Parsing Ellipsis Efficiently

Gregory M. Kobele

University of Chicago

April 23, 2017

Abstract

I show how a transformational account of ellipsis allows for a sound and complete parsing algorithm which allows for the efficient processing of discourses containing elliptical sentences. The result relies not just on formal properties of the grammar formalism (*formal universals*), but also crucially on linguistic assumptions (*substantive universals*).

Keywords: ellipsis, parsing, minimalist grammars, computational complexity

1 Introduction

The problem of recovering a meaning from a linguistic expression is one that we solve by and large effortlessly every day of our lives. A main way to approach this problem formally has been to identify the relevant linguistic expressions with sentences, and to develop algorithms which recover parse trees for sentences.

For many interesting linguistic grammar formalisms, this problem turns out to be efficiently exactly solvable: the set of *all* possible parses of a sentence can be computed in time polynomial in the length of the sentence.

Focussing on recovering parses is in general a reasonable approximation to recovering meanings; a parse tree typically uniquely identifies a meaning representation (via a compositional semantics). However, this is not always the case. In particular, in anaphoric constructions typically more information is required to map the parse tree into a meaning representation than is represented in the parse tree itself. Thus the problem of identifying a parse tree for a sentence with anaphoric expressions is only a proper subpart of the problem of recovering a meaning for the sentence.

In addition to making the complexity of recovering a meaning potentially more complex than that of recovering a parse tree, anaphoric constructions also disrupt the simplifying assumption that discourses are 'bags of sentences.' This assumption guarantees that the cost of parsing a discourse is simply the sum of the costs of parsing each of its sentences. The problem posed by anaphoric dependencies is that whether a sentence has a fully fleshed out meaning can depend on whether there is available in the surrounding discourse a legitimate antecedent.[1]

In this paper I focus on ellipsis[2] in the framework of minimalist grammars

[1] A similar problem arises with discourse relations, which impose more structure on discourses than is present in the bag-of-sentences model. One possible difference between these two cases is that it does not seem that different discourse structurings can render sentences *meaningless* in the same way that lack of an antecedent can. Still, especially under theories which treat discourse structure as an extension of syntactic structure, the number of parses of the discourse grows as a polynomial function of the number of words in the discourse.

[2] The status of ellipsis as an anaphoric construction is not uncontroversial, at least in the sense here of its structure not uniquely determining its meaning. A long-standing proposal

[Stabler, 1997], in particular the theory of Kobele [2015]. I show that the problem of parsing sentences with ellipsis sites (i.e., the problem of recovering all parse trees for a given sentence) is efficiently solvable, and that, given a parse tree, and a set of possible antecedents, determining a legitimate resolution of its ellipsis sites, if one exists, can be done in constant time. Thus in the system of Kobele [2015], the problem of recovering a meaning for a string in a particular discourse context can be solved in polynomial time, even in the presence of ellipsis. I then show that the problem of updating a discourse context with the newly available antecedents of a freshly parsed sentence is also solvable in polynomial time, given standard linguistic assumptions about the licensing of ellipsis sites [Lobeck, 1995], and features of the grammar formalism particular to linguistic grammar formalisms. From this, it follows that the cost of parsing discourses with ellipsis reduces to the sum of the costs of parsing their component sentences.

While couched in terms of minimalist grammars, the approach to ellipsis developed in Kobele [2015] can be extended to any grammar formalism characterizable in terms of second order ACGs [de Groote, 2001, de Groote and Pogodalla, 2004]. However, the linguistic applications touted in Kobele

in the transformational community (going back to at least Lees [1960]; Merchant [2001] is a more recent champion) has it that the ellipsis site has a fully articulated structure appropriate to its meaning. I believe that this controversy is irrelevant for two reasons. From a purely pragmatic standpoint, the anaphoric ('proform') treatment provides a clear divide between the cost of parsing (plus identifying ellipsis sites), which is known, and the cost of resolving the ellipses. Syntactic theories of ellipsis mix these two, seemingly leaving us with no clear way to investigate the problem, short of developing completely novel theories of parsing. From a more empirical perspective, Kobele [2015] argues that the standard arguments in favor of refined syntactic structures inside of ellipsis sites are in fact compatible with proform theories, given a slightly more sophisticated treatment of syntactic categories. Note that Kobele [2012a] argues that syntactic theories are simply alternative notations for proform theories.

[2015] depend crucially on particularities of the analysis therein, which seem difficult to replicate outside of the minimalist grammar formalism.

2 Ellipsis in Minimalist Grammars

The formal framework of minimalist grammars was developed by Stabler [1997] as a formalization of the core aspects of Chomsky's then nascent minimalist program [Chomsky, 1995]. It was proven shortly thereafter [Michaelis, 1998] that there is a constructive procedure to transform a minimalist grammar into a strongly equivalent[3] multiple context-free grammar [Seki et al., 1991], and thus that minimalist grammars enjoy an efficiently solvable parsing problem [Harkema, 2001].

Minimalist grammars have proven to be a good formalization of the minimalist program, in that virtually all extensions thereof and amendments thereto either already have been or at least appear to be directly and faithfully implemented in this framework.[4]

I adopt in section 2.1 a particularly spare, 'chain-based' version of minimalist grammars [Stabler and Keenan, 2003]. This version, viewed from the perspective of theoretical linguistics, commits to a strong version of 'cyclic spellout' [Chomsky, 2000], where spell-out is effected at each derivational step (i.e. compositionally). Other versions are admissible, so long as they do not affect the fact that the set of derivation trees is regular, as discussed in section 2.2. Relevant aspects of the theory of ellipsis in Kobele [2015] are

[3] I use *strongly equivalent* in the sense that the *derivation trees* assigned by both grammars to strings are isomorphic.

[4] In some cases, the straightforward implementation increases the generative capacity of the framework, even to turing completeness [Kobele and Michaelis, 2005].

given in section 2.3.

