On Natural Language Processing and Plan Recognition

Christopher W. Geib and Mark Steedman

School of Informatics
University of Edinburgh
Edinburgh EH8 9LW, Scotland
cgeib@inf.ed.ac.uk

Abstract

The research areas of plan recognition and natural language parsing share many common features and even algorithms. However, the dialog between these two disciplines has not been effective. Specifically, significant recent results in parsing mildly context sensitive grammars have not been leveraged in the state of the art plan recognition systems. This paper will outline the relations between natural language processing(NLP) and plan recognition(PR), argue that each of them can effectively inform the other, and then focus on key recent research results in NLP and argue for their applicability to PR.

1 Introduction

Without performing a careful literature search one could easily imagine that the fields of *Plan Recognition*(PR) and *Natural Language Processing*(NLP) are two separate fields that have little in common. There are few papers in either discipline that directly cite work done in the other. While there are exceptions,[Carberry, 1990; Blaylock and Allen, 2003; Pynadath and Wellman, 2000; Vilain, 1991], even these papers often are only citing NLP in passing and not making use of recent research results.

Interestingly, many researchers do see these two areas as very related, but are still not taking the recent lessons learned in one area and applying them to the other. In an effort to rectify this lack, this paper will outline the commonalities between PR and NLP, argue why the results from each of these research areas should be used to inform the other, and then outline some recent research results that could inform a unified view of these two tasks.

2 Commonalities

In this section we will sketch the similarities at the surface and algorithmic levels between PR and NLP before more formally drawing their representations together in the Section 3. We will start this process by laying out some terminology so that we can see the common parts of NLP and PR.

Both PR and NLP take as input a set of *observations*. In PR these are observations of action executions and in NLP these are individual words or utterances. In both cases, the observations are used to create a higher level structure. In NLP these higher level structures may be parse trees [Collins, 1997] or logical forms [Bos *et al.*, 2004]. In PR they are usually a hierarchical plan structure[Kautz and Allen, 1986; Kaminka *et al.*, 2001; Geib and Goldman, 2003] or at least a high level root goal[Horvitz *et al.*, 1998]. In either case, both NLP and PR construct a higher level knowledge structure that relates the meanings of each of the individual observations to a meaning for the collection of observations as a whole.

For the purposes of this discussion it will aid us to abstract away from the specific details of the higher level structure that is built by this process. To simplify this discussion we will talk about these systems as if they were creating an hierarchical data structure that captures the meaning of the collection of observations. We will use the PR terminology and call this structure an *explanation* and following the NLP terminology call the process of producing a single explanation *parsing*.

In order to parse a set of observations into an explanation both PR and NLP must specify the patterns of observations they are willing to accept or the rules that govern how the observations can be combined. In PR this specification is done in the form of a library of plans, while in NLP this is done through a grammar. In Section 3 we will argue that there is no significant distinction between PR plan libraries and NLP grammars. Therefore, in this paper we will call all such specifications of the rules for acceptable combination of observations *grammars*.

With this terminology in place, we can now describe both NLP and PR as taking in as inputs a set of observations and a grammar specifying the acceptable sets of observations. Both NLP and PR then parse these observations to produce explanations that organize the observations into a structured representation of the meaning of the collection.

Given this level similarity, it is not surprising that grammars in both NLP and PR can result in multiple explanations for a given set of observations. However, it is of interest that in both disciplines this ambiguity has been resolved using very similar probabilistic methods. In both areas, the state of the art methods are based on weighted model counting. These systems build the set of possible explanations and establish a probability distribution over the set in order to determine the most likely explanation.

The work in NLP often uses probability models derived

from an annotated corpus of text[Clark and Curran, 2004] while the probability models from PR have been based on Markov models of the world dynamics [Bui *et al.*, 2002] or probabilistic models of plan execution [Geib and Goldman, 2003]. While space prohibits a full exposition of these very different probability models, it is still telling that a weighted model counting method is the state of the art in both fields.

Beyond these surface and algorithmic similarities there are psycholinguistic reasons for believing that PR and NLP are very closely tied process that should inform one another. For example, consider indirect speech acts like asking someone "Do you know what time it is?" To correctly understand and respond to this question requires both NLP and PR.

