June 1991

Generation and Synchronous Tree-Adjoining Grammars

Stuart M. Shieber  
Harvard University  

Yves Schabes  
University of Pennsylvania  

Follow this and additional works at: http://repository.upenn.edu/cis_reports
Abstract
The recent history of grammar reversing can be viewed as an effort to recover seine notion of semantic locality on which to base a generation process. For instance, Wedekind (1988) requires a property of a grammar that he refers to as connectedness, which specifies that complements be semantically connected to their head. Shieber (1981) defines a notion of semantic monotonicity, a kind of compositionality property that guarantees that it can be locally determined whether phrases can contribute to forming an expression with a given meaning. Generation schemes that reorder top-down generation (Dymetman and Isabelle, 1988; Strzalkowski, 1989) so as to make available information that well-founds the top-down recursion also fall into the mold of localizing semantic information. Semantic-head-driven generation (Shieber et al., forthcoming; Calder et al., 1989) uses semantic heads and their complements as a locus of semantic locality.

Comments

This technical report is available at ScholarlyCommons: http://repository.upenn.edu/cis_reports/412
Generation and Synchronous Tree-Adjoining Grammars

MS-CIS-91-42
LINC LAB 202

Stuart M. Shieber
(University of Harvard)

Yves Schabes
(University of Pennsylvania)

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104-6389

May 1991

To appear in Computational Intelligence, 1991
Generation and
Synchronous Tree-Adjoining Grammars

Stuart M. Shieber
Aiken Computation Laboratory
Division of Applied Sciences
Harvard University
Cambridge, MA 02138

Yves Schabes
Department of Computer and
Information Science
University of Pennsylvania
Philadelphia, PA 19104
1 Introduction

The recent history of grammar reversing can be viewed as an effort to recover some notion of semantic locality on which to base a generation process. For instance, Wedekind (1988) requires a property of a grammar that he refers to as connectedness, which specifies that complements be semantically connected to their head. Shieber (1988) defines a notion of semantic monotonicity, a kind of compositionality property that guarantees that it can be locally determined whether phrases can contribute to forming an expression with a given meaning. Generation schemes that reorder top-down generation (Dymetman and Isabelle, 1988; Strzalkowski, 1989) so as to make available information that well-founds the top-down recursion also fall into the mold of localizing semantic information. Semantic-head-driven generation (Shieber et al., forthcoming; Calder et al., 1989) uses semantic heads and their complements as a locus of semantic locality. Joshi (1987) points out that tree-adjoining grammars (TAG) may be an especially appropriate formalism for generation because of their syntactic locality properties, which, intuitively at least, ought to correlate with some notion of semantic locality. The same observation runs as an undercurrent in the work of McDonald and Pustejovsky (1985), who apply TAGs to the task of generation. As these researchers note, the properties of TAGs for describing the syntactic structuring of a natural language mesh quite naturally with the requirements of natural-language generation. Nonetheless, generation is not, as typically viewed, a problem in natural-language syntax. Any system that attempts to use the TAG formalism as a substrate upon which to build a generation component must devise some mechanism by which a TAG can articulate appropriately with semantic
information. In this paper, we discuss one such mechanism, synchronous TAGs, which we have previously proposed in the arena of semantic interpretation and automatic translation, and examine how it might underlie a generation system of the sort proposed by Joshi and McDonald and Pustejovsky. In particular, synchronous TAGs allow for a precise notion of semantic locality corresponding to the syntactic locality of pure TAGs.

2 Scope of the Paper

The portion of the full-blown generation problem that we address here is what has been referred to as the tactical as opposed to the strategic generation problem (Thompson, 1977). That is, we are concerned only with how to compute instances of a well-defined relation between strings and canonical logical forms\(^1\) in the direction from logical forms to strings, a problem that is sometimes referred to as “reversing” a grammar. This aspect of the generation problem, which ignores the crucial issues in determining what content to communicate, what predicates to use in the communication, and so forth, can be seen as the reverse of the problem of parsing natural language to derive a semantic representation. The separation of generation into tactical and strategic components is a part of many, if not most, natural-language generation systems. McKeown (1985, Chapter 6) provides an excellent overview of previous research in tactical generation and the relation to strategic generation. The citations in the first paragraph can also serve to place the issue in its research context. The other truly difficult issues of general natural-language production are well beyond the scope of this paper, but

\(^1\)This issue of canonicality of logical forms is discussed by Shieber (1988).
we return to the issue of how a synchronous-TAG tactical component might fit into a full natural-language-production system in Section 7.