2.1 Minimalist Grammars

A minimalist grammar (over some alphabet Σ) is determined by a finite set $\texttt{AtFeat}$ of *atomic features*, and a finite set $\texttt{Lex}$ of lexical items. A lexical item $\ell = \langle w, \delta \rangle$ is a pair consisting of a word w (over Σ) and a feature bundle δ. A feature bundle is a finite list $\delta = f_1, \ldots, f_n$ of features, which are elements of the finite set $\texttt{Feat}$ defined below.[5]

$$\mathsf{Feat} := \{\texttt{f}, \texttt{=f}, \texttt{+f}, \texttt{-f} : f \in \mathsf{AtFeat}\}$$

A feature is either an *attractor* ($\texttt{=f}, \texttt{+f}$) or an *attractee* ($\texttt{f}, \texttt{-f}$) and is either a *selection* feature ($\texttt{=f}, \texttt{f}$) or a *licensing* feature ($\texttt{+f}, \texttt{-f}$). A feature bundle is a complex structured category, as in categorial grammar, although feature bundles are structured as unary trees (strings).

A minimalist grammar defines a set of minimalist expressions, which are finite sequences of *chains*. A chain $\phi = \langle w, \delta \rangle$ is a pair of a string (over Σ) and a feature bundle. The expressions E derivable by a minimalist grammar, written $\vdash E$, are neatly presented in terms of a set of inference rules. In the rules below, it is assumed that δ is non-empty.

$$\frac{\ell \in \mathsf{Lex}}{\vdash \ell}\,\textbf{Select}_{\ell}$$

[5] A list $a_1, \ldots, a_n$ has a head a_1 and a tail $a_2, \ldots, a_n$, and can be thought of as a unary branching tree. It can be implemented as a pair, whose first element is the head of the list, and whose second element is the tail of the list. A finite sequence is a (flat) tuple of elements. Despite being structurally distinct, I systematically confuse lists and sequences notationally throughout this paper.

$$\frac{\vdash \langle u, \texttt{=x}\gamma\rangle, \phi_1, \ldots, \phi_i \qquad \vdash \langle v, \texttt{x}\rangle, \psi_1, \ldots, \psi_j}{\vdash \langle uv, \gamma\rangle, \phi_1, \ldots, \phi_i, \psi_1, \ldots, \psi_j}\ \textbf{Merge}_1$$

$$\frac{\vdash \langle u, \texttt{=x}\gamma\rangle, \phi_1, \ldots, \phi_i \qquad \vdash \langle v, \texttt{x}\delta\rangle, \psi_1, \ldots, \psi_j}{\vdash \langle u, \gamma\rangle, \phi_1, \ldots, \phi_i, \langle v, \delta\rangle, \psi_1, \ldots, \psi_j}\ \textbf{Merge}_2$$

$$\frac{\vdash \langle u, \texttt{+x}\gamma\rangle, \phi_1, \ldots, \phi_{i-1}, \langle v, \texttt{-x}\rangle, \phi_{i+1}, \ldots, \phi_j}{\vdash \langle vu, \gamma\rangle, \phi_1, \ldots, \phi_{i-1}, \phi_{i+1}, \ldots, \phi_j}\ \textbf{Move}_1$$

$$\frac{\vdash \langle u, \texttt{+x}\gamma\rangle, \phi_1, \ldots, \phi_{i-1}, \langle v, \texttt{-x}\delta\rangle, \phi_{i+1}, \ldots, \phi_j}{\vdash \langle vu, \gamma\rangle, \phi_1, \ldots, \phi_{i-1}, \langle v, \delta\rangle, \phi_{i+1}, \ldots, \phi_j}\ \textbf{Move}_2$$

The language of a minimalist grammar G at a feature f is defined to be

$$L(G, f) := \{w : \vdash \langle w, \texttt{f}\rangle\}$$

This version of minimalist grammars was shown in Salvati [2011] to be extremely complex: its membership problem is reducible to provability in multiplicative exponential linear logic, which, though decidable [Bimbó, 2015], is at least ExpSpace-hard [Lincoln, 1995]. A polynomial membership problem is obtained by imposing the following well-formedness constraint, called the **SMC** [Stabler, 1997], on the inputs and outputs of the operations defined previously.[6]

$$\frac{f_1, \ldots, f_i \text{ are pairwise distinct}}{\textbf{SMC}\ (\langle w_1, f_1\delta_1\rangle, \ldots, \langle w_i, f_i\delta_i\rangle)}$$

[6] Salvati [2011] shows that it is simply keeping track of unboundedly many unchecked features which is the source of the computational difficulty. The **SMC** relieves the computational system of this burden by imposing an upper bound on the number of unchecked features in any derivable expression. As there are finitely many features, any expression satisfying **SMC** has length of at most k, where $k = |\mathsf{Feat}|$. As the length of a feature bundle never increases after any rule application, this places an upper bound of kn on the number of unchecked features in a derivable expression satisfying **SMC**, where k is as before, and $n = \max(\{|\delta| : \langle w, \delta\rangle \in \mathsf{Lex}\})$. Any reasonable way of imposing a finite bound on the number of unchecked features in expressions leads to a similarly restricted system.

The **SMC** holds of an expression if all of its chains have distinct first features. When the **SMC** is imposed, the operations defined previously become partial, as, even when restricted to inputs satisfying **SMC**, their outputs might not. From this point on, I will assume that all expressions satisfy **SMC**, and thus that the operations have been so restricted, without explicit mention.

2.2 Derivational structure

I define a *category* $c = \delta_1, \ldots, \delta_i$ to be a sequence of feature bundles. Given an expression $e = \phi_1, \ldots, \phi_i$, its category $\texttt{cat}(e) := \pi_2\phi_1, \ldots, \pi_2\phi_i$ is the sequence of the same length which has at each position the feature bundle of the chain at the same position in e. The symbol π_i denotes the i^{th} projection function: the function which maps a tuple to its i^{th} component.

Whether an operation may apply to a sequence of expressions $e_1, \ldots, e_i$ is completely determined by the sequence of their categories $\texttt{cat}(e_1), \ldots, \texttt{cat}(e_i)$, as is the category of the result. For **op** an operation on minimalist expressions, I write $\texttt{cat}(\mathbf{op})$ for the relation between categories obtained by extracting the categories of the expressions in the graph of **op**. The previous observation amounts to saying that $\texttt{cat}(\mathbf{op})$ is functional (although partial), for each $\mathbf{op} \in \{\mathbf{Select}_\ell, \mathbf{Merge}_i, \mathbf{Move}_i : 1 \leq i \leq 2, \ell \in \mathsf{Lex}\}$.

One observes, by inspecting the definitions of the operations, that each feature bundle in an expression either remains the same, or decreases in size (by losing its head) as it undergoes an operation. From this one derives the consequence that there are *only finitely many* categories which can be involved in any proof of a derivable expression [Michaelis, 2001].

