The Frege–Map is not Function Application

Marcus Kracht *
II. Institute of Mathematics
Arnimallee 3
D-14195 Berlin
Germany

February 11, 2008

Abstract

The Frege-Function is the map that constructs the meaning of a complex expression out of that of its immediate parts. With MONTAGUE most linguists believe the Frege-Function is simply function appplication. However, given some basic assumptions about phrase structure and the configurational possibilities of natural languages we will show that this assumption is untenable, precisely because the λ-abstract prejudicates the structure of the projected phrase. We propose an alternative semantics based on referent systems of [9] that eliminates this problem.

Categorial Grammar standardly assumes its justification in the dictum by FREGE which in its modern version says that the meaning of an expression is the result of applying a (uniform) function to the meaning of its immediate parts. Let us call this function the *Frege-Function*. MONTAGUE has put forward the proposal that the Frege–function be none other than function application. A simple predicate is then nothing but a function from individuals to truth–values, and a quantified noun–phrase a function from predicates into sentences. Verbs with more than one argument pose a slight problem for the theory inasfar that one has to get the association right between syntactic arguments and the variables of the expression.

*This work was carried out as part of the *Innovationskolleg 'Formale Modelle Kognitiver Komplexität'* at the University of Potsdam, funded by the Deutsche Forschungsgemeinschaft.

In this paper we will focus on precisely this problem. It will help to illustrate what we believe to be a fundamental error in the design of modern semantics, namely that the assumption that the Frege–function is nothing but function application.

The argument works only on certain assumptions about the relationship between syntax and semantics, which are not uncontroversial. Nevertheless, avoiding them would in part mean that one considers the way meanings get assigned to sentences an *ignoramus*. We will motivate our claims by syntactic evidence. First, we believe strongly that the semantics of an expression should not prejudicate its syntactic behaviour. This is a non–obvious assumption. MONTAGUE has argued conversely, namely that the syntactic behaviour of linguistic expressions is a reflex of their semantics. For a long time this has been the philosophy behind Categorial Grammar but has been tacitly abandoned in face of overwhelming evidence to the contrary. Below we will outline some evidence. Nevertheless, there is sense to the claim that syntax follows semantics, namely because we expect that we should not be able to combine syntactically what is not combinable semantically. We will therefore advance MONTAGUE's dictum in the following weak form.

> AUTONOMY OF SEMANTICS. *A syntactic item does not receive different semantic analyses if it has several syntactic patterns, unless they result in a (traceable) semantic difference in the meaning of that item.*

In this form we will adhere to the uniqueness of semantic form, but will give up to determine the syntactic behaviour of a lexical item from its semantics – at least in the way it has been done before. We will be precise below. Notice that the strategy of forced type raising (referred to as *raising to the worst case* in [7]) has been introduced by MONTAGUE precisely in order to save the uniqueness of the semantic analysis. If it so turns out that some dps (= determiner phrases) [1] cannot be analyzed as arguments of the verb (such as quantified expressions) then none of them should be arguments, but they should all be functions instead. The second assumption we want to make is that there is a unique *Frege–map*, which is a homomorphism from the syntactic algebra into the meaning algebra. Additionally, there is another map, a homomorphism from the syntactic algebra into the algebra of strings, determining how syntactic composition is *spelled out*. (See [7] for the architecture of Categorial Grammar.) Our assumption is again

[1] We prefer to speak of determiner phrases to refer to what used to be noun phrases. This, apart from being an established practice in GB, has the advantage that we get the interpretation of quantified noun phrases as functors for free without type raising, assuming that a quantifier (in fact any determiner) is a generalized quantifier.

that there is a unique such map, and that it is string–concatenation. It may be that syntax does not define the relative ordering explicitly ([6] and others) and leaves that to PF. That is a matter of close investigation, which we put aside here.