3 Semantics in Generation

Although Joshi discusses at length the properties of TAGs advantageous to the generation task (1987), he does not address the issue of characterizing a semantic representation off of which generation can proceed. McDonald and Pustejovsky do mention this issue. Because TAGs break up complex syntactic structures into elementary structures in a particular way, their semantic representation follows this structuring by breaking up the logical form into corresponding parts. McDonald and Pustejovsky consider the sentence

(1) How many ships did Reuters report that Iraq had said it attacked?

Its semantic representation follows the decomposition of the sentence into its elementary TAG trees—corresponding (roughly) to “How many ships ... it attacked”, “did Reuters report that ...”, “Iraq had said ...”. McDonald and Pustejovsky describe their semantic representation: “The representation we use ... amounts to breaking up the logical expression into individual units and allowing them to include references to each other.” The units for the example at hand would be:

\[
U_1 = \lambda(quantity-of-ships).
    \text{attack}(\text{Iraq}, quantity-of-ships)
\]
\[
U_2 = \text{say}(\text{Iraq}, U_1)
\]
\[
U_3 = \text{report}(\text{Reuters}, U_2)
\]
By composing the units using substitution of equals for equals, a more conventional logical form representation is revealed:

\[
\begin{align*}
\text{(2) } & \quad \text{report(Reuters,} \\
& \quad \quad \text{say(Iraq,} \\
& \quad \quad \quad \lambda(\text{quantity-of-ships).} \\
& \quad \quad \quad \text{attack(Iraq,quantity-of-ships}))}
\end{align*}
\]

The full logical form that is being realized is thus composed from more primitive units that are appropriate for separate linguistic realization. Three problems present themselves with respect to the composition (or, conversely, decomposition) of the logical form: How is the decomposition determined? What composition operations are possible? Where are compositions performed?

**How is the decomposition determined?**

The simplest scheme for determining the decomposition is that chosen by McDonald and Pustejovsky: the particular decomposition of the full semantic form must be explicitly specified as part of the input to the generation system. With any other scheme, some method of “parsing” the semantic representation into subparts is needed. Although McDonald and Pustejovsky do not provide such a parsing method, we examine two possibilities here (besides that provided by synchronous TAGs).

A natural, but too restrictive, proposal is to assume that each atom in the logical form—each predicate or constant, say—is a single, linguistically realizable unit. That is, the tree representation of the logical form is broken up into units corresponding to a node and its immediate children, as in Figure 1. This method severely and unduly restricts the semantic representations that can be associated
Figure 1: A simple, but too restrictive, method of “parsing” a logical form into linguistically realizable units is to break up local sets of nodes in the tree representation.

with linguistic constructs, since a construct may not map to a constellation of semantic atoms. For instance, an agentless passive could not be represented with overt existential quantification over the missing agent position; this would associate a tree of depth two with a single linguistic construct. Similarly, an analysis of the transitive verb want would be disallowed in which its semantics includes an implicit relation of possession that can be independently modified (as in the sentence “Reuters wants the report tomorrow”, in which the temporal adverb modifies the possession, not the wanting). (See Figure 2 for the required parse, and the discussion by McCawley (1979, pages 84–86) of the phenomenon and an analysis along these lines.)

Second, as will be seen in the next section, there are cases in which semantic material associated with a single linguistic unit can be distributed arbitrarily far apart in the semantic form, a phenomenon that would be disallowed under this proposal. The want example can serve to demonstrate this problem as well, as shown in Figure 3.
Figure 2: An analysis in which the meaning of the transitive verb *wants* incorporates an implicit notion of possession requires that the “parse” of the logical form allow multiple levels of structure to correspond to a single linguistically realizable unit.

