Scaling Cognitive Linguistics: Formalisms for Language Understanding

Nancy Chang[1], Jerome Feldman[1], Robert Porzel[2] and Keith Sanders[1]

[1]International Computer Science Institute
1947 Center Street
Berkeley, CA 94704 (USA)
{nchang,jfeldman,ksanders}@icsi.berkeley.edu

[2]European Media Laboratory
Schloss-Wolfsbrunnenweg 33
D-69118 Heidelberg (Germany)
porzel@eml.org

Abstract

Research in cognitive linguistics has yielded valuable semantic and pragmatic insights that should be incorporated by language understanding systems. These insights have not been expressed in a rigorous enough form to be implemented. We provide a unified formalism for representing four conceptual primitives: schemas, constructions, mental spaces, and maps. These primitives are motivated by the cognitive linguistics literature and can serve as the basis for scalable deep understanding systems.

1 Introduction

The incorporation of principles of semantics and pragmatics is essential for building systems that can reasonably be said to understand natural language. Research in cognitive linguistics has yielded valuable insights in these areas, but not yet in a form that is rigorous enough for implementation. Moreover, although current logical and statistical approaches to natural language, especially unification-based approaches (Shieber, 1986), provide a useful starting point for capturing these insights, additional representational tools and techniques are needed. In this paper we outline mechanisms for formalizing what we take to be the four conceptual primitives of cognitive linguistics; together, these primitives provide a basis for scalable deep understanding systems.

Cognitively motivated approaches to linguistics have sought to demonstrate how diverse phenomena affecting language use are grounded in the rest of cognition. The meanings of linguistic units are subject to category effects (Lakoff, 1987); largely based on abstractions over sensorimotor patterns, called *image schemas* (Johnson, 1987; Lakoff and Johnson, 1980) or *force-dynamic* schemas (Talmy, 1988); and often defined against a constellation of related concepts captured in a *frame* (Fillmore, 1982). Apparently exotic phenomena, including metaphorical inference (Lakoff and Johnson, 1980) and *mental space* phenomena (Fauconnier, 1985), are taken as reflecting basic facts of cognitive organization (such as the prevalence of cross-domain mappings). In general, linguistic knowledge is seen as a collection of conventionalized pairings between form and meaning (Langacker, 1987), or *constructions* (Goldberg, 1995; Fillmore, 1988).

The key to scalability in any paradigm is compositionality; our goal in language understanding, then, is to systematically combine the heterogeneous structures posited in cognitive linguistics to yield overall interpretations. We identify four conceptual primitives that we believe capture most of the proposed structures and thus suffice for building scalable language understanding systems: **schemas**, **constructions**, **spaces** and **maps**. After discussing some general background assumptions (Section 2), we describe each primitive using a common formalism based on that used in the Embodied Construction Grammar (ECG) framework. The unified representation of these four primitives provides an overarching computational framework for identifying the underlying conceptual relations between diverse linguistic phenomena.

In this paper we concentrate on basic representational issues, but the general framework for which ECG was designed assumes that understanding an utterance involves an analysis process – during which the best-fitting set of constructions is determined and the corresponding network of conceptual schemas is evoked – followed by some appropriate action, or **enactment**; one form of enactment is mental **simulation** of the evoked schemas in context to produce a rich set of inferences.[1] These efforts are all part of broader research in the Neural Theory of Language project,[2] which addresses the biological and neural basis of language understanding and use (see Section 7); for discussion of how the formalism is used in language understanding and learning systems, see (Narayanan, 1997).

2 Embodied Construction Grammar

Each of the four primitives defined in this paper corresponds to a collection of schematic structures organized hierarchically in a lattice induced by the **subcase** relation (between a **base** structure and its more specific subcase). A preliminary synopsis of these follows:

- A **schema** is a conceptual structure associated with a set of **roles** (or parameters), plus constraints on those roles and the relations between them.
- A **construction**, the basic linguistic unit, is a pairing of form and meaning, where the paired elements can be entities or relations.
- A **map** identifies correspondences across a pair of conceptual domains (e.g., between two schemas, or between two spaces).
- A **space** is a conceptual domain containing entities and relations among them.