Correctly responding requires not merely parsing the sentence to understand that it is a request about ones ability to provide a piece of information. It also requires recognizing that asking the question of someone else is the first step in a two part plan for finding out a piece of information by asking someone else. PR allows one to conclude that if someone is following this plan they most likely have the goal of knowing the piece of information (the current time in this case) and that providing the desired information will be more helpful than answering the literal question asked.

Given the similarities between the two areas, it seems reasonable that work in one area should inform the other. However important results in each area are not being leveraged in the other community. In the next section we will more formally specify the relation between these two areas to help researchers take advantage of the results in both areas.

3 Plans as Grammars

Our argument that PR and NLP should inform one another would be significantly strengthened if we could show, as we have asserted above, that the plan libraries used by PR systems are equivalent to the grammars used by NLP systems. In the following section we will show the parallels between these two constructs and a mapping between them.

Almost all PR work has been done on traditional hierarchical plans.[1] While much of the work in plan recognition has not provided formal specifications for their plan representations they can all generally be seen as special cases of Hierarchical Task Networks (HTN) as defined in [Ghallab *et al.*, 2004].

According to Ghallab the actions of an HTN domain are defined as either *operators* or *methods*. An operator corresponds to an action that can be executed in the world. Following Ghallab we will define them as a triple $(n, add-list, delete-list)$ where n is the name of the operator, $add-list$ is a list of predicates that are made true or added to the world by the operator, and $delete-list$ is the set of predicates that are made false or deleted from the world by the operator.

A method on the other hand represents a higher level action and is represented as a 4-tuple $(name, T, \{st_0, ..., st_n\}, C)$ such that *name* is a unique identifier for the method, T names the higher level action this method decomposes, and $\{st_0, ..., st_n\}$ identifies the set of sub-tasks that must be performed for the higher level task to be performed. Finally, C represents a set of ordering constraints that have to hold between the subtasks for the method to be effective.

We will draw a parallel between HTNs and context free grammars (CFGs). Following Aho and Ullman[Aho and Ullman, 1992] we define a CFG, G, as a 4-tuple $G = (N, \Sigma, P, S)$ where

- N is a finite set of *nonterminal symbols*,
- Σ is a finite set of *terminal symbols* disjoint from N,
- P is a set of *production rules* that have the form $n \rightarrow \omega$ where $n \in N$ and $\omega \in (\Sigma \cup N)^*$, and
- S is a distinguished $S \in N$ that is the start symbol.

Given these definitions, we would like to map the plans represented as an HTN into an equivalent CFG. We first consider the case of a collection of HTN plans that are totally ordered. That is, we assume that for every method definition the constraints on the subtasks $st_0, ..., st_n$ define a total ordering over the subtasks. Without loss of generality, we assume that the subtasks' subscripts represent this ordering.

To encode the HTN as a CFG, we first consider the operators. The processing for these is quite simple. We identify the names of each operator as a terminal symbols in our new grammar, and attach the add and delete lists to the nonterminal as features. Next we consider mapping the method definitions into productions within the grammar.

Given a totally ordered method definition, we can add the task to be decomposed to the set of non-terminal symbols. Then we define a new production rule with this task its left hand side. We then define the right hand side of the rule as the ordered set of subtasks. Thus, the method definition $(name, T, \{st_0, ..., st_n\}, C)$ is rewritten as the CFG production rule: $T \rightarrow st_0, ..., st_n$ and T is added to the set of nonterminals.

For example, consider the very simple HTN method, $m1$ for acquiring shoes:

$$(m1, \quad acquire(shoes), \\ \{goto(store), choose(shoes), buy(shoes)\}, \\ \{(1 \prec 2), (2 \prec 3)\})$$

where the constraints $(1 \prec 2)$ indicates the task $goto(store)$ must precede the task $choose(shoes)$ and $(2 \prec 3)$ indicates that $choose(shoes)$ must precede $buy(shoes)$. This is very easily captured with the CFG production:

$$acquire(shoes) \rightarrow \\ goto(store), choose(shoes), buy(shoes)$$

This process of converting each method definition into a production rule and adding the task to be decomposed to the set of non-terminals is repeated for every method in the HTN to produce the CFG for the plans. Now we turn to the question of partial ordering.

