The Affordance-Based Concept

by

Peter John Gorniak

B.Sc., University of British Columbia (1998)
M.Sc., University of British Columbia (2000)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2005

Author⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯
Program in Media Arts and Sciences
August 5, 2005

Certified by⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯
Deb K. Roy
Associate Professor
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯
Andrew B. Lippman
Chair, Department Committee on Graduate Students
Program in Media Arts and Sciences

The Affordance-Based Concept

by

Peter John Gorniak

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on August 5, 2005, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Media Arts and Sciences

Abstract

Natural language use relies on situational context. The meaning of words and utterances depend on the physical environment and the goals and plans of communication partners. These facts should be central to theories of language and automatic language understanding systems. Instead, they are often ignored, leading to partial theories and systems that cannot fully interpret linguistic meaning.

I introduce a new computational theory of conceptual structure that has as its core claim that concepts are neither internal nor external to the language user, but instead span the objective-subjective boundary. This theory proposes interaction and prediction as a central theme, rather than solely emphasizing deducing, sensing or acting. To capture the possible interactions between subject and object, the theory relies on the notion of *perceived affordances*: structured units of interaction that can be used for prediction at certain levels of abstraction. By using perceived affordances as a basis for language understanding, the theory accounts for many aspects of the situated nature of human language use. It provides a unified solution to a number of other demands on a theory of language understanding including conceptual combination, prototypicality effects, and the generative nature of lexical items.

To support the theory, I describe an implementation that relies on probabilistic hierarchical plan recognition to predict possible interactions. The elements of a recognized plan provide an instance of perceived affordances which are used by a linguistic parser to ground the meaning of words and grammatical constituents. Evaluations performed in a multiuser role playing game environment show that this implementation captures the meaning of free-form spontaneous directive speech acts that cannot be understood without taking into account the intentional and physical situation of speaker and listener.

Thesis Supervisor: Deb K. Roy
Title: Associate Professor, Program in Media Arts and Sciences

The Affordance-Based Concept

by

Peter John Gorniak

Thesis Committee:

Advisor

Deb K. Roy
Associate Professor
Program in Media Arts and Sciences
Massachusetts Institute of Technology

Thesis Reader

Professor Daniel C. Dennett
Professor of Philosophy
Tufts University

Thesis Reader

Allen L. Gorin
Director, Human Language Technology Research
National Security Agency
U.S. Department of Defense

Thesis Reader _____

Professor Leslie P. Kaelbling
Professor of Computer Science and Electrical Engineering
Massachusetts Institute of Technology

Acknowledgements

Most importantly, I must thank my thesis supervisor, Deb Roy. He has given me the chance to help define and explore a new and exciting research focus. His ability to maintain his independence from established specializations and yet draw from and give back to a rich set of collaborators and communities continues to inspire me. Daniel Dennett has over the years offered his broad and deep perspective on my research endeavours, questioning my assumptions and relaying the exciting relevant theories he and others have put forward. Leslie Kaelbling and Allen Gorin critiqued this thesis and its accompanying defence talk and ensured it was appropriate for all the its embedding research areas.

The MIT Media Laboratory provided a unique and always inspiring research home for my Ph.D. work. Its research and artistic freedom as well as its sheer diversity remain unsurpassed. I would especially like to thank members and mentors of the Cognitive Machines, Synthetic Characters, Robotic Life, Affective Computing, Human Dynamics and Music Mind and Machine groups, as well as the Student Committee.

David Thomson deserves credit for commenting on the first draft of this thesis, and for playing endless hours of Neverwinter Nights with me, which both gave me the idea for the studies presented here and carried me through my degree without losing sight of what this is all about. The speech group at Sun Microsystems, especially Philip Kwok, helped with many of the speech recognition aspects of my work.

I would also like to take this opportunity to thank some of my mentors and friends who have provided me with opportunities and inspiration over the years, and who I have to thank for preparing me for my work here at MIT. They include Josh Tenenbaum, David Poole, Alan Mackworth, Alan Richardson, Julyet Benbasat, Mark Maclean and Anette and Ian Niefiendt-Umlauff.

My parents have been enthusiastic and proud supporters of my academic work and early on instilled a love for research and the mind in me.

Finally, I can only hope to be as invaluable for my wife Cydney's time at MIT and future ahead as she has been and will be for mine.

Contents

List of Figures

11

List of Tables

Chapter 1

Introduction

Much of human language speaks about the world. We easily refer to objects using words such as "door" or "the blue thing for making pizza that I gave you yesterday." The relationship that holds between words and the world, variously and differently described by such terms as *reference*, *intentionality* and *aboutness*, has been the subject of many theories and debates. Most of these theories posit an intermediary step between words and the world, usually labelled as a *concept*. However, theories differ even on fundamental matters such as whether a concept is a mental construct of the language user, or an independent abstract entity. Only very few of these theories have computational instantiations that have been used to build larger scale natural language processing systems.

There are two highly interrelated parts to any theory of concepts: a description of the internal structure (or lack of internal structure) of a concept, and an account of how this structure comes to be about the world. In many cases, theories focus on the first and neglect the second, or at best give a vague answer to the second. Why is this? I suggest that at issue is the problem of autonomy. Human beings are autonomous in a very specific way - we interact with our immediate world ourselves, and maintain our own concepts about this world. However, we neither try to fully internalize a complete representation of the world [Brooks, 1991], nor do we individually maintain all possible concepts of our language community [Putnam, 1975]. If we are to build a machine that uses language at the human level, such a machine needs to show the same type of autonomy: it needs to be able to maintain its own concepts about its own experience, yet rely on its environment and its community to maintain most of the state of the world and the meaning of language in

general.

In building machines other than language using ones, the need for this type of autonomy is obvious: a walking robot would not be considered a walking robot if it needed to be carried by a human being to locomote. In building language using machines, however, this type of limitation is widely accepted: human beings provide input and output interpretation for these machines, alleviating the need for the machine itself to maintain any sort of intentionality. Thus, a machine connecting words to other words is considered language using, though it relies on human beings to establish any sort of external meaning for its words. This is not only a failure of language using machines: theories of concepts make exactly the same mistake. It is true that for other problems the human solution may not always be the optimal one: perhaps a driving robot can get around better than a walking one in many cases. However, there are good reasons to build walking robots: human beings have shaped their environment to be amenable to the type of locomotion they are good at, so if a robot wants to share this environment, it better learn how to walk. The same requirement is true of language: there may be more efficient and clearer ways for machines to define concepts without the requirement that they connect to the world like human ones do, but if we are to build language using machines that speak human language, their concepts and intentionality must be of the same type and quality as the human equivalents.

The lack of a link between language using systems and the world they are supposed to speak about supports the use of theories of concepts that neglect the intentional aspects of concepts. These theories may define words in terms of other words or symbols and call these definitions concepts. I believe that if we are to build a machine that is a true language using machine in all the ways a human being is, we need to start from scratch with a new theory of concepts that emphasizes from the ground up the importance of intentionality and tightly couples the internal structure of concepts with their need to be about the world. Any theory that draws a clear line between concepts and the world is doomed to support detached concepts that lose their intentionality. In this thesis, I provide a theory that refuses to draw such a line. Instead, it proposes that every structural element of every concept must cross over from the concept to the world, that every concept is both a property of the language using system, and of its relation to the the embedding world. These structural elements are called *affordances*, yielding a theory of *Affordance-Based Concepts*.

The Affordance-Based Concept provides a solution to the need to take into account the intentional link between the language user and the world by making predicted interac-

tions its core elements. By doing so, it also yields a substrate that addresses many other demands of a theory of concepts that have been only addressed individually before. For example, perceived affordances are naturally ranked according to typicality and context, addressing the prototypicality effects often exhibited by human concepts. Similarly, the rich predictions made by Affordance-Based Concepts naturally lend themselves to conceptual composition. In fact, as I will show in the implementation provided here, conceptual composition can be cast as a filtering process on the complete set of affordances a situation yields. Finally, hierarchical sets of affordances give an intuitive framework for performing conceptual generalization and abstraction.

The computational realization of this theory employs plan recognition to model the link between the language user and the world. In recognizing a language user's plans, it maintains sets of plan states that capture predictions about the language user relative to the structure of the world at a specific level of abstraction. These hierarchically organized probabilistic state sets correspond to the notion of affordances just introduced. In understanding language, then, the implementation introduced in this thesis understands speech by linking grammatical constituents to sets of plan recognition states. By doing so, it grounds language in a substrate that naturally represents concrete and abstract objects together with their possible interactions as sets of affordances. An evaluation using a probabilistic Earley parser as a plan recognizer to understand situated commands in a multiuser computer role playing game shows that this implementation leverages the perceived affordances to understand situated directives. It also provides examples of reasoning over past affordances to understand complex utterances.

1.1   Roadmap and Contributions

This thesis casts the problem of understanding situated language as that of using words and linguistic structure to filter the set of perceived affordances relevant at the time of an utterance. Perceived affordances, mental representations that summarize the past and predict the future at a single level of abstraction, represent a subject's possible past, current and future interactions with the situation. Language, or in the case of the studies presented here a linguistic command, filters the state of all perceived affordances down to those implied by the utterance used, resulting in a concept consisting of perceived affordances. This

concept can be used to predict the next action of someone listening to the command (as it is in the studies presented here), or more generally to capture the intended effect of an utterance taking into account the physical situation and intentional context of speaker and listener. In the course of casting concepts anew as bundles of perceived affordances this thesis sketches solutions to standing problems such as conceptual abstraction and composition, and that of reasoning about other minds, and it provides constrained instantiations of the proposed solutions in the context of understanding commands in a computer role playing game. This thesis thus moves beyond the current state of the art in language understanding by tying language to the elements of a dynamic, intentional representation of its embedding situation.

Chapter 2 To support the need for a new theory of concepts, I critique other proposed solutions, especially ones attempting to bridge the gap between concepts and the world and show how they fall short of embedding the concept in the world. With support from other recent work in Philosophy and Cognitive Science, I arrive at an interaction based theory of concepts that forces every conceptual element to represent a part of the world by making a prediction about it. In this way, concepts are the basic elements of reasoning and can be falsified when their predictions do not come true. I draw on the theory of affordances, which focuses on perceived possible interactions between an agent and the world, to flesh out the theory.

Chapter 3 To support the new theory, I then provide a computational implementation of the proposed theory using the notion of plan recognition via hierarchical parsing. The implementation supports global hierarchical plan recognition of two actors in a constrained environment, and proposes the idea of using linguistic analysis to filter the results of plan recognition to achieve understanding. It performs plan recognition via probabilistic Earley parsing, and ties the notion of an affordance to an Earley state used to summarize past actions and predict the future at one level of the parse forest hierarchy. Language is also analysed by a parser, which builds up a filter expression used to limit the set of Earley states considered to be a valid interpretation of an utterance.

Chapter 4 This chapter applies the implementation described in Chapter 3 to the concrete problem of understanding spontaneous language used by players of online role playing games. Such games provide a rich interaction environment embedding the play-

ers' characters, and the theory introduced in Chapter 2 proves useful in modelling the affordances of the environment and linking its action possibilities to the language used by the players. Specifically, the studies address the problem of understanding directives in a co-operative two player puzzle by predicting the listener's next action based on plan recognition filtered by linguistic analysis. The chapter also sketches how the implementation might be extended to cover descriptions and questions.

Chapter 2

The Problem of Concept Detachment

Adopting any given theory of concepts imposes a bias, because every theory emphasizes certain features of concepts to the detriment of others. There are thus always a host of objections to any theory. Some of these objections simply note that the theory does not cover some features, for example that a definitional theory of concepts does not explain the existence of prototypical concepts. Other criticisms are offered at a more abstract level, claiming that there are insurmountable theoretical problems with a given solution. An example of such an attack is the claim that any solution that represents concepts by their relation to each other leads to meaning holism, where any change to one concept changes all concepts at once. Laurence and Margolis as well as Prinz give good overviews of current theories of concepts and the debates surrounding them [Laurence and Margolis, 1999; Prinz, 2002].

In my view, however, many problems with traditional theories stem from a fundamental bias of their creators: that the use of concepts by an actual language using system, and thus their connection to the world, is secondary to their internal structure and formal properties. This is a persistent human bias visible also from the first attempts to create intelligent computational systems: human beings believe that what is easy for them, namely perceiving and acting on the physical world, must generally be easy problems. More "abstract" mental feats, however, such as playing chess or providing word definitions, seem to be harder problems to us. Artificial intelligence quickly foundered on this bias, because building systems that could sense and act like human beings turned out to be very difficult, and still remains an unsolved problem today. Theories of concepts do not have this built-in bias

detector, because most of the people proposing new theories do not implement them in an artificial system with wide coverage, but rather pick and choose examples from human language use that they can explain. To compound this problem, researchers actually building language using systems have a cheat available that those building sensing and acting machines do not have: they can de-couple the more abstract mental use of concepts from their connection to the world, by using a conveniently available natural mechanism to substitute for the ability of their system to make this connection: actual human beings. Thus, current day artificial language using systems live in a world of symbols which connect to other symbols in a myriad ways. At the beginning and at the end, however, there is always a human being feeding in the original symbols and interpreting the final ones. Thus, internet search engines [Brin and Page, 1998], automatic essay grading programs [Valenti *et al.*, 2003] and text summarization tools [Paice, 1990] show impressive performance and are very useful. They cannot, however, autonomously use language without the human being serving as input and output converters.

I believe that many of the criticisms and recently proposed theoretical solutions are concerned with this underlying problem of *Concept Detachment* that both traditional theories of concepts and computational language systems share. In the following, I will relay some of these criticisms and proposals in my own words, and show how they relate to the problem of concept detachment.

2.1 Non-Mental Concepts and Human Intentionality

At its most extreme, concept detachment takes the form of the claim that concepts are not mental constructs at all, but rather are abstract entities that attach to language independent of a specific language user. While some state this view explicitly, such as Frege does when presenting his notion of *sense*, it is implicit in a number of theories by virtue of their lack of attention to the mechanisms of conceptual attachment [Frege, 1892]. Alternatively, a number of theorists claim that there is something special about the way that human beings attach their concepts to the world, so special that a computational machine could not possibly have the same kind of intentionality. Thought experiments like Searle's Chinese Room scenario purport to show this impossibility [Searle, 1980]. Both the view that concepts attach to the world independently of the language user, or that they attach through the language user, but

in a way that is not computationally explainable, lead theorists to feel like they do not have to tell a story about how concepts connect to the world. This is not the place for a full refutation of these arguments, especially as such a refutation has been eloquently given by others (many of these arguments are included with Searle's original publication). Suffice it to say that if one is interested in building machines that attain human level autonomous language use, they are not arguments that lead forward. To use Dennett's phrase, these thought experiments only work due to a failure of imagination [Dennett, 1992]. I thus assume a position very similar to Jackendoff's and Barwise and Perry's: that there is no magic in how concepts attach to the world, and that this attachment is a fully computationally explainable, if complicated, relationship between the language user and the world [Jackendoff, 2002; Barwise and Perry, 1983]. It is this attachment that should be the main subject of study if we are to move forward with a useful theory of concepts. It follows that if it is computationally explainable, it is also computationally implementable and there is no reason in principle why a machine cannot have exactly the same kinds of intentionality that human beings have.

Interestingly, most computational manifestations of concepts reveal the same implicit assumptions by not providing any details of how the data structures used as concepts attach to the physical or virtual world they are about. As these programs are often billed as language understanding systems, this either means that even computer scientists do not believe that machines can have intentionality, or that their hope is that language attaches to the world independently of language users. The first belief seems like giving up the quest for human level intelligent machines without a fight, whereas the second has only led to systems that need human interpreters for input and output. It is hard to see how either justification allows room for autonomous language using machines.

2.2 Requirements for Conceptual Attachment

Accepting the position that concepts are mental constructs of the language user, there are two overarching requirements for a theory of concepts. It must detail the structure of a concept and how it can be used computationally, and it must tell a story about how this mental structure attaches to the world through the language user's perception and action. These two requirements are inextricable, and as we will see in the following it is a mistake

to treat one without the other.

2.2.1 Sensory Grounding

The most straightforward attempt to fulfill the two requirements of detailing the structure of concepts and explaining how they are attached to the world consists of taking an existing theory of concepts, such as the definitional one, and adding a link from each symbol to some sensing machinery. This is exactly Harnad's scheme [Harnad, 1990]. Here, a set of basic symbols is supported by sensory categorization machinery – a connectionist network, in Harnad's proposal. Each of these symbols thus becomes connected to the world by its categorizer's ability to pick out members of the concept from sensory input. There are a number of systems that have been built according to this paradigm, including some of our own [Roy *et al.*, 2002; Roy, 2002]. These systems often address additional aspects of this type of grounding, such as how categories are learned by the language user [Roy, 2003] and how concepts grounded in this way can be combined [Gorniak and Roy, 2004].

There is no denying that sensory grounding is a very important aspect of language understanding. One can note immediately, however, that the systems resulting from this paradigm are severely limited in several ways. While it is to be expected that at least initially only basic words can be directly grounded in sensory perception, it is more interesting that these systems are almost entirely without a purpose of their own: they are programmed to interpret or produce descriptions, and have no choice but to interpret or produce descriptions. Practically, thus, these systems are severely limited by having no desires of their own, and no ability to model other's desires. Responding to anything other than hardcoded commands, or acting autonomously to reach goals, are thus out of the question. In fact, even representing actions and understanding language about actions is at best an afterthought in these systems.

There are a few examples of language using systems that use words to label actions. Some of them simply perceive actions of others and label them, making them in a way equivalent to systems that label other features of the world they perceive [Siskind, 2001]. Other systems are based purely on a representation of action, but do not perform what we would normally call perception - in fact, while they represent action they do not actually act themselves, but rather understand language in terms of their built-in action representations [Bailey, 1997; Narayanan, 1997].

2.2.2 Prediction and Interactivity

There remain looming theoretical and practical problems for proposals of language grounding systems that treat perception and action separately. Separating perception and action leads one to see the perception process as a type of encoding. That is, a mental representation of an external object functions as a representation due to the fact that there is a close correspondence between object and representation. The theoretical objection levelled by critics like Millikan and Bickhard is that any correspondence-based theory does not allow room for errors [Millikan, 1993; Bickhard, 2001]. A correspondence representation either corresponds or does not correspond to an object – there is no sense in which it corresponds to the wrong object. Being wrong, however, is clearly a feature of mental representations, one that is crucial for basic processes like reasoning and learning.

This objection may seem unimportant to practical systems. After all, several of them learn correspondences without problems. It is important to note, however, that when they learn, the correctness of the correspondence is judged by a human observer. Take, for example, Toco, a small robot that learns an audio-visual word dictionary by listening to a teacher's speech and observing various objects named somewhere in the speech [Roy, 2003]. The robot is equipped with an algorithm that lets it find the English sounds for words like "ball" or "red" from the speech stream, together with their visual correspondences. The fact that this is the *correct* correspondence only stems from the fact that its designer meant for Toco to learn English words corresponding to visual features. If Toco randomly paired sounds with visual features, or, for that matter, if Toco were to spontaneously start dancing while reciting Kant in the original German, this would not be wrong behaviour for Toco itself, but only wrong given the context of Toco's purpose imposed externally by its designer. The problem that stems from viewing representation as correspondence is thus very similar to the original problem of designing language using systems that do not connect words to the world: such systems need a human being as a judge of whether their actions are correct and meaningful or not. Meaning, therefore, stays external to the system.

