

**Project no.:** 027657  
**Project full title:** Perception, Action & Cognition through learning of Object-Action Complexes  
**Project Acronym:** PACO-PLUS  
**Deliverable no.:** D5.1  
**Title of the deliverable:** LDEC and Its Relation to Communicative Action Including Planned and Purposeful Linguistic Utterance

<b>Contractual Date of Delivery to the CEC:</b>	31 January 2007
<b>Actual Date of Delivery to the CEC:</b>	30 January 2007
<b>Organisation name of lead contractor for this deliverable:</b>	UEDIN
<b>Author(s):</b> Chris Geib, Ron Petrick, and Mark Steedman	
<b>Participant(s):</b> UEDIN, UL, AAU, BCCN	
<b>Work package contributing to the deliverable:</b>	WP4,WP5
<b>Nature:</b>	R
<b>Version:</b>	Final
<b>Total number of pages:</b>	45
<b>Start date of project:</b>	1 <sup>st</sup> Feb. 2006
	<b>Duration:</b> 48 month

Project co-funded by the European Commission within the Sixth Framework Programme (2002–2006)	
Dissemination Level	
<b>PU</b>	Public <span style="color: red;">X</span>
<b>PP</b>	Restricted to other programme participants (including the Commission Services)
<b>RE</b>	Restricted to a group specified by the consortium (including the Commission Services)
<b>CO</b>	Confidential, only for members of the consortium (including the Commission Services)

### Abstract:

The core focus of WP5 is the generalization of the action representation developed in WP2, WP3 and WP4 to cover communicative acts, and the formalization of syntax and semantics for communication and interaction in natural language with situated purposeful agents, together with mechanisms for the acquisition of grammar from sentence-meaning pairs. The deliverable and the attached papers show how the LDEC action representation and the associated PKS planner developed under WP4 and described in D4.3.1 can both be induced from lower-level representations of states and state transitions, and provide a basis for natural language semantics at the higher level of Combinatory Categorial Grammar. The associated deliverable 5.2 describes the computational problem of natural language acquisition.

**Keyword list:** Combinatory Categorial Grammar (CCG); Plans and the Structure of Grammar; Dialog Planning; Planning with Knowledge and Sensing (PKS); Linear Dynamic Event Calculus (LDEC)



## Table of Contents

<b>1. EXECUTIVE SUMMARY .....</b>	<b>5</b>
<b>2. ROLE OF PKS/LDEC AND SITUATED DIALOG IN PACOPLUS .....</b>	<b>6</b>
<b>3. RELATION TO DEMONSTRATOR 8.1.....</b>	<b>6</b>
<b>4. PRINCIPAL SCIENTIFIC RESULTS.....</b>	<b>6</b>
<b>5. FUTURE WORK .....</b>	<b>6</b>
<b>6. PUBLICATIONS ASSOCIATED WITH D5.1 .....</b>	<b>7</b>
<b>REFERENCES .....</b>	<b>7</b>
<b>7. ANNEXES .....</b>	<b>8</b>
<b>A. FOUNDATIONS OF UNIVERSAL GRAMMAR IN PLANNED ACTION .....</b>	<b>9</b>
<b>B. PLANNING DIALOG ACTIONS .....</b>	<b>39</b>



## 1. Executive Summary

---

The core focus of WP5 is the generalization of the action representation developed in WP2, WP3, and WP4 to cover communicative acts, and the formalization of syntax and semantics for communication and interaction in natural language with situated purposeful agents, together with mechanisms for the acquisition of grammar from sentence-meaning pairs. The deliverable and the attached papers show how the LDEC action representation and the associated PKS planner developed under WP4 and described in D4.3.1 can both be induced from lower-level representations of states and state transitions, and provide a basis for natural language semantics at the higher level of Combinatory Categorial Grammar. The associated deliverable 5.2 describes the computational problem of natural language acquisition. Both of these papers are theoretical and look ahead to the next phase of the project, since at this stage, as was anticipated in the plan of work in the annex, the grounded linguistic semantics that will provide the basis for learning is yet to be developed.

Combinatory Categorial Grammar (CCG, Steedman 2000) is a theory of grammar according to which all language-specific grammatical information resides in the lexicon. A small universal set of strictly type-driven, non-structure dependent, syntactic rules (based on Curry's combinators **B**, **S**, and **T**) then "projects" lexical items into sentence-meaning pairs and defines the mapping from one to the other.

Steedman (2002b,a) showed how the same set of combinatory operations were involved in human and animal non-linguistic planning, and defined a Linear Dynamic version of the Event Calculus (LDEC) as a notation for such a planner. Work by UEDIN under PACO-PLUS support reported under deliverable D4.3.1 implements LDEC as a high-level symbolic planner under the PKS framework of Petrick and Bacchus (2002, 2004).

The present report analyzes the problem of connecting this planner to observable state-changes brought about by robot sensory-motor systems, on the one hand, and the lexicon and the grammar used by the language system on the other. It is proposed to use an associative memory such as the associative net of Willshaw (1981) both to associate affordance concepts with representations of initial conditions of actions, and to represent state-change or STRIPS-style action updates Fikes and Nilsson (1971). This representation is unusual in mapping the sensory map or manifold onto symbolic names for actions and their effects, as a basis for learning from experience. These actions are of the type that can be reasoned about by the PKS/LDEC planner described in Paco-Plus deliverable D4.3.1. This report describes the process whereby they will form the basis of the linguistic semantics that forms the basis for language learning described in deliverable D5.2, once the low level action domain is specified in terms of the Object Action Complex (OAC) representation. Certain extensions to the planner to handle specific types of action involved in speech acts for constructing dialogs to create shared plans are also described.

The document comprises two papers that describe this work, as follows:

- A: Foundations of Universal Grammar in Planned Action (to appear in Christiansen, Collins and Edelman 2007, *Language Universals*, Oxford University Press). This paper sketches the complete path between an attentional manifold of localized facts and a representation of action and change to the level of symbolic action, the nature of the conceptual system for the planner, and its relation to the lexicon and grammar. The low-level associative representation is the subject of continuing research and is yet to be linked to the specific representations used in the various robot platforms of the project. The high level grammar that will be used in the project itself awaits further specification of the robot action domains. However, the paper establishes that the CCG grammar formalism is transparent to the action representation. The significance of this link is that a variety of efficient applicable parsers for CCG already exist, and will be readily adaptable to the demands of PacoPlus.
  - B: Planning Dialog Actions (draft, to be submitted to AAAI 2007). This paper applies the PKS and LDEC formalisms developed under deliverable 4.3.1 to the analysis of speech acts to support dialog planning in PacoPlus. It shows that dialog acts can be treated analogously to perceptual sensing
-

acts in standard planning using PKS. The problem of dialog planning is treated with full generality, and a number of examples involving indirect speech acts and so-called conversational implicature are shown to fall out of formalism, including some important asymmetries in such effects that have not hitherto been commented on in the literature. While some of the effects discussed here (such as sarcasm) are unlikely to be directly involved in dialog with PacoPlus robotic agents, it is the general human tendency to interpolate unconstrained inference concerning indirection and implicature that notoriously causes mixed-initiative dialog to diverge and collapse. It follows that this capability needs to be represented in our systems.

## **2. Role of PKS/LDEC and situated dialog in PacoPlus**

---

PKS/LDEC provides a unifying theoretical framework for the various low-level action representations used in PacoPlus, and a practical mechanism for their integration with high level planning and interaction in natural language.

## **3. Relation to Demonstrator 8.1**

---

The lower-level concerns of the representation discussed in this report have been influential in determining the form of representations used in WP8.1 as reported in D8.1.1. The dialog component is relevant to future interactive demos.

## **4. Principal Scientific Results**

---

The work described in this report completes the theoretical path between low-level sensory-related and high-level plan-related and language-related representations for the Paco-Plus project as a whole. It embodies the knowledge base for reasoning about shared plans and actions grounded in the system developed under WPs 2,3 and 4. The claim is that it is a logical necessity that the various low-level representations of the project compile into intermediate level representations of the kind represented here in order to interface with cognitive systems for shared planning and language interaction. Because of the transparent relation to the low-level representations, the high level representation will in turn be shaped and informed by its distinctively embodied and grounded character. A second result is to show that such an action representation may be robust to the conversational inference processes of users that tend otherwise to make mixed-initiative dialog systems fail in practise.