Let us now proceed to the facts. We are interested in a very simple construction, namely verbs taking a number of arguments. We will not deal with tense, or mood etc. In that case a transitive verb like *eat* assumes a translation which is usually written $\mathbf{eat}'(x, y)$, where $\mathbf{eat}'$ is a two-place predicate and x and y are variables. In λ-calculus we also write $\lambda y.\lambda x.\mathbf{eat}'(x, y)$. Notice that $\mathbf{eat}'$ is already a function, so the technical apparatus of λ-calculus is not needed at this point. Also, the choice of this translation rather than $\lambda x.\lambda y.\mathbf{eat}'(x, y)$ is technically motivated by the fact that the verb forms a constituent with the object, not with the subject (at least in English), but that has no explanation inside Categorial Grammar. That is to say, it actually does not conform to the philosophical principles of MONTAGUE; there is no formal distinction between the two meanings other than the syntactic behaviour that is being reflected in them. We will see that this is a great problem. However, notice that choosing λ-calculus has the advantage that the choice of the variables becomes immaterial, and the variable handling is taken care of at least partly by the machine of λ-calculus. Assuming all that we get that transitive verbs are seeking two arguments, whose denotation must be an object, i. e. which can fill the places of x and y. That much must be assumed by any theory. Following MONTAGUE we will then assume that the syntactic type of the verb is for example $(\mathsf{dp}\backslash\mathsf{s})/\mathsf{dp}$. We can work here with directional slashes $\backslash$ and $/$, or with the undirectional slash, written $\multimap$ in analogy with linear logic. Either way, it is known that this analysis cannot work for quantified noun–phrases (alias dps), and that the latter must be analysed as functions rather than arguments. Let us assume then, by the principle of semantic uniqueness discussed above, that *all* dps are functions. Then the distinction between object and subject is in the type of argument they seek, whether it is a transitive verb or an intransitive verb. Let us agree on the following abbreviations. Start with the semantic types s, sentence, and e, object.

v_0	:=	s	sentence
v_1	:=	$\mathsf{e} \multimap \mathsf{v}_0$	intransitive verb
v_2	:=	$\mathsf{e} \multimap \mathsf{v}_1$	transitive verb
v_3	:=	$\mathsf{e} \multimap \mathsf{v}_2$	ditransitive verb
su	:=	$\mathsf{v}_1 \multimap \mathsf{v}_0$	subject
do	:=	$\mathsf{v}_2 \multimap \mathsf{v}_1$	direct object
io	:=	$\mathsf{v}_3 \multimap \mathsf{v}_2$	indirect object

These are more or less standard. Here now is the first dilemma. If composition is conditional on adjacency (that is, if we assume the phonological map to be string concatenation), this allows only the following basic word orders and constituents in the AJDUKIEWICZ-BAR HILLEL system.

$$\begin{array}{c} \mathsf{su}\ (\mathsf{v}_2\ \mathsf{do}) \\ \mathsf{su}\ (\mathsf{do}\ \mathsf{v}_2) \\ (\mathsf{v}_2\ \mathsf{do})\ \mathsf{su} \\ (\mathsf{do}\ \mathsf{v}_2)\ \mathsf{su} \end{array}$$

This is not such a favourable situation, because it does not include the not so infrequent VSO order. If we forbid ourselves type raising then we can only resort to passing to a stronger deductive system, for example the LAMBEK system. This solution has been advocated by many people, for various reasons. The strongest such system is that of MARK STEEDMAN [8] who argues that there are additional modes of syntactic combination, paired with different Frege-maps. One of these the additional mode is forward/backward composition, alias GEACH-rule. As STEEDMAN observes, this gives the additional benefit of getting the gapping facts right. [2] However, his solution, even though parsing all six basic word orders, fails to assign a parse to the following word orders.

$$\begin{array}{c} \mathsf{do}\ \mathsf{v}_3\ \mathsf{su}\ \mathsf{io} \\ \mathsf{io}\ \mathsf{su}\ \mathsf{v}_3\ \mathsf{do} \end{array}$$

In languages where the arguments of the verb are basically free, such as German and Latin, such word orders exist. One may argue that German falls out of this list here because the verb is sentence final, but the problem remains if we take verbs with more than three arguments. Thus, STEEDMAN's solution is not viable, failing to parse basic sentences.