We will define each of these in more detail in the remaining sections of the paper. In terms of the language understanding framework mentioned in the last section, the analysis process makes use of the construction lattice to build a **semantic specification**, or **semspec**, that is used during enactment; the semspec consists of a set of interconnected schemas (from the schema lattice). These two primitives are defined in Section 3 and Section 4, in a form close to that in (Bergen and Chang, 2002) and described in more detail there. The remaining two primitives, maps and spaces, are involved in analyses producing semspecs with more complex internal structure. Our description of these two primitives in Section 5 and Section 6 is less detailed since the processes in which they are involved have not been worked out as thoroughly.

Although the various ECG lattices all encode concepts, they are not intended to exhaustively recreate world knowledge. Rather, they contain only the structures needed for language analysis.[3] However, subtle interactions between linguistic and world knowledge are clearly a general characteristic of language understanding. Thus, some of the structures may make reference to one or more external ontologies (i.e., domain representations) that contain encyclopedic world knowledge. A limited interface between the ECG conceptual primitives and the external ontologies allows types and predicates defined in the ontologies to serve as constraints. Also, ontology relations play a central role in metonymy and other phenomena related to processes of **construal** (in which the semantic interpretation of some linguistic element is dependent on contextual factors), though we will not address this further here.

In addition to general knowledge represented in the ontology, an evolving belief net captures the understander's beliefs about the discourse situation (Narayanan, 1997). The structures involved in language understanding cannot be composed independent of context; although in this paper we do not focus on contextual features, all ECG primitives are subject to contextually dependent effects.

The formalism descriptions to follow all exhibit similar structure: each is a template for defining the relevant conceptual primitive. In each case, keywords of the description language are shown in bold, and the first line names the structure being defined. This is followed by a series of declarations and/or blocks stating the various constraints and relations that hold of the structure; these vary according to the particular primitive to be defined. In all the definitions, a **subcase of** tag indicates any subcase rela-

[1] An earlier version of the formalism designed for use in a simulation-based model of language understanding is described in (Bergen and Chang, 2002).

[2] `http://www.icsi.berkeley.edu/NTL/`

[3] From the neural-psychological perspective, this means that only part of human knowledge is schematized for language.

tions that may hold of the structure being defined; the keyword **self** refers to the structure being defined. Also, the **evokes** relation makes the structures and constraints of the evoked conceptual structure (referred to by a local name within the current definition) accessible to the structure being defined. The evokes relation implies neither full inheritance of the evoked structure's roles nor containment in either direction.

The definition templates use a few additional notations: a left square bracket ([) marks optional blocks; curly braces ({ }) enclose a set of options for the statements that may appear (possibly multiple times) in the block. Angle brackets (<>) denote a reference to some structure or role (either locally declared or accessible through local structures, where standard slot-chain notation is used to refer to a role y of a structure x as x.y), or a predicate on accessible structures. We also use '//' to introduce italicized comments. We defer additional details for the examples below.

3 Schemas

Schemas serve as the basic building blocks for all semantic representation. Each schema formalizes some conceptual or semantic structure as a set of **roles**, which serve to parameterize that structure; these are similar to features or attributes in a framework like HPSG (Pollard and Sag, 1994). They are used to represent image schemas (Lakoff, 1987) and frames (Fillmore, 1982), as well as the action representations known as **executing schemas** (or **x-schemas**) (Narayanan, 1997), and other mechanisms of cognitive linguistics. Schemas are used to parameterize both static and dynamic situations. Figure 1 provides a template for schema definitions; Figure 2 provides illustrative examples of the formalism details we describe in this section.

In schema definitions, only the **roles** block, declaring and constraining the schema's local roles, is required. As mentioned earlier, the **subcase of** relation indicates that the current structure specializes its base structure and inherits its roles. These roles are accessible to other schemas. Schemas listed in the **evokes** block serve as background for the current schema; their roles are not inherited, and evoked schemas are not accessible to other structures using the schema being defined. The **constraints** block lists constraints that apply to locally accessible roles.

```
schema <name>
[subcase of <schema>
[evokes
   { <schema> as <local-name> }
 roles
   { <local-role>
     <local-role> : <restriction>
     <local-role> ⟷ <role>
     <local-role> ⟷ <role> : <restriction> }
[constraints
   { <role> ⟵ <value>
     <role> ⟷ <role>
     <predicate>
     <setting> :: <predicate>
     <setting> :: <role> ⟷ <role> }
```

Figure 1: Schema definition formalism.

Figure 2 shows two example schemas. The SPG schema encodes the concept of a directed curve between two points, and an object (or trajector) positioned along that curve; Translational-Motion is defined as a subcase of Motion but also makes crucial reference to (i.e., **evokes**) the static configuration described in SPG, although the latter is neither a **role** in the former, nor a base structure from which it inherits its roles and constraints.