## **5. Future Work**

---

A number of questions remain open at the time of this report and constitute further work.

1. While the input to the system is described as a map or manifold of located structured meanings the precise inventory of such features is left open in order to accommodate the various needs of the various low-level platforms. This picture will be refined over the coming months with partners in WP2,3, and 4.
2. While the device that maps state changes in such feature maps across time onto affordances or action concepts is currently assumed to be an associative net, issues such as storage efficiency may call for

other models of associative memory, such as the Holographic Reduced Representations (HRR) of Plate (1994) may be called for.

3. While the learning mechanism for the associative memory is described in terms of a generalization to the Perceptron Learning Algorithm (possibly using the “Kernel Trick” discussed by Freund and Shapire 1999) the generalization to the LDEC rule representation learning remains incomplete at the time of reporting.
4. The implementation of the dialog planner is incomplete at the time of reporting.
5. The next phase of the planner development will include incorporation of probabilistic models appropriate to the types of nondeterminacy that will undoubtedly arise from low level perception. There is work of this kind in the US using a very different planner framework by Leslie Kaelbling’s group at MIT (Zettlemoyer, Pasula and Kaelbling 2005), which we are following closely.

## 6. Publications Associated with D5.1

---

1. M. Steedman, 2006: “Foundations of Universal Grammar in Planned Action” in Christiansen, Collins and Edelman, (eds.), *Language Universals*, Oxford University Press.
2. M. Steedman and R. Petrick, 2007: “Planning Dialog Actions,” to be submitted to AAAI 2007.

## References

---

- Christiansen, Morten, Chris Collins, and Shimon Edelman, eds. 2007. *Language Universals*. Oxford: Oxford University Press. To appear.
- Fikes, Richard, and Nils Nilsson. 1971. “STRIPS: a New Approach to the Application of Theorem Proving to Problem Solving.” *Artificial Intelligence*, 2, 189–208.
- Freund, Yoav, and Robert Shapire. 1999. “Large Margin Classification Using the Perceptron Algorithm.” *Machine Learning*, 37, 277–296.
- Petrick, Ronald P. A., and Fahiem Bacchus. 2002. “A Knowledge-Based Approach to Planning with Incomplete Information and Sensing.” In Malik Ghallab, Joachim Hertzberg, and Paolo Traverso, eds., *Proceedings of the Sixth International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2002)*, 212–221. Menlo Park, CA: AAAI Press.
- Petrick, Ronald P. A., and Fahiem Bacchus. 2004. “Extending the Knowledge-Based Approach to Planning with Incomplete Information and Sensing.” In Shlomo Zilberstein, Jana Koehler, and Sven Koenig, eds., *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS-04)*, 2–11. Menlo Park, CA: AAAI Press.
- Plate, Tony. 1994. *Distributed Representations and Nested Compositional Structure*. Ph.D. thesis, University of Toronto.
- Steedman, Mark. 2000. *The Syntactic Process*. Cambridge, MA: MIT Press.
- Steedman, Mark. 2002a. “Formalizing Affordance.” In *Proceedings of the 24th Annual Meeting of the Cognitive Science Society, Fairfax VA, August*, 834–839. Mahwah NJ: Lawrence Erlbaum.
- Steedman, Mark. 2002b. “Plans, Affordances, and Combinatory Grammar.” *Linguistics and Philosophy*, 25, 723–753.

Willshaw, David. 1981. “Holography, Association and Induction.” In Geoffrey Hinton and James Anderson, eds., *Parallel Models of Associative Memory*, 83–104. Hillsdale, NJ: Erlbaum.

Zettlemoyer, Luke S., Hanna M. Pasula, and Leslie Pack Kaelbling. 2005. “Learning Planning Rules in Noisy Stochastic Worlds.” In *National Conference on Artificial Intelligence (AAAI)*,. AAAI.

## **7. Annexes**

---

### **A. Foundations of Universal Grammar in Planned Action**

---

#### **Mark Steedman**

This paper attempts to link the specific form taken by the universal grammatical mechanism that projects the finite lexicon of any given language onto the infinite set of strings of words paired with meanings that constitute that language to a more primitive capacity for *planning*, or constructing sequences of actions that culminate in an intended goal. A central consideration in defining this system is that of how action representations can be learned from interaction with the physical world.

### **B. Planning Dialog Actions**

---

#### **Mark Steedman and Ron Petrick**

The problem of planning dialog moves can be viewed as an instance of a more general AI problem of planning with sensing actions. Planning with sensing actions is complicated by the fact that such actions engender potentially infinite state-spaces. We adapt the PKS planner and the linear dynamic event calculus to the representation of dialog acts, and show potential beneficial consequences for planning mixed-initiative collaborative discourse.

## **Annex A**

---

# **Foundations of Universal Grammar in Planned Action\***

Mark Steedman  
University of Edinburgh  
[steedman@inf.ed.ac.uk](mailto:steedman@inf.ed.ac.uk)

Draft 1.0, January 29, 2007

This paper attempts to link the specific form taken by the universal grammatical mechanism that projects the finite lexicon of any given language onto the infinite set of strings of words paired with meanings that constitute that language to a more primitive capacity for *planning*, or constructing sequences of actions that culminate in an intended goal. A central consideration in defining this system is that of how action representations can be learned from interaction with the physical world.

### **1 Universal Grammar**

Two rather different kinds of phenomenon trade under the name of linguistic universals. The first are often expressed as implicational rules of the form “if a language has property P it has property Q”. An example is Greenberg’s (1963) Universal 3, “Languages with dominant VSO order are always prepositional”. While sometimes stated as deterministic laws, such rules almost always admit exceptions (as Greenberg 3 does—Dryer 1992:83), and should be regarded as probabilistic, arising either from the origins of most prepositions as verbs rather than adnominals, or from a requirement for efficient encoding to ensure easy learnability of the grammar as a whole, rather than as rules of universal grammar as such. Languages are free to violate such constraints, just so long as they do not violate so many of them as to make life unreasonably difficult for the child language learner.

The second kind often take the form of claims of the form “No natural language does X” or “every natural language does Y”, and seem more like strict constraints on human language, such as that every language has nouns, or transitive verbs, or relative clauses. This second type of universal is further divided into three types: “substantive” universals, “functional” universals, and “formal” universals, although there is some confusion in the literature concerning the definition of these types.<sup>1</sup>

---

\*Thanks to Chris Geib, Kira Mourão, Ron Petrick and Matthew Stone. The work was supported in part by the SE Edinburgh-Stanford Link grant Sounds of Discourse and EU IST grant FP6-2004-IST-4-27657 PACO-PLUS.

<sup>1</sup>The following distinctions follow Chomsky 1995. Chomsky 1965:27-30 distinguishes only between substantive and formal universals. However, the specific instances of formal universal cited there include some that under the definition of Chomsky 1995:54-55 would be classified as substantive or functional. To the extent that formal universals are discussed at all in Chomsky 1995:16,222, it is clear that the definition is the restricted one stated below, in contrast to that in Lasnik and Uriagereka 2005:12, where functional universals are referred to in passing as “formal,” threatening to lose an important distinction.

Substantive Universals, such as the ubiquity of nouns and transitive verbs are to do with content, and are determined by ontology, or the way our interactions with the physical and mental world structure mental representations, and hence semantics, into categories like mothers, dogs, and grasping. Functional Universals, such as the ubiquity of complementizers, case, tense, definiteness and the like, are determined by relations among substantive entities. Both substantive and functional categories are represented lexically by morphemes, although at least some functional categories are almost always morphologically implicit or “unmarked” in any given language. This distinction therefore corresponds quite closely to traditional notions of “open class” vs. “closed class” items, or “stems” vs. “inflections” and “function words.”

The third class, the Formal Universals, are rather different. These relate to the inventory of syntactic operations that combine substantive and functional categories, and project their characteristics and meanings onto sentences and logical forms. Such universals concern the mathematical or automata-theoretic class of operations that are countenanced in the theory of grammar, and take the form of statements such as “Natural languages fall outside the class of context-free languages” (Chomsky 1957). Such universals are not statistical in nature: one example of a natural language (or in this case, natural language constructions, as in Huybrechts 1984 and Shieber 1985) that is provably non-context-free proves the claim, even if natural language constructions in general are, in fact “with overwhelmingly greater than chance frequency,” context free.