[2] There is a slight problem with gapping in SVO languages that STEEDMAN discusses, namely when we take

William met the President and his friend the Vice-President.

where we must assume that the second conjunct, consisting of SO must be a constituent. In the left hand conjunct there is no such constituent. STEEDMAN solves this by allowing the left hand side to split up after being composed. Even if we accept this, the constructions given below constitute strong counterexamples because they do not even form constituents. Notice that the following construction is legitimate in German, showing that we can coordinate what is a nonconstituent in STEEDMANS analysis.

Hans gibt Maria ein Buch und Johann seiner Freundin eine Blume.

Let us pause and see what this gets us. Assuming that the verb discharges the arguments in a fixed order and under the syntactic adjacency (which is assumed in standard categorial grammars) we are bound to encounter hard problems. Whenever there are more than two arguments, free word order can block all possible parses. We must give up one of the principles here. Giving up the first is easy, we just fix an order in the lexicon and make verbs sufficiently polymorphous by adding lexical rules. These are then language dependent rules allowing the verb to switch the order in which it takes its arguments. This is a viable alternative, but from a lot of viewpoints unsactisfactory. It is precisely these lexical rules that constitute the body of language particular rules, and it is also an implicit commitment to non–lexicalism. Radical lexicalists cannot be happy with this. Giving up the second is harder, but more fruitful. It is not enough, though, to just split up the syntactic parse and the semantic interpretation; we also have to say in what ways they are related. One approach was to allow wrapping as an alternative mode of phonological combination, and that has been explored, too. (For example [7] and references therein.) We will not discuss that proposal here.

Let us pursue a particular solution, which is motivated by Government and Binding and points to a rather surprising solution of the paradoxe. It is assumed in GB that the verb discharges its arguments at D-structure in a canonical order. After this discharge the arguments are allowed to adjoin to vp in German (and other languages). This movement is referred to as *scrambling*. We assume here without questioning that this solution gets at least the word–order facts right. This might not be unproblematic, but the challenge we want to take up is the question of semantic translation. Let us also make the specific assumption (which is not necessary) that subjects originate inside the vp, so that after the discharge of the arguments we essentially get a sentence (which motivated the notation v_0). It follows that vp-adjuncts must be semantically similar to modal operators. The verb phrase has no free object variables which the adverb can quantify over. Notice however that if an argument scrambles and adjoins to vp it is syntactically an adjunct, and therefore, assuming a uniform Frege–function, must be a modal operator. How can this be possible? Here is an answer. The verb phrase is an open proposition, containing a variable x_i, which was freed by applying the verb to what is now the trace of the argument. So, dp-traces are of semantic type e, and basically just feed a variable to the verb. We then end up with an open expression. The raised dp is a generalized quantifier into which a particular variable is already inserted. If that variable is the same as the variable of the trace then the resulting translation is the desired one. If not, the quantifier quantifies over the wrong variable, in the worst case over a variable that is not free in the matrix ex-

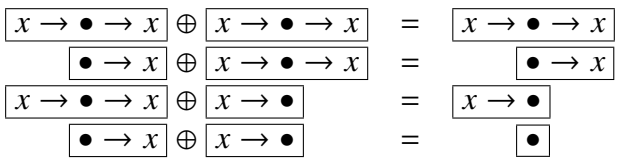
pression. Here is a simple example: the subject originates inside the verb phrase and scrambles. (This is just for illustration.)