Given a lexicon `Lex`, the set of *derivation trees* over `Lex` is the smallest set $\mathtt{Der}(\mathsf{Lex})$ closed under the following operations.

$$\frac{\ell \in \mathsf{Lex}}{\mathbf{Select}_\ell \in \mathtt{Der}(\mathsf{Lex})} \qquad \frac{t \in \mathtt{Der}(\mathsf{Lex}) \qquad 1 \leq i \leq 2}{\mathbf{Move}_i(t) \in \mathtt{Der}(\mathsf{Lex})}$$

$$\frac{t_1 \in \mathtt{Der}(\mathsf{Lex}) \qquad t_2 \in \mathtt{Der}(\mathsf{Lex}) \qquad 1 \leq i \leq 2}{\mathbf{Merge}_i(t_1, t_2) \in \mathtt{Der}(\mathsf{Lex})}$$

A derivation tree is *well-formed* just in case it describes a derivation of an expression. Whether or not a derivation tree is well-formed can be determined by a simple bottom-up procedure: replace each subtree $t = \mathbf{op}(c_1, \ldots, c_n)$ with the category $\mathtt{cat}(\mathbf{op})(c_1, \ldots, c_n)$, beginning with the leaves (necessarily of the form $\mathbf{Select}_\ell$, where $\mathtt{cat}(\mathbf{Select}_\ell)() = \pi_2\ell$). A derivation tree t is well-formed iff it is replaced with a category c by this procedure.[7] As noted in Kobele et al. [2007], this coïncides with the definition of a *bottom-up tree automaton* [Comon et al., 2002], whence the set of well-formed derivation trees for a given minimalist grammar is *regular*. This is in stark contrast to the *non-regularity* of the string languages of minimalist grammars. As a derivation tree completely determines the expression it is the derivation of, the problem of parsing reduces to finding a derivation tree underlying an input string; being regular, the space of derivation trees has a simpler structure than the space of well-formed expressions.

2.3 Ellipsis

A common perspective on ellipsis is that elliptical sentences are derived with nothing in the 'ellipsis site.'[8] In analyses in the tradition of transformational

[7] A tree t fails to be well-formed if it contains a subtree $t' = \mathbf{op}(c_1, \ldots, c_n)$ for which $c_1, \ldots, c_n$ is not in the domain of the operation $\mathtt{cat}(\mathbf{op})$.

[8] For a concise overview of ellipsis see Kobele [2012b].

grammar, there are many constructions in which expressions which appear in the surface string would normally be analyzed as having been moved out from inside of the ellipsis site. This tradition adopts a different perspective on ellipsis—elliptical sentences are derived with the full syntactic structure of a non-elided constitutent in the ellipsis site. This of course allows expressions to move out from inside of the ellipsis site. The fact that ellipsis sites are not pronounced is dealt with by means of an operation which phonologically deletes the remaining material inside of the ellipsis site.[9] Kobele [2015] proposes to treat ellipsis as a family of grammatical operations; intuitively, one for each way of building up, and then deleting, an ellipsis site. To define the behaviour of these operations, I first extend the notion of derivation to allow for hypotheses. Intuitively, I will extend the derivational system to allow expressions to be constructed which are missing one or more constituents (represented by hypotheses). Building up an expression, and silencing all of it except for some moving pieces, is reconceptualized as building an expression which is missing some parts, and silencing all of it. The resulting expression can be thought of as specifying a way of changing the features, and rearranging the phonetic components, of its missing parts; i.e. an operation on expressions. Even though there are (in general) infinitely many ways to construct an expression which is missing a fixed array of parts, these boil down to only finitely many distinct operations. An ellipsis operation will be seen as realizing the effect of this construction and deletion process, without actually needing to construct or delete anything. The use of hypothetical derivations is solely at the meta-level, to define the behaviour of ellipsis oper-

[9] See Merchant [2001] for a modern exposition of this theory.

ations; I am not here enriching (or proposing to enrich) minimalist grammars with hypothetical reasoning .

2.3.1 Hypothetical Derivations

A hypothetical expression of category $c = \delta_1, \ldots, \delta_i$ is of the form $\mathsf{hyp}(c) = \langle \mathbf{x}_1, \delta_1\rangle, \ldots, \langle \mathbf{x}_i, \delta_i\rangle$, where $\mathbf{x}_1, \ldots, \mathbf{x}_i$ are pairwise distinct *variables.* I will write $\mathbf{x} : \mathsf{hyp}(c)$ to indicate that the variables used in $\mathsf{hyp}(c)$ will be referred to as $\mathbf{x}_1, \ldots, \mathbf{x}_i$. Derivability is extended from a property of expressions ($\vdash e$) to a relation between hypothetical expressions and expressions ($\Gamma \vdash e$) in the following way.

$$\frac{\ell \in \mathsf{Lex}}{\vdash \ell}\,\textbf{Select}_\ell \qquad\qquad \frac{}{\mathbf{x} : \mathsf{hyp}(c) \vdash \mathsf{hyp}(c)}\,\textbf{Axiom}_c$$

$$\frac{\Gamma \vdash \langle u, \texttt{=x}\gamma\rangle, \phi_1, \ldots, \phi_i \qquad \Delta \vdash \langle v, \texttt{x}\rangle, \psi_1, \ldots, \psi_j}{\Gamma, \Delta \vdash \langle uv, \gamma\rangle, \phi_1, \ldots, \phi_i, \psi_1, \ldots, \psi_j}\,\textbf{Merge}_1$$

$$\frac{\Gamma \vdash \langle u, \texttt{=x}\gamma\rangle, \phi_1, \ldots, \phi_i \qquad \Delta \vdash \langle v, \texttt{x}\delta\rangle, \psi_1, \ldots, \psi_j}{\Gamma, \Delta \vdash \langle u, \gamma\rangle, \phi_1, \ldots, \phi_i, \langle v, \delta\rangle, \psi_1, \ldots, \psi_j}\,\textbf{Merge}_2$$

$$\frac{\Gamma \vdash \langle u, \texttt{+x}\gamma\rangle, \phi_1, \ldots, \phi_{i-1}, \langle v, \texttt{-x}\rangle, \phi_{i+1}, \ldots, \phi_j}{\Gamma \vdash \langle vu, \gamma\rangle, \phi_1, \ldots, \phi_{i-1}, \phi_{i+1}, \ldots, \phi_j}\,\textbf{Move}_1$$

$$\frac{\Gamma \vdash \langle u, \texttt{+x}\gamma\rangle, \phi_1, \ldots, \phi_{i-1}, \langle v, \texttt{-x}\delta\rangle, \phi_{i+1}, \ldots, \phi_j}{\Gamma \vdash \langle vu, \gamma\rangle, \phi_1, \ldots, \phi_{i-1}, \langle v, \delta\rangle, \phi_{i+1}, \ldots, \phi_j}\,\textbf{Move}_2$$