Now, perhaps this is not a problem. Millikan argues that normative function, which lets you say that a heart is *for* pumping blood and that "red" *is meant* to refer to a specific perceived colour is determined by evolutionary history for natural systems. Similarly, we could argue that Toco's use of "red" is correct if Toco was designed to learn English. Here we run into another problem, however. While it is imaginable that we design a system that uses the

English language with greater sophistication than Toco, what we are after is ultimately a system that uses a natural language just like human beings use a natural language. We are thus attempting to align our design process with the outcomes of evolution. While this is abstractly true, it does not provide much guidance for the detailed design process.

A more useful view of normative function, put forward by Smith and Bickhard, proposes that representation arises from the need for prediction, and stays intimately coupled with prediction [Smith, 1996; Bickhard, 2001]. This view is useful, because it allows for normativity within the conceptual system itself: the system makes a prediction based upon its representation of the world, and the world either develops according to the prediction, or it does not. Smith goes as far as to claim that this use of mental representations for prediction is what leads to the distinction between subject and object, between representer and represented in the first place. He argues that subject and object must engage in an *intentional dance* to be subject and object: the subject must internalize some structure of the world and make a prediction based on this internalization, and the world has to be structured such that it allows for predictions to be successful, thus becoming an object. Most importantly, representation is cast as an active process that fuses perception, representation and action into a unified conceptual system where one cannot exist without the other two.

There are aspects of this unified conceptual structure in some artificial systems: some systems learn by reinforcement, thus evaluating their predictions and model of the world based upon feedback from the world [Sutton and Barto, 1998]. However, especially in symbolic reasoning systems, which language using systems must on some level be due to the symbolic nature of language, it seems a common trend to separate perception and action. As discussed, these systems assume a more or less simple correspondence between perception and the symbols used, and maintain a separate system for making decisions and acting, if they have one at all. While the perception and the action systems obviously communicate in some way to produce reasonable behaviours, this communication is an afterthought. I have argued in this chapter that the link between perception, representation and action should be the central design issue for a language using system. To move forward based on this premise, however, we need a basic representation unit that can be used for perception and prediction. In the following, I argue that the notion of an *affordance* is exactly such a unit.

While the theory introduced in the next chapter is general in nature, it should be seen as a proposal and outline with partial support from the implementation and studies that follow in the subsequent chapters. Many of the linguistic aspects of the implementation are

simple, and blatantly ignore discourse history to focus on taking into account intentional and physical history. This is a deliberate decision, because discourse history has been proposed as a way to analyse intentions and recover plans before, whereas intentional and physical history has been left unaddressed [Allen and Perrault, 1980; Litman and Allen, 1984; Stone, 2001]. This decision means that treatment of anaphora, and linguistic analysis in general are somewhat simplistic in favour of emphasizing the connection to the situation model given by the physical and intentional analysis. Similarly, while suggestive and used as a model to analyse and predict human behaviour in this thesis, there is not yet evidence for the proposed theory beyond the studies shown here, so its psychological reality should be considered in that light.

2.3 Existing Computational Approaches

Winograd's SHRDLU was one of the first situated language understanding systems [Winograd, 1970]. In fact, it still stands today as one of the most sophisticated ones, without much followup work to surpass it. SHRDLU uses a relatively static, symbolic representation of the situation and keeps the user's plans distinct from the physical (logical) situation. Plans in SHRDLU are only implicitly encoded in the form of procedures applied due to the language used. In the work presented here, the situation includes a noisy estimate of the language user's plans in a highly dynamic situation. The situation thus requires categorization and representation in order to be tied to language, which in turn requires interaction and prediction on the part of the language understanding system. SHRDLU thus commits to the problematic assumption of the separation of linguistic concepts from the world they are about that was discussed in the previous sections.

Chapman's work describes a semi-autonomous agent in a game that follows simple linguistic instructions [Chapman, 1991]. While touching on elements of interaction and planning, this work de-emphasizes the linguistic component in favour of focusing on a model for interactivity. This thesis expand on that work by introducing a strong language element to cast the elements of interactivity and prediction themselves as the basis for a linguistic system.

In our own work, we have introduced both visually situated language understanding systems [Gorniak and Roy, 2004] as well as interactive conversational robotic systems [Hsiao

et al., 2003]. While grounded in visual perception and pioneering the linguistic parsing and incremental grounding strategies that lead to the work under discussion here, the former only considers language with little purpose. All of its interactions are pure visually referring expression with the single purpose of communicating their referent. Here, I propose that determining the purpose behind an utterance is of prime importance to understanding its meaning. Along similar lines, our robotics work has led Roy to propose a theory for grounding linguistic concepts in physical interaction [Roy, 2005]. That work complements that presented here as a proposal for linguistic meaning based on interactions with the world at a far more detailed and fine grained of experience than considered here. In the future, we hope to give an account that encompasses both the level of representation discussed there as well the more abstract and broader interactions under investigation here.

Modern non-situated spoken language understanding systems usually attempt to fill slots in queries necessary to perform database retrievals, such as for providing flight or weather information [Zue *et al.*, 2000; Schwartz *et al.*, 2004]. Any recognition of the user's plans relies solely on the language used, and does not take into account the dynamics of an evolving situation - mainly because these systems do not allow the situation to evolve in interesting ways. Similarly, it is unclear how to tie a richer yet static database of symbolic knowledge such as Cyc to the dynamic situations in which most speech occurs [Lenat, 1995]. Similar to the work presented here other authors have proposed plans and plan recognition as important elements for language understanding [Pollack, 1986; Litman and Allen, 1984; Allen and Perrault, 1980] . Again, however, their view of plan recognition includes only the language used in a discourse, not the dynamic physical and intentional external situation of the discourse.

Horswill's approach to merging symbolic computation with realtime perception and robotic action has an important parallel to the work described here in its use of tags [Horswill, 2001]. Tags connect different parts of the framework by marking operations as being about the same object or role. Implicitly, this connects a set of operations that can perceive, predict and act on a structural element of the world, making them similar to the notion of perceived affordances introduced here. The difference lies in the fact that affordances are represented explicitly and based on plan recognition in the work presented here, whereas they connect different sub-systems in Horswill's work. Having different representations of the same concept is an important topic not addressed here, but having an explicit structural element called an affordances allows more sophisticated reasoning and prediction of the

type employed in language understanding in the following chapters. The two contributions are thus complimentary, and it stands to reason that the higher level reasoning and understanding employed here would benefit from integration with a lower level perception and action system in the case of a robotic platform.

In the field of language parsing, there exist many efforts to map parse structures to logical form [Zettlemoyer and Collins, 2005; Haddock, 1989; Schuler, 2003]. While this problem is very similar to the mapping performed here between parse structures and functional call structures, it is only one aspect of the overall problem of taking an interactive external situation into account to perform language understanding. None of these works address anything but static, symbolic (usually logical) language groundings.

The implementation introduced here relies on hierarchical plan recognition based on observing a sequence of actions given a generative model to perform planning. While much work and many systems exist that produce hierarchical plans given goals, especially in the popular framework of HTN (Hierarchical Transition Network) planning [Erol *et al.*, 1994; Nau *et al.*, 2003], there exists considerably less work on applying similarly expressive and structured models to probabilistic plan recognition. Except for the use of parsers employed in the work presented here [Bobick and Ivanov, 1998; Pynadath and Wellman, 2000], the use of Abstract Hidden Markov models has been suggested, which does not produce the type of modularity required here [Bui *et al.*, 2002]. A promising new candidate is Geib and Goldman's execution model based plan recognition framework, which maintains pending action sets that could be used instead of the Earley state sets on which the work here is based [Geib and Goldman, 2005]. The advantage of a plan library based approach using HTN style methods would be a better parametrization of the plan library, and thus easier creation of and reasoning about possible plans.

Except for work explicitly related to planning and plan recognition, some authors have proposed other predictive representations for learning and acting. Drescher uses structural elements that assemble themselves into hierarchies while interacting with a simple world [Drescher, 1991]. While strongly related to the notion of affordances used here, this work does not connect to language and it is unclear how it scales to a problem of the size tackled in the studies presented in later chapters. The work does contain many insights into how affordances might be learned and organized by interacting with a situation. More recently, Littman *et al.* have proposed a stochastic representation of an agent's state based upon predictions of the outcome of a series of actions the agent could take [Littman *et al.*, 2001].

These proposed representations are promising candidates for computational instantiations of affordances. However, in the implementation presented here we rely on a known plan recognition paradigm that is suitable for the complexity and structure of the scenario investigated. In other situations, for example in the robotic case where action and perception are unreliable, but plans may be less complex, these other ways of working with affordances may be more suitable.

Finally, there exists work on computationally modelling affordances more abstractly as a theoretical tool to explore linguistic mechanisms [Steedman, 2002], as well as in a non-linguistic setting to model a robot's interactions with the real world [Stoytchev, 2005]. While both research areas are relevant to the work presented here, they do not address the need for a theory linking perceived affordance to linguistic concepts in an implementable fashion. They do, however, suggest other ways to encode and reason about affordances, which could enrich the work presented here in the future.

Chapter 3

The ABC Theory

The theory of Affordance-Based Concepts provides a solution to the problem of concept detachment outlined in the last chapter. The nature of its basic units, perceived affordances, ensures that it provides the linked triplet of perception, representation and prediction at the most basic level. The theory therefore produces concepts connected to the real world in the strongest possible sense, doing away with problems of passive perception and lack of normativity. I describe the theory in this chapter, and a computational instantiation that captures many aspects of the theory in the next chapter.

3.1 Affordances

The last chapter ended by pointing out the need for a mental structure that incorporates perception, representation and prediction aspects into a coherent unit. This section introduces the notion of a *perceived affordance* to fulfill this need.

3.1.1 Affordances and Perceived Affordances

The term *affordance* was coined by Gibson in 1977 [Gibson, 1977]. Working in the field of visual perception, Gibson was responding to what I have called correspondence theories of perception. Rather than focusing on image-like representations that are similar to, or

correspond to, the light information impinging on the retina, he proposed that perception encodes what the external world affords the perceiver. Thus, extended surfaces are perceived to provide support for walking on, if the surface is of an appropriate size relative to the perceiver and sturdy enough to hold the perceiver's weight, and the perceiver is actually able to walk. However, affordances are not necessarily perceived. They are relationships between an actor and the environment embedding the actor that hold independently of the actor perceiving them. I therefore distinguish between affordances and perceived affordances – those that the actor perceives and thus mentally represents.

Affordances are unique in that they are primitive aspects of the physical makeup of the world that are neither objective nor subjective. They span the objective-subjective boundary. There is no sense in which a chair affords sitting on, unless we assume someone who is doing the sitting relative to the chair: the sitter must be of the right size and weight to get onto the chair and be supported by it. Thus, a human sized chair affords sitting for an adult human actor, but not for a horse. A chair might also afford picking up and throwing for adult humans, but not if it is bolted to the floor. The set of all affordances of an individual in an environment contains all possible interactions of the individual with the environment. This set is not identical to the set of perceived affordances of the individual. Neither is the set of perceived affordances a subset of the set of all affordances, because the individual may be wrong about what the environment affords it. If a person attempts (and fails) to sit on a cunningly designed object that looks like a wooden chair but is actually made out of paper, the individual perceived an affordance that did not actually hold.

Perceived affordances, as I have described them here, fulfill the requirements of a representation I arrived at in the last chapter: they are the product of perception of the world, they encode some aspect of the structure of the world relative to the perceiver, and they predict a possible interaction between perceiver and world. By implying a prediction, they can be falsified. However, not every wrong perceived affordance must be falsified. If in the preceding example the perceiver never decides to use the prediction and does not sit on the paper chair, the perceived affordance, though wrong, will never be falsified. The distinction between true and false perceived affordances is not necessarily a binary one. Agents may have degrees of belief in the validity of perceived affordances, and in fact the implementation presented in Chapter 4 maintains exactly such degrees of belief.

32

3.1.2 The Structure of Perceived Affordances

An affordance concerns possible interactions between an actor and an environment, and an interaction necessarily includes a temporal element. Given a joint state of actor and environment an affordance is a possible future interaction and thus concerns at least two points in time: the current moment, and the future point of interaction, which may also be extended in time. Remember that affordances in general are not representations - they are sets of possible interactions and thus exist simply because of the physical state of the system that includes the state of the environment and the state of the actor – in short, because of the *situation*. Here, we are more interested in perceived affordances, which are mental representations, and thus must be finitely describable without requiring a complete description of the situation. Due to what Smith calls the *flex and slop* of the world, namely the property that in the macroscopic world of everyday experience effects die off with distance, it is generally possible to produce a state description of the situation that suffices to make good predictions without describing it completely. The *Markov Assumption* of a state in a model proposes much the same thing: that it is possible to predict the future behaviour of the system given only a simplified encoding of its current state. Perceived affordances thus include an encoding of some aspects of the current situation. There are many examples of such state encodings in the current day literature concerning decision making for artificial agents [Boutilier *et al.*, 1999].

In addition to a state encoding, an affordance predicts a possible interaction. This prediction may be representationally explicit, such as a list of possible ways to pick up a cup, or it may be implicit, such as an encoding of the cup's geometry together with a model of possible hand movements and configurations. Both representational styles have their place at different levels of affordances. It seems unlikely that a list is a good way to represent the myriad ways to pick up a cup, but it may serve well for thinking about what to have for breakfast. In general, as Minsky points out, there are many styles of representation that are amenable for different ways of thinking about different things, or thinking differently about the same thing [Minsky, 1985]. As long as they encode state and serve to predict possible interactions, they are candidates for affordances.

An affordance addresses the possible action prediction problem at a single level of representation. In the previous example, the possible ways to pick up a cup and the choice of breakfast foods are on very different levels of representation. They are connected, however,

in that a possible breakfast choice may include pouring a cup of milk, and thus picking up a cup. To make mental representation feasible it is important to keep these levels of affordances related yet distinct. Keeping them distinct allows one to reason on a single level, to achieve more concise yet still approximately Markovian state encodings and to employ the representation and reasoning methods that are best for that level. Keeping them loosely connected, on the other hand, allows for predictions that span levels and lets one fill in the details of high level plans, creating a hierarchy of perceived affordances.

3.2 Affordance-Based Concepts

3.2.1 Objects

Note that so far I have not invoked the notion of objects per se – perceived affordances are about the structure of the world that can be exploited to make predictions. These structures can be below the level of everyday objects, for example when they concern the topology of a graspable surface, which may or may not be part of a larger structure that we usually label "cup". Having replaced the notion of objects with the notion of structural elements called affordances, we can now re-introduce objects as bundles of affordances. Due to the distinction between affordances and perceived affordances, we need to distinguish between objects and concepts of objects. A cup becomes the set of interactions a cup affords, as determined by its physical properties and the agent's abilities. Due the the subject-relative nature of affordances, objects thus only exist relative to subjects. Here, as in Smith's metaphysical view, an object is only an object due to its being pinned down in the structure of the world by a subject's possible interactions with the object. When we engage in an active process of representation to distinguish objects within the structure of the world, we carve out a set of local affordances in the world and consider it an object. This process is not arbitrary, however, as it exploits the pre-existing structure of the world, including our own abilities. Thus while concepts of objects are the product of our perception, representation and actions, and while we may decide to cut up the world into different sets of objects at different times, we are externally constrained in our object categorizations by our own structure and that of our environment. In that sense, objects exist in the world. Affordances and perceived affordances thus jointly address the metaphysical and the psychological aspects of the existence of objects. The world must provide affordances to allow

representation of objects by a subject via perceived affordances, yielding concepts of objects. As this thesis concerns a theory of and mechanism for concepts, which are assumed to be mental structures, the psychological claim is of prime importance here.

3.2.2 Concepts and Compositions

Concepts of Objects are instances of the more general class of structures I call concepts. Each concept is a bundle of perceived affordances. In addition to representing concrete everyday objects at various levels, as described in the last section, concepts can represent any other sets of structures in the world. Allowing arbitrary bundles of affordances gives the Affordance Based Concept theory a unique representational power, but the use of affordances imposes limits as it is subject to constraints imposed by the framing structure of subject and environment. One aspect of this power is the ability to represent abstraction. If a bundle of affordances corresponds to the perceived possible interactions with a particular red cup, one meter in front of the subject, filled with hot tea, it is only a matter of adding and dropping other affordances in a coherent manner to arrive at interactions possible with hot tea cups of all colours, filled and non-filled cups, cups I drank from yesterday, containers, objects that have handles, cup-sized objects and physical objects in general. This is not a claim that the way in which affordances are added and dropped is simple, but we will see in the next chapters that such feats of abstraction can indeed be instantiated computationally. Importantly, however, using affordances gives the power to perform these abstractions, yet maintains the notions of prediction and interactivity I have emphasized throughout.

Concepts may be labelled or unlabelled if they occur in language using creatures. A dog may have mostly unlabelled concepts, with perhaps a few labels taught to it by interaction with human beings. We, on the other hand, have labels for many of our concepts. Human language is so highly spontaneously productive and shapes our thinking so much that we can come up with labels for many abstract concepts that we have never labelled before. I have already used some such labels ("objects that have handles", "the blue thing for making pizza I gave you yesterday"). It is worth noting, and I will expand on this greatly in the following chapters, that representing concepts as bundles of affordances is amenable to performing linguistically driven conceptual combination. Thus, in parsing "objects that have handles" I propose that we perform an online conceptual specification and abstraction, starting with the sets of affordances labelled by some of the individual words and

combining these according to syntactic and conceptual combination rules to arrive at a set of affordances that is the meaning of the whole phrase. Importantly, also, the concepts attached to each word already contain an immense amount of information about possible interactions, thus suggesting many rich complex combinations, such as in Pustejovsky's examples of "fast car", "fast road" and "fast food" [Pustejovsky, 1995]. Many similar combinations will be computationally explained in the following chapters.

3.2.3   Abstract Concepts, Agents and Subjects

ABCs also extend to non-physical concepts. Some labelled concepts have intuitively clear constraints on interaction possibilities associated with them, such as "mass" or "ease of use". But I believe there is even a story of levels of affordances to be told about a concept like "freedom". As said, I do not claim that a single type of mental representation suffices to account for all possible levels and types of affordances. The following chapters introduce one type of framework to maintain hierarchical levels of affordances and to perform language understanding in terms of these affordances. Some meanings of a word like "freedom" might be representable in that framework, within the limited domain addressed. We will need to develop a representationally richer framework that relies less on explicit enumeration of affordances to cover the full human meaning of a word in terms of affordances. I do believe this is possible in the framework of possibilities of interaction, and not out of reach computationally.