Roles can be declared with simply a name (<local-role>), or they may be accompanied by role restrictions (a schema or ontology type, with an optional cardinality restriction) and/or **identification** constraints (indicated with the double-headed arrow (⟷). The latter (which may appear in either the **roles** or the **constraints** block) causes its arguments to be unified, such that their roles and constraints are shared. In the example, the Translational-Motion schema inherits its mover role (as well as before and after roles; see below) from its base Motion schema and adds source and goal roles that are bound to the specified roles of the evoked instance of SPG.

The assignment of a particular value to a role's filler is expressed with a left arrow (⟵). Values can include numbers and strings, as well as constant-like values for grammatical features such as number and person. The notation <predicate> refers to predicates modeling particular semantic relations that hold in a given schema (and later in a construction, map, etc.). These relations are restricted to

```
schema SPG
  roles
    source : Place
    path : Directed-Curve
    goal : Place
    trajector : Entity
```

```
schema Translational-Motion
  subcase of Motion
  evokes SPG as s
  roles
    mover ⟷ s.trajector
    source ⟷ s.source
    goal ⟷ s.goal
  constraints
    before :: mover.location ⟷ source
    after :: mover.location ⟷ goal
```

Figure 2: Example schemas: SPG and Translational-Motion.

a fixed set that can be evaluated against the external ontology and internal belief structure. Among others, these would include relations expressing relative positions along a scale, as well as inequalities like greater(x, y) and less(x, y). For instance, the constraints on a Containment schema would include a predicate like greater(container.volume, contents.volume).

The final constraint type allowed uses the '::' notation to assert that the specified condition – either an identification of two or more roles, or an ontological predicate – must hold at a particular phase of enactment or simulation. The <setting> names come from a fixed set of roles, e.g., before, after. For example, in Figure 2 the schema for Translational-Motion is a dynamic schema, some of whose roles (e.g., before and after) refer to particular phases in the action (and eventually in the corresponding simulation or enactment. The listed constraints indicate the different locations of the mover before and after the motion.

The '::' notation thus captures the distinction between permanent constraints and ones that are more transitory or episodic: predicates can correspond to either persistent properties (similar to Kratzer's (1995) *individual-level* properties), or, when marked by a '::' prefix, transitory (*stage-level*) properties.

4 Constructions

Constructions, the basic unit of linguistic knowledge, serve as the link between form and meaning (Langacker, 1987; Goldberg, 1995). Linguistic units of varying complexity (including morphological, lexical, phrasal, clausal, and intonational constructions) associate elements of form (describing properties of the speech signal, text stream, gesture, etc.) with elements of meaning (any conceptual matter, described in the last section in terms of various kinds of schemas).

All constructions can thus be characterized as *bipolar* (Langacker, 1987), having a *form pole* and a *meaning pole*. A simple example would be a lexical construction whose form pole is the phonological or orthographic form "house" and whose meaning pole is the corresponding conceptual representation (say, the category of House). Some constructions have additional internal structure: they may have **constituents** that can be instantiated by other constructions, including but not necessarily limited to straightforward cases like *in the house*, where *the house* forms an NP constituent which then participates in a higher PP construction.

Our construction definition notation (shown in Figure 3) uses some of the same formal tools as used to define schemas. After the identifying first line and the subcase declaration (again denoting inheritance relations, but this time in the construction lattice), it has three main blocks, each of which can be seen as consisting of elements (of various kinds) and constraints. Those in the (optional) **constructional** block are cross-domain, involving both form and meaning (and their constructional relationship), while those in the **form** and **meaning** blocks involve only their respective domains.

The first of these, the **constructional** block, optionally indicates any **constituents**, along with any applicable type restrictions. As with schemas, the **evokes** block allows the activation of other constructions that are related in a variety of ways to **self**. Note that constituents themselves are constructions and thus have form and meaning poles; these poles are implicitly accessible for reference in the form or meaning blocks to follow, and can be referred to by a subscripted f or m. The example SPATIAL-PP shown in Figure 4 is similar to that defined in

```
construction <name>
 [subcase of <construction>
 constructional
   evokes
   { <construction>  as <local-name> }
  constituents
    { <local-name> : <construction> }
   constraints
     // like schema constraints
 form
   elements
     // like schema roles
  constraints
    // like schema constraints
 meaning
   // evoked, roles and constraints
   // defined as in schema
```

Figure 3: Construction definition formalism.