The category of a hypothetical expression $E = \Gamma \vdash e$ is $\texttt{cat}(E) := c_1 \times \cdots \times c_n \rightarrow \texttt{cat}(e)$, for $\Gamma = \mathsf{hyp}(c_1), \ldots \mathsf{hyp}(c_n)$.[10] The notion of a derivation tree is extended so as to include leaves labeled with $\textbf{Axiom}_c$.

As is standard, we require that hypothetical contexts have at most one hypothesis associated with a particular variable. The use of variables in this

[10] When $\Gamma = \emptyset$ I write simply $\texttt{cat}(e)$ instead of $\mathbf{1} \rightarrow \texttt{cat}(e)$, where $\mathbf{1}$ is the unit for ×.

system is intended to be *linear*, which means that variables on the left of the turnstile occur exactly once on its right. (This is implicitly assumed in the **Merge** rules.) I am implicitly assuming that each use of the **Axiom** rule introduces a *globally fresh* variable; in particular, when speaking of multiple derivation trees $t_1, \dots, t_n$, it is assumed that the expressions to which they evaluate have no variables in common.

Note that there is no rule of hypothesis discharge (i.e. no way of moving hypotheses across the turnstile); the role of hypotheses is simply to specify the behaviour of incomplete derivations (derivation *contexts* as opposed to derivation *trees*).

2.3.2 Ellipsis operations

Given two hypothetical expressions $E = \Gamma, \mathbf{x} : \mathsf{hyp}(c), \Gamma' \vdash e$ and $E' = \Delta \vdash e'$ such that $\mathtt{cat}(e') = c$, we define the *substitution* of E' for $\mathsf{hyp}(c)$ in E to be the expression $E[E'/\mathbf{x}] := \Gamma, \Delta, \Gamma' \vdash e[\mathbf{x}_1/\pi_1 {e'}_1 \dots, \mathbf{x}_i/\pi_1 {e'}_i]$, where $[\mathbf{x}_1/w_1 \dots, \mathbf{x}_i/w_i]$ is the *simultaneous substitution* of strings $w_1, \dots, w_i$ for variables $\mathbf{x}_1, \dots, \mathbf{x}_i$, extended over pairs $\langle w, \delta \rangle$, and sequences $\phi_1, \dots, \phi_n$ in the obvious way. Let $t[\mathbf{Axiom}_c]$ be a derivation tree with a designated leaf labeled with $\mathbf{Axiom}_c$, and let t' be a derivation tree. Then $t[t']$ is the derivation tree obtained from $t[\mathbf{Axiom}_c]$ by replacing the designated leaf with t'. If $t[\mathbf{Axiom}_c]$ evaluates to $E = \Gamma, \mathbf{x} : \mathsf{hyp}(c), \Gamma' \vdash e$ and t' evaluates to $E' = \Delta \vdash e'$ such that $\mathtt{cat}(e') = c$, then $t[t']$ evaluates to $E[E'/\mathbf{x}]$.

Finally, the *phonological deletion* of a hypothetical expression $E = \Gamma \vdash e$ is the expression $\mathtt{delete}(E) = \Gamma \vdash \mathtt{delete}(e)$, where $\mathtt{delete}$ maps strings over Σ^* to the empty string, and variables to themselves, and is extended in the obvious way over chains and expressions. Note that for any hypothetical category C, the set $\mathsf{Delete}_C := \{\mathtt{delete}(E) : \mathtt{cat}(E) = C\}$ is finite modulo renaming of variables—there are just finitely many ways of assigning the variables in the hypotheses to chains. Elements of the set $\bigcup_C \mathsf{Delete}_C$ will be called *deletion profiles*; note that a deletion profile θ is an element of exactly one Delete_C.

I write $\Gamma \vdash e : c$ to mean that $\Gamma \vdash e$ and that $\mathtt{cat}(e) = c$. To the basic operations of minimalist grammars I now add the following, for any categories $c, c_1, \ldots, c_n$ and $\theta \in \mathsf{Delete}_{(c_1 \times \cdots \times c_n) \to c}$.

$$\frac{\vdash e_1 : c_1 \qquad \cdots \qquad \vdash e_n : c_n}{\vdash \theta[\mathbf{x}_1/e_1 \ldots, \mathbf{x}_n/e_n] : c}\ \mathbf{e}_\theta$$

This family of operations faithfully implements the idea that ellipsis is a matter of deleting the phonological material in an expression previously derived. However, whereas phonological deletion must be conditioned on some often complex (even undecidable[11]) relation holding between an antcedent and an ellipsis site, there are no such conditions on the family e_θ; an operation e_θ may apply in a context-free way. The fact that the meaning of an elliptical sentence is somehow parasitic on some other linguistic expression in a discourse is accounted for by treating e_θ as semantically anaphoric; an ellipsis site e_θ requires a salient semantic antecedent *which is the meaning of a hypothetical derivation which has occurred in the surrounding discourse.* This property, described in more detail in section 2.3.3 and demonstrated in 3, can be efficiently tracked and updated during the processing of a discourse.

[11] Merchant [2001] proposes that *mutual entailment* must hold between (the existential closures of) the respective meanings of antecedent and ellipsis site.

Kobele [2015] defines (in effect) a minimalist grammar with ellipsis to be given by a finite set $\mathsf{Ellipsis} \subset \bigcup_{n\in\mathbb{N}} \bigcup_{c_1,\ldots c_n,c} \mathsf{Delete}((c_1 \times \cdots \times c_n \to c))$, in addition to `AtFeat` and `Lex`, which determines the operations e_θ usable by the grammar. He motivates this empirically by noting the (currently) brute fact that different languages seem to have different elliptical processes at their disposal; for example, German has no verb phrase ellipsis.