Figure 3: The ability to modify the embedded proposition of possession in the logical form for the sentence “Reuters wants the report tomorrow” demonstrates that the subparts of the logical form parcelled out to a single linguistically realizable unit need not even be connected in the tree. This calls for adjunction as a primitive operation in the semantics.

Another common technique is to specify the “parsing” procedurally. The main predicate in the semantic form invokes a procedure that traverses the semantic form, picking out material that corresponds to the semantic unit that will be linguistically realized, and recursively invoking similar processes on certain subforms. Hovy’s PAULINE system (1987a) works in much this way. In keeping with the underlying motivations for grammatical formalisms such as TAG, we
would prefer to specify this decomposition uniformly and declaratively, rather
than on the basis of particular procedures, if possible. The remainder of this
paper can be taken as a demonstration that it is.

What composition operations are possible?

The basic operation that is used (implicitly) to compose the individual parts
of (2), namely substitution, does not parallel the primitive operation that TAGs
make available, namely adjunction. In the particular example, this latter problem
is revealed in the scope of the quantity quantifier being inside the say predicate—
a scoping more appropriate for a sentence like “Reuters reported that Iraq had
said how many ships it attacked.” The more standard representation of scoping
would be akin to

(3) \( \lambda(\text{quantity-of-ships}). \)

\[
\begin{align*}
& \text{report( Reuters,} \\
& \text{say( Iraq, attack( Iraq,} \\
& \text{quantity-of-ships)))}
\end{align*}
\]

but this requires one of the elementary semantic units to be “broken up”. Conse-
quently, McDonald and Pustejovsky note that they cannot have the logical form
(3) as the source of the example sentence (1).\(^2\)

Where are compositions performed?

Finally, under the analysis of McDonald and Pustejovsky, the grammatical infor-
mentation alone does not determine where adjunctions should occur. They allude

\(^2\)The synchronous TAG analysis to be presented does construct the logical form (3) as the
source of the example sentence (1). Nothing in the formalism precludes the inner-scoped logical
form (2) from being associated with this or other sentences if desired.
to this problem when they note that “the [generator] must have some principle by which to judge where to start.” In their own example, they say that “the two pending units, $U_2$ and $U_3$, are then attached to this matrix . . . into complement positions,” but do not specify how the particular attachment positions within the elementary trees are chosen (which of course has an impact on the semantics). The relationship between syntax and semantics that they propose links elementary trees with units of the realization specification. Apparently, a more finely structured representation is needed.

In the case of Mumble, McDonald’s tactical generation system (Meteer et al., 1987), this fine structure is realized procedurally through LISP code. (See for instance, the discussion by McDonald and Meteer (1988).) Here also, one would want such linguistic information concerning the relationship between syntax and semantics, to be presented in a way that exhibits its uniform nature and where it could be used uniformly, rather than being hidden in procedures.

4 Synchronous TAGs

In order to provide an explicit representation for the semantics of strings generated by a TAG, and in so doing provide a foundation for the generation efforts of Joshi and McDonald and Pustejovsky, we present an extension to TAGs, synchronous TAGs, which was originally developed just to characterize the declarative relationship between strings and representations of their semantics. The formalism allows us to circumvent some of the problems discussed above.

The idea underlying synchronous TAGs is simple. One can characterize both a natural language and a logical form language with TAGs. The relation between
strings in the two languages (sentences and logical forms, respectively) can then be rigorously stated by pairing the elementary trees of the two grammars and linking the corresponding nodes, forming a new grammar whose elements are linked pairs of elementary trees.

The synchronous TAG formalism addresses all three of the problems mentioned above. First, a synchronous TAG characterizes a relation between languages. Thus, we need not assume that the sentences of the logical form language come pre-packaged into their constituent units (just as in the case of sentence parsing, where we need not assume that sentences come pre-bracketed). Second, the operations that are used to build the two structures—natural language sentences and semantic representations—are stated using the same kinds of operations, as they are both characterized by TAGs. Third, the linking of individual nodes in the elementary trees of a synchronous TAG provides just the fine-grained relationship between syntax and semantics that allows decisions about where to perform semantic operations to be well-defined.