Limited cases of partial orderness could be handled in CFGs by expanding the grammar with a production rules for each possible ordering. However, as the NLP community has realized this can result in an unacceptable increase in the size of the grammar, and the related runtime of the parsing algorithm[Barton, 1985].

[1] See [Bui *et al.*, 2002] for an exception that works on hierarchical Markov models

So instead, to address this, the NLP community has produced a number of different grammar formalisms that allow the grammar to separately express decomposition and ordering. This includes the work of Shieber on ID/LP grammars [Shieber, 1984], Nederhof on poms-CFGs [Nederhof *et al.*, 2003], and Hoffman[Hoffman, 1995] and Baldridge[Baldridge, 2002] on partial orderness in Combinatory Catagorial Grammars. All of of these are attempts to include partial orderness within the grammar formalism (and parsing mechanism) without the exponential increase in the grammar size and runtime. Since each of these formalisms use very different representations, rather than presenting examples, we refer the reader to the cited papers. It suffices to say that these grammar formalisms introduce notational additions to denote partial orderness within the production rules and to explicitly specify the ordering relations that are required in each production. These formalisms can be used to capture HTN plan domains that require partial ordering.

It should be clear from this exposition that the grammar formalisms found in the NLP literature are sufficient to cover the method definitions found in most if not all of the PR literature. However, to the best of our knowledge no one has used any of the relatively recent grammar formalisms and their associated parsing machinery for plan recognition. Making use of these grammatical formalisms would also allow the use of their associated formal complexity results as well, something that has often been lacking in the work in PR.

Thus, we propose that NLP and PR could be unified by the use of the same underlying grammatical formalisms for representing the constraints on observations, and using a common parsing mechanism. In the case of probabilistic NLP and PR systems, we believe these systems may need to retaining separate methods for computing their probability distributions, however the parsing of observations into explanations could share a common framework. In the next section we will advocate a specific class of grammars for this task.

4 New Grammar Formalisms for PR

Given that researchers in NLP have been working on the close relationship between grammars, parsers, and language expressiveness it shouldn't be a surprise that results from this work could inform the work in PR. There are some classes of grammars that are too computationally expensive to parse for real world application. For example, the well known complexity results for parsing context sensitive grammars (CSGs) have all but ruled them out for NLP work. Likewise we expect poor performance for applications of CSGs to PR. Unfortunately, PR researchers have used these results as a motivation to build their own algorithms for parsing, often without even considering the limitations of the existing parsing algorithms. Examples include graph covering[Kautz and Allen, 1986] and Bayes nets[Bui *et al.*, 2002], that trade one np-hard problem for another. What has been largely ignored by the PR community is the NLP work in extending context free grammars and their efficient parsing algorithms.

Recent work in NLP has expanded the language hierarchy with grammars that have a complexity that falls between context free and context sensitive. Examples, include IL/LP Grammars[Shieber, 1984], Tree Adjunction Grammars(TAG)[Joshi and Schabes, 1997], and Combinatory Catagorial Grammars(CCG)[Steedman, 2000; Hockenmaier, 2003; Clark and Curran, 2004]. These "mildly context sensitive grammars"(MCSGs) have a number of properties that make them attractive for NLP including greater expressiveness than CFGs but still having polynomial algorithms for parsing. These properties also make them attractive for adoption by the PR community.

While these grammars are of scientific interest, we should justify their use, since it is not clear that PR requires grammars that are more expressive than CFGs. Such a claim would rest on the empirical need for plans that are not context free. If nothing more than a CFG is needed for PR, then a well known parsing algorithm like CKY that has a cubic complexity seems to be the obvious choices for application to PR. However, if there are PR problems that require recognizing plans that are not within the class of CFG plans, this would provide a convincing argument that PR requires a grammar that is not context free. In the following we will provide just such an example. While there are a number of different classes of MCSGs with different expressiveness results, and the exploration of all of them may prove useful for PR research, we will focus on the subclass of MCSGs that includes CCGs and TAGs called Linear Index Grammars(LIG).

Steedman[Steedman, 2000] has argued convincingly that CCGs and other LIGs are able to capture phenomena beyond CFGs that are essential to real world language use. Given the parallels we have already demonstrated between NLP and PR, we argue that if this class is necessary for NLP it shouldn't be surprising to us if this class of grammars captured essential phenomena in PR as well. In this light, Steedman shows that CCGs provide for *crossing dependencies* in NLP, a critical extension that context free grammars cannot capture. Likewise, if we find that such crossing dependencies are necessary for recognizing plans we would have a strong argument that PR requires a grammar that is in the MCSG family.

