New Directions in Type-Theoretic Grammars

Reinhard Muskens

Published online: 20 December 2009
© The Author(s) 2009. This article is published with open access at Springerlink.com

Abstract   This paper argues for the idea that in describing language we should follow Haskell Curry in distinguishing between the structure of an expression and its appearance or manifestation. It is explained how making this distinction obviates the need for directed types in type-theoretic grammars and a simple grammatical formalism is sketched in which representations at all levels are lambda terms. The lambda term representing the abstract structure of an expression is homomorphically translated to a lambda term representing its manifestation, but also to a lambda term representing its semantics.

Keywords   Lambda grammar · Abstract categorial grammar · Tectogrammatics · Phenogrammatics

1 Introduction

In 1961 Haskell Curry published his by now famous paper on ‘Some Logical Aspects of Grammatical Structure’ (Curry 1961). In this paper, large parts of which had already been written in the 1940s, he made a distinction between what he called the tectogrammatics and the phenogrammatics of language. The latter is language as it appears or manifests itself; the former language as it is built, its underlying structure.¹ The

¹ Classical Greek τέκτων means builder (compare architect), while φαίνω means appear or shine (phenotype, phenomenon). I am grateful to David Dowty for pointing this out.

The author wishes to thank the Netherlands Organisation for Scientific Research (NWO) for supporting the Workshop on New Directions in Type-theoretic Grammars, held in Dublin, 6–10 August, 2007.

R. Muskens (✉)
Department of Philosophy, Tilburg University, Tilburg, The Netherlands
e-mail: r.a.muskens@uvt.nl
distinction between these two levels—very similar to the more recent distinction between abstract syntax and concrete syntax in compiler theory—enabled Curry to get rid of directionality in the type system in categorial grammar. In 1953 Bar-Hillel had introduced a distinction between categories of expressions seeking material to their right and of those seeking material to their left (Bar-Hillel 1953) and this distinction had been taken over by Lambek (1958), but Curry criticises the resulting type system for its ‘admixture of phenogrammatics’.

While Curry uses a functional type system on the tectogrammatical level similar to the system introduced by Ajdukiewicz (1935) (and to the one still in use in simple type theory), he models phenogrammatics with the help of *functors*, which are ‘means of combining phrases to form other phrases’. Any kind of operation from sequences of phrases to phrases is allowed here. The transformations of Transformational Grammar are given as an example, but one could also think of rules like ‘put past morphology on the head of X’, or ‘attach Y just before the head of Z’. In Curry’s paper, moreover, much use is made of *incomplete* phrases. In (1) we give some examples.

\begin{itemize}
  \item a. both $-1$ and $-2$
  \item b. $-1$ were eaten by the children
  \item c. $-1$ is between $-2$ and $-3$
\end{itemize}

The items in (1) are functors, with the blanks indicating where arguments are to be inserted, and the subscripts constraining the order in which these insertions can take place. (1c), for example, combines the phrases *Paris*, *London*, and *Berlin* (in that order) to form the sentence *Paris is between London and Berlin*.

Curry also considers a type system\(^2\) for functors, with types as in Table 1 (some of the examples are mine, not Curry’s) and gives a rule that makes an obvious connection with Lambek’s system:

\[ \ldots \text{Lambek’s ‘‘} f \text{‘‘ is an } N/S \text{’’ would mean the same as ‘‘} f -1 \text{‘‘ is an } FSN \text{’’}, \]
\[ \text{whereas his ‘‘} f \text{‘‘ is an } N\setminus S \text{’’ would mean the same as my ‘‘} f -1 \text{‘‘ is an } FNS \text{’’}. \]

Thus Lambek’s conception has an admixture of phenogrammatics. Moreover it seems to break down completely with reference to functors which are not either prefixes or suffixes.

\(^2\) Curry’s notation for types will be unfamiliar to many readers, but can easily be translated to a more familiar one: Delete all the \(F\)s (subscripted or not) and restore parentheses using the rule that \(ABC\) is short for \(A(BC)\). So \(F_2(FNN)(FNN)(FNN)\) is \((NN)(NN)(NN)\), which is short for \((NN)((NN)(NN))\), for example.
Table 1  Some functors and their types

| Type          | Functors                          |
|---------------|-----------------------------------|
| $N$           | John, Sue, Fred, he               |
| $F_{NS}$      | $-1$ smokes, $-1$ dances, $-1$ is running |
| $F_{2NNS}$    | $-1$ loves $-2$, $-1$ admires $-2$ |
| $F_{3NNNS}$   | $-1$ introduces $-2$ to $-3$, $-1$ is between $-2$ and $-3$ |
| $F_{2}(F_{NN})(F_{NN})(F_{NN})$ | both $-1$ and $-2$               |