5 An Example Synchronous TAG

We introduce synchronous TAGs by example, continuing with an exegesis of the sentence that McDonald and Pustejovsky focus on, and following roughly the structure of their TAG analysis.\(^3\)

A synchronous TAG sufficient for this example includes the three pairings of

\(^3\)The linguistic analysis implicit in the TAG English fragment that we present is not proposed as an appropriate one in general. It merely provides sufficient structure to make the points with respect to generation. Furthermore, the trees that we present here for expository purposes as elementary should actually themselves be built from more primitive trees. Finally, we gloss over details such as features necessary to control for agreement or verb-form checking, and we replace the pronoun with its proper noun antecedent to finesse issues in pronominal interpretation.
trees (labeled $\alpha$, $\beta_1$, and $\beta_2$) found in Figure 4. Note that the first components of the three pairs constitute a TAG grammar sufficient to generate the sentence “How many ships did Reuters report that Iraq attacked” or “How many ships did Reuters report that Iraq said that Iraq attacked”. The second components generate strings in a logical form language. The syntax of that language includes such phrase types as formula ($F$) or abstracted property ($\lambda$). The obvious linearization of such trees will be assumed, so that the logical form in given for the sample sentence is in the language. Some of the nodes in the pairs are linked;
formally, the interpretation of these links is that operations on the tree pairs must occur at both ends of a link. For simplicity, we have marked only those links that will be needed for the derivation of the sample sentence.

Derivation in the synchronous grammar proceeds by choosing a pairing of initial trees from the grammar and repeatedly updating it by the following three-step process:

1. Choose a link to act upon.
2. Choose a pairing such that the two trees can respectively act on (substitute at or adjoin at) the respective ends of the link chosen in Step 1.
3. Remove the chosen link from the trees being updated and perform the two operations, one in each of the trees. If the trees in the chosen pairing themselves have links, these are preserved in the result.

For instance, we might start with the initial tree pair $\alpha$ from Figure 4. We choose the sole link in $\alpha$, and choose $\beta_1$ as the tree pair to operate with, as the first component of $\beta_1$ can operate (by adjunction) on an $S$ node, and the second on an $F$ node as required by the chosen link. The result of performing the adjunctions is the pairing given as $\alpha + \beta_1$ in Figure 5. The link in the $\beta_1$ pair is preserved in the resultant, and can serve as the chosen link in the next round of the derivation. This time, we use $\beta_2$ to operate at each end of the link resulting in the pairing labeled $\alpha + \beta_1 + \beta_2$. This pairing manifests the association between the English string “How many ships did Reuters report that Iraq said that Iraq attacked” and the logical form representation in (3).

---

4A fuller description of the formal aspects of synchronous TAGs can be found in a previous paper (Shieber and Schabes, 1990).
Figure 5: The results of synchronous derivation steps. The $\beta_1$ pair of trees is adjoined at the two ends of the sole link in $\alpha$, and the $\beta_2$ pair is then adjoined into the result forming the derived tree pair that encodes the association between sentence (1) and logical form (3).

Returning to the three issues cited previously, the synchronous TAG presented here:
1. Makes the decomposition of the logical forms implicit in the grammar just as the decomposition of the natural-language expressions are, by stating the structure of logical forms grammatically.

2. Allows the same operations to be used for composing both natural-language expressions and semantic representations as both are stated with the same grammatical tools.

3. Makes the fine-grained correspondence between expressions of natural language and their meanings explicit by the technique of node linking.

The strong notion of semantic locality that synchronous TAGs embody makes these results possible. This semantic locality, in turn, is only possible because the extended domain of locality found in pure TAGs makes it possible to localize dependencies that would otherwise be spread across several primitive structures.

6 Translation with Synchronous TAGs

Synchronous TAGs as informally described here declaratively characterize a relation over strings in two languages without priority of one of the languages over the other. Any method for computing this relation in one direction will perforce be applicable to the other direction as well. The distinction between parsing and generation is a purely informal one depending merely on which side of the relation one chooses to compute from: both are instances of a process of translating between two TAG languages appropriately synchronized.