While a full discussion of the handling of crossing dependencies in CCGs is beyond the scope of this paper, it will be helpful to understand their basic structure in order to identify them in PR contexts. Crossing dependencies occur when the words that make up a constituent (like a relative clause) are interleaved in the sentence with the elements of a different constituent. Steedman[Steedman, 2000] has argued that a particularly strong example of the naturalness of these constructs are Dutch verbs like *proberen* 'to try' which allow a number of scrambled word orders that that are outside of the expressiveness of CFGs.

For example the translation of the phrase "... because I try to teach Jan to sing the song." has four possible acceptable orderings [1 - 4] and a fifth that is more questionable.

1. ... omdat ik_1 Jan_2 het $lied_3$ $probeer_1$ te $leren_2$ $zingen_3$.
   ... because I Jan the song try to teach to sing.
2. ... omdat ik_1 $probeer_1$ Jan_2 het $lied_3$ te $leren_2$ $zingen_3$.
3. ... omdat ik_1 $probeer_1$ Jan_2 te $leren_2$ het $lied_3$ te $zingen_3$.
4. ... omdat ik_1 Jan_2 $probeer_1$ te $leren_2$ het $lied_3$ te $zingen_3$.
5. ?... omdat ik_1 Jan $probeer_1$ het lied te leren zingen

The subscripts are included to show the correspondence of the noun phrases to the verbs. For example in the first ordering the noun phrases are all introduced first followed by their verbs in the same order as their nouns. This produces the maximally crossed ordering for this sentence.

The realization of these kinds of crossed dependencies in a PR context is relatively straightforward. Its important to keep in mind the mapping that we are using between traditional language grammars and planning grammars will mean that dependencies in PR are not the same as in NLP. In NLP dependencies are features like gender, number or tense that must agree between different words within the sentence. In the PR context, dependencies are equivalent to causal links in traditional nonlinear planning[McAllester and Rosenblitt, 1991]. That is, they are states of the world that are produced by one action and consumed by another. Therefore, a plan with a crossing dependency would have the causal structure shown in Figure 1 where in *act*1 is found to produce the preconditions for actions *act*2 and *act*3 which each produce a precondition for *act*4. Such a structure requires that two different conditions be created and preserved across two different actions for their use. Note that while the actions are only partially ordered, there is no linearizion of them that will remove the crossing dependency. That is, *act*2 and *act*3 can be reordered but this will not remove the crossing dependency.

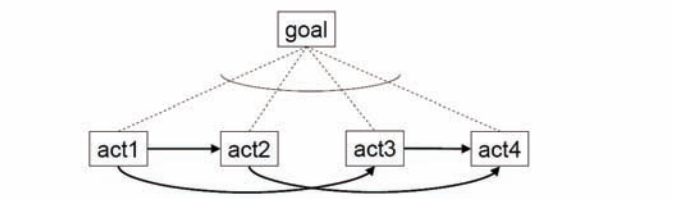

Figure 1: An abstract plan with a crossed dependency structure

The argument for the necessity of MCSGs for planning rests on real world examples of plans with this structure. Being able to describe what such a plan looks like is not compelling if they never occur in PR problem domains. Fortunately, examples of plans with this structure are relatively common. Consider recognizing the activities of a bank robber that has both his gun and ski-mask in a duffel bag and his goal is to rob a bank. He must open the bag, put on the mask and pick up the gun and enter the bank. Figure 2 shows this plan. This plan has exactly the same crossed dependency structure shown in Figure 1.

Note, that we could make this plan much more complex with out effecting the result. Actions could be added before opening the bag, after entering the bank, and even between putting on the ski-mask and picking up the gun so long as the critical causal links are not violated. The presence of plans with this structure and our desire to recognize such plans gives us a strong reason to look at the grammars that fall this class as a grammatical formalism for PR.

