interpreted at the lowest type possible, and in particular that conjoined expressions be interpreted at the lowest type they both share. Abbreviating the minimal type of eat, buy, etc. as TV_1 and the type of seek, need, etc. as TV_2, we would have the following patterns:[11]

(19)



Note that extensional transitive verbs in their basic forms are now of a type that takes e-type things as argument. Montague assimilated proper names and pronouns to the higher type needed for quantified noun phrases, but it would be consonant with our alternative schema to assign proper nouns and pronouns to type e (with their PTQ type interpretations available as predictable homonyms so that they can conjoin with quantified noun phrases and serve as objects of intensional verbs.) If pronouns are assigned to type e, the quantifying-in rule (which would not have to be changed at all) would permit quantified noun phrases to occur as objects of extensional verbs. Only with intensional verbs would quantified noun phrases be directly generated as direct objects.

In sum, we have a three-part proposal:

(i) Enter each verb lexically in its minimal type.

(ii) Provide lexical rules furnishing "higher"-type homonyms for "lower"-type elements.

(iii) Posit as a processing strategy that all expressions are interpreted at the lowest type possible, invoking higher-type homonyms only when needed for type coherence.

With these proposals, all of the judgments about conjoined verbs in sentences (12) - (15) agree with the predictions of the generalized conjunction schema. As an additional benefit, sentences which are intuitively simpler now have simpler translations involving lower types than sentences which are intuitively more complex.

B. Parallel issues with intransitive verbs. In PTQ and in much of the subsequent literature in the Montague tradition, the type of intransitive verbs, and of verb phrases in general, is <<s,e>,t>. Bennett (1974) proposed that IV (and CN) be assigned the type <e,t>, and

Dowty, Wall, and Peters (1980) follow Bennett's type system in their presentation of PTQ. Since we are ignoring the "s" parts of types in this presentation, we will regard Bennett's IV-type and the PTQ IV-type as equivalent, representing both as <e,t>. In either case, subjects take verb phrases as arguments to form an expression of type t.

The alternative, that verb phrases should be of type <type(T), t> and take the subject as argument, had been employed by Montague in Montague (1970) (UG), and has been argued for by Keenan and Faltz (1978) and Bach (1980a); it is also employed by Thomason (1976)/Hoepelman (1978), Gazdar & Sag (1981), and Bach and Partee (1980). Flynn (1981) considers the possibility that languages might differ from each other with respect to which of the subject and the verb phrase takes the other as argument. Bach (1980a) also discusses the possibility of assigning untensed IV's to the type <e,t> but tensed IV's to the type <type(T),t>.

Let us see what predictions the generalized conjunction schema makes about the types of IV's. Given a sentence of the form (20), if IV

(20) $T\ IV_1 \left\{\begin{matrix}\text{and}\\ \text{or}\end{matrix}\right\} IV_2$

is of type <e,t>, then (regardless of whether the subject is of type e or type <<e,t>,t>) the schema predicts that there will be a paraphrase of the form (21):

(21) T is such that he/it IV_1 $\left\{\begin{matrix}\text{and}\\ \text{or}\end{matrix}\right\}$ he/it IV_2.

If, on the other hand, IV is of type <type(T),t> and takes the subject as argument, the schema predicts that there will be a paraphrase of the form (22):

(22) $T\ IV_1 \left\{\begin{matrix}\text{and}\\ \text{or}\end{matrix}\right\} T\ IV_2$.

Examining sentences (23) - (26), we observe that the lower type gives the right result for (23) and (24), the higher type the right

result for (25) and (26).[12]

(23) A fish walked and talked.

(24) Every participant sent in an abstract or apologized.

(25) An easy model theory textbook is badly needed and will surely be written within this decade.

(26) A tropical storm was expected to form off the coast of Florida and did form there within a few days of the forecast.

This pattern of results is parallel to those with the extensional and intensional transitive verbs; the verb phrases in (25) are both intensional with respect to subject position, and (26) represents a conjunction of an intensional and an extensional verb phrase.[13] So in this case as well, the observed judgments accord with the hypothesis that each item is listed lexically with its minimal type, that there is a rule for generating higher type IV's from lower type ones, and that there is a processing strategy of trying to use the lowest types possible.

Note that in this case we are proposing that there is no fixed directionality to the function-argument structure of subject and predicate. If the verb phrase is a simple one of type <e,t>, it will be the function if the subject is of type e (as we suggested above for proper names and pronouns), but the argument if the subject is a quantified noun phrase. If the verb phrase is of type <<<e,t>,t>,t>, it will always be the function, assuming that we never find reason to posit term phrases of still higher type taking such IV's as argument.

Having argued that there are two IV types, we have to reconsider our TV types; a summary of all the newly proposed types is given in Appendix A.

C. <u>Psycholinguistic advantages</u>. As remarked above, an added advantage of our revision of Montague's uniform category-type correspondence is that the intuitively simpler cases now have simpler interpretations. The potential disadvantage of having multiple interpretations available for extensional verbs (via the lexical rule introducing higher-type homonyms) is offset by the processing strategy of trying the simplest type first. More work needs to be done to explore the workings of this strategy, particularly to see if extending the type-multiplicity to more categories leads to cases in which the minimal lexical types don't cohere but there is more than one alternative available to try next. In any case, the revised system seems like a step in the right direction for making Montague grammar more plausibly connectable to a performance model of language processing.

There is an additional potential advantage from the perspective of language acquisition. If, for example, children learn proper names and pronouns before they learn quantifiers, it would be natural to suppose they would assign them to type e. On the original PTQ model, they would have to revise their interpretation of proper names and pronouns when they learn the higher term-phrase type. Under our revised principles, they would keep the original interpretation as the basic one, and add a higher-type interpretation via a general rule. Similar considerations apply to extensional and intensional verbs.[14] In general under our assumptions more of the child's learning of semantics could take place by accretion, and there would be less need for restructuring. On the assumption that accretion is easier than restructuring, this also seems a welcome result.

V. Keenan and Faltz.

In their 'Logical Types for Natural Language' ('LT'), Keenan and Faltz ('K & F') describe a different way of endowing the types with Boolean structure. In this section, K & F's construction is summarized and related to our analysis. At some points, references to LT are substituted for proofs.

For K & F (as for Montague) term denotations (elements of D_T) are (in extensional models) sets of sets. The Boolean operations in this type are set union, intersection and complementation. (This is equivalent to the pointwise definition, for if X_A and X_B are the characteristic functions of A and B respectively, $X_A \cap X_B = X_{A\cap B}$.) K & F give a special status to the term denotations corresponding to individuals. If b is an element of the domain D, the corresponding term denotation I_b is $\{X \subseteq D \mid b \in X\}$. (These are sometimes called individual sublimations; see Dowty, Wall and Peters (1980).). The set $\{I_b \mid b \in D\}$ of term denotations corresponding to individuals is denoted I_D.