The question of generation with synchronous TAGs reverts then to one of whether this relation can be computed in general. There are many issues involved in answering this question, most importantly, what the underlying TAG formalism (the base formalism) is that the two linked TAGs are stated in. The example above required a particularly simple base formalism, namely pure TAGs
with adjunction as the only operation. The experience of grammar writers has demonstrated that substitution is a necessary operation to be added to the formalism, and that a limited form of feature structures with equations are helpful as well. Work on the use of synchronous TAGs to capture quantifier scoping possibilities makes use of so-called multi-component TAGs. Finally, the base TAGs may be lexicalized (Schabes et al., 1988) or not. (The reader is encouraged to refer to the thesis by Schabes (1990) for a full discussion of all of the issues involved, especially those concerning lexicalization.)

Once the base formalism has been decided upon (we currently are using lexicalized multi-component TAGs with substitution and adjunction), a simple translation strategy from a source string to a target is to parse the string using an appropriate TAG parser for the base formalism. Each derivation of the source string can be mapped according to the synchronizing links in the grammar to a target derivation. Such a target derivation defines a string in the target language which is a translate of the source string.

In the case of generation, the source string is a semantic representation, the target is a natural-language realization. For example, the logical form (3) has a single derivation in the pure TAG formed by projecting the synchronous TAG onto its semantic component. (We might notate the semantic components with α(sem), β₁(sem), and β₂(sem), and analogously for the syntactic components.) That derivation can be recovered by “parsing” the logical form with the projected logical form grammar, as depicted in Figure 6. The pairings whose semantic components were used in this derivation and the links operated on implicitly define a corresponding derivation on the syntactic side. The yield of this derivation is a string whose meaning is represented by the logical form that we started with.
λq : report(Reuters, 
said(Iraq, 
attack(Iraq,q))))

parse

α (sem)  α (syn)
| 2 linking | 2

β₁ (sem)  β₁ (syn)
| 0

β₂ (sem)  β₂ (syn)

yield

How many ships 
did Reuters report 
that Iraq had said 
Iraq attacked?

**Figure 6:** Generation proceeds by translation of a derivation. Conceptually, the logical form is parsed according to the semantic part of the grammar, the resulting derivation tree is canonicalized if necessary (it is not in this case) and translated into a syntactic grammar derivation by virtue of the links. The yield of the derived tree from this derivation is the generated sentence.
The target derivation might not, unlike in the example above, be in canonical form (as defined by Vijay-Shanker (1988)), and consequently must be normalized to put it into canonical form. Under certain configurations of links, the normalization process is nondeterministic; thus one source derivation (necessarily in canonical form by virtue of properties of the parsing algorithm) may be associated with several canonical target derivations. In translation from natural language to logical forms, the multiple translates typically correspond to scope ambiguities in the source sentence (as quantifier scope or scope of negation or adverbs). On the other hand, we have not observed the linking configurations that give rise to such ambiguities in translating in the other direction, that is, in performing generation.

7 Discussion

Inherence of Semantic Monotonicity in Synchronous TAGs

In previous work, one of us noted that generation according to an augmented context-free grammar can be made more efficient by requiring the grammar to be semantically monotonic (Shieber, 1988); the derived semantics for an expression must include, in an appropriate sense, the semantic material of all its subconstituents. It is interesting to note that synchronous TAGs are inherently semantically monotonic, and the computational advantages that accrue to such grammars apply to synchronous TAG generation as well. Furthermore, it is reasonable to require that the semantic component of a synchronous TAG be lexicalized (in the sense of Schabes et al. (1988)), allowing for more efficient parsing according to the semantic grammar and, consequently, more efficient generation. In the case of
augmented context-free grammars, the semantic monotonicity requirement precludes "lexicalization" of the semantics. It is not possible to require nontrivial semantics to be associated with each lexical item. This fact, and the inefficiencies of generation that follow from it, was the initial motivation for the move to semantic-head-driven generation (Shieber et al., forthcoming). The efficiencies that that algorithm gains for augmented-context-free generation inhere in the synchronous TAG generation process if the semantic grammar is lexicalized. In summary, just as lexicalization of the syntactic grammar aids parsing (Schabes and Joshi, 1989), so lexicalization of the semantic grammar aids generation.