directional grammars cannot represent functors other than prefixes or suffixes is spot on. Lambek categorial grammars essentially fail to deal with medial gaps. While they can easily handle extractions such as in the *boys Zelda admired*, they have problems with those such as in the *books Zelda bought in Paris* (Moortgat 1997). This is a direct consequence of the attempt to regulate word order on the level of the type system. In fact, a lot of research carried out within the Lambek paradigm can be seen as the invention of a series of epicycles needed to counter this architectural mistake. Lambek’s realization that grammatical extraction and hypothetical reasoning (or lambda abstraction) are one and the same thing has been absolutely pivotal to categorial grammar, but type directionality should be considered a red herring.

While the pheno\/tecto distinction proposed by Curry has never become mainstream in categorial grammar (let alone in linguistics), interest has not ceased to exist. Two highlights of the approach undoubtedly are Dowty’s (1982, 1995). In the first of these papers Dowty firmly associates tectogrammatics with those parts of a grammar which are language universal, while placing phenogrammatics in the language specific part. Modelling tectostructure with the help of a reduced form of Montague’s analysis trees (Montague 1973), Dowty lets (2) go proxy for the language universal structure of the sentence that in English is expressed as *John hits Mary*.

\[
\begin{array}{c}
\text{“John”}_T \\
[l] \\
\text{“hit”}_T \text{“Mary”}_T
\end{array}
\]

While Montague’s original analysis trees have phenogrammatic material decorating all their nodes, Dowty considers reduced forms where such material is lacking. Different languages may realize analysis trees in different ways, on the basis of their own versions of Montague’s ‘structural operations’. While English decorates (2) with the help of something akin to Montague’s Montague (1973) rules $F_4$ and $F_5$, resulting in

---

3 Dowty (1995) was presented at a conference in January 1990.

4 Curry already makes a remark going in this direction: “It is to be expected that grammatical structure will vary less from language to language than does the phenogrammatics.” (Curry 1961, p. 66)
(3a), Japanese, using different rules, arrives at (3b). Other languages have a completely different phenogrammatics.

The distinction between tectogrammatics and phenogrammatics is somewhat reminiscent of the division between syntax and “phonology” in current versions of the Minimalist Program (Chomsky 1995), but while “phonology” mainly seems to play the role of a wastebasket in the latter, Dowty (1995) provides detailed proposals for the phenogrammatics of fragments of English and Finnish. The basic data structures in Dowty’s proposal are unordered lists and the default operation on these is a simple merge, but order is constrained by linear precedence principles (borrowed from Generalised Phrase Structure Grammar, Gazdar et al. 1985), by rule specific operations (which are marked), and by the general rule that the expressions belonging to a bounding category cannot mingle with expressions outside of that category. The answer to the question which are bounding categories can vary across languages, on the other hand. The resulting system liberates word order considerations from the straitjacket of the ordered tree and redefines the rules of play for syntactic theory.

Dowty’s theory is best appreciated from a linguistic point of view, but there is a formal bonus too. Anyone who has ever tried to prove anything about the fragment in Montague (1973) will have noticed that a simple inductive definition of the notion of ‘analysis tree’ would come in extremely handy, but Montague (1973) gives no proper definition. Muskens (1995) therefore defines a language of labeled bracketings such as the one in (4a) (the labels follow the numbering of syntactic rules in Montague 1973). Analysis trees like this one provide the core of the system and subsequently defined functions homomorphically send analysis trees to strings and to semantic values. In particular, a function $\sigma$ is defined providing a phenogrammatics and a function $(\cdot)^\sigma$ sends analysis trees (tectostructures) to semantic representations. (4b) and (4c) give examples of how (4a) is translated twice.

\begin{align*}
\text{a.} & \quad [\text{a woman}]^3[\text{every man}]^3[\text{love he}_0]^5[4]^4\sigma^0 \\
\text{b.} & \quad \sigma(4a) = \text{every man loves a woman} \\
\text{c.} & \quad (4a)^\sigma = \lambda i \exists y \left( \text{woman } y_i \land \forall x \left( \text{man } x_i \rightarrow \text{love } yxi \right) \right)
\end{align*}
The purpose was to give an alternative definition of the fragment in Montague (1973) that was mathematically more perspicuous than the original while following it closely, but structures such as the one in (4a) are functionally equivalent to Dowty’s reduced analysis trees and language-dependent variations upon $\sigma$ may formalize various phenogrammatic realizations of these tectostructures.