I_D has a useful property: it is a set of free generators (cf. Halmos (1963) p. 40) for the Boolean algebra $\langle D_T, \cap, \cup, {}^c, \emptyset, D\rangle$ (the set $2^{(2^D)}$ with operations intersection, union, complementation, and constants $\emptyset$ (the identity for $\cup$) and D (the identity for $\cap$).)

This means that

(27) a. I_D generates D_T: each element of D_T can be represented as a combination of I_b's via the operations $\cap$, $\cup$ and ^{c}. (See Thm 8 of LT, page 91).

b. Any function g from I_D to a Boolean algebra B can be extended uniquely to a homomorphism h from D_T to B. (This is the Justification Theorem of LT, page 120).

Pictorially, (27b) can be represented as (28), where id is the identity map.

(28)
$$\begin{tikzcd} & B \\ I_b \arrow[ur, "g"] \arrow[r, "\underline{id}"'] & D_T \arrow[u, dashed, "h"'] \end{tikzcd} \qquad g = h \circ id$$

We can now describe the Boolean structure which K & F assign to the space of extensional VP's, which they take to be Hom $(D_T, 2)$, the set of homomorphisms from D_T to the set $2 = \{0,1\}$ of truth values.[15] We do this in the following way. First, a bijection between Hom $(D_T, 2)$ and 2^D is asserted to exist. Then this bijection is used to transfer the Boolean structure which 2^D has as an algebra of sets to Hom $(D_T, 2)$. Suppose f is a characteristic function for a set of individuals, and let I be the function mapping an element of D to the corresponding T denotation:

(29) $f\colon\ D \to 2 \qquad I\colon\ D \to I_D$

$\qquad\qquad\qquad\qquad b \rightsquigarrow I_b$

By fact (27b) $f \circ I^{-1}$ extends uniquely to a homomorphism $h\colon D_T \to 2$. So we can define a map M which maps a characteristic function f to the corresponding homomorphism:

(30) a.
$$\begin{tikzcd} D \arrow[r, "f"] \arrow[d, "I"'] & 2 \\ I_D \arrow[ur, "f \circ I^{-1}"] \arrow[r, "\underline{id}"'] & D_T \arrow[u, dashed, "h"'] \end{tikzcd}$$

b. $M\colon\ 2^D \to \text{Hom}\ (D_T, 2)$

$\qquad f \rightsquigarrow h$ (in the example above)

M is a function, because the extension licensed by (27) is unique. M is an injection (one to one), because if $f_1 \neq f_2$, $f_1 \circ I^{-1} \neq f_2 \circ I^{-1}$ and $M(f_1)$ and $M(f_2)$ extend $f_1 \circ I^{-1}$ and $f_2 \circ I^{-1}$ respectively. M is a surjection (onto) because if k is in $\text{Hom}(D_T, 2)$, $M(I \circ \underline{id} \circ k) = k$, again because the extension is unique. Thus M is a bijection between 2^D and $\text{Hom}(D_T, 2)$.

M can be used to define operations on Hom $(D_T, 2)$:

(31) $h \wedge g \underset{def}{=} M(M^{-1}(h) \wedge M^{-1}(g))$

$\qquad h \vee g \underset{def}{=} M(M^{-1}(h) \vee M^{-1}(g))$

$\qquad h^c \underset{def}{=} M([M^{-1}(h)]^c)$

It is easy to verify that Hom $(D_T, 2)$ with operations so defined is a Boolean algebra. The Boolean identities are verified by appealing to the corresponding identities in the set algebra 2^D. As an instance, take the distributive law:

(32) $h \wedge (g \vee k) = M(M^{-1}(h) \wedge M^{-1}M(M^{-1}(g) \vee M^{-1}(k)))$

$\qquad = M(M^{-1}(h) \wedge (M^{-1}(g) \vee M^{-1}(k)))$

$\qquad = M((M^{-1}(h) \wedge M^{-1}(g)) \vee (M^{-1}(h) \wedge M^{-1}(k)))$

$\qquad\qquad$ using the distributive law in 2^D

$\qquad = M(M^{-1}M(M^{-1}(h) \wedge M^{-1}(g)) \vee M^{-1}M(M^{-1}(h) \wedge M^{-1}(k)))$

$\qquad = (h \wedge g) \vee (h \wedge k)$

This completes our exposition of K & F's construction. Obviously, it is impossible to do justice to LT in a few pages. As partial motivation for the homomorphism construction, note that because an extensional verb like sing is a homomorphism, sing' (a-man' $\vee$ a-woman') = sing'(a-man') $\vee$ sing' (a-woman'), so that 'a man or a woman sings' is equivalent to 'a man sings or a woman sings', as desired.

In section 4, we suggested that the basic type for extensional VP's was <e,t>, with Boolean structure given by the pointwise definition. (As noted above, since elements of this type are characteristic functions, this is the structure of an algebra of sets.) The function M reveals the relation between this approach and K & F's homomorphism construction.

Notice that M is an isomorphism between 2^D and Hom $(D_T, 2)$:

(33) $M(f_1 \wedge f_2) = M(M^{-1}M(f_1) \wedge M^{-1}M(f_2)) = M(f_1) \wedge M(f_2)$

$M(f_1 \vee f_2) = M(M^{-1}M(f_1) \vee M^{-1}M(f_2)) = M(f_1) \vee M(f_2)$

$M(f_1{}^c) = M([M^{-1}M(F_1)]^c) = [M(f_1)]^c$

Since the algebra K & F use in translating extensional IVs is isomorphic to the one we use, we are in agreement here. The advantage of our approach, we would claim, is revealed when we consider conjunction of intensional IVs. As was shown above, the pointwise definition gives the right result here. While they do not include conjunction of intensional IVs (or intensional TVs) in their fragment, K & F agree with us on this point (see footnote 30, page 328 of LT). Hence in an appropriate extension of LT, <u>and</u> and <u>or</u> conjoining IVs are translated in two distinct ways: when they conjoin intensional IVs, they are interpreted via the pointwise definition; when they conjoin extensional IVs, they are interpreted as operations in Hom $(D_T, 2)$.

On the other hand, in section IV it was shown that by translating intensional and extensional IVs by objects of different type, a single cross-categorial definition for $\sqcap$ and $\sqcup$ can be employed. While we consider this an advantage, we do not believe that there is an empirical difference between an extended LT semantics employing features to differentiate the two conjunction modes and the analysis of section IV.