Objects with agency, be they human beings, animals, machines or other things one may want to assume the intentional stance towards [Dennett, 1989], can be treated exactly like other objects under the ABC paradigm. However, the possible interactions of one agent with another are often far more extensive than those between one agent and a non-agentive world structure. For human beings, this includes being able to speak to other human beings, folding the full interactive richness of natural language into the affordance framework. A special set of affordances forms the concept of the subject, the "self". It is clear that this set is distinct from those representing other agents, though similar in many ways. As our own affordances are quite similar to those of other agents, many of the same interaction possibilities exist in subject-subject interaction. Thus we can convince ourselves to do certain things, we can analyse our own abilities; in short, we can conceptualize ourselves. This level of reflection about one's own abilities is one of the most powerful aspects of human

thinking. All the aspects of Affordance-Based Concepts, from linking perception to prediction to supporting abstraction ("I'm good at thinking on the spot", "I have awful hand-eye coordination") play into the full range of human thought and consciousness necessary both for human thinking and for building human-level artificial thinking machines [Singh, 2005; Minsky, 1985; Minsky, to be published].

Chapter 4

An Implementation of the ABC

I now turn to a computational implementation of the Affordance-Based Concept theory. The implementation described here features all the aspects of ABCs:

- predictive units that capture the possible interactions at a given level of abstraction

- a hierarchy relating affordances at different levels of abstraction

- a mechanism to track the current situation in terms of perceived affordances of all levels

- a set of functions to form and combine concepts from the past and current perceived affordances

- the necessary relationships linking linguistic structure to ABCs to decode language into concepts given a situation.

As the first of its kind, however, the implementation is limited in scope. While its mechanisms are general and should be transferable to any domain and situation, it achieves coherent treatment of hierarchical perceived affordances through uniformity: each affordance is represented in the same way. While this particular representation is useful for a number of problems and domains, I claim in no way that perceived affordances should actually be uniformly represented, or that they are so in human beings.

4.1 Hierarchical Plans

This implementation of affordances hinges on the notion of a hierarchical plan. A plan is a sequence of one or more steps an agent takes or considers taking. A hierarchical plan is a plan in which a top level node is expanded out into lower level nodes, with the leaves of this plan forming a non-hierarchical plan of concrete steps the agent can actually take. We explicitly or implicitly maintain hierarchical plans all the time, such as when planning to buy milk, which expands into going to the store and purchasing milk, which in turn expands into walking to the car, getting in the car, driving to the store, and so on. Hierarchical plans have the advantage of making some independence assumptions: if your goal is to buy milk, how you get to the store does not matter – you could walk, drive or bike.

Plans and planning are intimately related to perceived affordances. In fact, perceived affordances are the basis for planning. The current situation must contain an affordance predicting I could go buy milk, as otherwise I would not plan for it. Similarly, I will only consider driving to the store, at a different level of affordances, if I actually have access to a car, and if my encoding of the situation contains the perceived affordance of driving. Perceived affordances are thus not the elements of a plan, but at each step they are the possible choices a planner faces when making decisions. Thus each planner must maintain sets of affordances to perform its planning, and a hierarchical planner maintains hierarchical trees of affordances.

Note that the activities of planning and plan recognition are tightly coupled. In fact, as soon as there are two agents involved in a plan, the two activities become one and the same - to plan for two people, each individual must recognize the other individual's plan and incorporate it. In the implementation presented here, I focus on hierarchical plan recognition, because it allows me to model two people's intertwined affordances, model their concepts and understand their speech externally. As we will see, however, elements of planning will be necessary to understand language as well, and when building an artificial language using machine, planning takes central stage. I will outline how to proceed to a fully autonomous language using machine after describing the computational modelling of the ABCs of human speakers via plan recognition.

4.1.1 Probabilistic Context Free Parsing

The machinery used in representing affordances in the implementation presented here is that of context free parsing, so this section gives a brief introduction to the relevant notions. A Context Free Grammar (CFG) is described by a set of rules of the form $X \rightarrow Y$ where X is a single symbol called a non-terminal, and Y is a string of symbols. Any symbol in Y (the *tail* of the rule) that does not appear on the left side of an arrow in the set of rules (is not the *head* of a rule) is called a terminal. Rules should be interpreted as re-write rules: X can be re-written as Y (or Y as X, depending on the direction of analysis). In a context free grammar the fact that every rule can only have one non-terminal as its head enforces that X can be replaced with Y independently of what symbols occur to the left or to the right of X, independent of X's context. Given a string of terminal symbols, the basic task in using a grammar is to apply re-write rules starting with the string of terminal symbols until a pre-specified top-level symbol, S, is produced. This process is called *parsing* and the tree of symbols produced due to rule applications is called a *parse tree*. Note that the combination of a given terminal string and a given grammar can produce many parse trees (a *forest*) due to ambiguity. There are a number of efficient parsing algorithms, which work either as described by starting with S and expanding it (*top-down*), or by starting with the given terminal symbols and applying rules by replacing the tail with the head until the top level symbol is produced (*bottom-up*), or a combination of top-down prediction and bottom-up parsing [Collins, 2003].

By making the same context-free assumption in a probabilistic context, namely that rules are expanded independently from each other given a non-terminal during the parsing process, a CFG parser can be turned into a Probabilistic Context Free Grammar (PCFG) parser by adding a probability p of rule expansion to each rule,

$$P(X \rightarrow Y).$$

The probability of a parse tree T is then given as

$$P(T) = \prod_{P(X \rightarrow Y) \in d(T)} p$$

where $d(T)$ is a derivation of the terminal string consisting of a sequence of rule appli-

41

R_RETRIEVE_KEY	→	R_ROOM_1_TO_ROOM_2 R_OPEN_CHEST R_TAKE_KEY
R_ROOM_1_TO_ROOM_2	→	I_MAKE_DOOR_PASSABLE R_ROOMCHANGE_ROOM_1_TO_ROOM_2
R_ROOMCHANGE_ROOM_1_TO_ROOM_2	→	R_THROUGH_DOOR R_ENTER_ROOM_2
I_MAKE_DOOR_PASSABLE	→	I_PULL_LEVER O_OPEN_DOOR
I_MAKE_DOOR_PASSABLE	→	I_BREAK_DOOR
I_MAKE_DOOR_PASSABLE	→	I_UNLOCK_DOOR I_OPEN_DOOR
R_OPEN_CHEST	→	R_UNLOCK_CHEST R_LIFT_LID
R_OPEN_CHEST	→	R_BREAK_CHEST

Table 4.1: Sample Plan Recognition Grammar Fragment

cations that produces T. There can be multiple derivations producing the same parse tree (depending on the order of rule applications), so usually a uniquely identifiable derivation is picked amongst them. Often this is a *leftmost derivation*, where the leftmost terminal is always replaced first. The likelihood of the terminal string with symbols $x_1...x_n$ given a grammar G, on the other hand, is the sum of all $P(T)$:

$$P(x_1...x_n|G) = \sum_{d(T)} P(T)$$

where $d(T)$ are the possible leftmost derivations producing $x_1...x_n$ from S. Using a PCFG instead of a CFG has the advantage that $P(T)$ can be used to distinguish between a set of possible parse trees, and that $P(x_1...x_n|G)$ can be used to compare the likelihood of different terminal symbol sequences.

The whole point of context free parsing is to recover hierarchical structures from a sequence of non-hierarchical observations, so it comes as no surprise that context free grammars, and especially PCFGs have been suggested as ideal paradigms for performing plan recognition [Bobick and Ivanov, 1998; Pynadath and Wellman, 2000]. In this case, the symbols in the terminal string correspond to observed events in a temporal sequence, and the grammar specifies possible higher level event structures. Let us turn to a simplified example from the studies that will be described in the next chapter. The example involves two players, Roirry (prefix 'R') and Isania (prefix 'I'), that engage in the short sequence of events depicted in Figure 4-1. Isania pulls a lever to open a door, and Roirry goes through the door and fetches a key from a chest in the next room. Table 4.1 shows a small grammar fragment covering this example event trace. Given the observation sequence given in Figure 4-1, a context

$0:$ $_0$ I_MAKE_DOOR_PASSABLE $\rightarrow$. I_PULL_LEVER O_OPEN_DOOR

$0:$ $_0$ I_MAKE_DOOR_PASSABLE $\rightarrow$. I_BREAK_DOOR

$0:$ $_0$ I_MAKE_DOOR_PASSABLE $\rightarrow$. I_UNLOCK_DOOR I_OPEN_DOOR

$0:$ $_0$ R_RETRIEVE_KEY $\rightarrow$. R_ROOM_1_TO_ROOM_2 R_OPEN_CHEST R_TAKE_KEY

$0:$ $_0$ R_ROOM_1_TO_ROOM_2 $\rightarrow$. I_MAKE_DOOR_PASSABLE R_ROOMCHANGE_ROOM_1_TO_ROOM_2

$1:$ $_0$ I_MAKE_DOOR_PASSABLE $\rightarrow$ I_PULL_LEVER . O_OPEN_DOOR

$2:$ $_0$ I_MAKE_DOOR_PASSABLE $\rightarrow$ I_PULL_LEVER O_OPEN_DOOR .

$2:$ $_0$ R_ROOM_1_TO_ROOM_2 $\rightarrow$ I_MAKE_DOOR_PASSABLE . R_ROOMCHANGE_ROOM_1_TO_ROOM_2

$2:$ $_2$ R_ROOMCHANGE_ROOM_1_TO_ROOM_2 $\rightarrow$. R_THROUGH_DOOR R_ENTER_ROOM_2

$3:$ $_2$ R_ROOMCHANGE_ROOM_1_TO_ROOM_2 $\rightarrow$ R_THROUGH_DOOR . R_ENTER_ROOM_2

$4:$ $_2$ R_ROOMCHANGE_ROOM_1_TO_ROOM_2 $\rightarrow$ R_THROUGH_DOOR R_ENTER_ROOM_2 .

$4:$ $_0$ R_ROOM_1_TO_ROOM_2 $\rightarrow$ I_MAKE_DOOR_PASSABLE R_ROOMCHANGE_ROOM_1_TO_ROOM_2 .

$4:$ $_0$ R_RETRIEVE_KEY $\rightarrow$ R_ROOM_1_TO_ROOM_2 . R_OPEN_CHEST R_TAKE_KEY

$4:$ $_4$ R_OPEN_CHEST $\rightarrow$. R_UNLOCK_CHEST R_LIFT_LID

$4:$ $_4$ R_OPEN_CHEST $\rightarrow$. R_BREAK_CHEST

$5:$ $_4$ R_OPEN_CHEST $\rightarrow$ R_UNLOCK_CHEST . R_LIFT_LID

$6:$ $_4$ R_OPEN_CHEST $\rightarrow$ R_UNLOCK_CHEST R_LIFT_LID .

$6:$ $_0$ R_RETRIEVE_KEY $\rightarrow$ R_ROOM_1_TO_ROOM_2 R_OPEN_CHEST . R_TAKE_KEY

$7:$ $_0$ R_RETRIEVE_KEY $\rightarrow$ R_ROOM_1_TO_ROOM_2 R_OPEN_CHEST R_TAKE_KEY .

Table 4.2: Earley States for the Plan Parsing Example

free grammar parser would recover the parse tree shown in Figure 4-2.

Figure 4-1: Sample Event Trace
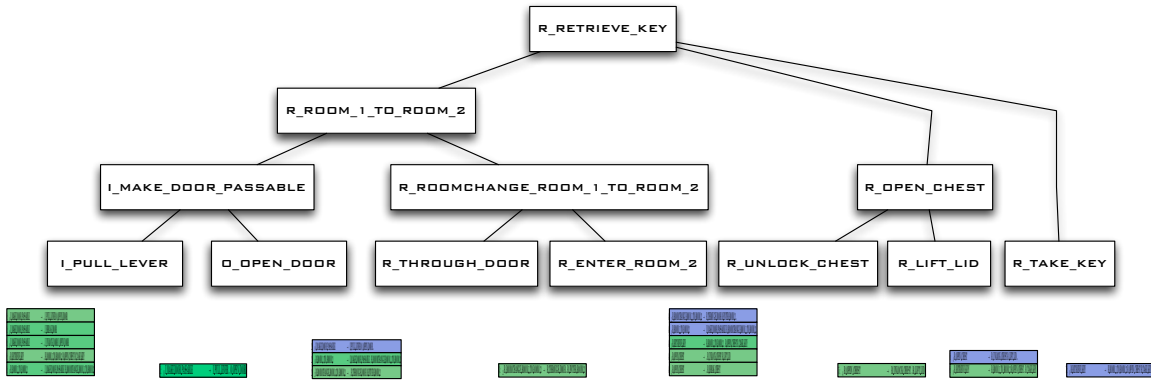
Figure 4-2: Sample Plan Parse Tree

4.1.2 Probabilistic Earley Parsing

I claimed before that a planner must consider affordances when making decisions. A plan recognizer has the luxury of a known terminal string (or at least a set of observed terminal strings). Many parsers take advantage of this fact by constraining their search for possible parse trees to only include those that actually include the terminal string. Any bottom up parser does this, because it incrementally combines symbols to form higher level structures, starting with the terminal symbols. However, such a parser also considers possible subtrees that cannot be used to derive the S symbol. For example, if we add the rule PASS_DOOR $\rightarrow$ O_OPEN_DOOR R_THROUGH_DOOR to the grammar fragment in Table 4.1, a bottom-up parser would construct a constituent PASS_DOOR, but never use it in the derivation for R_RETRIEVE_KEY. In contrast, a pure top-down parser would expand I_BREAK_DOOR, even though the terminal symbols do not end up supporting this sub-tree. In short, different parsing algorithms produce the same answers in terms of possible parse trees, but they do so with varying efficiency and by maintaining different internal states. As we aim to use the internal states of a plan recognizer to represent a set of affordances, we need to be careful to select an algorithm that does predict all possible interactions at all levels at any given point in time, but that uses the symbols observed to constrain its search. The ideal candidate for an efficient parser along these lines is an Earley parser, which performs a combination of top-down prediction and bottom-up completion of parse trees to optimize its search behaviour [Earley, 1970].

An Earley parser is based on the notion of an Earley state, a structure that concisely sum-

marizes the state of the parser at a particular point in the observation sequence, and at one level of the current parse. An Earley state is denoted as

$$i : \,_k X \to \lambda.\mu$$

which should be interpreted as the fact that when the parser was parsing position i in the observation sequence $x_0...x_{i-1} \, x_i...x_n$, it had started expanding non-terminal X at position k in the observation sequence, and that in using rule $X \to \lambda\mu$ it had advanced past λ in the tail of rule as indicated by the dot. For example, Table 4.2 shows the state sets an Earley parser would produce while producing the parse tree in Figure 4-2. These state sets are also visually represented as colour coded stacks below the leafs of the parse tree in Figure 4-2. Each state that has the dot to the right of the rule, meaning that it has successfully completed the rule, is coloured in blue, whereas states that still have predictions pending are coloured in green. The same colour scheme will be used to visualize more complex plan parses in the next chapter. In short, at any given position i in the parse, the Earley parser is predicting a set of next symbols, namely the symbols to the right of a dot in the set of states at i (from those states coloured in green). However, the parser does not produce all top-down parse trees, because it uses already present states to predict future states. Thus, a non-terminal will only be expanded at a given position if it occurs to the right of a dot, and each possible symbol will be only expanded once at a given position because the Earley parser re-uses produced sub-trees in a dynamic programming fashion.

An Earley parser can be turned into a probabilistic Earley parser by adding two quantities to the state description for state S:

$$i : \,_k X \to \lambda.\mu \ [\alpha, \gamma]$$

The quantities α and γ are called the forward and inner probability of an Earley state, respectively [Stolcke, 1995]. The forward probability (a misnomer, as it is an expected count, due to possible recursion in the grammar) represents the expected number of occurrences of a given state in state set i after symbols $x_0...x_{i-1}$ have been parsed. The inner probability is the probability of the parser being in the given state after parsing $x_k...x_{i-1}$, i.e. $\gamma = P(x_k...x_{i-1}, S|G)$ where G is the grammar the parser uses.

4.2   Earley States as Perceived Affordances

I now wish to claim that an Earley parse state,

$$i : {}_kX \rightarrow \lambda.\mu \; [\alpha, \gamma]$$

in an Earley parser used for plan recognition is an ideal candidate for a computational manifestation of a perceived affordance. Assuming that the parser is used to recognize the plans of a particular agent, it

- predicts possible future interactions with the world at a particular point in time (the symbols to the right of the dot in the state)

- ranks the likelihood of possible future interactions given the interaction seen so far through its forward probability

- applies to a particular level of abstraction, but is related to other levels due to the hierarchical nature of the grammar

- summarizes a segment of past interaction to predict the future.

As an Earley parser progresses, it maintains complete state sets for each point in time, thus providing a complete history of past actions and predictions in addition to currently relevant predictions. I call the grammar used by this Earley parser an *affordance grammar*. This grammar is a predictive model of the structure of the world, representing a certain agent's predictions about and possible interactions with the world.

In principle, the affordance grammar should include all possible interactions including verbal ones. Giving a command or asking a question is certainly an interaction with the world. In the affordance grammar and the studies presented in the next chapter, however, we face somewhat of a chicken-and-egg problem: using the affordance grammar for plan recognition provides a substrate for language understanding, but we need to understand language to write an affordance grammar that can include verbal actions. Once the initial analysis using an affordance grammar that does not take into account utterances is done, however, it should be possible to extend this grammar with possible utterance actions and treat utterances identically to other interactions with the world. The work presented here does not

include this last step, and thus treats utterances as events external to the affordance grammar. This in turn means that while the meaning of utterances can be resolved in terms of how they express interaction with the physical world, the meaning cannot include linguistic interactions such as commands or descriptions. These are therefore handled externally to the affordance parsing process in the current implementation. It should also be noted that other work exists that deals with the effect of past utterances on the understanding of future utterance [Litman and Allen, 1984], in fact, past utterance are often the only type of situation taken into account by other language understanding systems. I therefore intentionally focus the work here on taking into account the extra-linguistic situation first and foremost, rather than the linguistic one.

4.3   Concise Environment Descriptions

In a human being, I assume that the ability to perceive affordances partially developed through evolution, and is expanded and adapted via lifetime learning. While the representation for affordances presented in the preceding sections is amenable to learning, that is not the topic of this thesis. Instead, the many rules for the affordance grammar used to derive Earley states are specified concisely via a rule generation system. The rule generation system produces a full set of rules capturing the hierarchical structure in possible event sequences, so that events and sub-events can be recognized and predicted at varying levels of description. The generation system works from a set of meta-rules that concisely specify 1) the essential events of interest and the sequence in which they must be observed to form higher level events, 2) the hierarchical relationships between these events, 3) the times and types of possible extraneous event structures within other events (note that what is extraneous to recognizing one event sequence may be the core of another), 4) the physical structure of the space (e.g. room connectivity) and 5) the parameterization of event structure (e.g. which actors can be involved in which events). These aspects of the plan recognition problem are interrelated; for example, the physical space structure determines possible temporal event structures. However, specifying these constraints in relative isolation in a meta-language lets the designer work in terms of intuitive constraints on the events being modelled, and leaves the generation of the large space of detailed grammar rules from this specification to the machine.