2.3.3 Interpreting ellipsis

The approach to ellipsis presented in section 2.3.2 takes the syntax of ellipsis to be quite straightforward, which in turn forces the semantics of ellipsis to be non-trivial.[12] Kobele [2015] adopts a *pro-form* theory of ellipsis interpretation, whereby the meaning contribution of an ellipsis operation $\mathbf{e}_\theta$ is to act as an anaphor, being resolved to a salient meaning in the discourse context. Not just any salient meaning is an appropriate antecedent however; the meaning must in addition be the meaning of a hypothetical derivation occuring in the discourse context of appropriate type.[13] This is stated for-

[12] This is in contrast to approaches, such as the PF-deletion approach, which make the syntactic distribution of ellipsis quite complicated (ellipsis may only obtain if the sentence in question stands in a computationally non-trivial relation with another sentence in the discourse context), while making the semantics of ellipsis trivial.

[13] Schachter [1977] points out, arguing against a claim made by Hankamer and Sag [1976], that there are perfectly well formed elliptical constructions with no overt linguistic antecedent. Miller and Pullum [2014] provides a summary of the literature, as well as a corpus informed study of so-called *exophoric* verb phrase ellipsis. In the context of the present theory, there is a straightforward, but not particularly compelling, way of treating cases of exophoric ellipsis. This is to allow that an always available default antecedent exists, something along the lines of *do it*. I would need to have some way of making this default antecedent dispreferred; perhaps the default can only be used in case nothing else is available. This strategy of allowing for certain always available antecedents, while not pretty, can also be used to describe other cases of mismatch between antecedent and ellipsis site, such as in the example below (from Hardt [1993]):

- Decorating for the holidays is easy, if you know how ~~to decorate for the holidays~~

mally below, where, for convenience, I write $[\![\mathbf{e}_\theta]\!]^D \mapsto M$ to indicate that M is a possible meaning for $\mathbf{e}_\theta$ in the discourse context D. I take D to be a finite set of derivation trees; a derivation tree d is in D just in case d is the chosen parse of a sentence in the discourse context.

$$[\![\mathbf{e}_\theta]\!]^D \mapsto M \text{ iff } \exists e, e', d_1, \ldots, d_n. \begin{array}{ll} 1. & e[e'[d_1, \ldots, d_n]] \in D \\ 2. & [\![e']\!] = M \\ 3. & \theta = \mathtt{delete}(e') \end{array}$$

2.3.4 An example

Consider the following simplified dialogue between two parties, **A** and **B**.

A Oskar might sleep.

B Carl will not.

I analyze sentences **A** and **B** in terms of the lexical items in table 1. In this table, and elsewhere in this section, I separate features in feature bundles with a period to enhance legibility. Sentences **A** and **B** have derivations d_A and d_B in figures 1 and 2 respectively. Note that in both sentences the subject of the clause moves to its surface position. In the derivation tree for sentence **B**, the ellipsis operation $\mathbf{e}_\theta$ is such that $\theta \in \mathsf{Delete}_{\mathtt{d.-k} \to \mathtt{v,-k}}$. The exact identity of θ will be specified later in this example.

Table 1: Lexical items

sleep :: `=d.v`	will :: `=v.+k.t`	might :: `=v.+k.t`
Oskar :: `d.-k`	Carl :: `d.-k`	not :: `=v.v`

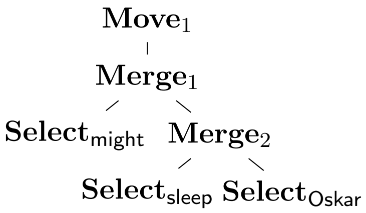

Figure 1: The derivation for **A**

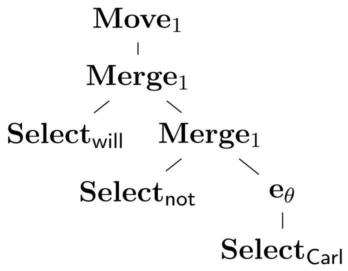

Figure 2: The derivation for **B**

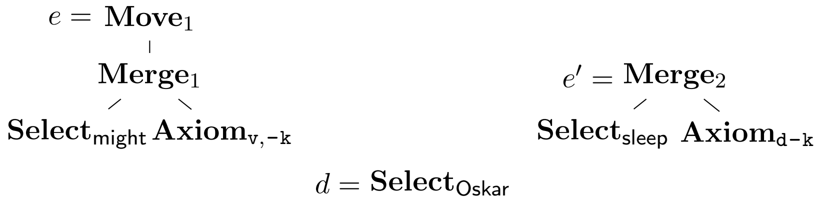

Figure 3: Breaking d_A into parts

Now assume that the discourse context D contains the derivation d_A for sentence **A**. This can be broken up into the three parts e, e', and d, shown in figure 3, such that $e[e'[d]] = d_A$. The hypothetical expression E' corresponding to the derivation e' is given below. Observe that $\texttt{cat}(E') = \texttt{d.-k} \to \texttt{v}, \texttt{-k}$ and that $\texttt{delete}(E') = x : (\langle x_1, \texttt{d.-k}\rangle) \vdash \langle \epsilon, \texttt{v}\rangle, \langle x_1, \texttt{-k}\rangle$.

$$\dfrac{\dfrac{\mathsf{sleep} :: \texttt{=d.v}}{\vdash \langle \mathsf{sleep}, \texttt{=d.v}\rangle}\,\textbf{Select} \qquad \dfrac{}{x : (\langle x_1, \texttt{d.-k}\rangle) \vdash \langle x_1, \texttt{d.-k}\rangle}\,\textbf{Axiom}}{x : (\langle x_1, \texttt{d.-k}\rangle) \vdash \langle \mathsf{sleep} : \texttt{v}\rangle, \langle x_1, \texttt{-k}\rangle}\,\textbf{Merge}_2$$

Returning to the issue of which $\mathbf{e}_\theta$ should be used in d_B, we choose $\theta = \texttt{delete}(E')$. We can see that d_B corresponds to the derivation below.