In a review of LT, Ballmer offers the following propositions (propositions (4) and (6) of Ballmer (1980)):

(34) a. "Especially, I disagree with what could be called the function-argument ideology, namely the assumption that the role of a grammatical category of being a function or an argument is invariably fixed."

b. "My most serious criticism is this. K & F base their algebraic semantics of NL [natural language -- BHP and MR] on some mathematical assumptions which are mistaken. They assume wrongly the existence of certain homomorphisms to link the Boolean algebras of certain grammatical categories and arrive therefore at false conclusions and results."

We agree with proposition a. In section IV, we suggested that an extensional IV like <u>sing</u> has type <e,t> and thus combines as an argument with <u>every man</u> (type <<e,t>,t>), while an intensional IV like <u>is believed to have left</u> has the type <<s,<<e,t>,t>>,t> and so combines as a function with <u>every man</u> (which is lifted to type <s,<<e,t>,t>> by one of the rules of Appendix A.)

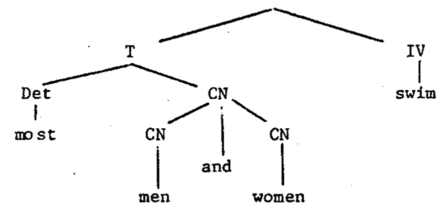
We do not agree with proposition b. Ballmer claims that in the LT system 'every man sings or dances' means the same as 'every man sings or every man dances'. To draw thus undesirable conclusion, he assumes that K & F use the pointwise definition, (Ballmer's (18), our (5)) to assign a Boolean structure to extensional IVs. If this were so, Ballmer's conclusion would indeed follow. But this assumption, and the replacement it sanctions are in error: "...the definition of $\vee$ [which is the homomorphism definition, not the pointwise definition -- BHP and MR] for the VP algebra only sanctions replacing a join of VP functions applied to a member of T_{DNP} by the corresponding join (union) in the formula algebra if that member of T_{DNP} is an I_b. We <u>cannot</u> do this in general for arbitrary members of T_{DNP}." (LT, pages 131-132; K & F's T_{DNP} is our D_T). Since <u>every man</u> is not an I_b, [sing' $\vee$ dance'](every-man') cannot be converted into sing'(every-man') $\vee$ dance'(every-man').

Thus, while we agree with proposition a (as well as with some of Ballmer's other points), we believe that his attempt to criticise LT based on the claim that K & F made 'serious mathematical errors' is misguided; we are aware of no such errors.

VI. Problem cases and conclusions.

Cooper (ms.) discusses a reading for terms containing conjoined CNs (common nouns) not predicted by the generalized conjunction definition (5). The reading predicted for (35a) is (35b), which can be paraphrased 'most hermaphrodites swim' (because man' and woman' denote characteristic functions, $\sqcap$ amounts to set intersection here.)

(35) (a)



(b) $most'(men' \sqcap women')(swim')$

Such readings are possible for CN-conjunction ('my friend and colleague', 'every wife and mother'), but a more natural reading is 'most men swim and most women swim'. Cooper notes that this reading can be obtained by reversing the function-argument order in the construction [Det CN]. (36a) defines new translations for man and woman which map determiner meanings to term meanings

(36) (a) $man'' = \lambda\mathscr{P}\mathscr{P}(man')$ $\quad$ $woman'' = \lambda\mathscr{P}\mathscr{P}(woman')$

(b) $man'' \sqcap woman'' = \lambda\mathscr{P}\mathscr{P}(man') \sqcap \lambda\mathscr{P}\mathscr{P}(woman')$

$= \lambda\mathscr{P}(\mathscr{P}(man') \sqcap \mathscr{P}(woman'))$ by fact (6c)

$[man'' \sqcap woman''](most') = \lambda\mathscr{P}(\mathscr{P}(man') \sqcap \mathscr{P}(woman'))(most')$

$= most'(man') \sqcap most(woman')$

$= \lambda P(most'(man')(P) \sqcap most'(woman')(P))$ by fact (6a)

where P is a variable of type <e,t>

$$[\text{man}'' \sqcap \text{woman}''](\text{most}')(\text{swim}') = \lambda P(\text{most}'(\text{man}')(P) \sqcap \text{most}'(\text{woman}')(P))(\text{swim}')$$
$$= \text{most}'(\text{man}')(\text{swim}') \sqcap \text{most}'(\text{woman}')(\text{swim}')$$

by lambda conversion

As the equalities in (36b) indicate, employing this elevated type for CNs while retaining the pointwise definition for conjunction yields the desired reading.

We will now discuss a similar ambiguity in intensional contexts and examine the possibility of extending Cooper's mechanism to cover it. The ambiguity is illustrated in (37), which has at least three readings.

(37) The department is looking for a phonologist or a phonetician.

In the de re reading, the department is looking for a specific person, and that person is a phonologist or a phonetician. In the normal de dicto reading, the department would be satisfied if they found a phonologist, and they would also be satisfied if they found a phonetician. The de re reading is obtained by quantifying the expression [a-phonologist' $\sqcup$ a-phonetician'] into the expression the-department'($\hat{y}$ look-for'($\hat{Q}$Q(x))(y)), abstracting over the variable x. The normal de dicto reading is obtained by combining look-for' with [a-phonologist' $\sqcup$ a-phonetician'] directly to yield the-department'($\hat{y}$ look-for'($^\wedge$[a-phonologist' $\sqcup$ a-phonetician'])(y)). The types employed here are those discussed in appendix A. Terms such as the department and a phonologist have type <<e,t>,t>>, and look-for' has type <<s,<<e,t>,t>>,<e,t>>. Q is a variable of type <e,t> and x and y are variables of type e.

The reading we are interested in, a second de dicto reading, is suggested by the continuation '... but I don't know which'. In this case, the department has a particular kind of person in mind, but the speaker doesn't know which

kind of person this is. This reading is equivalent to 'the department is looking for a phonologist or looking for a phonetician' with the object de dicto in both conjuncts.

The generalized form (38a) of (36a) allows us to derive this reading. This is indicated by the identities in (38b).

(38) (a) function-argument flip-flop

Let the phrase α have type a, and let b be any type. Then α has a translation α" of type <<a,b>,b> in addition to its translation α' of type a:

$\alpha'' = \lambda F F(\alpha')$, where F is a variable of type <a,b>

(b) (i) a-phonologist" = λFF(a-phonologist'), where F is a variable of type <<s,<<e,t>,t>>,<e,t>> (which is the type of look-for'.)