It is often quite hard to decide to what type a given universal claim should be assigned. Greenberg’s Universal 20 claims that only six of the twenty-four possible linear orderings of the categories Dem(onstrative), Num(ber), A(djective), and N(oun) exhibited in English NPs like *These five young lads* are universally attested. While Greenberg sampled only thirty languages, and eight further orders have since been attested (Hawkins 1983; Cinque 2005), they modify the statement of the universal itself, not its statistical strength.

Similarly, Ross (1970) described a universal relating “gapping” or deletion of the verb under coordination with base constituent order. The pattern can be summarized as follows for the three dominant sentential constituent orders (asterisks indicate the excluded cases):

- (1) SVO: \*SO and SVO SVO and SO
- VSO: \*SO and VSO VSO and SO
- SOV: SO and SOV \*SOV and SO

This observation can be generalized to individual constructions within a language: just about any construction in which an element apparently goes missing preserves canonical word order in an analogous fashion. For example, English ditransitive verbs subcategorize for two complements on their right, like VSO verbs. In the following “argument cluster” coordination, it is indeed in the right conjunct that the verb goes missing:

- (2) Give Thelma a book, and Louise a record.

At first glance, this observation looks like an implicational universal, and indeed there were early claims for exceptions, from languages like Dutch (SOV) and Zapotec (VSO, Rosenbaum 1977), which allow both patterns. However, both those languages can be claimed to have mixed base order, and if the claim is relativized to constructions it can be seen as making a claim about the universal apparatus for projecting lexically specified constructions onto sentences, and hence as a claim about a formal universal.

## 2 Universal Semantics

The only really plausible source for grammatical universals of the second, non-statistical, kind is the semantics, which in turn is determined by the specific nature of our interactions with the world, and the concepts that those interactions engender (Chomsky 1965:27-30; Pinker 1979; Newmeyer 2005). The reasoning is as follows. The only reason for natural language grammar to exist at all is to support semantic interpretation, as a basis for reasoning about joint action in the world with other members of a language community. Furthermore, we know that syntactic grammars for even the simplest language classes cannot be exactly induced on the basis of exposure to strings from the language alone (Gold 1967). (While Horning 1969 showed that grammars of any such class can technically be approximated to any desired degree of probable error by automatically induced statistical models, and such approximations are in fact quite practically applicable to problems such as word disambiguation for automatic speech recognition, such statistical approximation carries exponentially growing computational costs. It is also quite unclear how such approximations can support semantic interpretation.) We also know that exact induction of even quite high classes of (monotonic) grammar from strings paired with labeled trees corresponding to the yield of the grammar for that string is essentially trivial (apart from the problem of noise in the input and consequent error) (Buszkowski and Penn 1990; Siskind 1996; Villavicencio 2002; Zettlemoyer and Collins 2005). It follows that the simplest hypothesis concerning the way children acquire their native language is that they induce its syntactic grammar from pairings of strings and logical forms representing meaning. On this assumption, language universals must reflect the properties of a universal grammar of logical form, in which the structure of predicates and arguments carves nature (including our own being) at the joints in just one way, ideally suited to reasoning about it.

Of course, to say this much is not terribly helpful. The putative grammar of logical form itself has a syntax, which can in turn only be explained as arising from a semantics that must be specified in a much stronger sense, using a model theory whose details will ultimately be determined by the nature of our own and our remote non-human ancestor's interactions with the world. Worse still, our grasp on this kind of semantics is (as Chomsky never tires of pointing out) even shakier than our grasp on linguistic syntax, mainly because our formal and intuitive grasp of such dynamic systems is much weaker than that of static declarative systems. Nevertheless, this must be where linguistic universals originate.

This is easiest to see in terms of substantive and functional universals—that is, those that relate to content and category of morphemes, words, and constituents. For example, if it is the case that all natural languages have transitive verbs, or that no language has a verb allowing more than four arguments (Steedman 1993, 2000b; Newmeyer 2005:5, citing Pesetsky 1995), then the universal logical form must include all and only such relations.<sup>2</sup> If languages are nevertheless free to specify the position of the verb with respect to its arguments as initial, second position, or final, then we may suspect that the universal grammar of logical form specifies only dominance relations, not linear order.<sup>3</sup>

<sup>2</sup>I shall use the term “transitive” indiscriminately, to cover all verbs taking a second argument such as NP, PP, VP, or S in addition to the subject

<sup>3</sup>The fact that UG “cannot count beyond two”—that is, that no language requires its verb to be in third position, next-to-last position, etc. (Newmeyer 2005:4)—must also be semantic, say because of an association between first position and notions such as “topic”.

But it is also true of the formal universals—that is, those that govern the types of rule that combine constituents or categories, projecting their properties onto larger structures. For example, the main reason for believing in a formal universal to the effect that natural language grammar formalisms must be of at least the expressive power of context free grammars is not that intrinsically non-finite state fragments of languages like English can be identified. All attested and in fact humanly possible instances of such strings can be recognized by covering finite-state machines, and human beings must in some sense actually *be* finite state machines. The real reason is that no-one can see any way to parsimoniously capture the one part of the semantics that we do have a reasonably good understanding of, namely compositional projection of function-argument relations under constructions like complementization and relative clause-formation, governed by the particular type of transitive verbs that take sentences as complement, other than by simulating an infinite state, push-down automaton.<sup>4</sup>

Unfortunately, that is about as far as our intuitions take us. The way in which individual languages reflect the putative underlying universal is not very transparent to us as linguists (although it must be transparent to the child). For example, some languages like English lexicalize complex complex causatives like “he was running across the street” with special transitive versions of verbs like *run* taking PP complements. Other languages, like French, appear to lexicalize the elements of the underlying causative logical form more literally, in expressions like “Il était en train de traverser la rue à la course.”<sup>5</sup> Moreover, even such apparently painstakingly elaborated expressions do not seem to be anywhere near complete in explicitly identifying sufficient truth-conditions for such utterances about a specific situation (such as one in which the subject of the remark never reached the destination), and in fact it is very difficult to specify such truth conditions for any language. The reason is that such conditions seem to include the intentions which motivated the subject’s plan of action, together with the “normal” consequences that could be anticipated, as well as the physical action itself. This fact engenders the “imperfective paradox” that it is possible to truthfully say “he was running across the street” (but not “he ran across the street”), even if the person in question never reached the other side, just in case what he did would normally have resulted in his doing so (see Dowty 1979, and much subsequent work).

This paper argues that, if one wants to truly understand this semantics, and the form of the linguistic universals that it determines, it is necessary to simultaneously investigate the nature of action representations capable of supporting notions of teleology and change of state, together with the ways such representations can be learned in interaction with experience of the world, and the ways in which the specific form that human knowledge representations takes follows from that experience, and determines observed and predicted grammatical universals. The fact that we find it difficult to specify such knowledge representations using the logics that have been developed for other more mathematical inquiries should make us expect to find the form of such grounded and experientially induced knowledge representations quite surprising, and rather unlike the hand-built representations for common-sense knowledge or “naïve physics” that have been proposed in the AI literature (Hayes 1979, *passim*).

---

<sup>4</sup>In this sense, the emphasis in Hauser, Chomsky and Fitch 2002 on the evolution of recursion itself as the crucial element distinguishing human cognition and language from animal cognition may be misplaced. It must be the evolution of *concepts that intrinsically require recursive definitions* that separates us from other animals. Recursive concepts of mutual belief seem to be plausible candidates, as Tomasello 1999 has suggested.

<sup>5</sup>Many of these explicit elements like “à la course” are of course often elided in actual French utterance in context, making the problem of automatic translation much harder.

### 3 Representing Change and Reasoning About Action

The problem of planning is the problem of finding a sequence of actions  $\alpha, \beta$ , etc. through a state space of the kind represented in figure 1. This structure, in which blobs represent *states* (which

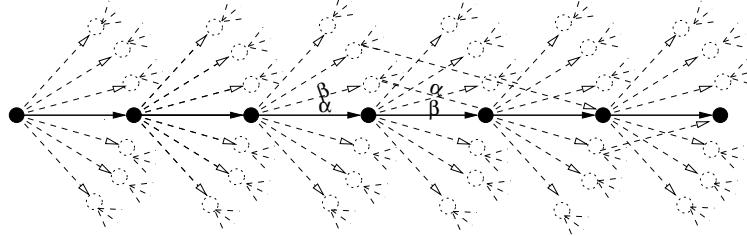


Figure 1: Kripke Model of Causal Accessibility Relation

we can think of as vectors of values of facts or propositions), and directed arcs represent *actions* that transform one state into another (which we can think of as finite-state transducers from one state vector to another), is known as a S4 Kripke model. We can define a planning calculus over such models as follows.