$$\dfrac{\dfrac{\dfrac{\mathsf{will} :: \mathtt{=v.+k.t}}{\vdash \langle \mathsf{will}, \mathtt{=v.+k.t}\rangle}\,\textbf{Select} \quad \dfrac{\dfrac{\mathsf{not} :: \mathtt{=v.v}}{\vdash \langle \mathsf{not}, \mathtt{=v.v}\rangle}\,\textbf{Select} \quad \dfrac{\dfrac{\mathsf{Carl} :: \mathtt{d.-k}}{\vdash \langle \mathsf{Carl}, \mathtt{d.-k}\rangle}\,\textbf{Select}}{\vdash \langle \epsilon, \mathtt{v}\rangle, \langle \mathsf{Carl}, \mathtt{-k}\rangle}\,\mathbf{e}_{\mathtt{Delete}(E')}}{\vdash \langle \mathsf{not} : \mathtt{v}\rangle, \langle \mathsf{Carl}, \mathtt{-k}\rangle}\,\textbf{Merge}_1}{\vdash \langle \mathsf{will\ not} : \mathtt{+k.t}\rangle, \langle \mathsf{Carl}, \mathtt{-k}\rangle}\,\textbf{Merge}_1}{\vdash \langle \mathsf{Carl\ will\ not} : \mathtt{t}\rangle}\,\textbf{Move}_1$$

3 The Complexity of Ellipsis Resolution

The problem of compreheding sentences involving ellipsis can be usefully broken up into three independent problems. The first is how to parse sentences (to their underlying structures) when the spate of grammatical operations includes ellipsis (section 3.1). The second is how the meaning of ellipsis sites (once postulated) is to be determined (section 3.2). The final problem is how to extend the previous discourse context with the new antecedents made available after parsing an additional sentence (section 3.3). The main result is that, in the context of the theory developed above, all of these problems are solvable in polynomial time. Or rather, common linguistic assumptions restrict the problem space in a way which makes linguistically possible solutions efficiently obtainable.

3.1 Parsing in the presence of ellipsis

Parsing will be here viewed as the problem of mapping a string to the set of its possible derivations given a grammar. As this set can be infinite, we must work with a finite representation thereof, the size of which should be related to the size of the input string in a reasonable way. Since Bar-Hillel et al. [1961], it is standard to view parsing as intersecting the input string (or more generally, a regular grammar) with the original context-free grammar. The derivations of the intersection grammar faithfully represent the derivations in the original grammar of the input string, and the size of the intersection grammar is a polynomial function of the sizes of the original grammar and of the input string. This perspective can be generalized to richer formal systems in a natural way [Kanazawa, 2007].

Sound and complete chart-parsing algorithms for minimalist grammars [Harkema, 2001] essentially compute the intersection of the regular grammar of minimalist derivations with the input string. In the context of a finite number of ellipsis operations $\mathbf{e}_\theta$, the derivation tree language remains regular, whence the standard parsing techniques apply without change. Although the problem of sound and complete parsing in the presence of ellipsis is therefore a trivial extension of previous results, it depends on the *linguistic assumption* below.

Linguistic Assumption 1. There are a fixed, finite number of ellipsis operations available in a given language.

3.2 Resolving ellipsis

Ellipsis resolution will be viewed as the problem of, given a *single* parse tree t, and a discourse context D, coming up with *one* way of interpreting the ellipsis operations in t given D, if one exists, and announcing failure, if none exist. A more realistic account of the resolution problem would take into account some metric of the *plausibility* of an antecedent.

I assume that a discourse context acts as a function mapping deletion profiles to lists of meanings of hypothetical expressions of that category.[14] As

[14] I show in section 3.3 how this assumption about the discourse context can be ensured

given Linguistic Assumption 1, there are finitely many ellipsis operations $\mathbf{e}_\theta$, a discourse context can be implemented as a finite map, and thus lookup can be done in time on the order of $\log n$, for n the number of ellipsis operations in the grammar.

To resolve an ellipsis site $\mathbf{e}_\theta$, simply take the head of the list returned at position $D(\theta)$, if it exists, and announce failure, otherwise.

There is a potential difficulty that I will now describe but not resolve (I return briefly to this in the conclusion). If the antecedent to an ellipsis site contains itself an unresolved ellipsis site, then there are pathological choices of resolutions which could lead to an infinite regress, thus dashing hopes of a polynomial time resolution strategy. I do not see a simple way of avoiding this problem, while permitting antecedents to contain unresolved ellipsis sites. Accordingly, I impose an *ad hoc* restriction on the number of resolution steps that may be undertaken in attempting to resolve a particular ellipsis site. A bound greater than 0 has the consequence that the entire list of possible antecedents for an ellipsis site may have to be explored to determine whether it can be resolved in the discourse context.

Linguistic Assumption 2. Antecedents may not contain unresolved ellipsis sites.

Linguistic Assumption 2 is, in contrast to the other lingusitic assumptions in this paper, *not* widely accepted. Indeed, recursive ellipsis resolution figures prominently in some analyses of sloppy VPE readings [Tomioka, 2008], as exemplified in figure 4, which is intended to be understood as meaning

during discourse processing.

the same as "*when I tried to kiss you, you didn't want me to kiss you, but when I tried to hug you, you did want me to hug you.*"

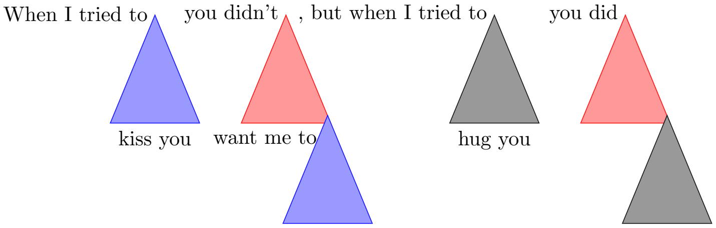

Figure 4: Sloppy VPE

Under Linguistic Assumption 2, the final "*you did*" of the sentence in the figure should be derived with *two* ellipsis operations, one (which should be resolved to *want me to*) taking the other (which should be resolved to *hug you*) as its argument.

3.3 Updating the discourse context

The main difficulty in parsing elliptical sentences to meanings in discourse revolves around updating the discourse context so as to ensure that ellipsis resolution can be done simply. In this section I show that this is possible to do efficiently, given certain otherwise motivated linguistic assumptions. The main result is the following:

Theorem 1. We can enumerate *all possible antecedents* in a derivation of w in time *polynomial* in $|w|$.