Incremental Generation with Synchronous TAGs

The simple generation algorithm that we have just presented seems to require that we completely analyze the logical form before generating the target string, as the process is a cascade of three subprocesses: parsing the logical form to a source derivation, mapping from source to target derivation, and computing the target derivation yield. As is common in such cases, portions of these computations can be interleaved, so that generation of the target string can proceed incrementally while traversing the source logical form. To what extent this incrementality can be achieved in practice depends on subtleties in the exact formal definition of synchronous TAG derivation and properties of particular grammars; a full explication is beyond the scope of this paper.

On the other hand, serious programmatic problems exist in the synchronous TAG framework as regards incrementality of generation, not in the sense discussed above, but in the following sense: Suppose that the semantic representation is being developed incrementally itself, perhaps as a result of the incremental
nature of the strategic planner. It is not easy to see how tactical generation based on a synchronous TAG could be interleaved with this incremental construction of the semantic form. Substructures that are local in the semantic derivation tree (the structure driving the linguistic realization) can be highly non-local in the semantic form: this is a side-effect of the use of adjunction as a composition operation. Thus incremental development of the logical form may not allow incremental development of the generated linguistic structure (unless it uses exactly the composition operations over the same set of elements). This is to be contrasted with the issue of interleaving analysis of the semantic form with generation of the linguistic realization. This latter, more conventional, sort of incremental generation can be done (pace considerations of the previous paragraph) if it is assumed that the entire content is known at the start.

The Synchronous TAG Tactical Framework in a Full Generation System

Many researchers have pointed out the need for interaction between strategic and tactical components in a full natural-language-generation system (Appelt, 1985; Hovy, 1987a; McKeown, 1985). The synchronous TAG framework specifies certain natural places at which this interaction might occur, and it is to this topic that we now turn.

Hovy (1987a) emphasizes the differing character of the hierarchical, top-down, global aspects of language generation planning and the interactive, bottom-up, local aspects. He uses the terms “prescriptive” and “restrictive” planning for the two aspects; the latter characterizes the on-line style of choice planning that must articulate with tactical generation in a tightly intertwined way. Many systems
make this important distinction by providing separate facilities for disambiguation of the tactical generator: the discrimination nets of Goldman's MARGIE (1975), the choosers of systemic grammar (Mann, 1983), PAULINE's limited commitment planner (Hovy, 1987a), the interleaved planning of the KAMP system (Appelt, 1985), and so forth. As a synchronous TAG will, in general, specify multiple realizations of a given semantic form—depending on choices among alternative tree pairs and alternative canonicalizations of derivation trees—the need for restrictive planning manifests itself in this framework as well. A natural approach would be to introduce communication with a restrictive planning component in order to disambiguate exactly these choices. Thus, tactical generation and planning would be interleaved through the interface of choice disambiguation, as is done in many of the above systems. Those aspects of language that are pertinent to the disambiguation process should be similar, regardless of whether the choice is embedded in a systemic grammar, say, or a TAG. As an example, McCoy et al. (1990) provide an approach to integrating TAGs and systemic grammar that is complementary to the tactical approach presented here in that it addresses some of these issues in integrating a TAG as the grammatical portion of a fuller generation system.

Two aspects of the framework of synchronous TAGs make it especially desirable from the standpoint of interaction with a restrictive planner: First, a lexicalized TAG specifies grammatical information separately and in a way that it can be directly manipulated. Appelt (1985, page 113) discusses this issue in more detail. Second, its lexical nature, as argued by Hovy (1987a, Section 6.2.1), is appropriate for the storage of disambiguation information. In this sense it is akin to the phrasal lexicons of Jacobs (1985), or Hovy (1987b) (and see references
cited therein).

8 Conclusion

The extended domain of locality that tree-adjoining grammars enjoy would seem to make them ideal candidates for the task of tactical generation, where semantic locality is of great importance. Synchronous TAGs, which extend pure TAGs to allow for mappings between languages, provide a formal foundation for this intuition by making explicit the semantic locality that generation requires.

9 Acknowledgements

This research was partially funded by ARO grant DAAG29-84-K-0061, DARPA grant N00014-85-K0018, and NSF grant MCS-82-19196 at the University of Pennsylvania. The authors are indebted to Aravind Joshi for his support of this research.

Bibliography