47
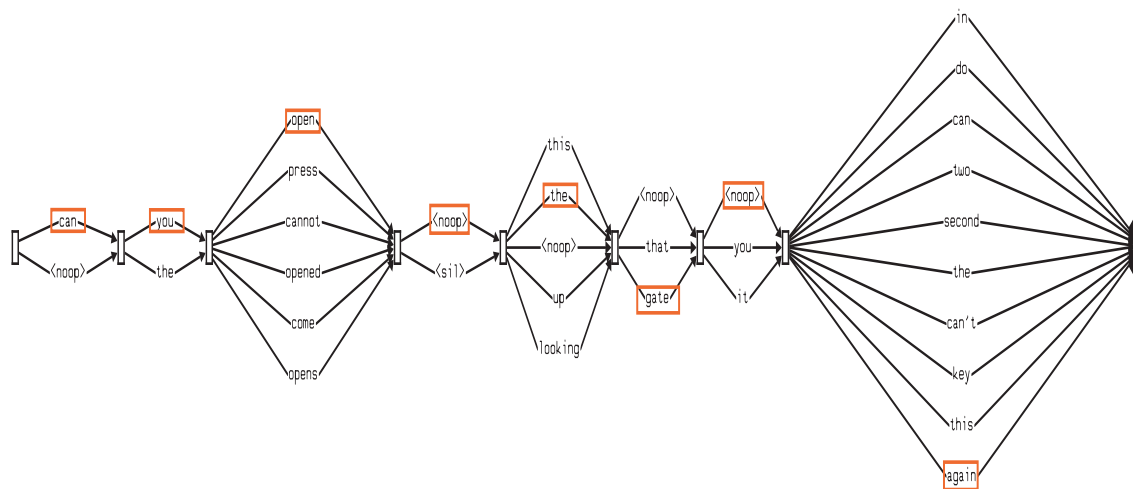
Figure 4-3: A part of a confusion network produced by Sphinx 4 for the utterance "Can you open the gate again".

4.4 A Framework for Understanding Situated Speech

Having identified probabilistic Earley states as the implementational correlates of perceived affordances, it is time to embed them in a framework for speech and language understanding. This framework maps the language signal, be it speech or typed text, onto Affordance Based Concepts using Earley states as affordances. The language signal is often ambiguous both in form (what was said) and content (what was meant). The goal of the Framework for Understanding Situated Speech (FUSS) presented here is to resolve these ambiguities without over-committing by discarding interpretation options at any stage of processing.

4.4.1 Speech and Text Recognition

The FUSS uses the Sphinx 4 speech recognizer[1] as a speech front end. I have augmented this speech recognizer with *confusion network* generation facilities. Confusion networks are compact representations of possible hypotheses [Mangu *et al.*, 1999]. Each link in the network is called a confusion set and spans exactly one word slot, containing all words that might have occurred over that period based on the speech recognizer's acoustic and

---

[1]http://cmusphinx.sourceforge.net/sphinx4/

language models. Each word hypothesis is associated with a corresponding posterior probability, where the posteriors of all possible hypotheses in one set sum to one. For the results reported here I used an efficient confusion network construction algorithm based on the maximum a posteriori path [Hakkani-Tur and Riccardi, 2003]. The resulting source code is now publicly available as part of the Sphinx 4 distribution. Figure 4-3 shows part of a network from the data for the spoken utterance "Can you open the gate again." Nodes are shown in order of decreasing probability from top to bottom with the correct node highlighted in each confusion set. "<noop>" and "<sil>" are special words that stand for a possible word skip and a silence word, respectively. The example shows that the correct word is often not the one with the highest probability, and that confusion varies from a single word choice to more than 10 choices.

Typed text is usually a less noisy signal signal than speech, but often contains spelling errors and out-of-vocabulary words. Similar to the way it handles speech, the FUSS transforms typed text into a confusion network by adding words with small string-edit distances to the typed word to the network link for that word. First, it checks that the word is part of the known vocabulary, and replaces it with the closest word from the vocabulary according to string edit distance if not. It then adds all words within a given string edit distance threshold to the confusion set, estimates probabilities for these words using a value of

$$\frac{1}{2^{stringedit(lw,tw)}}$$

where lw is the vocabulary word considered and tw is the word actually typed (or selected as closest from the vocabulary), and finally normalizes across all words in the confusion set.

4.4.2 Language Parsing

The linguistic parsing step of the FUSS uses the same Earley parser as described earlier. For the noisy confusion networks produced by speech and typed text, a few modifications to the standard Earley algorithm are necessary, some covered by Stolcke [1995]. The parser considers each word in a confusion set at position i as a possible word in that position, and multiplies a state's probabilities by the probability of the word in the confusion set. This incorporates the speech recognizer hypotheses directly into the parsing process and weighs

49
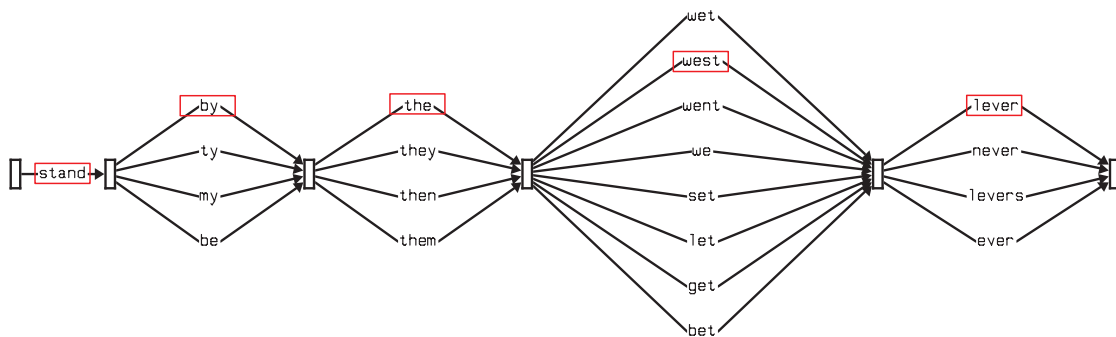
Figure 4-4: A confusion network produced by spelling correction for the utterance "Stand by the wet (sic) lever".

them by the speech recognizer's acoustic and language model, effectively conditioning all probabilities produced during parsing on these models.

As speech and typed text are often grammatically incorrect, the parser seeds each parse position with an initial state (one producing the S symbol), effectively causing the parser to work like a bottom-up parser so that it finds all grammatical substrings of the input. At the end of a parse the FUSS uses the most probable top level state that covers the largest portion of the confusion network as an interpretation of the sentence.

Finally, the framework automatically augments the grammar by splitting each rule $N \rightarrow t$, where t is a terminal, into three rules using a new non-terminal NOOP: the original rule, $N \rightarrow \text{NOOP}\,t$ and $N \rightarrow t\,\text{NOOP}$. The added rules $\text{NOOP} \rightarrow \text{<noop>}$ and $\text{NOOP} \rightarrow \text{NOOP}\,\text{NOOP}$ cover sequences of <noop> symbols. The probabilities of all these rules can be estimated by counting the number of individual and pairs of <noop> symbols along the best paths of all confusion networks. In effect, this allows every terminal to be replaced with any number of skips preceding or following the terminal.

4.4.3 Language Grounding

Whenever the parser produces a state that has the dot to the right of all symbols, that is, whenever it successfully applies a full grammatical rule and thus completes a constituent, it attempts to ground this constituent in terms of ABCs. For this purpose, constituents

$$\text{X} \longrightarrow \text{Y Z}.$$
$$\text{a}(\text{b}(\text{c}())) \blacktriangleleft \cdots\cdots \text{a}(?)\ \text{b}(\text{c}())$$
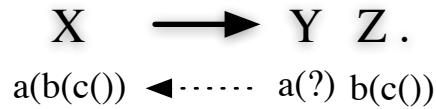
Figure 4-5: A possible composition schema for grounding a constituent in ABCs

can be associated with concept specifications. A concept specification takes the form of a nested function call specification, for example $a(b(c(), d(e)))$ expressing how the current set of perceived affordances is to be filtered to arrive at the ABC for this constituent. Every lexical item can be associated with a non-nested function call specification. Such a specification includes the name of the filtering function to apply to the set of affordances, and the argument positions used in the function call. It is possible to specify any number of function call specifications per lexical item. Upon completion of a grammatical rule, the parser walks along each symbol in the tail of the rule and checks whether the function specification for the symbol together with the arguments offered by the other symbols form a valid nested function call. Every tail symbol must be used, otherwise grounding fails for this constituent and an ungrounded constituent (one without a concept specification) is produced.

Figure 4-5 shows a successful grounding composition where needed arguments are indicated by a question mark. The parser here uses the complete concept specification for Z, $b(c())$ as an argument for Y's $a(?)$ specification to produce $a(b(c()))$ for the head X. Y can either require the argument to $a(?)$ to occur on its right, or leave the argument position unspecified. If a concept specification covers the tail but remains incomplete, for example if the specification for Y in Figure 4-5 was $a(?, ?)$, the parser can produce a still incomplete specification for the head, in the example $a(b(c()), ?)$. Note that composing concept specifications in this way is akin to how syntactic composition is driven in a Categorical Grammar [Steedman, 1988], but only applies to semantic composition as presented here. In my view, the semantic aspect of incremental composition is more important than the syntactic one, and might even explain many of the syntactic phenomena observed [Sedivy *et al.*, 1999].

This method of incremental composition driven by language syntax is akin to other work that associates grammatical rules with lambda calculus expressions [Schuler, 2003] and my own work that performs compositional grounding according to explicit composition rules

51

in the grammar [Gorniak and Roy, 2004]. The loose handling of compositions here has the advantage of pushing all the information into the lexicon, while leaving the grammar untouched. I take advantage of this fact in the next chapter, where the system is trained to use a relatively large probabilistic grammar without having to specify the compositional behaviour of each rule. If multiple interpretations are produced, they most probable longest (in terms of covering the most words in the utterance) candidate is selected. As for using concept specifications instead of the full visually grounded concepts of my previous work: this is simply a matter of efficiency. It was possible to consider all objects at every composition when there were at most 30 objects and few grammatical rules, leading to few completed constituents. Even in the restricted scenario presented in the next chapter, there can be tens of thousands of affordances to be considered, and hundreds of constituents completed during a single parse. It is thus prohibitive to perform a full composition whenever a constituent is completed. The concept specifications produced here are otherwise equivalent, but delay grounded composition until a final constituent is produced, though any other constituent could be explicitly grounded at any time. Interpreting a complex concept specification even with thousands of affordances being considered can be made speedy with suitable indexing of affordances. The interpretation of utterances in the study in the next chapter, for example, runs at a speed suitable for realtime in-game use.

4.4.4   Affordance Filtering

An utterance occurs at a specific point in time, and at that time the plan recognition Earley parsers will have a particular set of current and past Earley states under consideration. To interpret a concept specification, the nested function call they represent is interpreted as an incremental filter on the full set of perceived affordances. Thus, for example, a noun like "gate" might select all interactions involving opening, unlocking, breaking and walking through at all present and past points in time, whereas a verb like "open" might filter these to only include the possible and actual interactions of opening doors. This simple example is shown in Figure 4-6. Figure 4-7, on the other hand, shows the filter expressions from this simple parse tree applied to the previous affordance example. In sequence, the selected affordances for $select(DOOR)$, $select(OPEN)$ and $select(OPEN, select(DOOR))$ are highlighted. Each word specifies the possible number of arguments of the filter function attached to it, and the order of function applications is imposed by the order of rule appli-
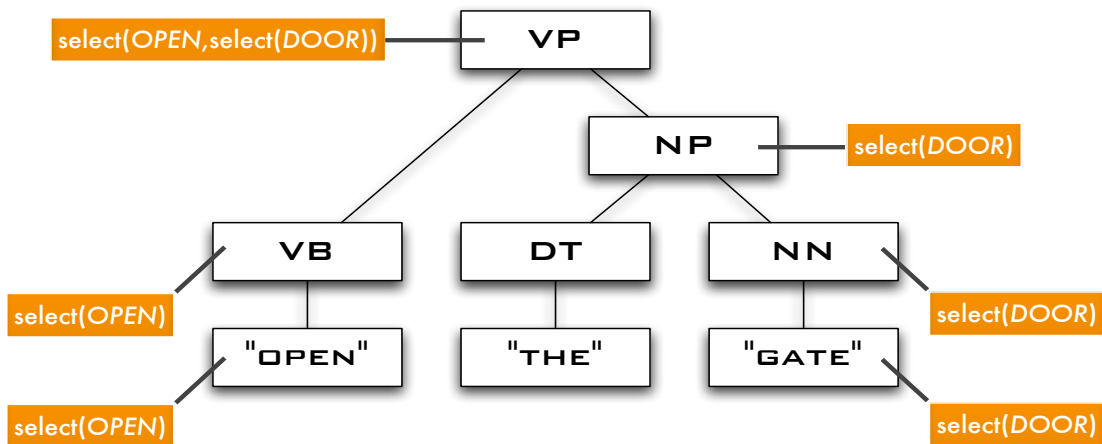
Figure 4-6: Simple parse tree example and affordance filters
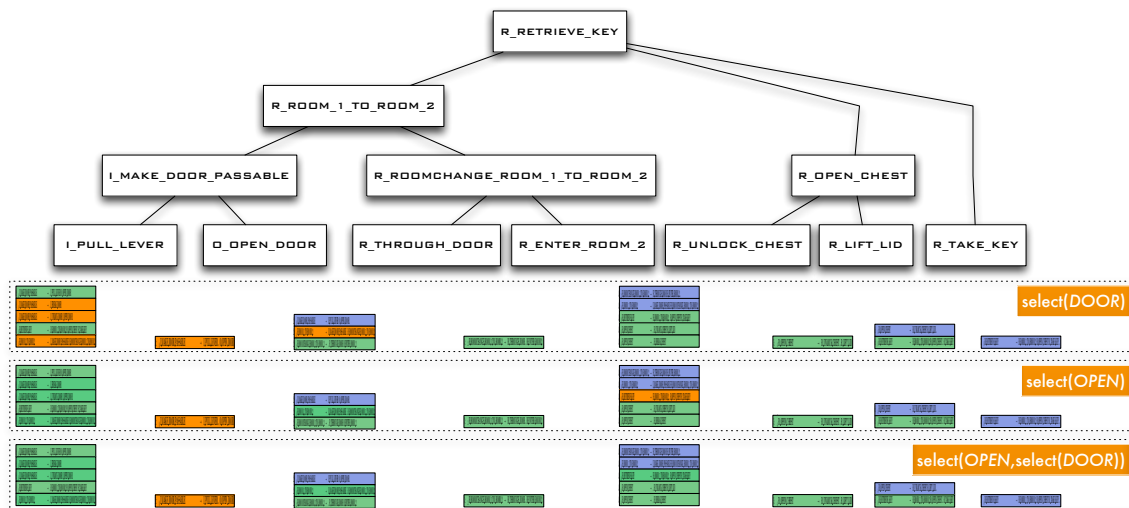
Figure 4-7: Filter functions applied to affordance example

cations. The order in which missing arguments are filled is determined first by the order of occurrence within a rule, and then by rule order. For example, "give" in "give this one a try" will first receive the filter function from "this one" as its leftmost argument, and then that for "a try". Filter functions that are still missing arguments are not considered as arguments themselves. Filter functions can be arbitrarily complex and in the next chapter we will encounter several examples that include filtering relative to a point in time, filtering by changing actor and filtering by planning.

Figure 4-8 shows a more realistic sample parse of the utterance "stand by the west lever" from the studies discussed in the next chapter. It shows both grounded and ungrounded constituents (those with grounding functions shown, and those without.) The figure does not show the probabilities associated with constituents, which are used to distinguish between possible parse trees. Incrementally, the parser builds up a grounding string for the final ("NONE") constituent at the top of the figure, which reads *plan_path(select_location(select(LEVER),.*ROOMCHANGE.*ROOM_1_[0-3]_TO_ROOM_0_[0-3].*))*. The functions involved will be explained in more detail in the next chapter, but in short this nested function call should be read as a filter on affordances that first selects all those concerning levers (including pulling and attacking levers), then amongst these all those that are in a location from which on can walk Eastward, and then plans a path through the game rooms to arrive in such a location.

stand   by   the   west   lever

NONE: plan_path(select_location(select(LEVER),,*ROOMCHANGE.*ROOM_1_[0-3]_TO_ROOM_0_[0-3].*))

VP: plan_path(select_location(select(LEVER),,*ROOMCHANGE.*ROOM_1_[0-3]_TO_ROOM_0_[0-3].*))

VP

FRAG

S

S: plan_path(select_location(select(LEVER),,*ROOMCHANGE.*ROOM_1_[0-3]_TO_ROOM_0_[0-3].*))

PP

PP: select_location(select(LEVER),,*ROOMCHANGE.*ROOM_1_[0-3]_TO_ROOM_0_[0-3].*)

VP

VP: select_location(select(LEVER),,*ROOMCHANGE.*ROOM_1_[0-3]_TO_ROOM_0_[0-3].*)

ADJP

NP

NP

NP: select_location(select(LEVER),,*ROOMCHANGE.*ROOM_1_[0-3]_TO_ROOM_0_[0-3].*)

NP: select_location(select(LEVER),,*ROOMCHANGE.*ROOM_1_[0-3]_TO_ROOM_0_[0-3].*)

NP

NN: select(LEVER)

SBAR: select_location(select(LEVER),,*ROOMCHANGE.*ROOM_1_[0-3]_TO_ROOM_0_[0-3].*)

SBAR

S

S: select_location(select(LEVER),,*ROOMCHANGE.*ROOM_1_[0-3]_TO_ROOM_0_[0-3].*)

FRAG: select_location(select(LEVER),,*ROOMCHANGE.*ROOM_1_[0-3]_TO_ROOM_0_[0-3].*)

FRAG

S

S: select_location(select(LEVER),,*ROOMCHANGE.*ROOM_1_[0-3]_TO_ROOM_0_[0-3].*)

ADJP

ADJP: select_location(select(LEVER),,*ROOMCHANGE.*ROOM_1_[0-3]_TO_ROOM_0_[0-3].*)

S: select_location(select(LEVER),,*ROOMCHANGE.*ROOM_1_[0-3]_TO_ROOM_0_[0-3].*)

ADJP: select_location(select(LEVER),,*ROOMCHANGE.*ROOM_1_[0-3]_TO_ROOM_0_[0-3].*)

ADVP: select_location(select(LEVER),,*ROOMCHANGE.*ROOM_1_[0-3]_TO_ROOM_0_[0-3].*)