(ii) a-phonetician" = λFF(a-phonetician')

(iii) a-phonologist" $\sqcup$ a-phonetician"
= λFF(a-phonologist') $\sqcup$ λFF(a-phonetician')
= λF(F(a-phonologist') $\sqcup$ F(a-phonetician')), using fact (6c)

(iv) [a-phonologist" $\sqcup$ a-phonetician"](look-for')
= look-for'(a-phonologist') $\sqcup$ look-for'(a-phonetician') by lambda conversion for F

Note that in (iv), [a-phonologist" $\sqcup$ a-phonetician] is the function and the TV translation look-for' is the argument. As desired, (iv) is also the translation of the phrase "look for a phonologist or look for a phonetician'.

Unfortunately, the flip-flop rule generates many undesired readings also, In section IV, we noted that a sentence with conjoined extensional IVs, such as (39a), is not equivalent to the corresponding conjoined sentence.

(39) (a) Every student failed or got a D.

(b) Every student failed or every student got a D.

We captured this by proposing that extensional IVs primitively have type <e,t> and conjoin at this level, whenever possible. But the derivations (36b) and (38b) are inconsistent with the principle of performing conjunction at the lowest possible level; they essentially involve conjunction at the higher type level generated by the flip-flop rule.

Consideration of other examples reveals a second inadequacy in the flip-flop approach. (40) is ambiguous in the same way as (37).

(40) John believes that a phonologist or a phonetician won.

However, the embedded subject a phonologist or a phonetician is already the function with respect to won, so flip-flop does not apply. While it is possible to pursue this approach further (cf. Lambek (1961)) we will not do so.

An alternative to the flip-flop rule which deals with (40) is a rule quantifying in terms of a higher type. This would yield a representation like (41) (ignoring tense).

(41) $[\lambda\Phi\Phi({}^{\wedge}\text{a-phonologist}') \sqcup \lambda\Phi\Phi({}^{\wedge}\text{a-phonetician}')]$

$(\lambda\mathcal{P}[\text{believe}'({}^{\wedge}[{}^{\vee}\mathcal{P}(\text{win}')])(j)])$ where type($\mathcal{P}$) = <s,<<e,t>,t>
and type(Φ) = <type($\mathcal{P}$),t>

Since $\lambda\Phi\Phi({}^{\wedge}\text{a-phonologist}') \sqcup \lambda\Phi\Phi({}^{\wedge}\text{a-phonetician}')$ $= \lambda\Phi[(\Phi({}^{\wedge}\text{a-phonologist}') \sqcup \Phi({}^{\wedge}\text{a-phonetician}')]$, lambda conversion yields the right results, namely:

$\text{believe}({}^{\wedge}\text{a-phonologist}'(\text{win}'))(j) \sqcup \text{believe}({}^{\wedge}\text{a-phonetician}'(\text{win}'))(j)$

While this solves the second problem, the first problem remains. It can be summarized as follows. While intuitions are far from clear, we believe that (42a) has a reading equivalent to (42c), but no reading equivalent to (42b).

(42) (a) Mary indicated that every student failed or got a D.

(b) Mary indicated that every student failed or every student got a D.

(c) Mary indicated that every student failed or indicated that every student got a D.

In this case, the element which is taking scope is a higher order translation of the IV failed or got a D. Our intuitions can be described by stating that it can take scope over the intensional operator indicate, but not simply over the term every man (cf. (39)).

We can see no non-ad hoc way of capturing this divergence between scope with respect to intensional operators and scope with respect to quantifiers.

The additional readings of sentences (37) and (40) that are not predicted by our generalized conjunction rules correspond to possible meanings of "conjunction-reduction" sources with two full sentences connected by or. As noted earlier, however, syntactic conjunction-reduction is notoriously non-meaning-preserving and would let in even more unwanted readings than either the function-argument flip-flop rule or the higher-type quantifying-in rule; nor do we see any way of restricting its application to allow the meaning-preserving cases.

An added difficulty facing any investigation of these problem cases is the unclarity of the data. As noted by Dowty (forthcoming) and Bach (1980a), intuitions are not always sharp and not always shared in cases where there is a phrasal conjunction and several other scope-bearing elements in the same sentence.

The problems discussed in this section are cases where there seem to be additional readings not predicted by the generalized conjunction schema. We believe that the generalized conjunction schema itself is not called into question by these cases, since it predicts all and only the right readings over a very wide range of clear cases, and predicts only good readings (but not quite all of them) in the problematical **cases just discussed**. It is simple and general, and leads to an argument for revising the association between syntactic categories and semantic types in a way which has the added advantage of making intuitively simpler sentences simpler in their type structure, and plausibly simpler to process and easier for children to acquire. We have given up Montague's requirement that each syntactic category have a single semantic type associated with it, but the set of types associated with each category is still predictable from its specified simplest type together with the "type-lifting" rules given in Appendix A. Similarly, there is no longer a single semantic interpretation rule associated with each syntactic rule, but an instruction such as "do function-argument application", which may require the application of "type-lifting" rules to one or both expressions to find the lowest types that fit the pattern <<u>a,b</u>>,<u>a</u> required for function-argument application to apply. The processing constraint, "use the lowest types possible", is essential to our account; without it, the type-lifting rules allow for additional unwanted readings in many cases. Insofar as the overall account is convincing, it provides an example of the importance of considering processing or "performance" issues in seeking explanatory accounts of the relation between form and meaning.

<u>Appendix A: Redundancy rules for predicting higher-type interpretations.</u>

The second part of our three-part proposal discussed in section IV was to provide redundancy rules for predicting the interpretation of the "higher-type" counterparts of expression whose simplest interpretation is at a lower type. Although we illustrated the process with lexical examples, the rules can apply to arbitrary expressions, and it is probably desirable to allow them to do so. Suppose, for example, we are conjoining an intensional and an extensional IV phrase, so we need to "lift" the extensional IV phrase to its corresponding higher type, and suppose the extensional IV consists of an extensional transitive verb plus an object. The processing will be simpler if we do the TV-object combination at the extensional level to build up the extensional IV interpretation and only then apply the redundancy rule to the IV-interpretation to get the corresponding higher-order IV-interpretation, rather than forcing the 'type-lifting' to be done on the lexical verb (particularly since an IV phrase can contain so many different kinds of lexical verbs, as well as verb-phrase adverbs).[16]

In this appendix we provide the formal rules for the three cases of type-lifing described in the paper: for term phrases, for IV-phrases, and for TV's. Then we offer a tentative characterization of what might be the full range of cases of type-lifting to be expected for natural languages.

