Advances in Logical Grammar: Background to Linear Grammar

Carl Pollard

June 13, 2012

Linear Grammar

Linear Grammar (LG) is a practical framework for linguistic analysis influenced by three traditions in linguistic theory:

- **categorial grammar (CG)**, a kind of syntactic analysis founded by Joachim Lambek (late 1950s) that treats lexical entries and grammar rules, respectively, as axioms and inference rules of a **proof theory**.
- **Montague semantics**, founded by Richard Montague (late 1960s), influenced by earlier philosophical logicians Frege (1892), Carnap (1947), and and Kripke (1963). Uses **type theory** (Church 1940, Henkin 1950) to analyze sentence meanings.
- **Dynamic semantics**, founded by Kamp (DRT, 1981), Heim (FCS, 1982), and others based on philosophical ideas of Stalnaker and Lewis (1960s and 1970s) about the role of context in the interpretation of multi-sentence discourses.

Sources of Linear Grammar (1/2)

LG is a synthesis based on recent developments in all three of these traditions:

- **curryesque** CG, which analyzes syntax using **linear logic** (Girard 1987). Inspired by programmatic ideas of Curry (1961) and technical innovations of Oehrle (1994).

  Includes de Groote's (2001) **abstract categorial grammar (ACG)** and Muskens' (2003, 2007) **lambda grammar**.
- **hyperintensional semantics**, a kind of type-theoretic semantics which proposes a more **fine-grained** analysis of sentence meaning than Montague's.

  An early form was Thomason's (1980) **intentional semantics**. More recent avatars are Muskens (2005) and Pollard (2008).

Sources of Linear Grammar (2/2)

- **Type-theoretic dynamic semantics**, which extends Montague's type-theoretic methods to analyze the kinds of discourse phenomena analyzed by DRT and FCS.
- Muskens (1994, 1996) pioneered this approach. More recent proposals are Beaver (2001), de Groote (2006), van Eijck and Unger (2010), and Martin and Pollard (2010, 2011).
- LG with dynamic semantics is called DyCG (dynamic categorial grammar), so when the term LG is used, the implication is that the semantics is static (i.e. not dynamic).

Curry

In a 1948 lecture, published in expanded form in 1961, Curry proposed that a linguistic expression should be analyzed as consisting of:

1. a **phenogrammatical** component: specifies the expression's superficial form
2. a **tectogrammatical** component: specifies the the expression's combinatory potential
3. a **semantic** component: specifies the expression's meaning

The Phenogrammatical Component

- usually abbreviated to just **pheno**
- Corresponds roughly to what computer scientists sometimes call **concrete syntax**
- Also corresponds roughly to what linguists call **phonology**, broadly construed to include word order and nonsegmental (or prosodic) aspects
- relates to what the expression sounds like (or in the case of sign language, looks like)

The Tectogrammatical Component

- usually abbreviated to just **tecto**
- Corresponds roughly to what computer scientists sometimes call **abstract syntax**
- Also corresponds roughly to what linguists call **syntactic category**.
- Relates to what other expressions the expression can combine with, and what results from the combination

Lambek

- Invented his **syntactic calculus** in 1958, later called the **Lambek calculus**.
- A Lambek calculus is a grammar written in the form of a **logical proof system**.
- The role of linguist's trees is taken over by **proof trees**.
- Words correspond to **axioms**.
- Grammar rules are replaced by logical **inference rules**.
- Well-formed linguistic expressions correspond to **theorems** of the proof system.
- Unlike earlier forms of categorial grammar (CG) due to Ajdukiewicz and Bar-Hillel), the Lambek calculus makes (crucial!) use of the rule of **hypothetical proof**, which we will explain soon.

Montague (1/2)

- In late 1960's, originated a style of CG influenced by ideas drawn from the philosophical logicians Gottlob Frege, Rudolph Carnap, Saul Kripke, and others.
- A Montague grammar recursively defines a set of triples, each of which consists of a **word string**, a **syntactic type**, and a typed lambda calculus term (TLC) denoting a meaning (often a function).
- In retrospect, we can relate Montague's string to Curry's pheno, and Montague's syntactic type to Curry's tecto.
- Some of the triples (**lexical entries**) are given, while the **rules** of the grammar produce new triples from old ones.
- Unlike Lambek calculus, Montague's CG was primitive (like Ajdukiewicz-Bar Hillel CG) in the sense of not having a rule like Hypothetical Proof.

Montague (2/2)

- Each rule include 'recipes' specifying how to construct the string and meaning of a new expression, respectively, from the strings and meanings of the expression's immediate constituents.
- The operation involved in constructing the new expression's string is usually **concatenation**.

- The operation involved in constructing the new expression's meaning is usually **function application**.
- On the semantic side, this last point is a version of Frege's notion of semantic **compositionality**.

Oehrle

- In the mid-to-late 1980's, categorial grammarians (such as van Benthem, Moortgat, Morrill) had the idea of combining Lambek calculus with Montague grammar.
- Within this setting, Oehrle introduced three technical innovations.
- The first was to replace the Lambek calculus with a simpler logic, namely **linear logic**.
- The second was to allow phenos to be not just strings, but also (possibly higher-order) functions over strings.
- The second innovation involved using TLC terms to denote phenos (not only for meanings as in Montague grammar).
- Oehrle's third innovation was a particular technique for analyzing quantified noun phrases ('quantifier lowering via β-reduction'), which we'll explain in due course.