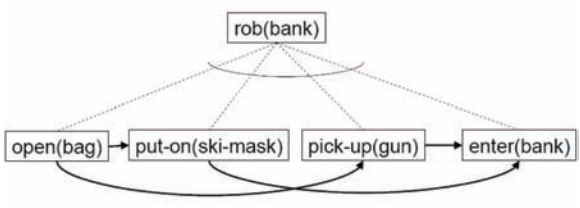

Figure 2: An example plan with crossing dependency structure

4.1 Why MCSGs?

Joshi[Joshi, 1985] first formally defined the class of MCSGs as those grammars that share four properties that are relevant for NLP:

- The class of languages included covers all context free languages.
- The languages in the class are polynomially parsable.
- The languages in the class only capture certain types of dependencies including nested (non-crossing) and crossed dependencies.
- The languages in the class have the *constant growth property* which requires that if all of the sentences in the language are sorted according to their length then any two consecutive sentences do not differ in their length by more than a constant factor determined by the grammar.

This set of properties are also relevant for defining the class of grammars that would work well for PR. We will argue for each of them in order.

First, we have just demonstrated the need for grammars that are more than context free for PR. Second, clearly polynomial parsing is desirable for PR. In order to use these algorithms in real world applications they will need to be extended to consider multiple possible interleaved goals and to handle partially observable domains[Geib and Goldman, 2003]. If a single goal can't be parsed in polynomial time what hope do we have for efficient algorithms for the needed extensions? Further, PR is needed in a great many applications that will not tolerate algorithms with greater complexity. For example, assistive systems are not useful if their advice comes to late.

Third, Plans do have structure that is captured in dependency structures. Therefore, it seems natural to try to restrict the grammar for plans to the kinds of dependency structures that are actually used. Whether or not the dependency restrictions that are consistent with NLP are the same set for PR is largely an empirical question. We have already seen evidence of crossing dependencies that required us to abandon CFGs in favor of MCSG's. While nested and crossing dependencies in the abstract can cover all the kinds of dependencies needed in planning, different MCSGs place different restrictions on the allowable depth of crossings and the number of nesting. This will have a significant impact on the expressiveness of a particular MCSG and its applicability to PR.

Fourth and finally, the requirement of the constant growth rate may be the hardest to understand. Intuitively in the PR

domain this means that if there is a plan of length n then there is another plan of length at most n+K where K is a constant for the specific domain. For example this rules out that the length of the next plan is a function of the length of the previous plan or some other external feature. Note this says nothing about the goals that are achieved by the plans. The plan of length n and length n+K may achieve very different goals but they are both acceptable plans within the grammar. This speaks to the intuition that given a plan one should be able to add a small fixed number of actions to the plan and get another plan. Again this seems to be the kind of property one expects to see in a PR domain and therefore in a plan grammar.

Now, while we believe we have made a strong argument for the use of MCSGs for PR, this is not the final word on the question. While we have presented an argument that we need at least the expressiveness of LIGs, it may be the case that still more powerful grammar formalisms are needed. The most promising method for proving such a result would require finding plans with dependency structures that are not in MCSG that our PR systems need to recognize. Thus, determining if MCSGs are sufficient for PR is an open research question for the community.

While there are well known languages that are not in MCSG it is difficult to see their relevance to planning domains. For example the language $\{a^{2^n}\}$, that is the language where in the length of any sentence of the language is a power of two, is not MCSG as it fails the constant growth requirement. It is possible to imagine contrived examples where this would be relevant for PR (Perhaps as part of some kind of athletic training regime, we want to recognize cases where in someone has run around the track a number of times that is a power of two.) However, this certainly seems anomalous and most likely should be dealt with by reasoning that falls outside of the grammar, like a counter and a simple test.

5 Conclusions

There are close ties between the process of natural language processing and plan recognition. This relation should allow these two processes to inform each other and allow the transfer of research results from one area to the other. However, much recent work in both fields has gone unnoticed by researchers in the other field. This paper begins the process of sharing these results, describing the isomorphism between the grammatical formalisms from NLP and plan representations for PR, arguing that like NLP, PR will require a grammatical formalism in the mildly context sensitive family, and finally that NLP and PR can form a common underlying task that can usefully be explored together.

Acknowledgments

The work described in this paper was conducted within the EU Cognitive Systems project PACO-PLUS (FP6-2004-IST-4-027657) funded by the European Commission.

References

[Aho and Ullman, 1992] Alfred V. Aho and Jeffrey D. Ullman. *Foundations of Computer Science*. W.H. Freeman/Computer Science Press, New York, NY, 1992.