#### 3.1 The Linear Dynamic Event Calculus

The Linear Dynamic Event Calculus (LDEC) combines the insights of the Event Calculus of Kowalski and Sergot (1986), itself a descendant of the Situation Calculus of McCarthy and Hayes (1969) and the STRIPS planner of Fikes and Nilsson (1971), with the Dynamic and Linear Logics that were developed by Harel (1984), Girard (1987) and others.

Dynamic logics are a form of modal logic in which the  $\Box$  and  $\Diamond$  modalities are relativized to particular events. For example, if a (possibly nondeterministic) program or command  $\alpha$  computes a function  $F$  over the integers, then we may write the following:<sup>6</sup>

$$(3) \ n \geq 0 \Rightarrow [\alpha](y = F(n))$$

This can be read as “if  $n \geq 0$ , executing the action  $\alpha$  always results in a situation in which  $y = F(n)$ ”. (dually) that “in any situation in which  $n \geq 0$ , there is an execution of  $\alpha$  that terminates with  $y = F(n)$ ”.

We can think of this modality as defining a logic whose models are Kripke diagrams in which accessibility between possible worlds corresponds to state-changing events. Such events can be defined as mappings between situations or partially specified possible worlds, defined in terms of conditions on the antecedent which must hold for them to apply (such as that  $n \geq 0$  in (3)), and consequences (such as that  $y = F(n)$ ) that hold in the consequent.

The particular dynamic logic that we are dealing with here is one that includes the following dynamic axiom, which says that the operator ; is *sequence*, an operation related to *functional*

---

<sup>6</sup>Dynamic Logic offers a dual “diamond” modality of possibility, as well as the “box” modality of necessity, such that the following means that “if  $n \geq 0$ , executing the action  $\alpha$  sometimes results in a situation in which  $y = F(n)$ .”

(i)  $n \geq 0 \Rightarrow \langle \alpha \rangle(y = F(n))$

*composition* over events, viewed as functions from situations to situations:

$$(4) [\alpha][\beta]P \Rightarrow [\alpha;\beta]P$$

Using this notation, we can conveniently represent, say, a plan for *getting outside* as the composition of *pushing* a door and then *going through* it, written *push';go-through'*.

Composition is one of the most primitive *combinators*, or operations combining functions, which Curry and Feys (1958) call **B**, writing the above sequence  $\alpha;\beta$  as  $\mathbf{B}\beta\alpha$ , where

$$(5) \mathbf{B}\beta\alpha \equiv \lambda s. \beta(\alpha(s))$$

Plans like *push';go-through'* could be written in Curry's notation as **B***go-through'push'*

### 3.2 Situation/Event Calculi and the Frame Problem

The situation calculi are heir to a problem known in the AI literature as the Frame Problem (McCarthy and Hayes 1969). This problem arises because the way that we structure our knowledge of change in the world is in terms of event-types that can be characterized (mostly) as affecting just a few fluents among a very large collection representing the state of the world. (Fluents are facts or propositions that are subject to change). Naive event representations which map entire situations to entire other situations are therefore representationally redundant and inferentially inefficient. A good representation of affordances must get around this problem.

To avoid the frame problem in both its representational and inferential aspects, we need a new form of logical implication, distinct from the standard or intuitionistic  $\Rightarrow$  we have used up till now. We will follow Bibel et al. (1989) and others in using *linear* logical implication  $\multimap$  rather than intuitionistic implication  $\Rightarrow$  in those rules that change the value of fluents.

For example, in Steedman 2002, events involving doors in a world (greatly simplified for purposes of exposition) in which there are two places *out* and *in* separated by a door which may be *open* or *shut*, as follows:<sup>7</sup>

$$(6) \text{affords}(\text{push}(y,x)) \wedge \text{shut}(x) \multimap [\text{push}(y,x)]\text{open}(x)$$

$$(7) \text{affords}(\text{go-through}(y,x)) \wedge \text{in}(y) \multimap [\text{go-through}(y,x)]\text{out}(y)$$

These rules say that if the situation affords you pushing something and the something is shut, then it stops being shut and starts being open, and that if the situation affords you going through something, and you are in, then you stop being in and start being out. Linear implication has the effect of building into the representation the update effects of actions—that once you apply the rule, the proposition in question is “used up”, and cannot take part in any further proofs, while a new fact is added. The formulae therefore say that if something is shut and you push it, it becomes open (and vice versa), and that if you are in and you go through something then you become out (and vice versa). This linear deletion effect is only defined for facts—that is ground literals.  $\text{affords}(\text{go-through}(y,x))$  is a derived proposition, so it will hold or not in the consequent state according to whether it can be proved or not in that state. The way we have defined affordance, it will hold. (However, we have not yet defined what happens if you go

---

<sup>7</sup>We follow a logic programming convention that all variables appearing in the consequent are implicitly universally quantified and all *other* variables are implicitly existentially quantified. Since in the real world doors don't always open when you push them, box must be read as *default* necessity, meaning “usually”.

through a door when you are *out*.)

In order to know when we can apply such rules, we also need to define the conditions that *afford* actions of pushing and going through. Here ordinary non-linear intuitionistic implication is appropriate:

- (8) a.  $\text{door}(x) \wedge \text{open}(x) \Rightarrow \text{affords}(\text{go-through}(y, x))$
- b.  $\text{door}(x) \wedge \text{shut}(x) \Rightarrow \text{affords}(\text{push}(y, x))$

These rules say (oversimplifying wildly) that if a thing is a door and is open then it's possible to go through it, and that if a thing is a door and it's shut, then it's possible to push it.

We also need to define the transitive property of the possibility relation, as follows, using the definition (4) of event sequence composition:

- (9)  $\text{affords}(\alpha) \wedge [\alpha]\text{affords}(\beta) \Rightarrow \text{affords}(\alpha; \beta)$

This says that any situation which affords an action  $\alpha$ , and in which actually doing  $\alpha$  gets you to a situation which affords an action  $\beta$ , is a situation in which afford  $\alpha$  *then*  $\beta$ .

To interpret linear implication as it is used here in terms of proof theory and proof search, we need to think of possible worlds in the Kripke diagram in figure 1 as states of a single updatable STRIPS database of facts. Rules like (6) and (7) can then be interpreted as (partial) functions over the states in the model that map states to other states by removing facts and adding other facts. Linear implication and the dynamic box operator are here essentially used as a single state-changing operator: you can't have one without the other.

The effect of such rules can be exemplified as follows. If the initial situation is that you are in and the door is shut:

- (10)  $\text{in}(you) \wedge \text{door}(d) \wedge \text{shut}(d)$

—then intuitionistic rule (8b) and the linear rule (6) mean that attempts to prove the following propositions concerning the state of the door in the situation that results from pushing the door will all succeed, since they are all facts in the database that results from the action  $\text{push}(you, d)$  in the initial situation (10):

- (11) a.  $[\text{push}(you, d)]\text{open}(d)$
- b.  $[\text{push}(you, d)]\text{door}(d)$
- c.  $[\text{push}(you, d)]\text{in}(you)$

On the other hand, an attempt to prove the proposition (12) will fail because rule (6a) removes the fact in question from the database that results from the action  $\text{push}(you, d)$ :<sup>8</sup>

- (12)  $[\text{push}(you, d)]\text{shut}(d)$

The advantage of interpreting linear implication in this way is that it builds the STRIPS treatment of the frame problem (Fikes and Nilsson 1971) into the proof theory, and entirely avoids the need for inferentially cumbersome reified frame axioms of the kind proposed by Kowalski (1979) and others (see Shanahan 1997).