1. <u>Term phrases</u>. The simplest term phrases (proper nouns and singular pronouns[17]) are now of type <u>e</u>. Montague's original type for term phrases is <<s,<<s,e>,t>>,t>. In the Bennett type-system employed in Dowty, Wall, and Peters (1980), the type for T phrases is <<s,<e,t>>,t>. Dowty, Wall and Peters suggest (footnote 14, p. 250) that there is no reason other than that of preserving Montague's uniform category-to-type mapping rules not to take the type of term phrases to be <<e,t>,t>; the intensionality

of verbs like <u>seek</u> could be adequately captured by assigning <u>seek</u>' the type $<<s,<<e,t>,t>>,<e,t>>$. Assuming that this is correct, then since we are giving up the uniform category-to-type mapping anyway, we will take the higher term-phrase type to be simply $<<e,t>,t>$.

The type-lifting rule for term phrases can be stated as follows:

If $\alpha \in P_T$ and α translates as $\alpha' \in ME_e$, then α also translates as α'' in $ME_{<<e,t>,t>}$, where $\alpha'' = \lambda P[P(\alpha')]$. ($P$ is a variable of type $<e,t>$.)

2. <u>IV phrases</u>. The type for first-order extensional IV phrases is $<e,t>$. The type for intensional IV phrases like <u>appears to be approaching</u> is $<<s,$(higher) type(T)$>, t>$, which by the line of reasoning followed above is $<<s,<<e,t>,t>>,t>$. If there were no <u>s</u> is this type, we would have an array of types for terms and IV-phrases that forms a sort of "type-ladder" from 0-order predicates (entities) to third-order predicates, as shown in Table 1.

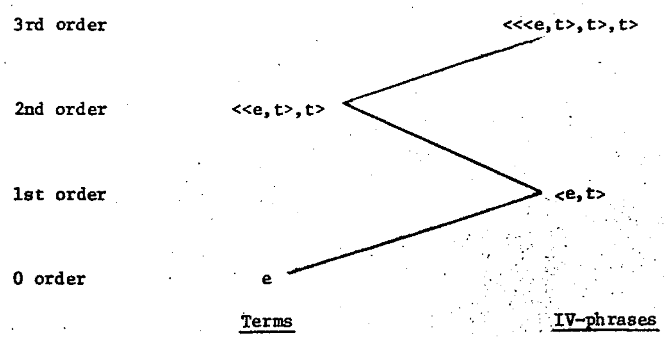


Table I

The lines connecting types in Table 1 show compatible function-argument types; in two of the cases, the IV is the function and the subject term phrase the argument, while in the middle case, the subject term phrase

is the function (as in PTQ).[18] If we wanted to combine a 3rd order IV with a 0 order term, we would have to first apply the term-phrase type-lifting rule to the term phrase, then use function-argument application.

In this purely extensional version, using δ as a variable of type $<<e,t>,t>$, the third-order translation δ'' corresponding to a given first-order IV translation δ' would simply be $\lambda\mathbb{P}[\mathbb{P}(\delta')]$, exactly analogous to the term-phrase lifting rule given above.

To reach the intensional type we actually want, the rule would instead be:

$\delta'' = \lambda\mathcal{P}[{}^{\vee}\mathcal{P}(\delta')]$ ($\mathcal{P}$ is of type $<s,<<e,t>,t>>$.)

3. <u>TV Phrases</u>. Transitive verbs (or TV phrases, such as <u>persuade to leave</u>, <u>consider intelligent</u>; see Bach (1980b)) are of syntactic category IV/T; we now have two semantic types for T and two for IV, so potentially we have at least four types for TV (eight if we allow as an independent variable whether functions take extensions or intensions as their arguments.) By the generalizations we present later in this Appendix, we could predict what all of the readings would be; here for simplicity we just concern ourselves with two: the fully extensional first-order type $<e,<e,t>>$, and a type where the argument is the intension of a second-order term phrase, i.e. $<s,<<e,t>,t>>$, and the result is a simple IV, type $<e,t>$, so the TV type is $<<s,<<e,t>,t>>,<e,t>>$. Then the rule for giving the higher-type TV translation δ'' from the lower-type one δ' is this:

$\delta'' = \lambda\mathcal{P}\lambda u[{}^{\vee}\mathcal{P}(\lambda v[\delta'(v)(u)])]$, where $\mathcal{P}$ is of type $<s,<<e,t>,t>>$, u,v of type e.

It appears to us that the actual array of kinds of type-liftings needed for TV's is limited in that there are no cases where TV takes an intension of a full term-phrase object to give an intensional IV. This reflects the apparent fact that there are no basic lexical verbs that are intensional with respect to both subject and object. We can passivize an intensional TV to get an intensional IV, but the result of passivization is no longer a TV (see Bach (1980b)); other means of producing intensional IV's involve the addition of modal auxiliaries or perhaps adverbs to extensional IV's. There seem to be a few simple intensional IV's such as 'be missing', as in 'one pencil is missing from this (new) box of pencils' but these do not include TV's. The only known candidates are what Postal (1970) called the "Psych-movement" verbs like <u>surprise</u>, <u>worry</u>, <u>intrigue</u>, etc.; whether or not they are really intensional with respect to subject position (and we are inclined to believe they are not), they are certainly extensional with respect to object position. So it appears that the only cases of type-lifting actually needed for TV's are versions of the case discussed in section IV, going from an argument of type e to a full term-phrase argument, with the resultant always an IV of type <e,t> (how many such cases there are then depends on the distribution of <u>s</u>'s in the higher term-phrase type(s). It may be that the rule given above is the only one needed.)

4. <u>Generalizations.</u> All of the rules stated above and more can be gotten by working from four basic "type-lifting" principles which we tentatively hypothesize as a complete set. The first three are perfectly general and follow from the interpretation of the type system; the fourth is a special one for term phrases.

TL1: <u>Extension-to-intension.</u>[19] Given α' of any type <u>a</u>, we can predict an interpretation α'' of type $\langle s,a\rangle$: $\alpha'' = {}^{\wedge}\alpha'$

TL2: <u>Extensional argument to intensional argument</u>. Given α' of type $\langle a,b\rangle$ for any a and b, we can predict α'' of type $\langle\langle s,a\rangle,b\rangle$: $\alpha'' = \lambda x[\alpha'({}^{\vee}x)]$, where x is of type $\langle s,a\rangle$.