(Bergen and Chang, 2002) for licensing PPs that describe spatial relations (*in the house*, *to the store*). As mentioned above, these can be analyzed as having constituent structure, where the constituents rel and lm are typed as SPATIAL-PREPOSITION (a subset of the standard prepositions) and REFERRING-EXPR (similar syntactically to NPs but named to reflect their referential function). The form and meaning poles of rel, for example, are referenced as rel_f and rel_m, respectively, elsewhere in the definition.

```
construction SPATIAL-PP
 subcase of PHRASE
 constructional
   constituents
     rel : SPATIAL-PREPOSITION
     lm : REFERRING-EXPR
   constraints
     rel.number ⟷ lm.number
     rel.case ⟷ lm.case
 form
   constraints
     rel_f < lm_f
 meaning
   constraints
     rel_m.landmark ⟷ lm_m
```

Figure 4: Example construction: SPATIAL-PP.

Note that constructional **features** may also be specified and used to capture phenomena like agreement. The two constructional constraints in Figure 4 are intended to express agreement constraints in English between number and case of the relation and the landmark: **among a cow* and **with they/their*. (These might be expressed more succinctly with a special feature that stands for all the (shared) features, e.g. with something like rel.features ⟷ lm.features.) Such expressions would not be licensed by this construction, assuming the appropriate features were specified in the constituent constructions with statements like **self**.number ⟵ PLURAL. Of course, such constraints might have alternate expression within the form or meaning blocks, but for current purposes the important point is that the formalism allows features that are constructional – that is, neither purely formal nor purely semantic.

Form constraints can apply to both the pure form **elements** specified and the form poles of the constituents (and, in turn, their constituents, through dotted names). The primary use of form constraints is for word ordering (as in Figure 4, which asserts that the form pole of the SPATIAL-PREPOSITION precedes that of the REFERRING-EXPR); there are a number of other order relations possible, as well as constraints on prosody or phonological form.

The meaning block of a construction is similar to a schema specification, except that no name or subcase declaration is needed, since the meaning pole is simply part of a construction, which is in the construction lattice. As described in more detail in (Bergen and Chang, 2002), the meaning constraints do most of the composition work needed to produce the evolving semspec.

5 Maps

Several kinds of cross-domain mappings have been proposed in the cognitive linguistics literature. These have been used to characterize phenomena including metonymic reference (e.g., referring to a restaurant customer by the associated order: *The ham sandwich at table 9 wants his check*) and metaphorical inference (e.g., applying inferences from the source domain of physical motion to a target economics domain: *France stumbled into recession* (Lakoff and Johnson, 1980; Narayanan, 1997)). Mental space phenomena have also been analyzed using maps; see Section 6.

All of these can be seen as involving loose correspondence relations between distinct conceptual entities such that, under restricted conditions, the entities can be treated as being mutually accessible and even referred to using the same expressions.

The ECG **map** primitive is intended to cover all such cross-domain mappings. As shown in Figure 5, maps are described using representational devices similar to those above, including a subcase declaration and blocks like those in the schema description. All maps link two conceptual domains (e.g., source and target for metaphor maps, trigger and target for metonymic maps) listed in the **roles** block.

In addition, the map has a **pairs** block for specifying individual (sub)mappings across the two mapped conceptual domains (expressed with $\mapsto$). The crucial distinction between this mapping relation and the identification ($\longleftrightarrow$) relation used elsewhere is that identification is used to unify two entities, while mapping expresses a correspondence between two entities that are identified. Note that in the current design, the subcase relation causes the map being defined to inherit roles but not pairs. Also, the roles and constraints apply to the map as a whole, and not to any particular pairing.

```
map <name>
 [subcase of <map>
 [evokes
 [{<map> as <local-name> }
 roles
   // like schema roles
 [constraints
 [  // like schema constraints
 pairs
   <role> ↦ <role>
```

Figure 5: Map definition formalism.

A simple metaphorical map is shown in Figure 6, corresponding to the LOVE IS A JOURNEY metaphor (Lakoff and Johnson, 1980) licensing sentences like "Our relationship is *at a dead end*" and "The marriage is *on the rocks*". It is defined as a subcase of the general EVENT STRUCTURE METAPHOR; like other metaphor maps, its roles are source and target and are bound to the general conceptual domains of Journey and Love. The **pairs** block lists some of the licensed mappings, where the relation x $\mapsto$ y maps x (defined with respect to source) to y (defined with respect to target).