S

ADJP

VP

S

PP

SBAR

S

NP

S

NP

ADJP

VP

ADVP

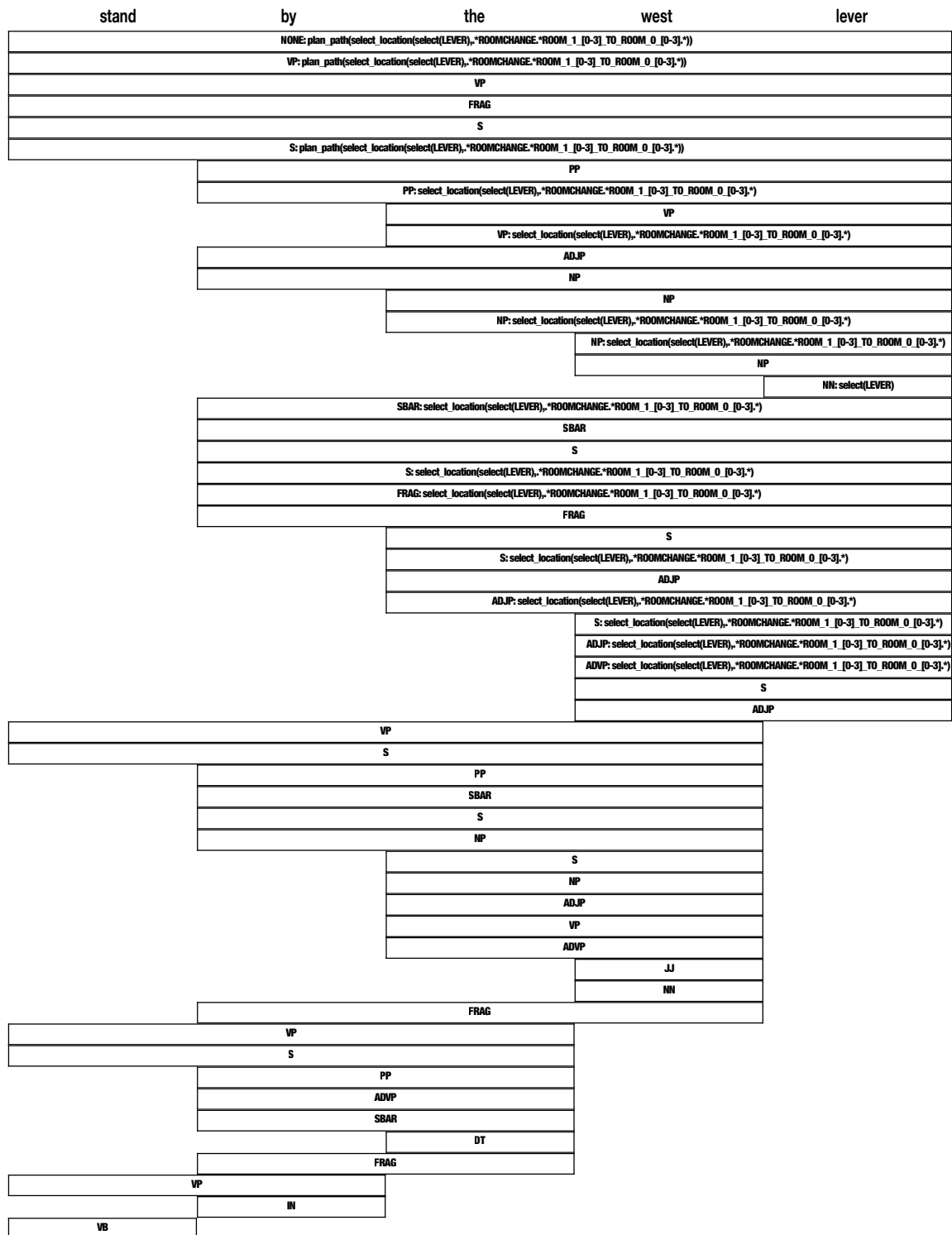JJ

NN

FRAG

VP

S

PP

ADVP

SBAR

DT

FRAG

VP

IN

VB

Figure 4-8: A sample parse of the utterance "stand by the west lever", including grounding functions

Chapter 5

ABCs in Computer Role Playing Games

I now apply the theory and implementation of Affordance-Based Concepts to the problem of understanding situated human language. To do so, it is not only necessary to record and analyse human language, but also to model the situation in which the language occurs using the machinery introduced in the last chapter. In prior work, we have used robots to access the same physical environment as human beings, and have studied language use in this environment [Roy *et al.*, 2002; Roy *et al.*, 2004]. Due to the inherent sensing and action problems robots face, however, such studies are necessarily limited in the complexity of the environment they can model, including limitations to the extent and detail of the physical space, the type of social relationships possible, and the ways in which the robot can affect the world. Here, I turn to multi-user graphical online role playing games to provide a rich and easily sensed world to support and capture human interaction.

5.1   The World of Neverwinter Nights

Current day multi-user graphical role playing games provide a rich interaction environment that includes rooms and exterior areas, everyday objects like chairs, doors and chests, possessions, character traits and other players' avatars. All of these can be acted upon by a player, be it through taking direct action on the world or through speaking with other players. Here, I describe a set of studies using a commercial game, Neverwinter Nights[1], that

---

[1] http://nwn.bioware.com

Figure 5-1: The in-game perspective of a player in Neverwinter Nights.

includes an editor allowing the creation of custom game worlds. A sample in-game view from the player's perspective in this game is shown in Figure 5-1.

I have instrumented the game to record complete transcripts of events in the game world, including player locations, actions such as pulling levers or opening doors, as well as all in-game text messaging between players. Figure 5-2 shows the map used for the study presented here. Dependencies between objects in the map are indicated with dotted arrows. The two players start at the South end of the map. There are two pre-designed in-game characters available for them to play. One of the characters is a rogue, with the ability to pick locks, whereas the other is a monk, who has the ability to destroy doors with her bare fists. However, the rogue can only unlock the doors and chests marked as unlockable on the map, whereas the monk can only break the doors marked as breakable. The levers each open one door for a short period of time, too short for the same character to pull the lever and run through the door him- or herself. Finally, the chests contain a key each, the first unlocking the other chest, the second unlocking the door behind the first chest. The only purpose of the puzzle is to reach the goal indicated on the map. When they start the puzzle,
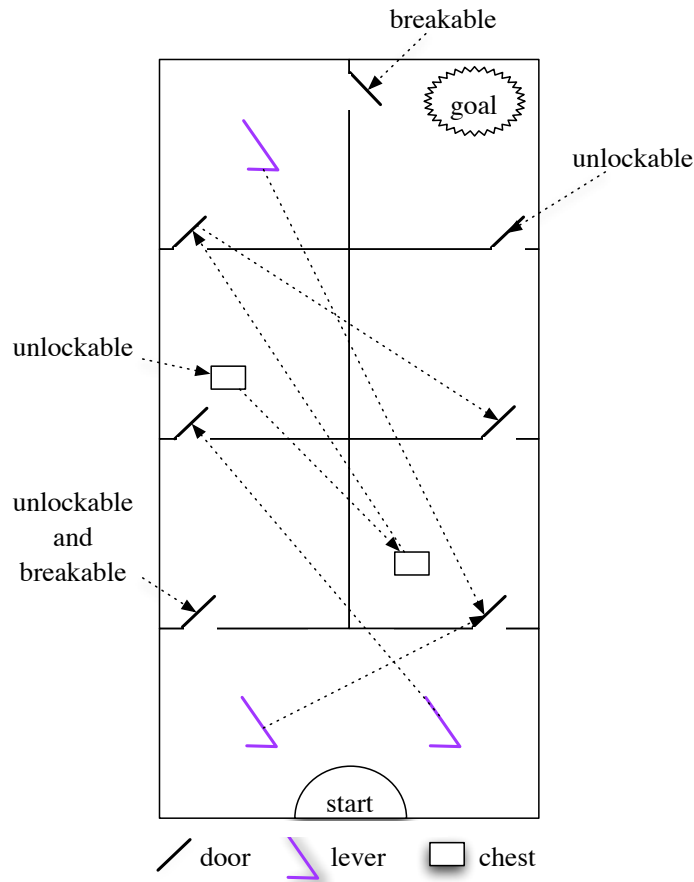
Figure 5-2: The map of the module used in studies.

players only know that there is a goal they need to step on somewhere in the module.

One possible puzzle solution plays out as follows: The rogue picks the lock on the South-West door. The monk opens the next door for him with the South-East lever, whereupon he picks the lock on the chest, obtains the key in it, and returns to the start with help from the monk. The monk now opens the South-East door for him, and he uses the key to open the chest here and obtains another key. Once more with help from the monk opening doors, he makes his way back to the room with the first chest and uses the key in the door leading from it (which also opens the center door in the East.) Opening doors for each other, the two characters now switch places and then reach the goal by unlocking or breaking their respective doors.

This puzzle is designed for players to separate and communicate their instructions and

goals by using language. As an added restriction, one of the players is randomly chosen in the beginning and forced to only used one of the following phrases instead of being able to speak freely:

- "Yes"

- "No"

- "I Can't"

- "Done"

- "Now"

- "What's going on?"

- "OK"

This limits the amount of dialogue phenomena possible, which are not the focus of the study.

5.2   Data Collection and Annotation

The study included 26 players who played in 13 dyads after responding to ads on the bulletin boards on the Neverwinter Nights website. 11 of these dyads completed the puzzle in times ranging from 25 minutes to 1 hour, whereas the others gave up after 1 hour. Even the two incomplete sessions completed most of the puzzle, except for both players entering the last room. While previous studies showed that the FUSS handles speech [Gorniak and Roy, 2005a; Gorniak and Roy, 2005b], this study only collected typed text to focus on the semantic problems at hand. 9 sessions served for development purposes, such as writing the affordance grammar and training the linguistic parser, and a group of 4 sessions formed an unbiased evaluation set. I first annotated the development data and built and trained the system, then annotated the evaluation data and tested on it.

Figure 5-3 shows the interface for browsing and annotating the data. At the bottom of the window we find a panel showing a timeline of the events that occurred during the session.
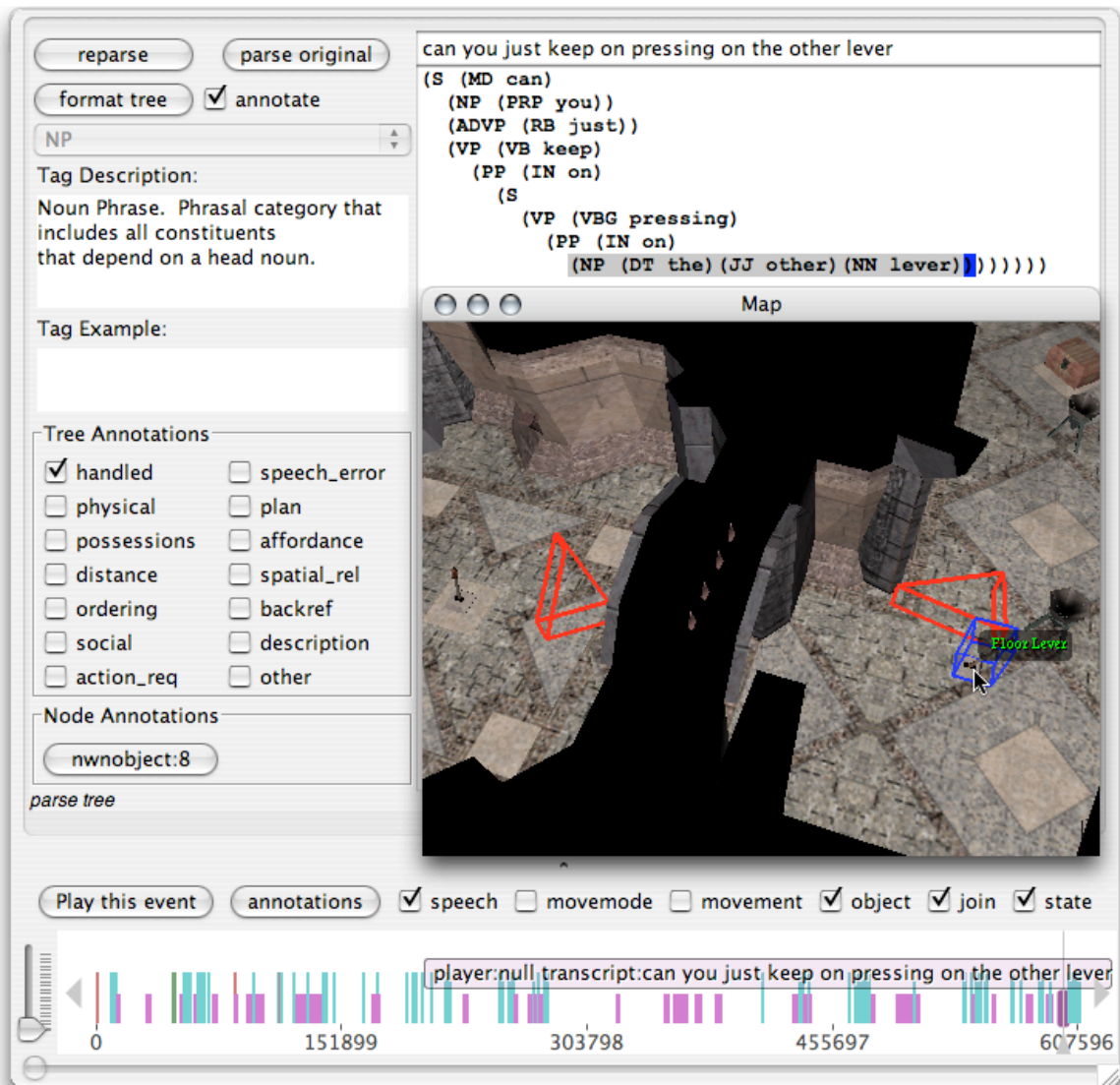
Figure 5-3: The annotation tool used to correct and annotate parse trees.

Events can be filtered using the check boxes above, quickly examined via overlaid information, and the annotator can zoom into and pan across different sections of the timeline. Furthermore, the timeline provides controls to step through a replay of the events, during which audio events are replayed and the map reflects player actions. The map panel is shown above and to the right of the timeline, and shows a picture very close to the one players see during game play, except that the camera can be arbitrarily controlled by the annotator and players are represented by red arrows.

61

The remainder of the window is filled with controls that let the user annotate a specific utterance. In the figure, the utterance "can you just keep on pressing on the other lever" has been selected. Above the map is a parse tree of the utterance. We initialize parse trees with the Stanford Parser [Klein and Manning, 2003] using a standard grammar for written English. This does not capture many of the phenomena encountered in spontaneous, situated text, and so the parse tree panel allows the annotator to correct the parse tree, which is interactively re-formatted. The controls on the left show information about the currently selected syntactic node, and allow for re-parsing of the original utterance. Below these controls are the annotation markers for the current utterance, letting the utterance be described as, for example, an action request. The node annotation button below lets the annotator select a referent for the currently highlighted syntactic constituent. In the case shown, the annotator has selected the noun phrase "the other lever" and used the map panel to indicate the lever this utterance refers to, which is translated into the appropriate reference indicator by the interface. This functionality is unused in the study presented here, because the current study involves the interpretation of complete parse trees rather than specific constituents.

For parsing, the detailed event trace yielded by the game can be abstracted into a simpler trace noting only the relevant changes in world state including

- object interactions (lever pulls, chest use, door interactions)

- room changes

- key acquisitions and exchange

- attempted actions such as attempted unlocks

Table 5.1 shows a sample event trace segment from one session. In this segment, one of the players (player 'R' for 'Roirry', the player character's name) unlocks the Southwest door (door 4), then attempts to unlock the next door (door 7) and fails. Player 'I' (for 'Isania') now first mistakenly pulls the Southwest lever (opening the Southeast door), but then opens the correct door for Roirry by pulling the Southeast lever (lever 9). Roirry enters the next room, lockpicks the chest in it and acquires the key from the chest. Event traces from the study sessions range between 450 and 2000 events in length.

High Level Events
I_ROOMCHANGE_ROOM_1_0_TO_ROOM_0_0
R_ROOMCHANGE_ROOM_1_0_TO_ROOM_0_0
R_ATTEMPT_UNLOCK_DOOR_4
R_UNLOCK_DOOR_4
R_ATTEMPT_UNLOCK_DOOR_4
R_OPENDOOR_DOOR_4
I_THROUGH_DOOR_4
I_ROOMCHANGE_ROOM_0_0_TO_ROOM_0_1
R_THROUGH_DOOR_4
R_ROOMCHANGE_ROOM_0_0_TO_ROOM_0_1
R_ATTEMPT_UNLOCK_DOOR_7
I_THROUGH_DOOR_4
I_ROOMCHANGE_ROOM_0_1_TO_ROOM_0_0
I_ACTIVATE_LEVER_10
O_OPENDOOR_DOOR_6
I_ROOMCHANGE_ROOM_0_0_TO_ROOM_1_0
O_CLOSEDOOR_DOOR_6
O_DEACTIVATE_LEVER_10
I_ACTIVATE_LEVER_9
O_OPENDOOR_DOOR_7
R_THROUGH_DOOR_7
R_ROOMCHANGE_ROOM_0_1_TO_ROOM_0_2
I_ROOMCHANGE_ROOM_1_0_TO_ROOM_0_0
O_CLOSEDOOR_DOOR_7
O_DEACTIVATE_LEVER_9
R_ATTEMPT_UNLOCK_CHEST_13
I_THROUGH_DOOR_4
I_ROOMCHANGE_ROOM_0_0_TO_ROOM_0_1
R_UNLOCK_CHEST_13
R_OPENPLACEABLE_CHEST_13
O_INVENTORY_CHEST_KEY_14

Table 5.1: A Sample Event Trace Segment from a Study Session

5.3   Language and Situation Modeling

The linguistic Earley parser uses a grammar estimated by counting the rules used in the corrected parse trees of the sessions' utterances. The concept specification for the lexical

entries will be further described below.

A set of 90 meta-rules specify the affordance grammar, which captures

- the physical makeup of the puzzle, including room and door connectivity, effects of levers, locations of chests

- the possible actions in every room, including moving to other rooms, pulling levers, unlocking doors, etc.