$$\begin{array}{ccccc} (\textit{every man}_i & & (t_i & \textit{walks})) & \\ [\lambda\phi(\forall x_i.\mathbf{man}'(x_i) \to \phi) & & [x_i & \lambda x.\mathbf{walk}'(x)]] & \rightsquigarrow \\ [\lambda\phi(\forall x_i.\mathbf{man}'(x_i) \to \phi) & & & '(x_i)] & \rightsquigarrow \\ \multicolumn{5}{c}{(\forall x_i)(\mathbf{man}'(x_i) \to \mathbf{walk}'(x_i))} \end{array}$$

This analysis has several advantages. From a GB point of view it explains the importance of indices as a syntactic device, not a semantic one. Notice that the index helps in identifying the variable, nothing else. However, if we now return to the question of semantic innocence, then we must ask what makes a scrambled dp semantically different from a non–scrambled dp. There is no intrinsic difference we can isolate. The only facts that get adduced are binding facts (see [5]). Moreover, since any of the base generated dps is allowed to scramble, there is no intrinsic quality of dps that would allow to deduce that some are modal operators, and some are not. Following the strategy of raising to the worst case, we are led to conclude that while all dps are generated inside the verb phrase, *they all have to move out before s-structure*. So, the verb phrase consists of the verb alone with numerous traces. This gets us a uniform interpretation for all dps, at the cost of relegating the association with the argument slots totally to the regime of movement. Notice that recent accounts of transformational grammar, such as the Minimalist Program [1] and Bare Phrase Structure [2] do assume this type of construction, though for different reasons. However, now we have reached another impasse here: if all dps move, there is no theory internal reason *a posteriori* to assume that they have been moved at all, except to give the verb a chance to project its constituent in the order it wants to have the arguments. Arguments from binding theory for a specific D-structure (see [5], for example) do not square well with this and have to be looked at again. This has been done in [4].

This last discussion highlights the problems that we face just because the semantic interpretation prejudicated the syntactic constituency. We have tried using transformations to get rid of this straightjacket. But as regards the semantic interpretation we are in fact back to square one. If all λ-calculus was designed to do is to get the regime of variables right, then in face of the empirical facts it simply breaks down, because – as we have seen – one arrives at just the same problem, this time in the language of movement chains. We have not bothered to spell this dilemma out in other frameworks; we strongly believe that they too have no better answer to that problem. Let us instead sketch a theory that would not make these assumptions and square better with the facts, we hope. The proposal is due to

Albert Visser and Kees Vermeulen, outlined in [9]. Generally, the idea is that meanings should be seen as cells which store the content of some expression using some tags, called *referents*. Referents are formal objects, and they are simple and unanalysable. They can only be distinct or identical, like members of a set. Cells can communicate the properties they store about referents only if they give them (temporary) names. This communication process is described by referent systems. A *referent system* is defined to be a triple $\mathfrak{R}_1 = \langle I, R, E\rangle$ where I is an injective partial function from variables (which function as temporary names for referents) to the set R of referents, and E an injective partial function from R to the set of variables. We say, a referent system *imports* r under the name x if $I(x) = r$, and exports it under the name y if $E(r) = y$. The merge $\oplus$ of two referent systems is defined as follows. Given $\mathfrak{R}_1 := \langle I_1, R_1, E_1\rangle$ and $\mathfrak{R}_2 := \langle I_2, R_2, E_2\rangle$ then the result of merging the first with the second consists in the referent system $\langle I_3, R_3, E_3\rangle$, where R_3 is the disjoint union of R_1 and R_2, with $r_1 \in R_1$ and $r_2 \in R_2$ identified if $r_2 = I_2(E_1(r_1))$. A referent is imported under the name x if it is imported under x in $\mathfrak{R}_1$ or it is not imported at all by $\mathfrak{R}_1$, but then imported by $\mathfrak{R}_2$ under the name x. A referent is exported under the name y if it exported under that name by $\mathfrak{R}_2$, and if it is not exported by $\mathfrak{R}_2$ if it is exported under y by $\mathfrak{R}_1$. We will not try to be faithful to this specific formulation. First of all, we assume that we do not have a mechanism of export and import, but a mechanism of *identification*. This is symmetric. One cannot import a referent under a different name than by which it is exported. This leaves the following basic scenarios for a given referent under merge. By the first system it may imported under a name x, but not exported. It then is not identified with any referent of the second referent system and continues to be imported under the name x after merge. It may only be exported under the name x in which case the second referent system can pick up that referent under the name x, if it exports a referent under that name. The two will then be identified as one and the same referent by the merge. Whether or not the referent is exported is determined by the second system. If so, however, it is exported under the name x. The following are possible combinations (the list is not exhaustive).