An antecedent is a hypothetical derivation E, and computing $\texttt{delete}(E)$ and inserting $[\![E]\!]$ in the antecedent database at the appropriate position is also efficient. Thus, Theorem 1 guarantees that discourse context update is efficiently computable in the length of the input sentence.

It is easy to see that the number of n-ary contexts in a tree t is bounded by $|t|^{n+1}$.[15] Thus, given Linguistic Assumption 1, the number of possible antecedents in a derivation tree d is polynomial in *the size of d*. There is, however, no bound on the size of a tree given its yield.[16] To demonstrate Theorem 1, I reduce first the dependency on $|t|$ to a dependency on the number of *leaves* of t, and then reduce this further to the number of *pronounced* leaves.

3.3.1 Eliding maximal projections

In a minimalist grammar *without* ellipsis, each leaf in the derivation tree (corresponding to a particular lexical item $\ell = \langle w, \delta \rangle$ uniquely determines the sequence of nodes dominating it, of which it is on a left branching path.[17] The length of this path is bounded by the number of features in the lexical item, and thus the ratio between the number of nodes and number of leaves in the derivation tree is bounded by a constant $c \leq |\delta|$ where δ is the largest feature bundle in the lexicon.

That the lexical items in a derivation uniquely determine it [Hale and Stabler, 2005] has been exploited (in [Graf, 2011, Kobele, 2011, Salvati, 2011]) to give an alternative, tree-adjoining grammar-like, representation of mini-

[15] One chooses one node of t to be the top of the context, and then n additional nodes to be the holes (and then $\binom{m}{n} \leq m^n$) Although this is clearly a very loose bound, a tighter one is not necessary for the purposes of this paper.

[16] This is due both to the possibility of unary branching, and of silent leaves.

[17] Speaking in terms of Gorn addresses, for any lexical item ℓ, there is some number n such that any leaf in a well-formed derivation labeled with Select_ℓ has Gorn address $u0^n$, where u is either empty or ends with a 1. Furthermore, the labels of all nodes with addresses $u0^m$, for $m \leq n$ are determined by ℓ.

malist derivations. Here, all nodes of the tree are labeled with lexical items ℓ (for various choices of ℓ). The *rank* of a lexical item $\ell = \langle w, \delta \rangle$ is the number of selector features (=x) in its feature bundle δ. This alternative representation equates the size of the tree with the number of lexical items it contains. Using this alternative representation for determining antecedents involves the following substantive (but uncontroversial) linguistic assumptions.

Linguistic Assumption 3. All ellipsis is lexicalized.

Linguistic Assumption 4. Only maximal projections can be elided.

Linguistic Assumption 3 means that each antecedent must contain at least one lexical item; concretely, $\mathsf{Move}(\mathsf{Assume}_C)$ is not a legitimate antecedent. Although Linguistic Assumption 4 is incompatible with early transformational analyses of so-called N-bar deletion, it *is* compatible with current theories and analyses, and seems to be, at least implicitly, uniformly assumed.

3.3.2 Bounding Ellipsis

The discussion in the previous section establishes that, without ellipsis, the size of derivation trees is a constant (depending on the grammar) factor of the number of lexical items used in the derivation. *With* ellipsis, however, there is no such bound; an ellipsis operation might map an expression of one category to another of the same (giving rise to a *cyclic* derivation). In analyses given by linguists, such a cyclic ellipsis configuration never arises. Lobeck [1995] claims that each ellipsis site must be *governed* by a particular head. This can be translated into the terminology of the paper as per the below.

Linguistic Assumption 5. Each ellipsis operation must be associated with a (unique) lexical item.

As a corollary of Linguistic Assumption 5, the ratio of ellipsis operations to lexical items in a well-formed derivation tree is bounded by a constant $e = 2$, and the number of possible n-ary antecedents is bounded by $((e + 1) \times |\mathsf{leaves}(t)|)^{n+1}$.

An alternative (stronger) way of formulating Linguistic Assumption 5 would be to require that every ellipsis operation be immediately dominated by some lexical item (in the tree-adjoining grammar-like representation of derivations).[18] Such a condition would appear to be incompatible with the analysis of sloppy VPE discussed in section 3.2, however.[19]

3.3.3 Stopping silence

The linguistic assumptions made thus far have brought the number of possible antecedents in a given derivation down to a polynomial of the number of leaves of that derivation. However, there is, given the possibility of silent lexical items, no connection between the number of words in a sentence and the number of lexical items used in its derivation.

While linguists in the transformational tradition postulate a great deal of silent lexical items, an analysis which allowed for structures of arbitrary size

[18] This stronger formulation is closer to what is intended by Lobeck [1995].

[19] It needn't be actually incompatible, so long as one were willing to posit an otherwise unmotivated silent lexical item in between the two ellipsis sites.

to be populated solely with silent lexical items would be considered bizarre. I instantiate this as a final linguistic assumption.

Linguistic Assumption 6. The ratio of silent lexical items to overt lexical items in a derivation is bounded by a constant k.

With this, there are at most $((e+1) \times (k+1) \times |w|)^{n+1}$ possible n-ary antecedents in a derivation tree for w, which is a polynomial function in w, as e, k, and n are constants fixed by the grammar.

Corollary 1. There are $\mathcal{O}(|w|)$ possible antecedents in a derivation tree of w.

4 Conclusion

I have shown that the theory of ellipsis in Kobele [2015] can be implemented efficiently in a parsing algorithm. In particular, maintaining and updating a discourse context sufficient to permit the resolution of ellipsis can be done efficiently during parsing *in the size of the to-be parsed input string.*

Although much of the discussion in section 3.2 (culminating in Linguistic Assumption 2) revolved around blocking pathological resolution dependencies, further study is necessary. In particular, such pathological dependencies do not seem to arise if the discourse context is populated with antecedents obtained during parsing. (Although some care is needed to ensure that a terminating resolution sequence is found quickly.) A weaker alternative to Linguistic Assumption 2, which would admit Tomioka's analysis of sloppy VPE, is to disallow an antecedent from being used more than once in resolving ellipsis sites (stemming from a particular one). While I worried in that section about the unbounded number of potential ellipsis sites, it seems that not all potentially possible antecedents are actually possible for a given ellipsis site; if there were some fixed upper bound on the number of possible antecedents which could be used at any given time, the efficiency results would still hold. This is provided, of course, that the process of keeping track of the accessibility of antecedents could itself be done efficiently.