TL3: <u>Argument-to-function flip-flop</u>. For any types <u>a</u> and <u>b</u>, given α' of type <u>a</u>, we can predict an interpretation α'' of type $\langle\langle a,b\rangle,b\rangle$: $\alpha'' = \lambda P\,[P(\alpha')]$, where P is of type $\langle a,b\rangle$

TL4: <u>Entity-argument-to-term phrase argument</u>: This rule is somewhat more complex and more closely tied to the type structure instantiated in PTQ than the first three. The rule concerns expressions of syntactic category A/T for any A (e.g. transitive verbs, double object verbs, prepositions), and is designed to lift their translations from type $\langle e, \text{type}(A)\rangle$ to type $\langle\langle\langle e,t\rangle,t\rangle, \text{type}(A)\rangle$. (Rule (ii) can then apply to make the argument intensional.) We rely crucially on the fact that every function category in PTQ, and in most extensions of PTQ, "ends in <u>t</u>"; that is, for every expression of α' of a category other than e, $\langle s,e\rangle$, or t that can arise as the translation of an English expression, there is some sequence of arguments $\beta'_1,\ldots,\beta'_n$, such that $\alpha'(\beta'_1)\ldots(\beta'_n)$ is of type t. For such types, we can state the rule as follows:

(TL4) For any type a that "ends in <u>t</u>", given α' of type $\langle e,a\rangle$, we can predict an interpretation α'' of type $\langle\langle\langle e,t\rangle,t\rangle,a\rangle$:

$$\alpha'' = \lambda\mathbb{P}\lambda v_1\ldots\lambda v_n[\mathbb{P}\,(\lambda u[\alpha'(u)(v_1)\ldots(v_n)])]$$

where $\mathbb{P}$ is of type $\langle\langle e,t\rangle,t\rangle$, u is of type <u>e</u> and $v_1\ldots v_n$ are of types such that $\alpha'(u)(v_1)\ldots(v_n)$ is of type <u>t</u>.

Building on a generalization of 'quantifying-in' rules developed in Rooth (1981), we can define this rule more formally by way of a recursive definition of a 'quantifying-in' schema. Define $Q(\gamma,\alpha,i)$ as follows: where γ is of type $\langle\langle e,t\rangle,t\rangle$, α is any expression, and i is an index:[20]

$$Q(\gamma,\alpha,i) = \begin{cases} \gamma(\lambda x_i\alpha) & \text{if type}(\alpha) = t \\ \lambda v_c\, Q(\gamma, \alpha(v_c), i) & \text{if type}(\alpha) = \langle c,d\rangle \end{cases}$$

Then we can restate the type-lifting rule as follows:

(TL4') $\alpha'' = \lambda \mathbb{P}[Q(\mathbb{P}, \alpha'(x_i), i)]$.

The TV rule given earlier was an instance of this schema; one can view Montague's meaning postulates for extensional first-order prepositions and for transitive verbs as instances of the inverse of the same schema.

One upshot of these rules is that in place of Montague's uniform translation for basic grammatical relations as $\alpha'(^\wedge\beta')$, we probably want to have just the instruction "combine α' and β' by function-argument application." This is to be interpreted in accordance with our processing strategy as an instruction to do the combination at the lowest types possible; if neither α' nor β' can take the other as argument directly, the minimum number of type-lifting rules should be applied to get types compatible for function-argument application.

Appendix B: Extensions based on Link's algebra

Godehard Link (this volume) proposes a Boolean algebra structure for the domain of entities E (D_e in Montague's terms). The set of atoms A of the Boolean algebra correspond to the ordinary "singular" individuals, and joins of these under the join operation $\sqcup_i$ correspond to "plural individuals". (There is a distinguished subset $D \subseteq A$, corresponding to "bits of matter", which has its own semilattice structure with a join operation corresponding to "material fusion"; the manner in which this "mass term" substructure is integrated with the larger Boolean algebra provides the basis for an elegant unified treatment of singulars, plurals, and mass nouns, but we ignore the mass term subsystem here.) If a and b denote elements of E, the "plural individual" denoted by a $\oplus$ b, namely $||a|| \sqcup_i ||b||$ is again an element of E, i.e. still corresponds to type e.

Suppose then that we say that there is a second basic English and which combines expressions of type e to give a new expression of type e, interpreted as Link's $\oplus$ operator. Intuitively, we can think of this and as the "group-reading" and as opposed to the "distributive-reading" and we have been concerned with in the rest of the paper. If we now think of our earlier recursive definition of conjoinable types as a definition of "t-conjoinable" types, we can give a parallel definition of e-conjoinable types:

Definition: (i) e is an e-conjoinable type

(ii) if b is an e-conjoinable type, <a,b> is an e-conjoinable type, for any type a.

However, the only e-conjoinable types that occur in PTQ are e and <s,e>, so although $\oplus$ is in principle just as generalizable as $\wedge$, it does not in fact generalize very far in English.

We can now provide a natural account of the ambiguous sentence (iii):

(iii) John and Mary lifted the piano.

Let us suppose that proper names are basically of type e; and we now have two generalized and's, which we can abbreviate as and_e and and_t. Syntactically, the conjoined term phrase has two analyses, (iv) and (v)

(iv) John and_e Mary, T
John, T and_e Mary, T

(v) John and_t Mary, T
John, T and_t Mary, T

Semantically, we first interpret John and Mary as being of type e, say j and m; in (iv) they are of the right types to combine immediately with and_e to give $j \oplus m$ as the translation of the result, giving the "group" reading, in which the lifting of the piano is predicated of the "plural individual" consisting of John and Mary. In (v), since e is not a t-conjoinable type, we cannot combine j and m directly with and_t; but we can lift j and m to the corresponding second-order term phrase interpretations $\lambda P[P(j)]$ and $\lambda P[P(m)]$ of type $\langle\langle e,t\rangle,t\rangle$, which is a t-conjoinable type. This gives the distributive reading (vi), which is logically equivalent to (vii).

(vi) $\lambda Q[\lambda P[P(j)](Q) \wedge \lambda P[P(m)](Q)]$ (lift-the-piano')

(vii) lift-the-piano' (j) $\wedge$ lift-the-piano'(m)

Note that we do not want the processing strategy of "trying the lowest possible types first" to prevent the second derivation, since the sentence is genuinely ambiguous. But it will not prevent it if we make the assumption that whenever we have homonyms with genuinely different meanings, as in the case of and_e and and_t, we can try either one; and for and_t, the full term-phrase type is the lowest type at which the meanings can combine.

Acknowledgements.

We have drawn heavily on the work of Gazdar (1980) and Keenan and Faltz (1978), and have been influenced as well by discussions with Robin Cooper. A number of the ideas developed here were independently arrived at by Dowty (forthcoming). We are grateful for discussions with colleagues here, in Konstanz, and in Prague, particularly Emmon Bach, Wynn Chao, Lyn Frazier, Roger Higgins, Eva Hajičova, Godehard Link, Petr Sgall, Steven Weisler. The first author is grateful to the Linguistics Department for a partial teaching release this year and to the Graduate School for a Faculty Fellowship; the second author was supported in part by NSF grant BNS 80-14326 and in part by a University Fellowship.