Moreover, the referents are identified by morphological or syntactical criteria. Thus, a verb specifies not a *name* for the referent but a syntactic property of the expression that it expects for that referent. Such syntactic properties can be

phonological (left of, right of) or morphological (case features, agreement features). Since the list of these properties is limited, we may ask what happens if the same property is used twice as identification (think of ditransitive verbs requiring two accusative objects). This is ruled out by definition of the referent systems; since we are dealing with properties, not names for referents, we shall have to allow for that. Let us however put that problem aside. A technical solution seems be possible, but we have none at hand. Let us concentrate instead on the basic mechanism.

Determiner phrases are assumed to *export* a referent only. With respect to this referent they are called *arguments*. Verbs only import them, and so are called *predicates*. Adjectives and adverbs do import and export a variable and are therefore called *adjuncts*. Notice that the terminology is relative to a given variable. The adjective *proud* takes a pp-complement and is a predicate with respect to that argument variable, but the projected ap, *proud of his sister* has another variable which is both imported and exported. With respect to the latter the adjective is an adjunct.

Given all these prerequisites, let us now return to basic phrase structure. If it is so that the type of a constituent consisting of a transitive verb and an accusative dp is basically the same as that of an intransitive verb but different from that of a constituent consisting of a verb with a nominative dp, and since we must furthermore assume that both types of constituents exist then it makes little sense to relegate the constituency entirely to the syntax, as in categorial grammar. Otherwise we are back to polymorphism. Moreover, there is no attestable difference between the various constituent boundaries in connection with adverb placement. This means that we either consider adverbs to be v^0-adjuncts, v^1-adjuncts as well as v^2-adjuncts, or we give up the distinction between phrasal levels altogether. This gets us close to the proposal of [2]. There is only one mode of combination for verbs if they combine with arguments, and this is

$$\mathsf{v} \to \mathsf{v} \quad \mathsf{d}$$

order irrelevant. In GB-terminology we would write dp rather than d, but notice that this distinction is now obsolete. What drives the syntax is the semantic merger. Namely, the principle underlying the application of a syntactic rule is the following.

> CONDITION ON RULE APPLICATION. *Each combination of two syntactic items* x *and* y *into a constituent* [x y] *or* [y x] *must be accompanied*

by a matching of an imported variable of one of the items and an exported variable of the other.

So the syntactic merger into a larger constituent goes hand in hand with a merge of the referent systems. It is the referent systems which play the role of the argument structure; they determine with which property referents get exported or imported. With [3] we also assume that the reason for a verb to have arguments is that most of the features are *uninterpretable* at LF. A classical candidate is *case*. Case is there mainly for the reason that a verb can distinguish between arguments. Case serves no interpretive purpose other than inducing or preventing a merge. Let us say also that case is an *export condition*, while agreement is an *import condition*.