- planning patterns for players, such as opening a door for the other player to enter a room

- the current state of the world, including which rooms the players are currently in and how much of the puzzle they have solved

The 90 rules expand to a full affordance grammar of about 6500 rules with 1300 non-terminal and terminal symbols. In essence, the meta rules parameterize entities like actors and rooms, whereas the full rule set produces a unique rule for each parameter setting. The lack of parameterization in the actual plan recognition mechanism is one of the shortcomings of using a pure context free grammar parser. However, the parser is efficient enough to run on the large rule set produced by the precomputed parameter expansion employed here. As already pointed out previously, it is desirable to move to a plan recognizer that employs a more concise description of the situation, but none of the existing paradigms near the efficiency and high quality algorithms that exist for parsing. Figure 5-4 shows 4 sample rules from the full grammar. Symbols consist of parts separated by underscores. These rules be read as follows: The initial part of each symbol, if it is **I** or **R** indicates the player performing the action (the character names in the modules are *Isania* the monk and *Roirry* the rogue.) These four rules describe actions assigned to Isania, because their head symbols start with **I**. The heads further tell us that in this action Isania moves from the South-West room (rooms are encoded in Cartesian coordinates, thus this is room 0,0) and moves to the second room on the East side. The last part of the head indicates that while this happens, the other player is in room 0,0. To perform this action, the other player (Roirry) must first open the door leading into room 1,1 (door 6) while being in room 0,0 (this action expands to pulling the South-East lever and the door opening) while Isania must then walk to room 1,0 and then to room 1,1. The last symbol is a roomchange sequence

rather than a simple room change because players can step back out of the target room and into it again before the door closes. By having a symbol for any sequence like this, the whole episode can be classified as a single room change event. The other three versions of this rule displayed here add room specific noise rules in all possible positions. These rules are marked as **NM** to indicate that they do not produce motion (room changes). The rule itself appears, amongst other places, in the tail of NOISE2_R_ROOM_0_0_I_ROOM_0_0 → I_2_ROOM_0_0_TO_ROOM_1_1_O_ROOM_0_0 NOISE2_R_ROOM_0_0_I_ROOM_1_1, showing how room noise rules transition between each other via movement rules.

```
I_2_ROOM_0_0_TO_ROOM_1_1_O_ROOM_0_0  ⟶
  R_2_OPN_DOR_6_R_ROOM_0_0_I_ROOM_0_0
  I_ROOMCHNG_ROOM_0_0_TO_ROOM_1_0
  I_ROOMCHNG_SEQ_2_2_ROOM_1_0_TO_ROOM_1_1_O_ROOM_0_0

I_2_ROOM_0_0_TO_ROOM_1_1_O_ROOM_0_0  ⟶
  R_2_OPN_DOR_6_R_ROOM_0_0_I_ROOM_0_0
  I_ROOMCHNG_ROOM_0_0_TO_ROOM_1_0
  NOISE2_NM_R_ROOM_0_0_I_ROOM_1_0
  I_ROOMCHNG_SEQ_2_2_ROOM_1_0_TO_ROOM_1_1_O_ROOM_0_0

I_2_ROOM_0_0_TO_ROOM_1_1_O_ROOM_0_0  ⟶
  R_2_OPN_DOR_6_R_ROOM_0_0_I_ROOM_0_0
  NOISE2_NM_R_ROOM_0_0_I_ROOM_0_0
  I_ROOMCHNG_ROOM_0_0_TO_ROOM_1_0
  I_ROOMCHNG_SEQ_2_2_ROOM_1_0_TO_ROOM_1_1_O_ROOM_0_0

I_2_ROOM_0_0_TO_ROOM_1_1_O_ROOM_0_0  ⟶
  R_2_OPN_DOR_6_R_ROOM_0_0_I_ROOM_0_0
  NOISE2_NM_R_ROOM_0_0_I_ROOM_0_0
  I_ROOMCHNG_ROOM_0_0_TO_ROOM_1_0
  NOISE2_NM_R_ROOM_0_0_I_ROOM_1_0
  I_ROOMCHNG_SEQ_2_2_ROOM_1_0_TO_ROOM_1_1_O_ROOM_0_0
```

Figure 5-4: A sample of 4 rules from the expanded affordance grammar.

The probabilities for the rules stem from counting the number of rule applications in the maximum likelihood parse trees for the development sessions. Not all of the rules produced by the meta-rules are actually used in the development sessions (remember that rules are produced for all possible parameter settings), therefore two forms of discounting are needed to produce probability estimates for the remaining rules. First, Witten-Bell discounting

assigns probabilities to rules whose heads have occurred, but whose tails have not, by estimating how likely a new rule with this head is to be seen [Witten and Bell, 1991]. This smoothing method uses the number of types of rules with a given head to estimate how likely one is to see another new rule with this head, and divides this probability amongst all the rules with this head that were not seen in the development data. This works for rules whose heads were seen in the training data, but leaves those rules with heads that were not seen. Absolute discounting reserves a fixed probability mass for these rules, and subtracts the mass proportionally from all the rules that were seen or received a probability via Witten-Bell discounting.

5.4   Communication Strategies

Players employ many different types of speech acts to communicate with each other about the puzzle, and each type further subdivides into different strategies for expressing intentions. Broadly, these strategies can be broken down into 3 types of speech acts,

directives "pull the east lever", "open", "go into the room with the chest and the locked door"

descriptions "there's a lever here", "my switch opens your door", "none of these doors can be lockpicked", "I'm in the entry room"

questions "you're not trapped in the west room are you?", "does it open?", "where have you been?"

Players also produce utterances that have little to do with the actual puzzle solution, such as "it's cold and dark in here", "mutter" or "KILL THE PROGRAMMER!". In the following, the focus lies largely on directives because their effect on the second human player is relatively easy to measure, and they are probably the most important category of utterance that a synthetic player would be expected to understand. Furthermore, as pointed out in the last chapter, it is a limitation of the current implementation that the affordance grammar does not include possible interactions via language, because it is used to interpret these interactions in the first place. To distinguish between speech acts within the framework presented here it is necessary to add speech acts as possible interactions into the affordance grammar

itself, so that the system can reason about them. By dealing mainly with directives I avoid this problem for now and interpret the produced grounding for an utterance as a directive by selecting those affordances selected that pertain to the listener (i.e. those the listener could take advantage of at the point in time the utterance occurs) and considering them as likely actions. I do, however, sketch possible ways to interpret descriptions and questions below, after presenting the results on directives.

Players typed a total of 1742 utterances in the development sessions, and 689 utterances in the test sessions. I annotated 1320 of the development session utterances as being on-topic, that is, relevant to solving the puzzle. 302 of these can be considered directives, whereas the remaining utterances are evenly split between questions and descriptions - a distribution to be expected in a puzzle designed to separate players while solving a puzzle. Similarly, the test sessions contain 69 directives out of 427 utterances.

5.5   Affordance Filters

As described in Section 4.4.3, the final result of linguistic interpretation is an affordance filter specification in the form of a nested function call. The affordance filtering process has two stages. First, the final concept specification is interpreted as a filtering function on the current set of affordances, producing another set of affordances that is the interpretation of the utterance at hand in terms of possible physical actions and their abstractions. Second, the utterance is interpreted as a speech act, which involves deciding on the type of speech act and taking any measures to treat it as such, which may involve planning to get the character into a situation in which he or she can perform the action predicted.

5.5.1   Filter Functions

In addition to the affordance set arguments they take as described in Section 4.4.3, filters are further parameterized with static parameters specified in the lexicon to re-use the same filter for different words (for example "east" uses the same filter function as "west" with different parameters). Many words have multiple meanings, of course, even in the limited world of these studies. Some examples of several meanings (for example for "that") occur

67

below, but not all meanings are covered by the system. I discuss failures due to missing meanings in Section 5.6.1.

Simple Selection The simplest filtering function, *select*, selects affordances by substrings in their predicted next symbols. Thus, a word like "open" selects all affordances involving opening of chests or doors.

Actor Selection The *actor_selection* filter can select either the speaker ("I"), the listening character ("you"), or both characters ("us","'s") by filtering affordances for the initial actor string in their predicted symbols.

Indexicality The *expand_set* filter uses the currently predicted set of affordances for the speaker as a source set, and selects a target set selecting either all affordances that specify the same interaction but for any actor. This is the filter associated with the word "this", selecting, for example, all the possible interactions with a lever next to the speaker for the fragment "this lever". For the word "other", the same filter selects affordances of either actor of the same type (e.g. opening doors or pulling levers) that are not currently available to the speaker (that are, for example, not in the current room.)

The *select_distant* filter, on the other hand, collects affordances that were encountered by the speaker at some point in the past and are not available in the speaker's current state. It grounds, for example, one use of "that" as in "What about that lever?" where the speaker is standing next to one lever, but referring to another one with this utterance.

Movement Planning The *plan_path* filter plans a path from the current set of affordances to another by assuming that location changes are enough to bring about the target set. This is largely a valid assumption in the puzzle discussed here: players can usually interact with the things around them, though some plans produced this way may be invalid because the players have not yet advanced far enough in the puzzle. For example, they may not have managed to open a door yet that is necessary to enter a target room. Movement planning takes into account the rules of the puzzle, such that players have to open doors for each other to get into certain rooms. This filter is used for words like "go" (as in "can you go stand by the other lever") or "run". The same planning functionality is also used when interpreting an utterance as a directive, which is discussed below.

Discourse Reference For every utterance, the parser stores the affordance set of the last filter call that filters by neither actor or planning. A back reference filter (*back_ref*) simply re-activates this set of affordance for words like "it".

Past Interactions The *select_past* filter finds those perceived affordances that were actually taken advantage of by the agent in the past. This yields another use of the word "that" as in "Let's try that again."

Location Reference The *select_location* filter selects affordance sets by the possible room changes they predict. This is used, for example, to ground "left" and "West" by selecting for those sets of affordances that predict a room change interaction in which the target room has an x value of 1. Note that this means that locations are defined by how one leaves them (i.e. "west" is a location from which on can walk East.) Again, this is obviously not the most general and only meaning of location references, but it works very well in the scenario discussed here.

Possession Players tend to use "my" and "your" to refer to objects they interacted with recently, thus the *select_recent* filter selects the most recently used affordances in the current set.

Table 5.2 lists the words grounded via filter functions used in the studies, together with the their filter function and the number of arguments they take on the left and on the right.

Word	Function	Left Arity	Right Arity
's	select_actor	0	1
chest	select	0	0
come	plan_path	0	0
door	select	0	0
east	select_location	0	1
east	select_location	1	0
exit	select	0	0
give	intersect	0	2
go	plan_path	1	1
			continued on next page

Word	Function	Left Arity	Right Arity
i	select_actor	0	0
it	backref	0	0
l	select_location	0	1
left	select_location	0	0
left	select_location	0	1
left	select_location	1	0
lever	select	0	0
lh	select_location	0	1
my	select_recent	0	1
north	select_location	1	0
northwest	select_location	0	1
one	select	0	0
open	select	0	0
open	select	0	1
opening	select	0	1
other	select_distant	0	1
out	select	0	1
press	select	0	1
pull	select	0	0
pull	select	0	1
pulling	select	0	1
r	select_location	0	1
right	select_location	0	1
right	select_location	1	0
room	select	0	0
stand	plan_path	0	1
switch	select	0	0
that	select_distant	0	1
that	select_past	0	0
then	select_arg	1	1
			continued on next page

70

Word	Function	Left Arity	Right Arity
this	expand_set	0	0
this	expand_set	0	1
throw	select	0	1
try	select	0	0
unlock	select	0	1
upper	select_location	0	1
use	select	0	1
west	select_location	0	1
west	select_location	1	0
you	select_actor	0	0
you	select_actor	0	1
your	select_recent	0	1

Table 5.2: Words with Filter Functions

5.5.2 Speech Act Interpretation

For a directive, the FUSS first applies the concept specification provided by the linguistic parser to produce a set of affordances grounding the utterance. It then translates the resulting set of affordances into a predicted next action by finding the most recent affordances in the set and checking whether any are also available for the listener in the currently predicted set. If they are, they are turned into the basic actions they predict (that is, actions the player can actually take), by walking down the affordance grammar until a lexical item is reached. If they are not currently available, but are known to be available in other situations, the FUSS will plan a path to the room in which such an affordance would be available, and make the first action in this plan its immediate prediction. Note that such a plan not only includes movement steps, but also the steps necessary to gain passage such as pulling levers to open doors for other players. If no predictions are produced in this way, it might be due to the fact that the next action predicted is not the listener's to take, for example in the case where the speaker must open the door for a listener to walk through. Thus, the FUSS now proceeds with a depth first search for the next action of the listener starting with the currently predicted symbols in the rules contained in the selected affordance states. If any of these steps produce multiple predictions, they are ranked by the sum of the forward prob-

abilities in the Earley states producing them, and the most probable action is used as the prediction.

5.6   Results

Whenever one player gives the other a directive, the utterance is turned into a confusion network and parsed by the language parser to produce an affordance filter specification. The plan recognizer then runs this filter specification on the complete set of affordances produced up to this point in the game, which yields a filtered set of affordances. These are then interpreted as described in the previous section to yield a single best prediction. To measure performance, this prediction is compared to the next action the player in question actually takes, and counted as correct if it matches.

Table 5.6 shows the overall results of language understanding using this method. All results are split between the development and the test set to show generalization to unseen data. The first row (*All Directives (AD)*) shows the performance on the complete set of 302 directives in the development sessions and 69 directives in the testing sessions. However, players do not always follow instructions, so the second row (*Followed Directives (FD)*) shows performance only on the 281 cases where the player actually performs an action that matches the directive as determined by the annotator (64 in the testing session). Half of the directives players used and followed correctly are what I will call *action markers*: single word utterances that do not significantly restrict the nature of the action to be performed, but rather mark the time at which the obvious action should be performed. Such utterances include "now", "go", "lever" and "open". While the high frequency of such action markers supports the claim made here that the interactive situation determines much of the meaning of language (sometime so much that language becomes unnecessary), the performance of the linguistic component of the FUSS is not evaluated in these utterances. *Followed Long Directives (FLD)* in Table 5.6 therefore shows performance on the half of the directives that contain more than one word. The average length of the total set of directives lies at 3.6 words, but rises to 6.2 words when restricted to the set of development directives employing more than one word (4.5 vs. 6.5 in the test set). Performance on the set of linguistically interesting directives is generally lower because the language groundings used in this study do not cover all of the meanings that occur (omissions and problems are discussed further

below). However, the gap to the pure plan recognition baseline widens significantly on this utterance set, showing that the FUSS can understand more complex language and produce the correct concept for many of these directives.

Table 5.6 shows a number of prediction baseline results for the same data sets. The *Hierarchical Plan Recognition* figure shows the performance if language is ignored - that is, if we simply pick the most probable prediction of the plan recognizer at the point an utterance occurs, without paying attention to the words in the utterance. As above, *Plan Recognition (FD)* and *Plan Recognition (FLD)* restrict the pure plan recognition baseline to those directives that were correctly acted upon by the listener (FD), and then further to those that use more than one word (FLD), respectively. *State Based Maximum* counts the actions players took when they were in a specific combination of two rooms, and in response to a directive predicts the action taken most often in this combination. Finally, *State Based Random* randomly picks amongst all the actions players were ever observed to perform in a room combination.

Selected Utterances	Accuracy - Development	Accuracy - Test
All Directives (AD)	70%	68%
Followed Directives (FD)	72%	70%
Followed Long Directives (FLD)	61%	68%

Table 5.3: Results of Understanding Directives in the Neverwinter Nights Puzzle Scenario

Prediction Type	Accuracy - Development	Accuracy - Test
Hierarchical Plan Recognition (AD)	65%	63%
Hierarchical Plan Recognition (FD)	66%	64%
Hierarchical Plan Recognition (FLD)	50%	60%
State Based Maximum (AD)	42%	48%
State Based Random (AD)	15%	17%

Table 5.4: Prediction Baselines for the Neverwinter Nights Puzzle Scenario

When interpreting these results, it is important to keep in mind that perfect prediction cannot and should not be achieved in any of these cases. The puzzle naturally causes much exploration by the players, and, as will be discussed further below, situations and directives often do not limit players to a single next action. Some amount of variability is thus inherent in the scenario.

72% constitutes the best measure of overall performance of the complete system. Given the complexity of the problem and the leeway players appear to give each other in following their own utterance, this figure indicates that the theory and implementation presented in previous chapters make for an effective substrate for language understanding systems.

It is clear from these results that the hierarchical plan recognizer captures important aspects of the puzzle solution: it shows over 20% improvement in predictions compared to a simple predictor baseline. Prediction is also no simple task, as the low random baseline shows (even this baseline does not pick amongst all possible actions, but only those players performed in the development data). Language understanding heavily relies on plan recognition - often the meaning of an utterance is highly constrained by the player's states and plans. Taking the words into account, however, improves again on the pure plan recognition performance. The best measure of this improvement is the 11% gain (8% in the test set) seen when considering the set of correctly followed directives longer than one word. The percentage performance gain is smaller when considering all utterances because performance is dominated by action markers, for which linguistic content plays little role, and thus yields no improvement in performance. Not all action markers are acknowledged by the simple rule of considering one word utterances to be action markers: "go for it", "go go go", and other multi-word action markers occur in the data, but they occur rarely.

Performance on the test utterances is entirely comparable to that on the development utterances, showing that the plan recognition grammar and linguistic parser, while restricted in their coverage, generalize well to unseen data. Of note is that as already discussed, individual sessions differ greatly in playing and communication style. In fact, there is a single session in the test set that contains very repetitive and easily predicted player behaviour. When it is omitted, the test set performance baselines are equals to or lower than the development set baselines.

5.6.1 Detailed Performance and Mistakes

Examining the utterances in detail yields clues as to the benefits and shortcoming of the implementation presented.

Action Markers I call utterances that impose next to no restrictions on the action to be performed via their words *action markers*. The most common ones (about half the data)

are "go", "now", "open" and "lever". There is an external bias imposed favouring "now" because it was one of the only action markers available to the non-speaking character. For this class of utterances, performance of the utterance understanding algorithm can only be as good as predictions made by the plan recognizer. However, the performance figure here also underestimates the performance of the FUSS: it seems that in many cases players do not have an exact action in mind. For example "open" might really be taken to mean "open anything and everything you can" or "open something" in several cases, especially when players cannot see each others' characters. Sometimes players even explicitly indicate this as in "try something else". I will discuss performance of the plan recognizer further below.

Simple Selection Almost every utterance that is not simply an action marker uses at least one content word involving simple selection of affordances (and even an action marker like "lever" or "open" does). The overall performances speaks to the usefulness of the affordance filtering approach in understanding directives in a plan recognition context.

Location Reference These include utterances like "throw the one to the west" and "now head to the east lever". These occur a significant amount in the data (35 utterances in the development data) and are correctly understood if in combination with a simple request. 4 of the 35 are incorrectly understood because they involve constructions or commands not covered by the affordance filters, such as "can you try thief' [sic] picking either the chest or north lock".

Discourse Reference 7 out of 11 uses of "it" (as in "I need you to pull it" in the development data were correctly understood via the *back_ref* filter. The remaining suggest that there are influences on the use of "it" in this context beyond the discourse one.

Indexicality Indexicals including "this", "that" and "other" were understood correctly in half of the cases (14 out of 28). In the 4 (out of 9) misunderstood cases of "this" the mistakes are due to problems with actor attribution, not with indexicality, as they are all of the form "throw it and i'll throw this one" or "let me go down this way once more ... not saying it'll help". "That" is correctly interpreted in 5/7 cases and "other" in 5/12. This only partially indicates problems with their current groundings, as some of the mistakes are due to other words in the utterance such as in "can you

75

try to open from the other side somehow?", which lacks groundings for "side" and "from" at minimum.

Movement Planning Is not only used for phrases like "go to" and "stand by", but also to interpret any utterance that produces affordances not available to the listener in his or her current location. As such, it is involved in understanding most utterances and performs extremely well.

Other communication strategies occurred too rarely to allow for meaningful analysis. There are a few overarching problems and omissions with the implementation presented here.

Missing Meanings There are a few classes of meanings that occur in the data for directives that the implementation currently does not handle at all. There are a number of idioms like "go for it" and "come back" that perhaps should be handled as idioms and not analysed word for word. Sometimes complicated linguistic structures occur, often expressing temporal dependencies and causality. These can even be intermixed with descriptions such as in "I need you to pull it when I open the door for you ... I think it opens the door on the other side". However, constructions this complex are rare.

Spatial Coarseness Spatial locations in the structural grammar are purely room based, and thus relatively coarse. For distance based directives, for example those including "that", utterances can be misunderstood because the player considers him- or herself distant from an object and uses "that", but is still considered to be in the same room as the object by the affordance grammar.