The same formalism can be used to specify other kinds of maps. The Order-FOR-Customer metonymy, for example, might be treated as relating a trigger (a FoodItem to a target (a Human), where the participant identified as the target is constrained to have ordered the food referred to by the trigger. Other maps are used to specify the relation between mental spaces, as we will discuss next.

```
map Love-IS-Journey
 subcase of Event-Structure-Metaphor
 roles
   source : Journey
   target : Love
 pairs
   source.traveler ↦ target.lover
   source.destination ↦ target.goal
   source.vehicle ↦ target.relationship
   source.impediment ↦ target.difficulty
```

Figure 6: Example map definition: Love-IS-Journey.

6 Mental spaces and their maps

The term **mental spaces** (Fauconnier, 1985) refers to a conceptual domain built up during discourse, in its most general form simply a set of entities and relations among them. In our language understanding framework, a mental space is a major partition of the overarching conceptual space that characterizes the (speaker's or hearer's) representation of the current discourse; it functions as a domain of reference and predication, such that the referents and predications built up by linguistic expressions must be assigned to or associated with some particular space for enactment to occur. Each space has its own belief context, history, inferences, etc. Phenomena for which mental space analyses have been proposed include referential ambiguities (such as in *In Len's painting, the girl with blue eyes has green eyes*), as well as presuppositions and counterfactuals.

During analysis, certain linguistic constructions (traditionally called *space builders*) may evoke a new mental space, along with the map (or maps) specifying how it is related to some other space (typically the current discourse space D). Figure 7 shows how the formalism used to define the other primitives can be straightforwardly extended to define (mental) spaces. The major addition is that mental spaces list at least one map, often to D.

Mental spaces are generally associated with a specific kind of map. Figure 8 shows a Depiction-Map and Depiction-Space, intended to capture the relationship of depiction (similar to Fauconnier's *image*

```
space <name>
 subcase of <space>
  evokes
   <space> as local name
  maps
   // like schema roles
  roles
   // like schema roles
  constraints
   // like schema constraints
```

Figure 7: Space definition formalism.

connector) that holds between, for example, a picture and the entities it depicts. In addition to roles, Depiction-Map has pairs that map entities and relations in the model domain to corresponding entities and relations in the artifact domain. (The representation of such highly general cross-domain maps remains a topic of our continuing work.) The Depiction-Space is then defined as involving a map of the specified kind; its roles are identified with roles in the map.

```
map Depiction-Map
 roles
   model : Situation
   artifact : Depiction-Artifact
   author : Human
   medium : Depictive-Medium
 pairs
   model.Entity ↦ artifact.Entity
   model.Relation ↦ artifact.Relation
```

```
space Depiction-Space
 maps
   dm : Depiction-Map
 roles
   model ⟷ dm.model
   author ⟷ dm.author
   medium ⟷ dm.medium
 constraints
   self ⟷ dm.artifact
```

Figure 8: Space example: Depiction.

The pairs of a Depiction-Map can be further elaborated by continuing discourse, and the formalism has additional notation for supporting this. For example, during the analysis of *In Harry's painting of Paris, the Eiffel Tower is only half finished*, the space builder *painting* along with the definite description *the Eiffel Tower* would lead to the instantiation of a Depiction-Space related to the current discourse by an instance of Depiction-Map. The reference to the Eiffel Tower would introduce a pair mapping a particular region of paint in the painting to the real-life Eiffel Tower (blob11 ↦ Eiffel-Tower). The set of pairs in a map (and of entities in a space) can be incremented by ongoing discourse, so that as new referents are introduced, entities are added to spaces along with mappings between them (blob23 ↦ Arc-de-Triomphe). Relations between entities can also be mapped across spaces, as in larger(blob11,blob23) ↦ taller(Eiffel-Tower,Arc-de-Triomphe).

Note that the Depiction-Map without the pairs resembles a schema defining the depiction relation, useful for understanding the simple sentence *Harry painted a picture of Paris*. Such a relationship between a simplified schema description and space-building map appears quite general and is the topic of ongoing research.