It may seem that all these considerations are somewhat particular to Montague Grammar, which, while it is loosely based on categorial grammar, is certainly not its most general formulation. But note the proximity of (4a) to the lambda term in (5a).

\[
\begin{align*}
&\text{a. } ((\text{a woman})\lambda \xi ((\text{every man}) (\text{love } \xi))) \\
&\text{b. } \varphi(5a) = ((\text{every } \bullet \text{ man}) \bullet (\text{loves } \bullet (\text{a } \bullet \text{ woman}))) \\
&\text{c. } \mu(5a) = \lambda i \exists y (\text{woman } yi \land \forall x (\text{man } xi \rightarrow \text{love } yxi))
\end{align*}
\]

All other analysis trees defined in Muskens (1995) have similar resemblances to lambda terms, with indexed pronouns acting as variables and certain superscripts acting as binders.

Now consider the following observation. If the reduced analysis trees in (2) or (4a) could be replaced by linear lambda terms (lambda terms in which each binder binds exactly one variable) such as the one in (5a), we would be back at the heart of categorial grammar. Linear lambda terms, after all, are the proof terms for Lambek’s calculus and are in one-to-one correspondence with proofs in the undirected version L*P of this calculus, studied in van Benthem (1986, 1991). Their adoption as a representation of tectostructure therefore would mean that Lambek’s idea to treat extraction as hypothetical reasoning is embraced again.

That this can indeed be done was shown independently by de Groote (2001) and Muskens (2001, 2003) (see www.loria.fr/equipes/calligramme/acg/ for more references), who provide closely related formalisms (called Abstract Categorial Grammar and Lambda Grammar, respectively) that embody the phen/o/tecto distinction but also enable the ‘gap management’ that we saw to be lacking in Curry (1961). I will explain their system on the basis of a very small fragment.

First some technicalities. If $B$ is some set of basic types, we write $TYP(B)$ for the smallest set containing $B$ such that $(\alpha \beta) \in TYP(B)$ whenever $\alpha, \beta \in TYP(B)$. A function $\eta$ from types to types is said to be a type homomorphism if $\eta(AB) = (\eta(A)\eta(B))$ whenever $\eta(AB)$ is defined. It is clear that a type homomorphism $\eta$ with domain $TYP(B)$ is completely determined by the values of $\eta$ for types $\alpha \in B$. For example, let $B = \{D, N, S\}$ (D stands for determiner phrases, N for noun phrases, S for sentences) and let $\delta$ be a type homomorphism such that $\delta(D) = e, \delta(N) = est,$ and $\delta(S) = st$ (here $e$ is for entities, and $s$ for possible worlds). This is the function illustrated in Table 3.

5 We let $v$ stand for the type of nodes, and $t$ for the type of truth values, so that $vt$ is the type of sets of nodes. Associating this type with phenogrammatical objects makes it possible to define certain unary and binary modal operators over them. See Muskens (2007) for detailed motivation.

6 As before, we omit brackets in types on the understanding that association is to the right.
Note that this implies that \( \vartheta(\ ) \) the constants occurring in the first column of Table 2, with types as in the second as was stated in (5b). A second term homomorphism sent to a term that is \( \beta\eta \) from lambda terms to lambda terms is a term homomorphism based on between its arguments, and the variables \( t \) and \( T \) are of types \( vt \) and \( (vt)vt \), respectively.

| Table 2 | An abstract categorial grammar connecting tectostructure with phenomenogrammatics |
|---------|----------------------------------|
| constant \( c \) | type \( \tau \) | \( \varphi(c) \) | \( \delta(\tau) \) |
| john | \( D \) | John | \( vt \) |
| woman | \( N \) | woman | \( vt \) |
| smokes | \( DS \) | \( \lambda t.(t \bullet \text{smokes}) \) | \( (vt)vt \) |
| loves | \( DDS \) | \( \lambda t'.(t \bullet (\text{loves} \bullet t')) \) | \( (vt)(vt)vt \) |
| believes | \( SDS \) | \( \lambda t'.(t \bullet (\text{believes} \bullet t')) \) | \( (vt)(vt)vt \) |
| every | \( N(DS)S \) | \( \lambda t.\text{T}.(\text{every} \bullet t) \) | \( (vt)((vt)vt)vt \) |
| a | \( N(DS)S \) | \( \lambda t.\text{T}.(a \bullet t) \) | \( (vt)((vt)vt)vt \) |

| Table 3 | An abstract categorial grammar connecting tectostructure with meaning |
|---------|----------------------------------|
| constant \( c \) | type \( \tau \) | \( \mu(c) \) | \( \gamma(\tau) \) |
| john | \( D \) | john | \( e \) |
| woman | \( N \) | woman | est |
| smokes | \( DS \) | smoke | est |
| loves | \( DDS \) | love | est |
| believes | \( SDS \) | \( \lambda p\lambda x\lambda i\forall j(Bxij \rightarrow pj) \) | \( (st)est \) |
| every | \( N(DS)S \) | \( \lambda P'\lambda PLx.\forall x(P'xi \rightarrow Pxi) \) | \( (est)(est)st \) |
| a | \( N(DS)S \) | \( \lambda P'\lambda P\lambda i.\exists x(P'xi \land Pxi) \) | \( (est)(est)st \) |