Several conclusions of this architecture fall out. We can show that an adjective needs to agree with a noun phrase it qualifies in gender, number and case simply because it is a semantic adjunct. To be more precise we have to say that if an adjective can at all morphologically agree with a noun it also has to, because it must pass on the export property of the noun. This follows because the noun phrase exports the variable under some property, and the adjective imports it under that property, otherwise there is no match of the referent systems. Since import and export conditions coincide, the adjective exports the variable under identical properties. A second conclusion is that checking generally takes place under sisterhood, not under spec-head-agreement. Checking is also not a property of the syntactic merge but of the merge of referent systems. This explains why cancellation of agreement features happens if a verb combines with a noun-phrase but not if an adjective phrase modifies a noun. The difference is that the checking features are connected with (formal) variables alias referents, and not with syntactic items. No movement is needed to check off features. This squares well with recent considerations that free word order in Germanic languages is not the effect of scrambling, but may well be base generated ([4]). In addition, we assume that the merge of referent systems goes hand in hand with the syntactic merge. There is no mechanism to delay semantic merger, because it is simply speaking neither needed (we have no movement) nor necessary. This has several consequences for parsing. For we assume that the merge of referent systems is automatic in case it yields a definite result. If a verb looks for only one argument with dative case then feeding such an argument will induce a merger. The bijection between argument places and syntactic arguments is an effect of the feature system of LF. [3] At LF no

[3] We equate LF here with the end of construction and/or derivation. If there is no movement, LF is simply the overt structure. The account here is neutral with respect to the question whether syntax is monostratal or polystratal. What we argue is that not only movement, but also the syntactic

uninterpretable features may be present in the structure. If features survive, it is either because some constituent is missing (or misplaced) or because there are too many arguments.

The basic attraction of this proposal is that it not only gets the basic sentence structure right, it also does that with a minimum effort. We can greatly reduce the categorial type system as well as large parts of the movement apparatus which has become an abundant and from a complexity point of view rather worrying aspect of transformational grammar. Nevertheless, one should not be led to think that with this semantics there is no movement. There are various intriguing cases of movement that have no apparent explanation in the present proposal. This is not to say that there can be none, but to find one is conditional on a proper understanding of the semantics. Examples are *wh-movement* and *head-movement*. We have to leave that for future research. Notice also that agreement in coordination induces a number of quirks. The phrase

William and the President

consists of two singular noun phrases but is itself a plural noun-phrase. The reason is not hard to find. We have a group variable which is introduced by the coordination, and it does not inherit the export properties of the individual conjunct, since they supply different variables. However, we are left with the question as to why all three noun-phrases have to agree in case. The answer that suggests itself is that agreement is interpretable at LF and so behaves differently from case, which is uninterpretable. Another solution is to say that while case is an export condition, agreement is an import condition. Moreover, one needs to stipulate that import conditions show distinct behaviour from export conditions. (Two verbs coordinated likewise do not constitute a plural verb.)

References

[1] Noam Chomsky. A minimalist program for linguistic theory. In K. Hale and Keyser S. J., editors, *The View from Building 20: Essays in Honour of Sylvain Bromberger*, pages 1 – 52. MIT Press, 1993.

[2] Noam Chomsky. Bare Phrase Structure. In Gert Webelhuth, editor, *Government and Binding Theory and the Minimalist Program*, pages 385 – 439. Blackwell, 1995.

merge can result in checking off a feature.

[3] Noam Chomsky. *The Minimalist Program*. MIT Press, 1995.

[4] Gisbert Fanselow. The final return of the base generators. *Linguistics in Potsdam*, 2, 1995.

[5] Hubert Haider. *Deutsche Syntax – generativ. Vorstudien zur Theorie einer projektiven Grammatik*. Gunter Narr Verlag, Tübingen, 1993.

[6] Richard S. Kayne. *The Antisymmetry of Syntax*. Number 25 in Linguistic Inquiry Monographs. MIT Press, 1994.

[7] Glyn V. Morrill. *Type Logical Grammar. Categorial Logic of Signs*. Kluwer Academic Publishers, Dordrecht, 1994.

[8] Mark Steedman. Gapping as Constituent Coordination. *Linguistics and Philosophy*, 13:207 – 263, 1990.

[9] Kees F. M. Vermeulen. Merging without Mystery or: Variables in Dynamic Semantics. *Journal of Philosophical Logic*, 24:405 – 450, 1995.