7 Conclusions

Cognitive linguistics has, over several decades, provided powerful new theories and mechanisms for the kind of deep semantic analysis needed for language understanding. The relative lack of computational work exploiting these insights can be largely attributed to the absence of an adequate formalization. This paper has attempted to address that need by outlining formal mechanisms for representing what we take to be the four building blocks of cognitive linguistics: schemas, constructions, maps, and mental spaces. The formalism will undoubtedly evolve as more applications are worked out. Nevertheless, the current formalism appears adequate for representing a wide array of cognitive linguistic analyses and for supporting inference and other applications in a scalable fashion.

It is important to understand the limitations of the formalism. The distinction between analysis and enactment provides a convenient separation between conventionalized conceptual and linguistic knowledge on the one hand and context-dependent inferential phenomena on the other. Even a complete ECG grammar only specifies the possible analyses of an utterance in context, without committing to any particular analysis. In principle, corpus techniques could be used to estimate conditional probabilities and build a Bayes-optimal parser. Some promising results have been achieved in pilot studies (Narayanan and Jurafsky, 1998), but there are limitations to such probabilistic approaches.

Moreover, within our broader framework of language understanding as enactment, language analysis merely serves as a prerequisite to some activity. The formalism as specified in this paper makes few direct references to enactment, but the need for action places major constraints on the formalism and accompanying analysis processes (Bergen and Chang, 2002; Narayanan, 1997).

Most importantly, any computational formalism like that we have described is only a crude approximation to the continuously active interconnected neural structures that support human language processing. Some crucial aspects of human language understanding, including priming, construal, and context sensitivity, cannot be explained or modeled at all in either symbolic or probabilistic formalisms. These issues can be studied using structured connectionist models (Shastri et al., 1999), but we do not yet know how to build scalable versions of these. From this perspective, an ECG grammar may be viewed as providing general indications of what neural connections are present without specifying any connection weights or system dynamics.

Our current goal is to use the ECG formalism to capture and extend the insights of cognitive linguistics. We are starting to build ECG analyzers that work well enough for relatively simple language, producing semantic specifications that can be used for a wide range of tasks. The primitives we have defined can all be easily composed, and there does not yet appear to be any barrier to the construction of large-scale systems. All of this remains to be shown, but a formalism like that we have defined in this paper is a necessary first step.

References

Benjamin K. Bergen and Nancy C. Chang. 2002. Embodied Construction Grammar in simulation-based language understanding. Technical Report TR-02-004, International Computer Science Institute.

Gilles Fauconnier. 1985. *Mental Spaces: Aspects of Meaning Construction in Natural Language*. MIT Press/Bradford, Cambridge, Mass. and London.

Charles J. Fillmore. 1982. Frame semantics. In Linguistic Society of Korea, editor, *Linguistics in the Morning Calm*, pages 111–38. Hanshin, Seoul.

Charles Fillmore. 1988. The mechanisms of construction grammar. In *Berkeley Linguistics Society*, volume 14, pages 35–55.

Adele E. Goldberg. 1995. *Constructions: A Construction Grammar Approach to Argument Structure*. University of Chicago Press.

Mark Johnson. 1987. *The Body in the Mind*. University of Chicago Press.

Angelika Kratzer. 1995. Stage-level and individual-level predicates. In Gregory N. Carlson and Francis Jeffry Pelletier, editors, *The generic book*, pages 125–75. University of Chicago Press, Chicago.

George Lakoff and Mark Johnson. 1980. *Metaphors We Live By*. University of Chicago Press.

George Lakoff. 1987. *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*. University of Chicago Press.

Ronald W. Langacker. 1987. *Foundations of Cognitive Grammar, Vol. 1*. Stanford University Press.

Srini Narayanan and Daniel Jurafsky. 1998. Bayesian models of human sentence processing. In *Proc. 20th Cognitive Science Society Conference*, pages 84–90. Lawrence Erlbaum Associates.

Srini Narayanan. 1997. *Knowledge-based Action Representations for Metaphor and Aspect (KARMA)*. Ph.D. thesis, University of California at Berkeley.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

L. Shastri, D. Grannes, S. Narayanan, and J. Feldman. 1999. A connectionist encoding of parameterized schemas and reactive plans. In G.K. Kraetzschmar and G. Palm, editors, *Hybrid Information Processing in Adaptive Autonomous Vehicles*, Lecture Notes in Computer Science, Lecture Notes in Artifi cial Intelligence. Springer-Verlag, Berlin.

Stuart M. Shieber. 1986. *An Introduction to Unification-Based Approaches to Grammar*. University of Chicago Press, Chicago.

Leonard Talmy. 1988. Force dynamics in language and cognition. *Cognitive Science*, (12):49–100.