Multiple Interpretations The particular implementation discussed here uses the best interpretation of an utterance exclusively. In previous work we have shown ways to consider multiple weighted interpretations simultaneously by probabilistically mixing the linguistic elements from the language parser with the affordances produced by the structural grammar [Gorniak and Roy, 2005a]. It would clearly be beneficial to adapt those methods to the system described here to consider multiple word and constituent meanings and their interpretations simultaneously.

Learning The paradigm presented here lends itself to supporting learning by a synthetic character. Possible learning targets include the weights and rules of the structural

grammar, the function bindings for words, and the interpretation of words in terms of affordances. Especially together with a coherent framework for considering multiple interpretations such a learning framework would likely improve robustness of the understanding system over the partially handcrafted approach taken here.

Omniscience vs. Player Modelling The plan recognizer used here models both players simultaneously and is informed of the structure of the puzzle. This eases recognition of interdependent actions by the players (such as pulling a lever to let the other person through a door), and increases prediction accuracy by taking into account the actual puzzle structure. However, when interpreted as perceived affordances, the plan states should correspond to those maintained by an individual player attempting to solve the puzzle, not to an omniscient planner for both players. For many directives this is not a problem, because "pull the east lever" can be understood in either model. Problems arise when players are mistaken about how to solve the puzzle, for example when they assume that levers act differently when pulled simultaneously. This presents two problems, one for directives and one for descriptions, discussed below. An utterance like "let's try that again" might refer to the joint action of the characters pulling their respective levers, which is not modelled in the plan used. In the particular puzzle there are few directives of this sort, but the effect on performance of the plan recognizer, which does not acknowledge these falsely perceived structures, may be degrading performance.

Descriptions The second problem with an omniscient plan recognizer is that it makes it hard to interpret descriptions. A player utters a description to inform the other player of the physical makeup of the puzzle ("there's a chest and a locked door in this room"), his or her mental model of how the puzzle works ("they both open opposite doors"), or the effects of actions ("both door and chest remain locked"). Intuitively, each should produce a change in the listener's mental model of the situation: he or she might consider new affordances or discard ones previously thought to be available. As all and only the correct affordances are available in the omniscient plan recognizer, it is impossible to model this effect. However, the filtering mechanisms proposed here lend themselves to exactly this type of effect when run on a different type of plan recognizer – one that is uninformed about the puzzle structure and has limited perception of the other player's actions.

Questions Questions are in content very much like descriptions in the data collected for

these studies, because the listener could respond only with primitive utterances. Thus, they usually read like a description in question form, for example "is the door back there locked?", in effect filling in the questioner's model of the puzzle workings and world state via the response.

Plan Recognition Beside the problem of whether to use an omniscient or several player-specific plan recognizers (or both in tandem), there are other problems with the plan recognizer used here. As Pynadath and Wellman point out, while successful in estimating hierarchical plans of agents, grammar based plan recognizers are not naturally parameterized in an intuitive or useful way. For example, many of the thousands of rules used in the plan recognizer here are due to the fact that they are largely conditioned on the rooms the players find themselves in. Rather than being parameters, these rooms are part of the symbols used in the grammar rules, and are explicitly produced by the meta-rules. The meta-rules are in essence a parameterization of the grammar, but they are not used during the actual plan recognition. To more easily derive and estimate affordance grammars, and also to reason directly about the underlying state variables, it seems advisable to go to a combined model of a grammar and an underlying state model that are linked but represented separately [Pynadath and Wellman, 2000].

5.7   Examples of Utterance Understanding

Figures 5-5 through 5-8 show visualizations of the concepts produced while understanding the utterance "Can you go to the other lever again please?". Each figure shows the full set of grounded constituents produced during the linguistic parse in the upper left hand corner, with the currently selected constituent highlighted in red. The affordance specification corresponding to the selected constituent is displayed at the top of each figure. The main part of each figure depicts the complete set of affordances encountered so far in the session with time running left to right. The utterance in this example occurs at time step 60. Each column corresponds to the set of affordances at one time step, with predicted affordances (those that still have symbols remaining to be parsed) shown in green, and completed ones (those that have been fully parsed and have the dot to the right of the rule) shown in blue. The rightmost column of affordances in each figure corresponds to the time at which the

utterance occurs. The set of affordances corresponding to the selected constituent is highlighted in red, and each figures magnifies an example affordance from the selected set and shows the symbol it predicts. The bottom of each figure notes the symbol that actually occurs at the time step of the magnified example affordance. Note that most of the filter functions only consider predicted affordances, because each completed affordance has a corresponding predicted affordance at an earlier time step.

The character named "Roirry" is the listener in this example. Figure 5-5 shows the concept for "you", which contains all the predicted affordances of the listener at all past and present time steps. For example, Roirry could have attempted to unlock door 6 at a time in the past, but rather chose to use lever 9. Figure 5-6 shows the affordances for "lever" and highlights Isania's option of pulling lever 9. Both "you" and "lever" correspond to simple selections. Figure 5-7 shows the affordances grounding "the other lever". The affordance specification selects those affordances involving levers that were available to the Roirry at some points in the past, but are not currently available to him. "You go to the other lever" in Figure 5-8, finally, plans a path from the current time to a situation in which Roirry could pull the lever selected by "the other lever". Highlighted and magnified is the single affordance corresponding to the first step in this plan, namely a room change into the adjacent room.

5.8 Future Steps towards Individuals' Mental Models

The single most important limitation of the model of perceived affordances presented in this thesis is that it employs a global and correct plan recognizer. While this allows for accurate predictions, it makes the flawed assumption that player's mental models of the situation are omniscient and correct. As pointed out before, this assumption makes it impossible to understand descriptions and questions, because they concern the updating of flawed or incomplete mental models.

A series of steps is necessary to alleviate the problems caused by this assumption. They range from simple extensions to the framework presented here to open research question. They are, however, covered by the general theory presented in earlier chapters. The first step consists of running one plan recognizer per player, rather than one per game. Alone, this step does not change the set of available affordances. The next step is to model players' limited perception by making their access to the world state incomplete. A simple start to

go to the other lever
lever
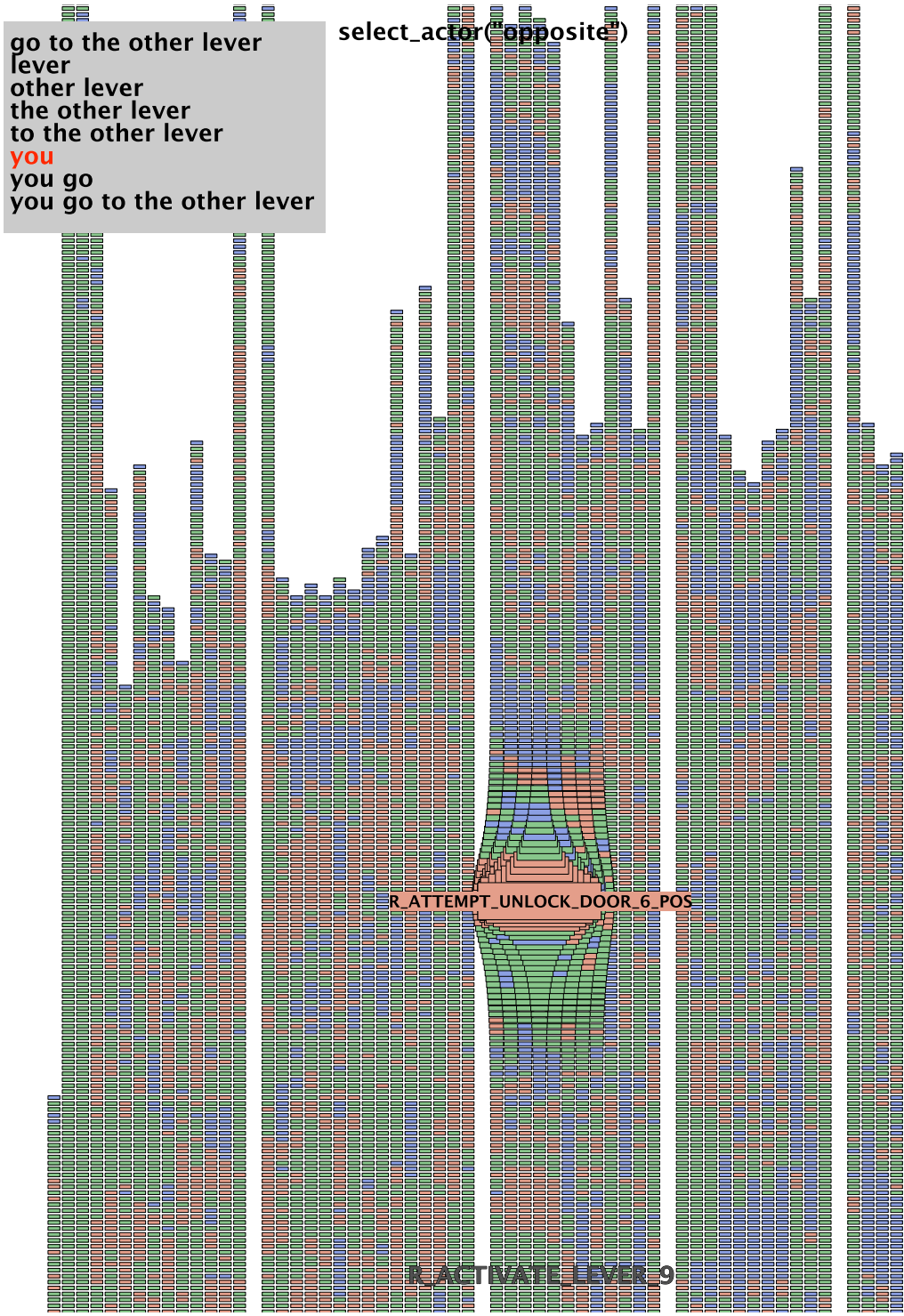other lever
the other lever
to the other lever
you
you go
you go to the other lever

select_actor("opposite")

R_ATTEMPT_UNLOCK_DOOR_6_POS

R_ACTIVATE_LEVER_9

Figure 5-5: Example: Understanding "can you go to the other lever again please?" - *you*

go to the other lever
lever
other lever
the other lever
to the other lever
you
you go
you go to the other lever

select("LEVER")

I_ACTIVATE_LEVER_9_POS

Q_OPENDOOR_DOOR_7
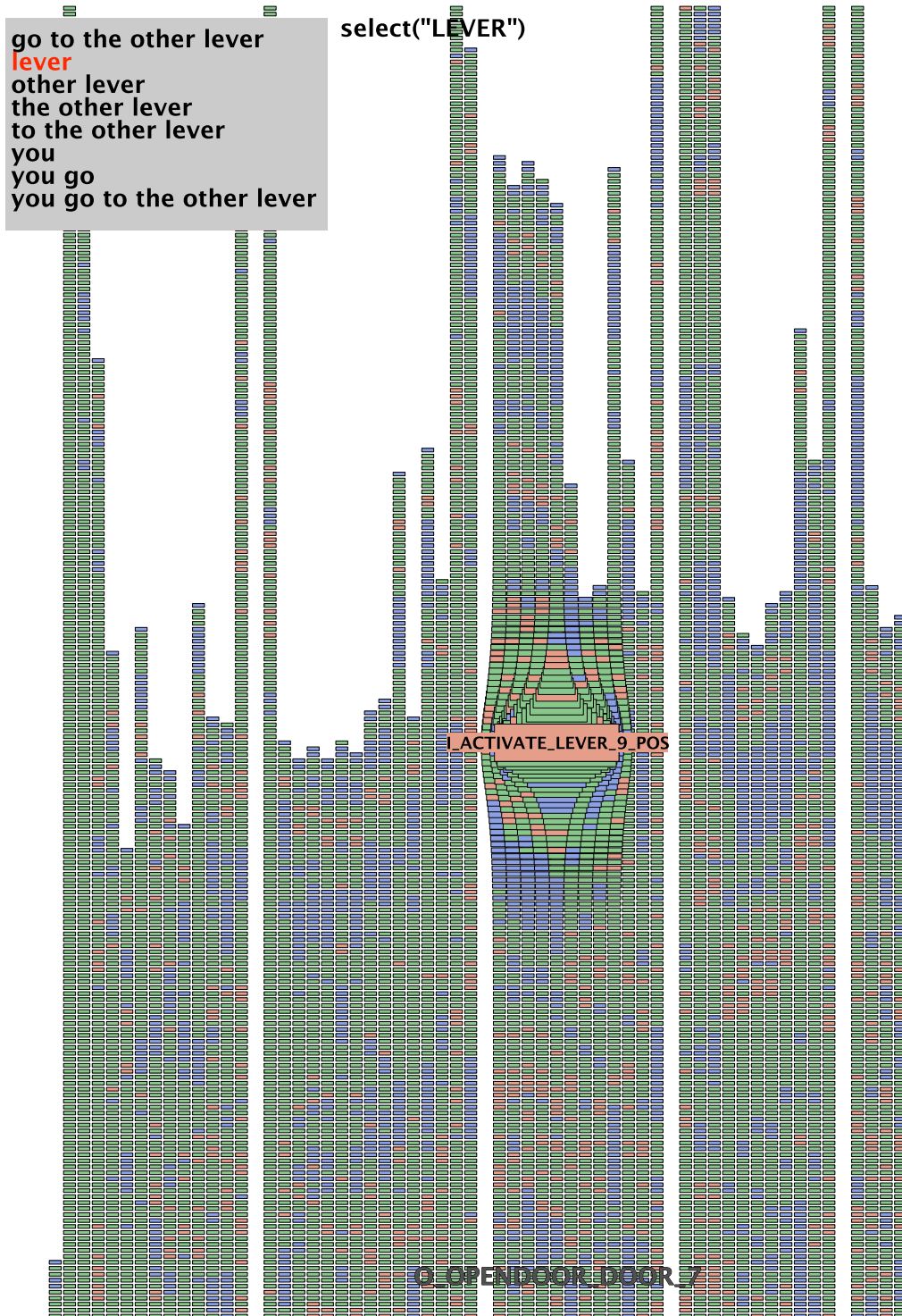
Figure 5-6: Example: Understanding "can you go to the other lever again please?" - *lever*

go to the other lever
lever
other lever
the other lever
to the other lever
you
you go
you go to the other lever

select_distant(select("LEVER"),"opposite")

R_ATTACK_LEVER_10_POS

O_OPENDOOR_DOOR_6
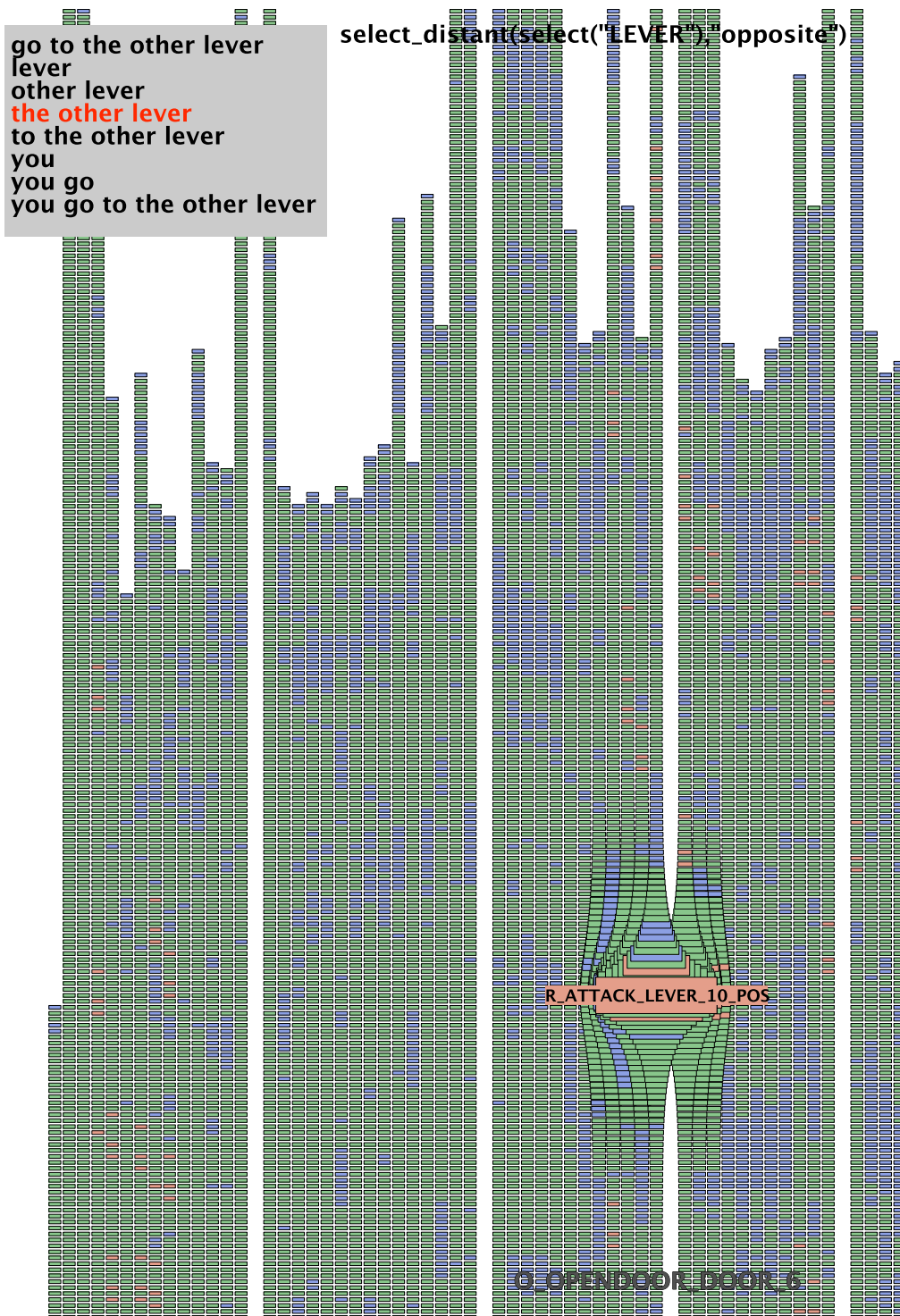
Figure 5-7: Example: Understanding "can you go to the other lever again please?" - *the other lever*

go to the other lever
lever
other lever
the other lever
to the other lever
you
you go
you go to the other lever

select_actor(plan_path(select_distant(select("LEVER"),"opposite")),"opposite")