This fragment gives us a simple planner in which starting from the world (13) in which you are *in*, and the door is *shut* and stating the goal (14) meaning “find a series of actions that

---

<sup>8</sup>We follow the logic programming convention of negation by failure, according to which a proposition is treated as false if it cannot be positively proved to be true.

the situation affords that will get you *out*,” can, given a suitable search control, be made to automatically deliver a constructive proof that one such plan is (15), the composition of *pushing*, and *going through*, the door:

$$(13) \quad \text{in}(you) \wedge \text{door}(d) \wedge \text{shut}(d)$$

$$(14) \quad \text{affords}(\alpha) \wedge [\alpha]\text{out}(you)$$

$$(15) \quad \alpha = \text{push}(you, d); \text{go-through}(you, d).$$

The situation that results from executing this plan in the start situation (10) is one in which the following conjunction of facts is directly represented by the database:

$$(16) \quad \text{out}(you) \wedge \text{door}(d) \wedge \text{open}(d)$$

Using linear implication (or the equivalent rewriting logic devices or state update axioms of Thielscher (1999) and Martí-Oliet and Meseguer (1999)) for STRIPS-like rules makes such frame axioms unnecessary. Instead, they are theorems concerning the linear logic representation. The further implications of the theory for extended forms of the frame problem considered by Hanks and McDermott (1986), Sandewall (1994) and Shanahan (1997) are discussed in Steedman 1997, 2000b.

Since we can regard actions as functions from situations to situations, then rule (9) defines *function composition* **B** as the basic plan-building operator of the system. Composition is one of the simplest of a small collection of combinators which Curry and Feys (1958) used to define the foundations of the  $\lambda$ -calculus and other applicative systems in which new concepts can be defined in terms of old. Since the knowledge representation that underlies human cognition and human language could hardly be anything *other* than an applicative system of some kind, we should not be surprised to see it turn up as one of the basic operations of planning systems.

This calculus is developed further in Steedman 1997, 2002 in application to more ambitious plans, and a number of generalizations of the frame problem, using a novel analysis of *durative* events extending over intervals of time, in which such events are represented by instantaneous *inceptive* and *culminative* events, which respectively add/remove facts about the event being in progress, and the consequences if any of its culmination. This representation has a number of advantages over more standard interval-based representations such as those of Allen (1984); Allen and Hayes (1989), including a solution to the imperfective paradox. These ramifications are passed over here.

By making the calculus affordance-based, we provide the basis for a simple forward-chaining reactive style of planning that seems to be characteristic of non linguistic animal planning. This kind of planning is not purely reactive in the sense of Brooks (1986) and Agre and Chapman (1987): the notion of state representation plays a central role, as Bryson has proposed within the Behavior-Based AI approach (2001, Bryson and Stein 2001).

There are two ways of thinking about computing plans with the Linear Dynamic Event Calculus. One is as a logic programming language, much like Prolog. Poole 1993 shows how the Horn clauses of such a representation can be associated with a Bayesian Network probability model. However, there are problems in scaling such logicist representations to realistically-sized cases. We noted earlier that STRIPS/LDEC operators can be thought of as finite-state transducers (FSTs) over state-space vectors. We can think of these operators, more specifically, as

FSTs over *sparse* vectors, since they treat most values as irrelevant, STRIPS-style. It follows that it is also possible to think of LDEC operators in terms of neural network representations, and in particular in terms of a very simple device called the Associative Network or Willshaw net, which is specialized to representing associations between sparse vectors. These two approaches are discussed next.

#### 4 Planning and Explanation-Based Learning with LDEC

Once an agent has learned a set of actions as LDEC operators, it is in a position to use them to form plans and learn more about the world. This process is often talked of in terms of “exploration”, as if it involved executing a random walk of actions in the real world storing action sequences in memory according to their good or bad. However, random walks make bad plans. Frequently, even if they end up in desirable states, they get there by way of detours and unproductive steps, so that they require critiquing and rejection or modification. They may even prove fatal. It is just as easy, and much safer, to critique potential action sequences off-line and ahead of time, a process which is usually called planning.

One simple way to do this efficiently for a set of operators  $\aleph$  is to consider the subset  $A_0 \subseteq \aleph$  of actions  $\alpha_{0_i} \in \aleph$  such that the current state  $\sigma_0$  affords  $\alpha_0$ , and generate the set  $\Sigma_{A_0}$  of states  $\sigma_{\alpha_{0_i}}$  that result from executing each  $\alpha_{0_i} \in A_0$  in state  $\sigma_0$ . Some of these states may be desirable goal states in their own right. However, assessing their desirability will often depend on considering what action, including actions by other agents, those states themselves afford. A wise agent will therefore consider the states  $\sigma_{\alpha_{0_i};\alpha_{1_j}} \in \Sigma_{\alpha_{0_i};A_{1_j}}$  that result from the sets  $A_{1_j}$  of further actions that each first-level state  $\sigma_{\alpha_{0_i}}$  affords, and so on recursively by breadth-first iterative deepening, applying dynamic programming methods to identify optimal plans (Bertsekas and Tsitsiklis 1996). Although the size of the state-space grows exponentially with depth, this method of growing the state-space has the advantage that the structure of the search space is isomorphic to the space of possible plans, potentially allowing planning using graph-based heuristics of the kind used by Hoffman and Nebel’s 2001 FastForward planner.

In this way we generate a set of plans of the form  $[\alpha_{0_i};\alpha_{1_j};\dots;\alpha_{m_x}]$ , where  $m \leq$  the depth of the tree. We can calculate the result state of each plan by applying the LDEC operators in sequence to the original state. We can assign a value to the plan by comparing the end state with the start state. We can also assign a cost to the plan in terms of the summed costs of the actions that it is made up of, and can choose among plans that end in the same state on a benefit/cost basis, eliminating wasteful plans such as those that include irrelevant or counterproductive actions. We can also assign an *a priori* reliability to the plan by computing the product of the reliability of its component actions.

Having identified a plan with a good benefit/cost ratio, we can add that plan to the set  $\aleph$  as a plan operator in its own right, and begin to collect observations on its actual reliability by applying it in the world. This process of adding action operators requires “flattening” the plan, making any preconditions and deletions of its elementary actions that are not explicitly added by an earlier elementary action conditions on situations that afford the plan, and making any additions that are not subsequently deleted be among the additions of the plan as a whole. Crucially, Finite State Transducers are closed under composition (Kaplan and Karttunen 1994).

Observations of frequency of use and reliability of operators will be essential to distinguish generally applicable plans from special-case plans and plans with inherent flaws arising from

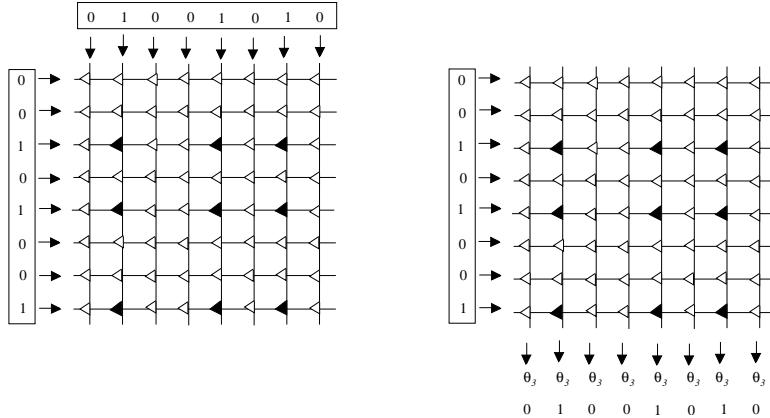


Figure 2: Hetero-associative net: Storage and Retrieval

inadequacies in the knowledge representation. Other methods of generalizing plans will be required. For example, the recognition that successful minimal plans for piling up specific numbers of boxes are all instances of the iterative plan  $[ (?there(l); puton(l, box); climbOn(box))^+]$  is an instance of a very powerful form of generalization. The successful selection of new useful additions to the set of actions  $\aleph$  achieves exponential savings in plan search, and is the key to solving the exponential growth of plan search spaces, also known as plan decomposition or hierarchical planning.

## 5 LDEC operators as Associative Networks

This section shows how we can represent the intuitionistic  $\Rightarrow$  LDEC inference rules relating situations to affordances using the Willshaw net in *auto-associative* mode, and represent the linear  $\multimap$  inference rules by a second *hetero-associative* Willshaw net, associating input state vectors with output state vectors.