A second notion we want to define is that of a term homomorphism. A function \( \theta \) from lambda terms to lambda terms is a term homomorphism based on \( \eta \) if \( \eta \) is a type homomorphism and, whenever \( M \) is in the domain of \( \theta \):

- \( \theta(M) \) is a term of type \( \eta(\tau) \), if \( M \) is a constant of type \( \tau \);
- \( \theta(M) \) is the \( n \)th variable of type \( \eta(\tau) \), if \( M \) is the \( n \)th variable of type \( \tau \);
- \( \theta(M) = (\theta(A)\theta(B)) \), if \( M \equiv (AB) \);
- \( \theta(M) = \lambda y.\theta(A) \), where \( y = \theta(x) \), if \( M \equiv (\lambda x.A) \).

Note that this implies that \( \theta(M) \) is a term of type \( \eta(\tau) \), if \( M \) is any term of type \( \tau \).

If \( C \) is some set of typed constants, we write \( \Lambda_0(C) \) for the set of all linear lambda terms with constants only from \( C \). Clearly, a term homomorphism \( \vartheta \) with domain \( \Lambda_0(C) \) is completely determined by the values \( \vartheta(c) \) for \( c \in C \). Consider, for example, the constants occurring in the first column of Table 2, with types as in the second column. A term homomorphism \( \varphi \) based on \( \delta \) is completely specified for each term in \( \Lambda_0(\{\text{john, woman, ...}\}) \) by the values for these constants given in the third column and the reader may verify that \( ((a \text{ woman})\lambda \xi.((\text{every} \text{ man})(\text{love} \ \xi))) \) is sent to a term that is \( \beta\eta \) equivalent with \( ((\text{every} \bullet \text{ man}) \bullet (\text{loves} \bullet (a \bullet \text{woman}))) \) as was stated in (5b). A second term homomorphism \( \mu \) based on \( \gamma \) is given in

---

7 Here the words in sans represent constants of type \( vt \), \( \bullet \) is an operator of type \( (vt)(vt)vt \) which we write between its arguments, and the variables \( t \) and \( T \) are of types \( vt \) and \( (vt)vt \), respectively.

8 In Tables 2 and 3 we have suppressed entries like that for \( \text{man} \) that can easily be constructed on the basis of other entries (like that for \( \text{woman} \)).
Table 3 and sends \( ((a \text{ woman})\lambda \xi ((\text{every man}) (\text{love } \xi )))) \) to \( \lambda i \exists y \left( \text{woman } y_i \land \forall x (\text{man } x_i \rightarrow \text{love } y_xi) \right) \). In fact Tables 2 and 3 together define a small fragment of natural language, in which phenogrammatics and meaning are coupled via tectostructure, the build they have in common.

This fragment could easily be extended, but should also be improved upon. The phenogrammatics defined in Table 2 assigns run-of-the-mill linguistic trees to the abstract terms of tectostructure while the linguistic benefits of the approach would be better exploited if a logic capturing the ideas in Dowty (1995) were defined. This could be done if the basic algebra behind Dowty’s phenogrammatics were given a purely logical formulation. This challenge will be left to a future occasion.

The line of research I have sketched here is but one strand in an interwoven texture in which many others are also present. One of them is the research program that since the 1980s has been pursued by Aarne Ranta and his coworkers and which has led to highlights such as Ranta (1994, 2004). Ranta’s work is based on Martin-Löf’s constructive version of type theory and a distinction between abstract syntax and concrete syntax has been present in it from the start. Another strand is Oehrle’s (1994, 1995, 1999) use of labeled deduction in linguistic description. Oehrle decorates proofs of the undirected Lambek calculus with (1) types, (2) terms that represent meaning, and (3) terms representing phenogrammatics. The system is very close to the one presented here (for more detailed comparison, see Muskens 2003). Other related formalisms are Lecomte’s Categorial Minimalism (2001) and the Convergent Grammar of Carl Pollard (see e.g. Mansfield et al. 2009). Since the turn of the century there has been a heightened activity within this collection of type-theoretical formalisms bearing family resemblances to one another. They are all in debt to Curry’s pheno/tecto distinction in one way or another. And so, while this distinction is far from new itself, it has led to new directions in type-theoretic grammars.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

Ajdukiewicz, K. (1935). Die syntaktische Konnexität. Studia Philosophica, 1, 1–27 English translation in Storrs McCall, ed., Polish Logic, 1920–1939, Oxford, 1967, 207–231.