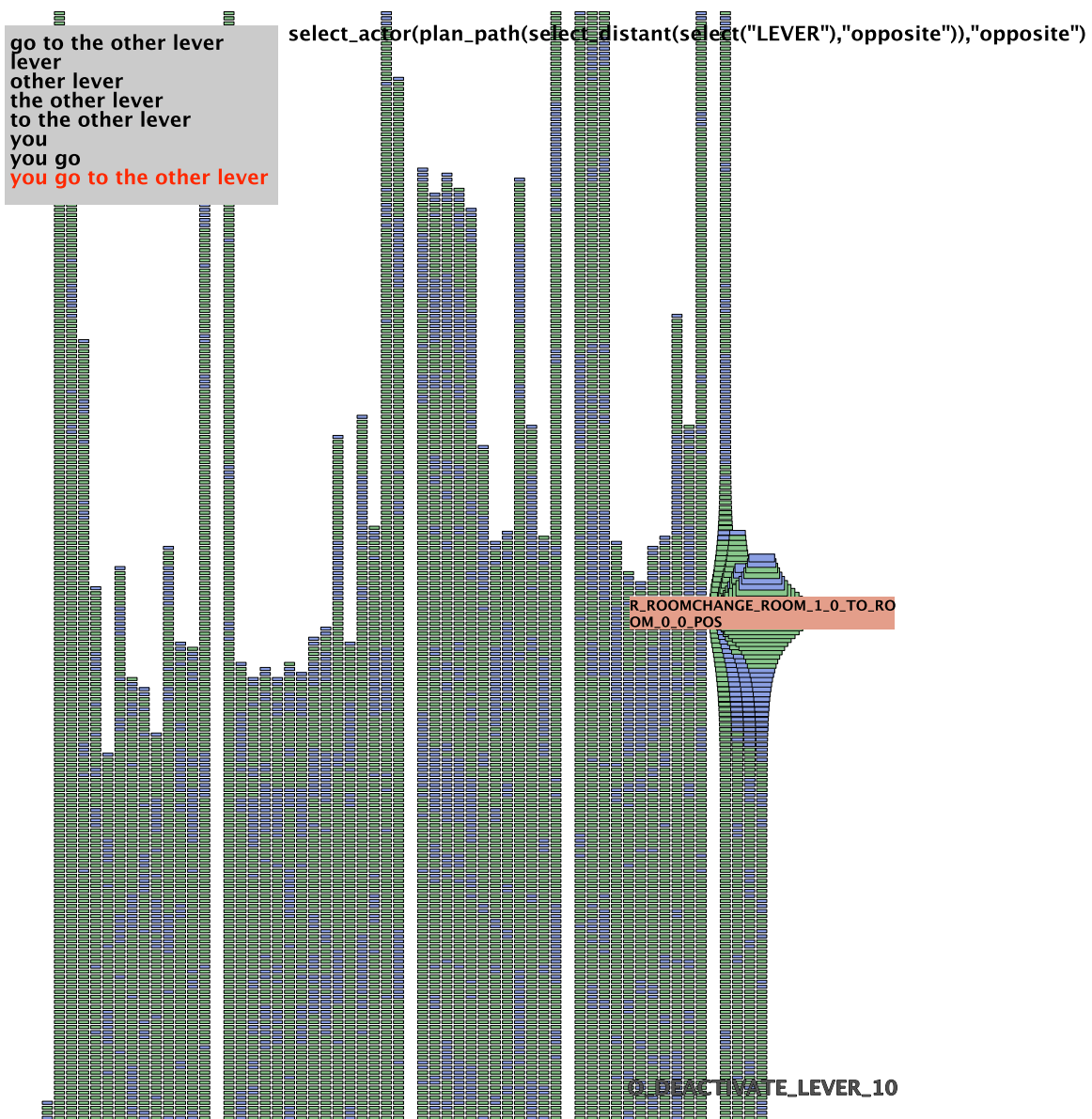R_ROOMCHANGE_ROOM_1_0_TO_RO
OM_0_0_POS

O_DEACTIVATE_LEVER_10

Figure 5-8: Example: Understanding "can you go to the other lever again please?" - *you go to the other lever*

this consists of replacing the symbol string representing events in the game world with a confusion network. When players are in the same room, the confusion sets of this network contain a single member because players can see each others' actions. When they are in different rooms, however, each confusion set representing an action by the other player contains all possible actions currently available to that player. Using confusion networks

83

would spread the probability assigned to the current world state over many possible states as players take actions without seeing each other act. This directly leads to the ability to interpret a subset of descriptions and questions such as "I'm in the Northwest room" or "Did you make it into the next room?". The descriptions would have the effect of narrowing the probability distribution over possible world states by raising the probability of the described state.

A further step towards creating more realistic mental models for players is to allow for lack of knowledge of the structure and functioning of the world. For example, when a player encounters a lever for the first time, his or her affordance grammar should predict a new set of interactions possible at this location, and hypothesize any number of effects these interactions might produce. In the world of Neverwinter Nights the set of possible effects is limited, and in most real situations the effects predicted are produced and constrained by experience. Encountering a switch on a wall leads us to predict only a few effects of the switch with high probability – lights or other electric appliances may turn on or open, but we do not expect the moon to rise or our friends to betray us as a result of flicking the switch. In fact, design meant to suggest and obvious and limited set of predicted interactions led to the introduction of the term "affordance" in the field of industrial design [Norman, 1988]. Predicting new affordances thus implies modifications to the structure of the affordance grammar - new interactions and their effects are predicted, but the possible interactions and most effects are specializations of more general categories of interaction based on experience. One might thus imagine that every new situation adds a set of affordances rules. For example, encountering a lever adds all possible interactions with levers as well as the possible effects of levers in the game, such as opening doors or unlocking chests. Of course, players can now be wrong about how the world works. This step covers utterances such as "there's a lever here", "I was wrong, it doesn't open this door" and "does anything happen when I pull this lever?". Both interaction with the world and utterances by other player can have the effect of pruning the possible affordances of new situations, for example by discovering that a lever pull seems to have no effect in the current room, or being told so by another player.

Finally, there remains the issue of modelling agents, both other players and the player in question him- or herself. Obviously, the perceived affordances captured by the affordance grammar are a model of the player. However, players reflect on and talk about their perceived affordance freely. Utterances like "This isn't working", "I think this lever opens the

South-East door, but I might be wrong" and "this is frustrating" refer not only to the player embedded in the affordances of the physical situation, but also to the thought processes and mental state of the player. Similarly, players share knowledge and comment on each others' mental states. There are two problems to be addressed here: First, utterances by other players should be events that can have all the effects of physical events and more. They can update a player's mental model of the situation, convey the other player's mental model and even communicate meta-comments, for example by categorizing a whole approach or mental model as invalid. Secondly, the model of other players should be rephrased similar to the way the self-model was rephrased here, namely by providing for uncertainty, ignorance and the influence of words. There are a number of open research questions along these lines, such as how to avoid regression in modelling others' models of oneself (though for the game playing purposes under investigation here one or two levels of regression are likely enough) and how to extend to the case where players are not co-operating and might lie. While an answer to these questions will involves machinery beyond the one track plan recognition paradigm presented here, I hope to have convinced the reader that the machinery presented should extend to handle more cases smoothly, and that even where it is flawed the contributions of this thesis in terms of viewing language as filters on the space of an affordance-based representation should underlie further implementational work.

Chapter 6

Conclusion

I hope to have convinced the reader at this point of four things, namely

- that language understanding depends on a mental representation designed for interaction with and prediction of the world

- that the notion of an *affordance* captures the crucial element of a theory of concepts that from the ground up acknowledges the need for interaction with the world

- that affordances make for powerful computational instantiations based on planning and plan recognition and lead to a new method for truly grounded computational language understanding

- and that, by example, this new method can feasibly be implemented and performs well in understanding spontaneous human language in a complex situation.

The implementation presented in this thesis provides a convenient framework for probabilistic hierarchical reasoning about affordances while understanding situated language. It will be important to integrate this framework with other approaches and views on affordances [Steedman, 2002; Roy, 2005] and to re-phrase existing approaches dealing with other aspects of grounded language understanding in an affordance-based framework.

The theory behind the implementation, as presented in the first two chapters of this thesis, I believe to be a fundamentally important and new view of mental representation of

concepts. It is unique in its strong ties to computational language understanding and its successful realization in a language understanding task dealing with spontaneous, situated human language. I hope that this pairing of theory and implementation speaks to those studying and thinking about human mental representation as well as those building artificial language understanding systems. The need for integration of the many insights available in the relevant fields fields into coherent, large scale theories and frameworks for language understanding is growing. I see this thesis as a necessary step to emphasize some of the aspects of grounding and intentionality that are much neglected in the computational disciplines, and to focus the work that does exist by acknowledging the importance that modern cognitive and philosophical insights about mental representation bear on synthetic systems. At the same time, I hope that providing a concrete implementation that performs well on spontaneous, situated human language shows that bridging the gap between theory and implementation is not only possible, but necessary for progress towards understanding language understanding.

Bibliography

[Allen and Perrault, 1980] James Allen and Raymond Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178, 1980.

[Bailey, 1997] David Bailey. *When Push Comes To Shove: A Computational Model of the Role of Motor Control in the Acquisition of Action Verbs*. PhD thesis, Univesity of California, Berkeley, 1997.

[Barwise and Perry, 1983] Jon Barwise and John Perry. *Situations and Attitudes*. MIT Press, Cambridge, MA, 1983.

[Bickhard, 2001] Mark H. Bickhard. Function, anticipation and representation. In D. M. Dubois, editor, *Computing Anticipatory Systems. CASYS 2000 - Fourth International Conference*, pages 459–469, Melville, NY, 2001. American Institute of Physics.

[Bobick and Ivanov, 1998] Aaron F. Bobick and Yuri A. Ivanov. Action recognition using probabilistic parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

[Boutilier *et al.*, 1999] Craig Boutilier, Thomas Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of AI Research*, 11:1–94, 1999.

[Brin and Page, 1998] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web seach engine. *Computer Networks and ISDN Systems*, 1998.

[Brooks, 1991] Rodney Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991.

[Bui *et al.*, 2002] Hung H. Bui, Svetha Venkatesh, and Geoff West. Policy recognition in the abstract hidden markov model. *Journal of Artificial Intelligence Research*, 17:451–499, 2002.

[Chapman, 1991] David Chapman. *Vision, Instruction and Action*. MIT Press, Cambridge, MA, 1991.

[Collins, 2003] Michael Collins. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 2003.

[Dennett, 1989] Daniel Dennett. *The Intentional Stance*. MIT Press, 1989.

[Dennett, 1992] Daniel Dennett. *Consciousness Explained*. Little, Brown and Company, 1992.

[Drescher, 1991] G. Drescher. *Made-up minds*. MIT Press, Cambridge, MA, 1991.

[Earley, 1970] Jay Earley. An efficient context-free parsing algorithm. *Communications of the ACM*, 6(8):451–455, 1970.

[Erol *et al.*, 1994] K Erol, JA Hendler, and DS Nau. Htn planning: Complexity and expressivity. In *Proceedings of the American Association for Artificial Intelligence*, 1994.

[Frege, 1892] Gottlob Frege. Über sinn und bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100:25–50, 1892.

[Geib and Goldman, 2005] Christopher Geib and Robert Goldman. Partial observability and probabilistic plan/goal recognition. In *IJCAI-05 workshop on Modeling Others from Observations*, 2005.

[Gibson, 1977] J.J. Gibson. The theory of affordances. In R. Shaw and J. Bransford, editors, *Perceiving, Acting and Knowing*, pages 67–82. Wiley, New York, 1977.

[Gorniak and Roy, 2004] Peter J. Gorniak and Deb Roy. Grounded semantic composition for visual scenes. *Journal of Artificial Intelligence Research*, 21:429–470, 2004.

[Gorniak and Roy, 2005a] Peter Gorniak and Deb Roy. Probabilistic grounding of situated speech using plan recognition and reference resolution. In *Proceedings of the International Conference on Multimodal Interfaces*, 2005.

[Gorniak and Roy, 2005b] Peter Gorniak and Deb Roy. Speaking with your sidekick: Understanding situated speech in computer role playing games. In *Proceedings of Artificial Intelligence and Digital Entertainment*, 2005.

[Haddock, 1989] N.J. Haddock. Computational models of incremental semantic interpretation. *Language and Cognitive Processes*, 4:337–368, 1989.

[Hakkani-Tur and Riccardi, 2003] D Hakkani-Tur and Guiseppe Riccardi. A general algorithm for word graph matrix decomposition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003)*, 2003.

[Harnad, 1990] Stevan Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[Horswill, 2001] Ian Horswill. Tagged behaviour-based architectures: Integrating cognition with embodied activity. *IEEE Intelligent Systems*, 16(5):30–38, September/October 2001.

[Hsiao *et al.*, 2003] K. Hsiao, N. Mavridis, and D. Roy. Coupling perception and simulation: Steps towards conversational robotics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.

[Jackendoff, 2002] Ray Jackendoff. *Foundations of Language: Brain, Meaning, Grammar, Evolution*. Oxford University Press, Oxford, UK, 2002.

[Klein and Manning, 2003] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of teh 41st Meeting of the Association of Computational Linguistics*, 2003.

[Laurence and Margolis, 1999] Stephen Laurence and Eric Margolis. Concepts and cognitive science. In Eric Margolis and Stephen Laurence, editors, *Concepts: Core Readings*, chapter 1, pages 3–81. MIT Press, 1999.

[Lenat, 1995] Douglas Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

[Litman and Allen, 1984] Diane J. Litman and James F. Allen. A plan recognition model for clarification subdialogues. In *COLING*, pages 302–311, 1984.

[Littman *et al.*, 2001] ML Littman, RS Sutton, and SP Singh. Predictive representations of state. In *NIPS*, 2001.

[Mangu *et al.*, 1999] L. Mangu, E. Brill, and A. Stolcke. Finding consensus among words: Lattice-based word error minimization. In *Proceedings of EUROSPEECH'99*, volume 1, pages 495–498, Budapest, 1999.

[Millikan, 1993] Ruth Millikan. *White Queen Psychology and other Essays for Alice*. MIT Press, 1993.

[Minsky, 1985] M. Minsky. *Society of Mind*. Simon and Schuster, New York, 1985.

[Minsky, to be published] Marvin Minsky. *The Emotion Machine*. Publisher: unknown, to be published. http://web.media.mit.edu/ minsky/E1/eb1.html.

[Narayanan, 1997] Srini Narayanan. *KARMA: Knowledge-based Action Representations for Metaphor and Aspect*. PhD thesis, University of California, Berkeley, 1997.

[Nau *et al.*, 2003] D Nau, TC Au, O Ilghami, U Kuter, W Murdock, and D Wu. Shop2: An HTN planning system. *Journal of Artificial Intelligence Research*, 2003.

[Norman, 1988] Donald A. Norman. *The Design of Everyday Things*. Doubleday, New York, 1988.

[Paice, 1990] C. D. Paice. Constructing literature abstracs by computer: Techniques and prospects. *Information Processing and Management*, 26(1):171–186, 1990.

[Pollack, 1986] M. E. Pollack. A model of plan inference that distinguishes between the beliefs of actors and observers. In *Proceedings of the 24th Meeting of the Association for Computational Linguistics*, 1986.

[Prinz, 2002] Jesse Prinz. *Furnishing the Mind: Concepts and their Perceptual Basis*. MIT Press, Cambridge, MA, USA, 2002.

[Pustejovsky, 1995] James Pustejovsky. *The Generative Lexicon*. MIT Press, Cambridge, MA, USA, 1995.

[Putnam, 1975] Hilary Putnam. The meaning of 'meaning'. In *Philosophical Papers, Vol. 2: Mind, Language and Reality*. Cambridge University Press, 1975.

[Pynadath and Wellman, 2000] David V. Pynadath and Michael P. Wellman. Probabilistic state-dependent grammars for plan recognition. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence, UAI2000*. Morgan Kaufmann Publishers, 2000.

[Roy *et al.*, 2002] Deb Roy, Peter J. Gorniak, Niloy Mukherjee, and Josh Juster. A trainable spoken language understanding system. In *Proceedings of the International Conference of Spoken Language Processing*, 2002.

[Roy *et al.*, 2004] Deb Roy, Kai-Yuh Hsiao, and Nikolaos Mavridis. Mental imagery for a conversational robot. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(3):1374–1383, 2004.

[Roy, 2002] Deb Roy. Learning words and syntax for a visual description task. *Computer Speech and Language*, 16:353–385, 2002.

[Roy, 2003] Deb Roy. Grounded spoken language acquisition: Experiments in word learning. *IEEE Transactions on Multimedia*, 5(2):197–209, June 2003.

[Roy, 2005] Deb Roy. Semiotic schemas: A framework for grounding language in action and perception. *Artificial Intelligence*, 2005.

[Schuler, 2003] William Schuler. Using model-theoretic semantic interpretation to guide statistical parsing and word recognition in a spoken language interface. In *Proceedings of the Association for Computational Linguistics*, 2003.

[Schwartz *et al.*, 2004] R. Schwartz, T. Colthurst, N. Duta, H. Gish, R. Iyer, C-L. Kao, D. Liu, O. Kimball, J. Ma, J. Makhoul, S. Matsoukas, L. Nguyen, M. Noamany, R. Prasad, B. Xiang, D-X. Xu, J-L. Gauvain, and L. Lamel. Speech recognition in multiple languages and domains: The 2003 bbn/limsi ears system. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pages III–753–756, Montreal, Canada, 2004.

[Searle, 1980] John Searle. Minds, brains, and programs. *The Behavioural and Brain Sciences*, 3, 1980.

[Sedivy *et al.*, 1999] Julie C. Sedivy, Michael K. Tanenhaus, Craig G. Chambers, and Gregory N.Carlson. Achieving incremental semantic interpretation through contextual representation. *Cognition*, 71:109–147, 1999.

[Singh, 2005] Push Singh. *EM-ONE: An Architecture for Reflective Commonsense Thinking*. PhD thesis, Massachusetts Institute of Technology, 2005.

[Siskind, 2001] Jeffrey Mark Siskind. Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of Artificial Intelligence Research*, 15:31–90, August 2001.

[Smith, 1996] Brian Cantwell Smith. *On the Origin of Objects*. MIT Press, Cambridge, MA, USA, 1996.

[Steedman, 1988] M. Steedman. Combinators and grammars. In Richard T. Oehrle, Emmon Bach, and Deirdre Wheeler, editors, *Categorical Grammars and Natural Language Structures*, pages 417–442. Kluwer Academic Publishers, 1988.

[Steedman, 2002] Mark Steedman. Formalizing affordance. In *roceedings of the 24th Annual Meeting of the Cognitive Science Society*, pages 834–839, 2002.

[Stolcke, 1995] Andreas Stolcke. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–201, 1995.

[Stone, 2001] Matthew Stone. Representing communicative intentions in collaborative conversational agents. In *AAAI Fall Symposium on Intent Inference for Collaborative Tasks*, 2001.

[Stoytchev, 2005] Alexander Stoytchev. Behavior-grounded representation of tool affordances. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, page ??, 2005.

[Sutton and Barto, 1998] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning*. MIT Press, 1998.

[Valenti *et al.*, 2003] Salvatore Valenti, Francesca Neri, and Alessandro Cucchiarelli. An overview of current research on automated essay grading. *Journal of Information Technology Education*, 2, 2003.

[Winograd, 1970] Terry Winograd. *Procedures as a representation for data in a computer program for understanding natural language*. PhD thesis, Massachusetts Institute of Technology, 1970.

[Witten and Bell, 1991] I. H. Witten and T. C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Trans. Information Theory*, 37(4):1085–1094, 1991.

[Zettlemoyer and Collins, 2005] Luke S. Zettlemoyer and Michael Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorical grammars. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2005.

[Zue *et al.*, 2000] V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. Hazen, and L. Hetherington. Jupiter: A telephone-based conversational inferface for weather information. *IEEE Transactions on Speech and Audio Processing*, 8(1):85–96, 2000.