### 5.1 The Associative Net

The Associative Net was invented by Willshaw, Buneman, and Longuet-Higgins (1969—see Willshaw (1981)), following early work by Steinbuch (1961) and Anderson (1968). This device illustrates three basic properties of network models which are characteristic of mechanisms involved in phenomena of human memory and attention:

- Non-localized storage (“Distributivity”)
- Ability to recover complete stored patterns from partial or noisy input (“Graceful Degradation”).
- Ability to work even in the face of damage (“Holographic Memory”).

A number of refinements relevant to practical application are proposed by Sommer and Palm (1998) and Plate (1991).

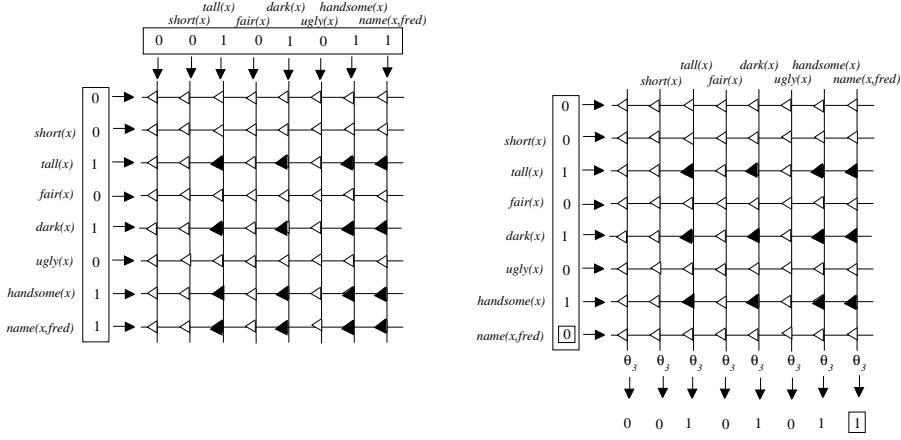


Figure 3: Auto-associative net: Storage and Augmented Retrieval

An associative net acts as a distributed memory associating pairs of input and output vectors, as in figure 2, which represents a grid of horizontal input lines and vertical output lines with binary switches (triangles) at the intersections. To store an association between the input vector on the left and the output vector along the top, switches are turned on (black triangles) at the intersection of lines which correspond to a 1 in both input and output patterns.

To retrieve the associate of the input, a signal is sent down each horizontal line corresponding to a 1 in the input. When such an input signal encounters an “on” switch, it increments the signal on the corresponding output line by one unit. These lines are then thresholded at a level corresponding to the number of on-bits in the input. With such thresholding, an associative memory can store a number of associations in a distributed fashion, with interesting properties of noise- and damage- resistance, provided that the 1s are relatively sparse.

For example, if one of the on-bits in the input goes off, so that we threshold at 2 rather than 3, we recover the entire associated pattern. Similarly if an off bit goes on we can similarly recover the correct association by reducing the threshold of 4 to 3. These properties depend on there being not too many similar patterns stored in the same net.

It follows that if patterns are “autoassociated,” or stored with themselves as output, associative nets can be used to complete partial patterns, as needed to recall perceptually non-evident properties of objects, such as the fact that the tall, dark and handsome person’s name is Fred, as in figure 3.

## 5.2 Associative Networks and the Hippocampal Associative Pathway

There is evidence for the involvement of associative and autoassociative networks in many functions of the brain. Marr’s seminal papers from 1969; 1970; 1971 propose a theory of the cerebellum, hippocampus, and neo-cortex using a related associative mechanism throughout.

There is further evidence from hippocampal patients for a dual path model of information processing in learning (Gluck and Myers 2000). Such patients are unable to learn associations

such as people’s names. However, they can exhibit classical conditioning to the extent of learning which of those people are nice to them (Damasio 1999:43-46,113-121). There seems to be an associative hippocampal path that is needed for both associative memory and operant conditioning, as well as a non associative non-hippocampal path supporting classical conditioning—as in the architecture shown in Figure 4.

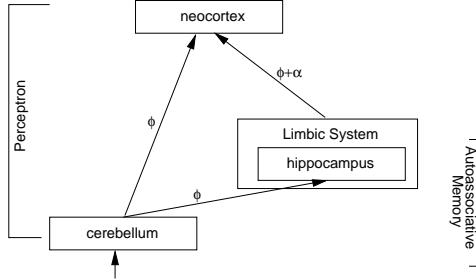


Figure 4: The basic Cerebellar-Hippocampo-Cortical dual-path circuit: (adapted from Gluck and Myers).

## 6 Associating Situations with Affordances

Forward-chaining hierarchical planning via plan compilation still has an exponential search space (albeit a search space of better plans), and for realistically sized problems and state-representations, an animal acting in real time cannot afford to find out what actions the situation affords by matching the input vectors of all its action operators against the situation. The vast majority of these operators will not be applicable, so we want *the state itself* to actively propose the operators that it affords.

We can do this by representing all the LDEC  $\Rightarrow$  rules as an auto-associative network and all the LDEC  $\rightarrow$  rules as second hetero-associative network.

We have already seen that the associative hippocampal pathway augments the state representation, adding facts about individuals (like their name) to vectors including observable facts about them (such as  $tall(x) \wedge dark(x) \wedge handsome(x)$ ). In the same way, the associative pathway can be made to add fluents like  $affords(go-through(x))$  to state vectors including  $door(x) \wedge open(x)$ . The presence of this fluent can then be used to directly access operators that have been applicable in similar situations in the past.

Thus, the *affords* literals of LDEC correspond to elements of the state vector that correspond to FST state transducer operators, which are added to each newly generated state by the hippocampal associative memory, and which are removed again by the application of any such operator. These operators and these alone form the set that generates the search space for plans.

This hippocampal associative network can in turn be thought of as relation between states and actions that they afford.

For example, suppose an animal has learned the operators *go-through* and *push*, and associated  $affords(go-through(y,x))$  with vectors subsuming  $in(y) \wedge door(x) \wedge open(x)$ , and  $affords(push(x,y))$  with those subsuming  $in(y) \wedge door(x) \wedge shut(x)$ . Suppose, moreover that agent

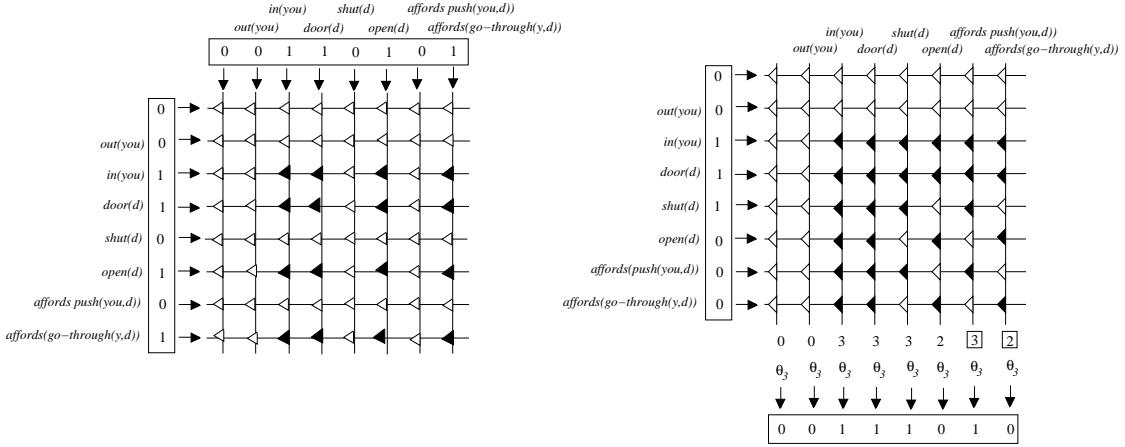


Figure 5: LDEC rules of affordance: Storing preconditions of *go-through* in an empty hippocampal auto-associative network, and retrieving the affordance of *push* from the loaded net.

has the goal of getting *out* but the door is shut.