Footnotes.

[1]We will have something to say about the "group-reading" and with noun phrases in Appendix B.

[2]The correct types, with s's, are:

$t - t$

$e - e$

$CN, IV - \langle\langle s,e\rangle,t\rangle$

$T - \langle\langle s,\langle\langle s,e\rangle,t\rangle\rangle,t\rangle$

$TV - \langle\langle s,\mathrm{type}(T)\rangle, \mathrm{type}(IV)\rangle$

[3]Actually he introduces only or for category T in order to avoid having to treat plurals. We ignore that and include the and rule for T as well (with the "distributive", not the "group" reading.)

[4]Somewhat more generally, if $\{B_i\}_{i \in A}$ is a family of Boolean algebras, than its product is the Cartesian product $\prod_{i \in A} B_i$ with operations ($\wedge$, $\vee$ and $'$) defined coordinate by coordinate. One way of defining the Cartesian product of $\{B_i\}_{i \in A}$ is as the set of all functions from A to $\bigcup_{i \in B} B_i$ such that for each i, $f(i) \in B_i$. In the present case, all the B_i's are simply D_b, and $\prod_{i \in D_A} D_b$ is simply $D_B{}^{D_A}$. For a more substantive presentation, see for instance Halmos (1963).

[5]Gazdar gets the recursion started in a slightly different way. He represents the truth values as sets, $0 = \{\ \}$ and $1 = \{\{\ \}\}$ and notes that the operations $\wedge$ and $\vee$ coincide with set intersection and set union, respectively. The denotations of type $\langle a,t\rangle$ are also considered sets (rather than characteristic functions); the denotations of type $\langle a,b\rangle$ with $b \neq t$ are functions as usual. He then defines

$x \sqcup y = x \cup y$ if x and y are sets

$= \{\langle z, x \sqcup y\rangle : \langle z,x\rangle \in x$ and $\langle z,y\rangle \in y\}$ if x and y are functions.

Similarly for $\sqcap$. This way of formulating the definition obscures the fact

that $D_{\langle a,t\rangle}$ inherits its operations from D_t in the same way that $D_{\langle a,b\rangle}$ inherits its operations from D_b, where $b \neq t$.

[6]The restriction requiring that ϕ and ψ be modally closed (i.e. that they not vary in their denotation from world to world) corresponds to the requirement in (6a) that ϕ and ψ contain no free occurrences of z. For in a sense, an expression which is not modally closed contains a free variable over worlds.

[7]We have been assured by native speakers that the same judgements in these and the following cases hold good for at least German, Czech, Hungarian, Portuguese, and Hebrew as well as English. In languages like Japanese with free "Pronoun-drop", the sentences are more likely to be judged ambiguous, but we would be very surprised to find a language in which the "wrong result" - "right result" judgements were the reverse of those for English.

[8]We owe the observation that the conjunction of an intensional and an extensional verb fits the predictions of the schema to Wynn Chao.

[9]The reading predicted by (8) for (6) and (7) may be an available reading, but as mentioned above, we should presumably get that reading (if at all) by quantifying in (or an equivalent interpretive procedure). The new type assignment would incorrectly predict the <u>un</u>availability of what is surely the more natural reading for these examples.

[10]With the full intensional type system, <u>buy</u> would still be entered lexically as type $\langle e,\langle e,t\rangle\rangle$. If we want $\underline{buy}'_2$ to be of Montague's TV type, $\langle\langle s,type(\mathscr{P})\rangle,\langle\langle s,e\rangle,t\rangle\rangle$, then

$$\underline{buy}'_2 = \lambda\mathscr{P}\lambda x[\mathscr{P}\{\lambda \underline{y[buy_1}({}^{\vee}y)({}^{\vee}x)]\}].$$

Note that this corresponds exactly to the relation between <u>eat</u>' and $\underline{eat}'_*$ in PTQ, but we are reversing which one is to be taken as basic. (The variables in this footnote correspond to Montague's notational conventions; those in the text are of corresponding s-less types.)

[11]In conjoining an intensional with an extensional verb, it seems much more natural to have the intensional one first, and very difficulty to interpret a conjunction with the extensional verb first. Lyn Frazier has suggested that this fact might have a natural processing explanation in our system: if the intensional verb is encountered first, the hearer knows immediately that the conjunction must be intensional and can apply type-lifting rules to the extensional verb immediately; the other order would produce a

temporary garden path and require backtracking. Petr Sgall has suggested that extensional-first is possible with sufficient prior context. His example, slightly modified, is this: "I often got gifts from my husband, but seldom did I get anything useful. I remember piles of handkerchiefs that I got and didn't need. But I got and did need a new coat at the beginning of winter."

The difficulty of constructing such examples may also be partly a reflection of the fact that in most cases the intensional verb denotes an attitude toward 'any old such-and-such' (such as wanting or needing) which generally is understood to precede the action denoted by the extensional verb, and there is a strong preference for preserving temporal order in the order of conjuncts.

[12]Again the data appears to be the same for the other languages mentioned in footnote 7 , and again we must add some caveats about the data. Where an intensional, or subject-as-argument, reading is predicted (paraphrase (22) above), an extensional or subject-as-function reading will generally be available as well via quantifying in. Where only an extensional reading is predicted (paraphrase (21) above), the other reading is predicted to be impossible,but sometimes the other reading seems possible if the intonation is highly marked and/or the context strongly supports that reading. For the examples given, assuming normal intonation and no special context, the judgements seem quite clear. For discussion of conflicting judgements about some other examples, see Bach (1980a), footnote 2.

[13]The move away from $\langle e,t \rangle$ as the type for IV's to a type which takes the intension of the subject term phrase as argument has been motivated in part by a move toward generating surface structure directly without using transformations such as Passive and Subject Raising. Montague noted the apparent intensionality of subject position in sentences such as (i), but assumed that such

(i) A unicorn appears to be approaching.

sentences would be accounted for indirectly. It is still an open question whether there are any basic verbs which are intensional with respect to subject position in simple active sentences; our examples of intensional IV's are built up by forming passives of intensional TV's (need, expect) or by adding modal operators (*will*, *surely*) to extensional IV's.

[14]Some intensional verbs like *want* appear very early, before quantifier phrases are mastered. We believe (without detailed investigation) that the earliest uses of *want* as an intensional transitive verb has just

indefinite noun phrases or bare common nouns as object ("want a cookie", "want cookie"), suggesting that at least at the outset, it may be that the object of intensional verbs is just a property-denoting expression. Whether this survives into the adult system is an interesting open question; if it does, it might help to explain why *every*-phrases seem to be harder than *a*-phrases to interpret intensionally when they occur as objects of verbs like *look for* and *want*.