[Baldridge, 2002] Jason Baldridge. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. PhD thesis, University of Edinburgh, 2002.

[Barton, 1985] G. Edward Barton. On the complexity of id/lp parsing. *Computational Linguistics*, 11(4):205–218, 1985.

[Blaylock and Allen, 2003] Nate Blaylock and James Allen. Corpus-based statistical goal recognition. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 1303–1308, 2003.

[Bos *et al.*, 2004] Johan Bos, Stephen Clark, Mark Steedman, James Curran, and Julia Hockenmaier. Wide-coverage semantic representations from a ccg parser. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING '04)*, 2004.

[Bui *et al.*, 2002] Hung H. Bui, Svetha Venkatesh, and Geoff West. Policy recognition in the abstract hidden markov model. In *Technical Report 4/2000 School of Computer Science, Curtin University of Technology*, 2002.

[Carberry, 1990] Sandra Carberry. *Plan Recognition in Natural Language Dialogue*. ACL-MIT Press Series in Natural Language Processing. MIT Press, 1990.

[Clark and Curran, 2004] Stephen Clark and James Curran. Parsing the wsj using ccg and log-linear models. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, 2004.

[Collins, 1997] Michael Collins. Three generative, lexicalized models for statistical parsing. In *ACL '97: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, 1997.

[Geib and Goldman, 2003] Christopher W. Geib and Robert P. Goldman. Recognizing plan/goal abandonment. In *Proceedings of IJCAI 2003*, 2003.

[Ghallab *et al.*, 2004] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.

[Hockenmaier, 2003] Julia Hockenmaier. *Data and Models for Statistical Parsing with Combinatory Catagorial Grammar*. PhD thesis, University of Edinburgh, 2003.

[Hoffman, 1995] Beryl Hoffman. Integrating 'free' word order syntax and information structure. In *Proceedings of the 1995 Conference of the European Chapter of Association for Computational Linguistics*, pages 245–252, 1995.

[Horvitz *et al.*, 1998] Eric Horvitz, Jack Breese, David Heckerman, David Hovel, and Koos Rommelse. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 1998.

[Joshi and Schabes, 1997] A. Joshi and Y. Schabes. Tree-adjoining grammars. In *Handbook of Formal Languages, Vol. 3*, pages 69–124. Springer Verlag, 1997.

[Joshi, 1985] Aravind Joshi. How much context-sensitivity is necessary for characterizing structural descriptions - tree adjoining grammars. In *Natural Language Processing - Theoretical, Computational, and Psychological Perspective*, pages 206–250. Cambridge University Press, 1985.

[Kaminka *et al.*, 2001] G. Kaminka, D.V. Pynadath, and M. Tambe. Monitoring deployed agent teams. In *Proceedings of the International Conference on Autonomous Agents*, pages 308–315, 2001.

[Kautz and Allen, 1986] Henry Kautz and James F. Allen. Generalized plan recognition. In *Proceedings of the Conference of the American Association of Artificial Intelligence 1986*, pages 32–38, 1986.

[McAllester and Rosenblitt, 1991] David McAllester and David Rosenblitt. Systematic nonlinear planning. In *Proceedings of the Conference of the American Association of Artificial Intelligence (1991)*, pages 634–639, 1991.

[Nederhof *et al.*, 2003] Mark-Jan Nederhof, Giorgio Satta, and Stuart M. Shieber. Partially ordered multiset context-free grammars and ID/LP parsing. In *Proceedings of the Eighth International Workshop on Parsing Technologies*, pages 171–182, Nancy, France, April 2003.

[Pynadath and Wellman, 2000] David Pynadath and Michael Wellman. Probabilistic state-dependent grammars for plan recognition. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI-'00)*, pages 507–514, 2000.

[Shieber, 1984] Stuart M. Shieber. Direct parsing of ID/LP grammars. *Linguistics and Philosophy*, 7(2):135–154, 1984.

[Steedman, 2000] Mark Steedman. *The Syntactic Process*. MIT Press, 2000.

[Vilain, 1991] Marc Vilain. Deduction as parsing. In *Proceedings of the Conference of the American Association of Artificial Intelligence (1991)*, pages 464–470, 1991.