If the agent matches the loaded affordance network to the situation, as in Figure 5, it will find that of its two learned operators, *push* is most strongly activated.<sup>9</sup>

The representation of the update effected by the action itself also uses associative networks in a subtly different way, illustrated in Figure 6. Weights and therefore outputs can take negative values, and outputs represent *changes* to the database, rather than truth, with  $-1$  meaning “set the feature-value to 0,” and  $+1$  meaning “set the feature-value to 1,” while 0 means “leave the feature value unchanged.” If the agent then considers the application of the *push* operator, using the “neo-cortical” change network in Figure 6, it will observe that it takes it to a state in which the door is open. A further application of the affordance net of Figure 1s 5 to the result state, restarting the whole cycle shown in Figure 7, will reveal that this state affords *go-through*, and that applying this operator will result in being *out*.

The change matrix in Figure 6 can be viewed as encoding the relation between affordances and the states that result from those affordances.

In terms of the logicians’ S4 model in Figure 1, the state-change matrix of Figure 6 encodes the *accessibility relation* defining the both the search space and the plan space. The hippocampal matrix 5 represents the relation between states and affordance of those actions.

The associative network representation assumes a solution to the Binding Problem (von der Malsberg 1995)—that is, the problem of representing the fact that the properties *door* and *open* are predicated of the *same* object *d*. This problem was first identified by Rosenblatt (1962), who noted that a perceptron trained to recognize triangles and squares anywhere in the image, and to recognize objects in the top half and bottom half of the image could nevertheless not distinguish a picture with a triangle above a square from its inverse. A number of solutions to the binding

<sup>9</sup>To save space, networks are shown more densely loaded than would be possible in practice. Note that, in STRIPS terms, both preconditions and deletions are included in the association. Under present simplifying assumptions, the two rules must have equal numbers of inputs: this assumption is non-essential.

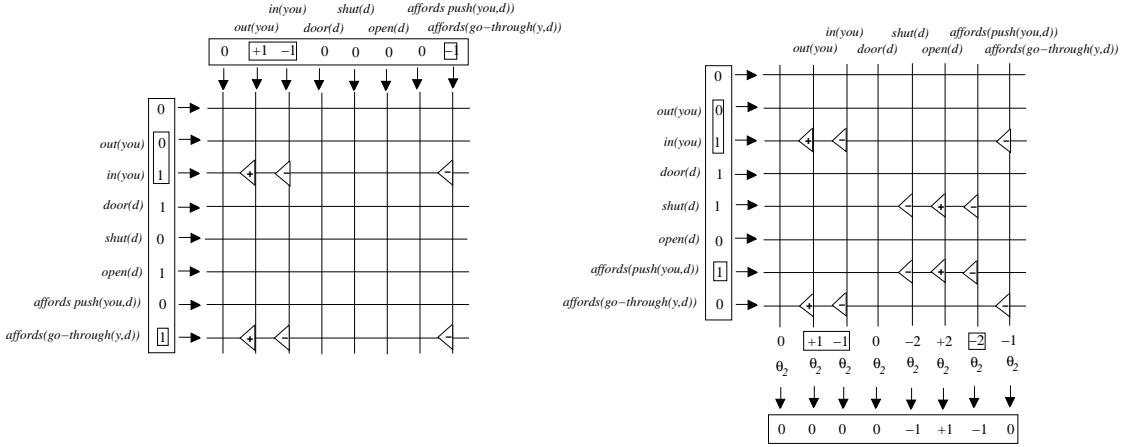


Figure 6: LDEC rules of change: Storing *go-through* in the neocortical hetero-associative network, and retrieving the *push* transduction from the loaded net.

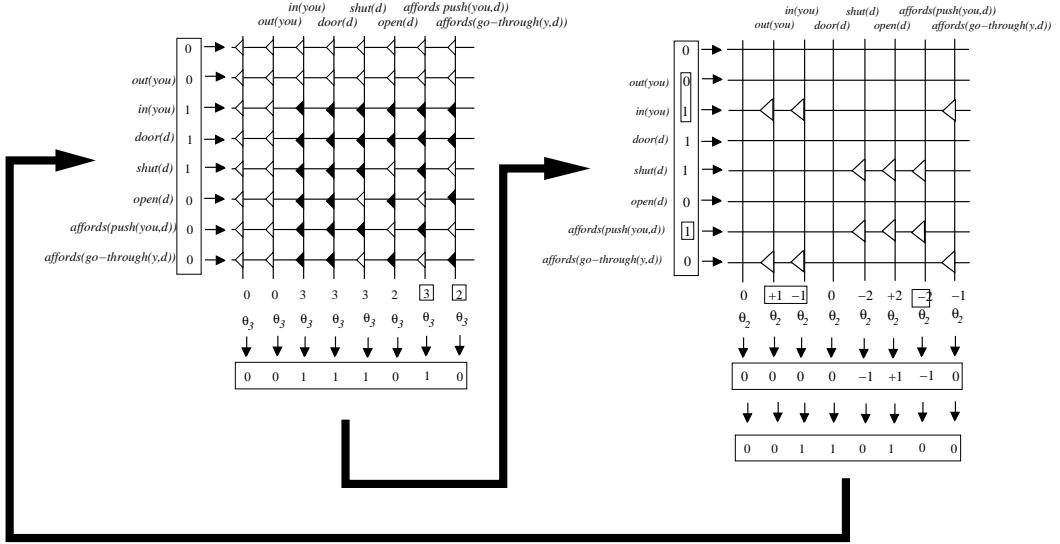


Figure 7: LDEC cycle: Retrieving the affordance of *push* from the hippocampal net, generating the next state from the neo-cortical net, and preparing to retrieve the affordance of *go-through* from the hippocampal net.

problem have been proposed, from the “deictic” solution of Agre and Chapman (1987) (which says that *door* and *open* are implicitly predicated of whatever object you are attending to) to the temporal encoding of variables in synchronous axonal firing rates of von der Malsberg (1995)

and Shastri and Ajjanagadde (1993). Others, including Rosenblatt himself, have suggested that the binding problem should be avoided by coding the combinations of properties that you need in the first place. Clearly, it is perverse to train on triangle features independent of place if place is important, and training for the combination of triangle features and upper-half features solves the problem. There is psychological evidence for combination coding of some but not all visual features (Triesman 1982) and for integrated object concepts (Luck and Beach 1998). While combination coding cannot be the solution to the entire action representation-induction problem, object-hood is such a fundamental requirement that it must be built into The lower-level input representation itself.

One way of thinking of integrated object concepts is to think of the input to the system as a map in which objects are represented by locations, and facts like  $door(d)$  and  $open(d)$  are represented by (sparse) set bits on a vector representing the value of all facts that can hold of that location. (The space need not be physical or perceptual space but its easiest to think of it that way for now.) Some part of the input vector to the associative network cascade then represents an object/location in the map: if it is a door-location and the door-location is a shut-location and the non-object specific part of the vector says the you-location is an in-location then there is a you-push-the-door-location affordance. This general picture seems in keeping with the observations of O'Keefe and Nadel (1978), Morris et al. (1982), O'Keefe (1989), and McNaughton (1989), concerning single-cell recording from rat hippocampus.

The associative cascade then becomes a function from object/locations to affordances and their results, in which propositions like  $door(d)$  and  $open(d)$  are simply the relevant bits of the vector and the identity of the object-location is implicit. One can either think of this function being applied to successive positions in a scan or (more likely) of the object/locations as proposing themselves by some autonomous salience mechanism.

The latter assumption is attractive, and makes object-concepts and their recognition central to the planning process. One can think of the scene as proposing locations for attention according to the a priori value of the objects they contain in terms of successful planning in the past. Such a representation is “deictic,” like that of Finney et al. 2002b,a, but differs in having an active attention/focus mechanism. The associative memory then has the effect of turning doors and other objects into functions from all and only the actions that are afforded by the situation and the things that they include onto the states that result from applying those actions to those things.

The operation of turning something into a function from the Functions-that-apply-to-it into the results-of-applying-them-to-it is the second major combinator that the planner is based on, namely *type-raising*, usually written as  $\mathbf{T}$ . The concept of a door can be defined as follows:

$$(17) \ door' = \lambda x_{door}. \mathbf{T}x$$

—where

$$(18) \ \mathbf{T}a_{type} \equiv \lambda p_{(type \rightarrow state)}. p(a)$$

The door concept (17) can then be thought of as a function from things of type *door* onto functions from functions-from-things-of-type-*door*-into-states-that-result-from-applying-those-functions-to-those-things.