[15]A homomorphism h between two Boolean algebras B_1 and B_2 is a function 'consistent with' the operations $\wedge$, $\vee$ and c and the constants 0 and 1; it is required to satisfy the following identities:

(for all $x, y \in B_1$):

$$h(x \wedge y) = h(x) \wedge h(y)$$
$$h(x \vee y) = h(x) \vee h(y)$$
$$h(x^c) = h(x)^c$$
$$h(0) = 0$$
$$h(1) = 1$$

The operations (and constants) on the left are in B_1, those on the right in B_2.

[16]The fact that type-lifting rules can be applied as easily to phrases as to lexical items may be an added advantage for these rules over Montague's meaning postulates. In the case of lexical items, our rules are in effect inverses of the meaning postulates that "lower" the types of various words. But meaning postulates, because they are constraints on possible interpretations, cannot refer to arbitrary expressions of a given semantic type, and it seems to be an open question

whether they can refer to 'any expression (of IL) which is a translation of an expression (of English) of syntactic category A'. A serious discussion of the proper characterization of meaning postulates is well beyond the scope of this paper, however,

[17] If we follow Montague's treatment of <u>price</u> and <u>temperature</u>, we would also want singular pronouns of type <s,e>; this could be accomodated in the generalizations we propose toward the end of this appendix.

[19] We have not exploited this rule directly in any of our examples. One natural use of it would be to eliminate Montague's uniform assignment of intensions in translation rules of functional application, and allow functor categories B/A to be interpreted either as of type <a,b> or of type <<s,a>,b>; then if a functor <<s,a>,b> is trying to combine with an argument of type <u>a</u>, this rule will lift the argument to type <s,a>. If on the other hand we have a functor of type <a,b> and an argument of type <s,a>, the next rule below will lift the type of the functor to <<s,a>,b>.

[20] The intensional version of this rule is

$$Q(\gamma,\alpha,i) = \begin{cases} \gamma({}^{\wedge}\lambda x_i \alpha) & \text{if type}(\alpha) = t \\ \lambda v_c\, Q(\gamma,\alpha(v_c),i) & \text{if type}(\alpha) = \langle c,d\rangle,\ c \neq s \\ Q(\gamma,{}^{\vee}\alpha,i) & \text{if type}(\alpha) = \langle s,d\rangle \end{cases}$$

where γ has type <<s,<e,t>>,t>

<u>new footnote 18</u>: If it turns out to be preferable to require that the IV-phrase always takes the subject as argument, we could disallow the middle case; then for a first-order IV to combine with a 2nd- order term, the IV would first have to be lifted to the 3rd-order type.

References

Bach, Emmon (1980a) Tenses and aspects as functions on verb phrases. In Christian Rohrer, ed., Time, Tense, and Quantifiers, Tuebingen; Niemeyer , 19-37.

Bach, Emmon (1980b) In defense of passive. Linguistics and Philosophy 3, 297-341.

Bach, Emmon and Barbara H. Partee (1980) Anaphora and semantic structure. In J. Kreiman and A. Ojeda, eds., Papers from the Parasession on Pronouns and Anaphora, Chicago Linguistics Society, Chicago, 1-28.

Ballmer, Thomas T. (1980) Is Keen an' Faltz Keen or false", Theoretical Linguistics 7, 155-170.

Bennett, Michael (1974) Some Extensions of a Montague Fragment of English, UCLA Ph.D. dissertation.

Cooper, Robin (ms. 1979) Model theory for a fragment of English syntax, draft chapter of a projected book, Madison, Wisconsin.

Dowty, David (forthcoming) Quantification and the lexicon: a reply to Fodor and Fodor, to appear in T. Hoekstra, H. van der Hulst, and M. Moortgat, eds., The Scope of Lexical Rules, Foris Publications, Dordrecht.

Dowty, D., R. Wall, and S. Peters (1980) Introduction to Montague Semantics, Synthese Language Library, Reidel, Dordrecht.

Flynn, Michael (1981) A Categorial Theory of the Base, unpublished Ph.D. dissertation, University of Massachusetts, Amherst.

Gazdar, Gerald (1980), A cross-categorial semantics for coordination, Linguistics and Philosophy 3, 407-409.

Gazdar, Gerald and Ivan Sag (1981) Passive and reflexives in phrase structure grammar, in J. Groenendijk, T. Janssen, and M. Stokhof, eds., Formal Methods in the Study of Language (Part I), Mathematisch Centrum, Amsterdam, 131-152.

Halmos, Paul (1963) Lectures on Boolean Algebras, Van Nostrand Reinhold Co., New York.

Hoepelman, J. (1978) The treatment of activity verbs in a Montague-type grammar, a first approximation, in F. Guenthner and C. Rohrer, eds., Studies in Formal Semantics, North-Holland, Amsterdam.

Keenan, Edward L. and Leonard M. Faltz (1978) Logical Types for Natural Language, UCLA Occasional Papers in Linguistics No. 3, Los Angeles.

Lambek, Joachim (1961) On the calculus of syntactic types, in Structure of Language and its Mathematical Aspects, Roman Jakobson, ed., Proceedings of Symposium in Applied Mathematics Vol. XII, American Mathematical Society.

Link, Godehard (this volume) The logical analysis of plurals and mass terms: a lattice-theoretical approach.

Montague, Richard (1970) Universal grammar, Theoria 36, 373-98. Reprinted in Montague (1974).

Montague, Richard (1973) The proper treatment of quantification in ordinary English. Reprinted in Montague (1974), 247-270.

Montague, Richard (1974) Formal Philosophy: Selected Papers of Richard Montague, edited and with an introduction by Richmond Thomason, Yale University Press, New Haven.

Parsons, Terence (1979) Type theory and ordinary language, in S. Davis and M. Mithun, eds., Linguistics, Philosophy, and Montague Grammar, University of Texas Press, Austin, 127-151.

Postal, Paul (1970) On the surface verb 'remind', Linguistic Inquiry 1.1, 37-120.

Rooth, Mats (1981) A comparison of three theories of verb phrase ellipsis, in University of Massachusetts Occasional Papers in Linguistics, Vol. VII, Wynn Chao and Deirdre Wheeler (eds.).

von Stechow, Arnim (1974) "ε-λ kontextfreie Sprachen: Ein Beitrag zu einer natürlichen formalen Semantik", Linguistische Berichte 34, (1-33).