Bar-Hillel, Y. (1953). A quasi-arithmetical notation for syntactic description. Language, 29, 47–58.

Chomsky, N. (1995). The minimalist program. Harvard: MIT Press.

Curry, H. B. (1961). Some logical aspects of grammatical structure. In O. J. Roman (Ed.), Structure of language and its mathematical aspects vol 12 of symposia on applied mathematics (pp. 56–68). Providence: American Mathematical Society.

de Groote, J (2001). Towards abstract categorial grammars. In Association for computational linguistics, 39th annual meeting and 10th conference of the European chapter, proceedings of the conference (pp. 148–155). Toulouse, France: ACL.

Dowty, D. R. (1982). Grammatical relations and Montague Grammar. In P. Jacobson & G. K. Pullum (Eds.), The nature of syntactic representation (pp. 79–130). Dordrecht: Reidel.

9 Here John, woman, etc are constants that are typed as in the fourth column, B is a constant of type est that stands for the relation of doxastic accessibility, p is a variable of type st, P a variable of type est, x a variable of type e, and i and j are variables of type s.
Dowty, D. R. (1995). Toward a minimalist theory of syntactic structure. In H. Bunt & A. Horckvam (Eds.), *Syntactic discontinuity* (pp. 11–62). The Hague: Mouton.

Gazdar, G., Klein, E., Pullum, G., & Sag, I. (1985). *Generalized Phrase Structure Grammar*. Cambridge, MA: Harvard University Press.

Lambek, J. (1958). The mathematics of sentence structure. *American Mathematical Monthly, 65*, 154–170.

Lecomte, A. (2001). Categorial minimalism. In M. Moortgat (Eds.), *LACL’98*, vol 2014 of LNAI (pp. 143–158).

Mansfield, L., Martin, S., Pollard, C., & Worth, C. (2009). *Phenogrammatical labelling in Convergent Grammar: the case of wrap* (unpublished manuscript).

Montague, R. (1973). The proper treatment of quantification in ordinary English. In J. Hintikka, J. Moravcsik, & P. Suppes (Eds.), *Approaches to natural language* (pp. 221–242). Dordrecht: Reidel. Reprinted in *Thomason* (1974).

Moortgat, M. (1997). Categorial type logics. In J. van Benthem, A. Ter Meulen (Eds.), *Handbook of logic and language* (pp. 93–177). Elsevier.

Muskens, R. A. (1995). *Meaning and partiality*. Stanford: CSLI.

Muskens, R. A. (2001). Categorial grammar and Lexical-Functional Grammar. In M. Butt, T. H. King (Eds.), *Proceedings of the LFG01 conference* (pp. 259–279). Stanford, CA: University of Hong Kong. CSLI Publications. http://cslipublications.stanford.edu/LFG/6/lfg01.html.

Muskens, R. A. (2003). Language, lambdas, and logic. In G.-J. Kruijff, R. Oehrle (Eds.), *Resource sensitivity in binding and anaphora studies in linguistics and philosophy* (pp. 23–54). Kluwer.

Muskens, R. (2007). Separating syntax and combinatorics in categorial grammar. *Research on Language and Computation, 5*(3), 267–285.

Oehrle, R. T. (1994). Term-labeled categorial type systems. *Linguistics and Philosophy, 17*, 633–678.

Oehrle, R. T. (1995). Some 3-dimensional systems of labelled deduction. *Bulletin of the IGPL, 3*, 429–448.

Oehrle, R. T. (1999). LFG as labeled deduction. In M. Dalrymple (Ed.), *Semantics and syntax in Lexical Functional Grammar, Chap. 9* (pp. 319–357). Cambridge, MA: MIT Press.

Ranta, A. (1994). *Type-theoretical grammar*. Oxford: Oxford UP.

Ranta, A. (2004). Grammatical framework: A type-theoretical grammar formalism. *Journal of Functional Programming, 14*(2), 145–189.

Thomason, R. (ed). (1974). *Formal philosophy, selected papers of Richard Montague*. Dordrecht: Reidel.

van Benthem, J. F. A. K. (1986). *Essays in logical semantics*. Dordrecht: Reidel.

van Benthem, J. F. A. K. (1991). *Language in action*. Amsterdam: North-Holland.