In order to use ellipsis resolution to influence the parser's online decisions, a natural idea is to include relevant information about antecedents in the discourse context. This information might include identities of lexical items (for topic modeling, or lexical priming effects), or the identites of the peripheral pronounced words (for bigram transition probabilities), etc. Of course, should this information become infinite, we would come into conflict with Linguistic Assumption 1, and with it could go the efficiency of parsing.

In this vein, it would be interesting to consider in more detail the actual implementation of a parser for minimalist grammars with ellipsis. In particular, various seemingly *ad hoc* properties of the formalism may appear different from the perspective of an on-line predictive parsing algorithm.

References

Y. Bar-Hillel, M. Perles, and E. Shamir. On formal properties of simple phrase structure grammars. *Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung*, 14(2):143–172, 1961.

K. Bimbó. The decidability of the intensional fragment of classical linear logic. *Theoretical Computer Science*, 597:1–17, 2015.

N. Chomsky. *The Minimalist Program.* MIT Press, Cambridge, Massachusetts, 1995.

N. Chomsky. Minimalist inquiries: The framework. In R. Martin, D. Michaels, and J. Uriagereka, editors, *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*, pages 89–155. MIT Press, Cambridge, Massachusetts, 2000.

H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available at `http://www.grappa.univ-lille3.fr/tata`, 2002.

P. de Groote. Towards abstract categorial grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, pages 148–155, 2001.

P. de Groote and S. Pogodalla. On the expressive power of Abstract Categorial Grammars: Representing Context-Free formalisms. *Journal of Logic, Language and Information*, 13(4):421–438, 2004.

T. Graf. Closure properties of minimalist derivation tree languages. In S. Pogodalla and J.-P. Prost, editors, *LACL 2011*, volume 6736 of *Lecture Notes in Artificial Intelligence*, pages 96–111, 2011.

J. T. Hale and E. P. Stabler. Strict deterministic aspects of minimalist grammars. In P. Blache, E. P. Stabler, J. Busquets, and R. Moot, editors, *Logical Aspects of Computational Linguistics*, volume 3492 of *Lecture Notes in Computer Science*, chapter 11, pages 162–176. Springer, 2005.

J. Hankamer and I. A. Sag. Deep and surface anaphora. *Linguistic Inquiry*, 7(3):391–428, 1976.

D. Hardt. *Verb Phrase Ellipsis: Form, Meaning, and Processing*. PhD thesis, University of Pennsylvania, 1993.

H. Harkema. *Parsing Minimalist Languages*. PhD thesis, University of California, Los Angeles, 2001.

M. Kanazawa. Parsing and generation as datalog queries. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 176–183, Prague, 2007. Association for Computational Linguistics.

G. M. Kobele. Minimalist tree languages are closed under intersection with recognizable tree languages. In S. Pogodalla and J.-P. Prost, editors, *LACL 2011*, volume 6736 of *Lecture Notes in Artificial Intelligence*, pages 129–144, 2011.

G. M. Kobele. Eliding the derivation: A minimalist formalization of ellipsis. In S. Müller, editor, *Proceedings of the 19th International Conference on Head-Driven Phrase Structure Grammar, Chungnam National University Daejeon*, pages 307–324, Stanford, 2012a. CSLI Publications.

G. M. Kobele. Ellipsis: computation of. *WIREs Cognitive Science*, 3(3): 411–418, 2012b.

G. M. Kobele. LF-copying without LF. *Lingua*, 166, part B:236–259, 2015.

G. M. Kobele and J. Michaelis. Two type 0-variants of minimalist grammars. In G. Jäger, P. Monachesi, G. Penn, J. Rogers, and S. Wintner, editors, *Proceedings of the 10th conference on Formal Grammar and the 9th Meeting on Mathematics of Language*, Edinburgh, 2005.

G. M. Kobele, C. Retoré, and S. Salvati. An automata theoretic approach to minimalism. In J. Rogers and S. Kepser, editors, *Proceedings of the Workshop Model-Theoretic Syntax at 10; ESSLLI '07*, Dublin, 2007.

R. B. Lees. *The grammar of English nominalizations.* Mouton, The Hague, 1960.

P. Lincoln. Deciding provability of linear logic formulas. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Note Series*, pages 109–122. Cambridge University Press, Cambridge, UK, 1995.

A. Lobeck. *Ellipsis: Functional Heads, Licensing, and Identification.* Oxford University Press, New York, 1995.

J. Merchant. *The Syntax of Silence: Sluicing, Islands, and the Theory of Ellipsis*, volume 1 of *Oxford Studies in Theoretical Linguistics.* Oxford University Press, New York, 2001.

J. Michaelis. Derivational minimalism is mildly context-sensitive. In M. Moortgat, editor, *Logical Aspects of Computational Linguistics, (LACL '98)*, volume 2014 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, Berlin, Heidelberg, Germany, 1998.

J. Michaelis. *On Formal Properties of Minimalist Grammars.* PhD thesis, Universität Potsdam, 2001.

P. H. Miller and G. K. Pullum. Exophoric VP ellipsis. In P. Hofmeister and E. Norcliffe, editors, *The Core and the Periphery: Data-Driven Perspectives on Syntax*, pages 5–32. CSLI Publications, Stanford, 2014.

S. Salvati. Minimalist grammars in the light of logic. In S. Pogodalla, M. Quatrini, and C. Retoré, editors, *Logic and Grammar: Essays Dedicated to Alain Lecomte on the Occasion of his 60th Birthday*, volume 6700 of *Lecture Notes in Artificial Intelligence*, pages 81–117. Springer, 2011.

P. Schachter. Does she or doesn't she? *Linguistic Inquiry*, 8(4):763–767, 1977.

H. Seki, T. Matsumura, M. Fujii, and T. Kasami. On multiple context-free grammars. *Theoretical Computer Science*, 88:191–229, 1991.

E. P. Stabler. Derivational minimalism. In C. Retoré, editor, *Logical Aspects of Computational Linguistics*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer-Verlag, Berlin, 1997.

E. P. Stabler and E. L. Keenan. Structural similarity within and among languages. *Theoretical Computer Science*, 293:345–363, 2003.

S. Tomioka. A step-by-step guide to ellipsis resolution. In K. Johnson, editor, *Topics in Ellipsis*, chapter 9, pages 210–228. Cambridge University Press, 2008.