Interestingly, there is more information in the affordance network than the above minimal planner is using. The unthresholded activation of *affords(go-through(you,d))* in Figure 6 is almost as high as that of *affords(push(you,d))*, reflecting the fact that the situation *nearly* allows

going through.

The information that the problem is the door being shut rather than open is potentially available from the network, and could in principle be used to direct the agent’s attention to those of the actions that the situation affords that specifically involve doors, such as *push*. Even if the agent doesn’t yet know the *push* action that changes closed doors to open doors, it has a good chance of discovering this operator by blind exploration of actions involving the door.

There may be objects other than doors in the situation which are *open*—bottles, say—which also have associated affordances that are activated by the situation—say, drinking. However, none of those actions will turn open bottles into open doors, so it will do no harm to ignore them, once it is clear that none of them bear directly on the goal.

The object-oriented nature of these associations is an example of a basic fact about animal ontology. We recognize the object-property status of *door* and *bottle*, as opposed to *open* by their consistent relation to actions. Doors consistently afford egress, and bottles consistently afford drinking, whereas the class of things that are open isn’t consistent in affordances. To see how the associative network can reveal this distinction, we must turn to the question of how the change network of sparse STRIPS FSTs can be learned from completely specified input-output states.

## 7 Perceptron-Associative Networks

The associative net can be regarded as a multiple-output Perceptron (Minsky and Papert 1969, 1988b).<sup>10</sup> In the autoassociative form as presented so far, it is a perceptron in which the initial weights are all zero and the gain is 1. However, in order to train such a device on STRIPS rules, we had to tell it explicitly which (sparse) bits were 1s and which 0s. We want the machine to work that out for itself, and associate situations including things with properties like *door*, *bottle*, and *open* with actions like *push*, *go-through* and *drink*.

The following is a proposal for how we might be able to do this using a version of the associative net in which weights are positive or negative real-valued, adjusted to minimize error from a random initial setting using some form the Perceptron Learning Algorithm (PLA), according to their positive or negative contribution to the decision of each bit in the output. This is work in progress, not a confirmed mechanism.

One of the properties of the PLA is that it will set weights on bits whose input value is irrelevant to zero. We will pass over the details of the PLA here, referring the reader to Rosenblatt 1962, Samuel 1959, and Russell and Norvig 2003:742, together with the important generalization of Freund and Shapire (1999) of the PLA to non-linear classification, using the “kernel trick” originated by Aizerman, Braverman and Rozonoer (1964). The way it is applied is as follows.

Every time an action  $\alpha$  is successfully executed, the input state vector with the bit representing  $affords(\alpha)$  is autoassociated, the weights being updated according to some version of the PLA. When learning is complete, and in offline planning, the associated affordance bit is retrieved and included as input to the change matrix, as in Figure 7

Every time an action  $\alpha$  is successfully executed, the change associative memory is also up-

<sup>10</sup>Just as there are multilayered perceptrons, so there are multilayered Associative Nets, such as Hopfield Nets (Hopfield 1982, Boltzmann Machines (Hinton and Sejnowski 1986), and Recursive AutoAssociative Memory (RAAM Pollock 1990). These devices may well also be applicable to the problem of learning and deploying STRIPS/LDEC rules. However, like multi-layered Perceptrons (Minsky and Papert 1988a), multi-layered Associative Networks are prey to false minima and are hard to train and generalize, and we will ignore their possibilities here.

dated according to the PLA, the input state vector with the bit representing  $\text{affords}(\alpha)$  set being associated with the output vector with the bit representing  $\text{affords}(\alpha)$  unset.

One interesting property of this proposal is that the weights potentially embody a distinction between object-properties like *door* and *bottle*, and other properties like *in* and *open*.

Object properties correspond to facts that are invariant under the effect of (most) actions. (That is to say that the fact in question holds as long as the object exists. Crucially, if the action destroys the object, or transmutes it, *all* the related facts and properties typically either change or entirely cease to hold.) They therefore have very high weights on the diagonal linking the same facts in input and output. Properties like *in* and *open* are sometimes changed by actions, so their weights on the diagonal are lesser.<sup>11</sup>

The emergence of the object-property distinction is important for the enterprise of mapping the action representation into natural language semantics. *Door* and *bottle* become properties of type  $e \rightarrow t$ , but *open* must be assigned the second-order type  $(e \rightarrow t) \rightarrow (e \rightarrow t)$ . This distinction is in turn grammaticalized as that between nouns *N* and adjectives *N/N*, and in turn provides the basis for a distinction between the “head” of the nominal construction and the “adjunct”.

This model is at present untested, and it is unclear whether it will be necessary to build more structure into the net—for example, to compensate for the fact that most fluents are not affected by most actions, and hence are highly auto-correlated. The way to actually test the hypothesis is to generate actual state spaces using a symbolic AI-style LDEC planner, then show that the network can learn the operators. This experiment is currently being actively pursued at Edinburgh using the PKS planner of Petrick and Bacchus (2002, 2004). The present account is put forward merely as an example of what a neurocomputational theory of planning might look like.

## 8 Languages which Lexicalize Affordance

Many North American Indian languages, such as the Athabaskan group that includes Navajo, are comparatively poorly-off for nouns. Many nouns for artefacts are morphological derivatives of verbs. For example, “door” is *ch’éétiiñ*, meaning “something has a path horizontally out”, a gloss which has an uncanny resemblance to (7). This process is completely productive: “towel” is *bee’ádít’oodí*, glossed as “one wipes oneself with it”, and “towelrack” is *bee’ádít’oodíbqáh dah náhidiiltsos*—roughly “one wipes oneself with it is repeatedly hung on it” (Young and Morgan 1987)

Such languages thus appear to lexicalize nouns as a *default affordance* (**T**), and to *compose* such affordances (**B**). Of course, we should avoid crassly Whorfian inferences about Navajo-speakers’ reasoning about objects. Though productive, these lexicalizations are as conventional as our own.<sup>12</sup>

Navajo nouns are also implicitly classified by animacy, shape, and consistency. However, rather than being realized via a rich gender system, as in some other Athabaskan languages such as Koyukon, this classification is in Navajo reflected in verbal morphology. For example, the classifier *-iltsos* on the verb *náhidiiltsos*, “hung,” marks it as predicated of flat, flexible things like towels. A belt-rack or a gun rack would have a different classifier.

<sup>11</sup>Such object concepts may nevertheless be quite abstract—holes are an example.

<sup>12</sup>Navajo-speakers probably find equally exotic the propensity of English to generate denominal verbs, like “table” and “pocket,” with equal productivity.

Wikipedia gives the following table of Navajo Classifiers (the orthographic conventions are slightly different from those used in the examples from Young and Morgan 1987).

(19) *Navaho Classifiers:*

Classifier+Stem Label	Explanation	Examples
-'q	SRO Solid Roundish Object	bottle, ball, boot, box, etc.
-yí	LPB Load, Pack, Burden	backpack, bundle, sack, saddle, etc.
-ł-jool	NCM Non-Compact Matter	bunch of hair or grass, cloud, fog, etc.
-lá	SFO Slender Flexible Object	rope, mittens, socks, pile of fried onions, etc.
-tq	SSO Slender Stiff Object	arrow, bracelet, skillet, saw, etc.
-ł-tsooz	FFO Flat Flexible Object	blanket, coat, sack of groceries, etc.
-tléé'	MM Mushy Matter	ice cream, mud, slumped-over drunken person, etc.
-nil	PLO1 Plural Objects 1	eggs, balls, animals, coins, etc.
-jaa'	PLO2 Plural Objects 2	marbles, seeds, sugar, bugs, etc.
-ká	OC Open Container	glass of milk, spoonful of food, handful of flour, etc.
-ł-tí	ANO Animate Object	microbe, person, corpse, doll, etc.

As a consequence, the English verb “give” is expressed by 11 different forms in Navajo, depending on the characteristics of the object given, including *níłjool* (give-NCM), used in “give me some hay” and *nítjih* (give-SSO), used in “give me a cigarette”.<sup>13</sup>

The appearance of such pronominal classifiers on the verb is an example of a “head marking” system of *case*, insofar as the final position of such classifiers “structurally” marks the fact that they are patients of the action (cf. Blake 2001:13). The interest of such classifiers and their reflex in Navajo nominalizations as a form of case marking agreement is twofold. First, if these classifiers appear explicitly in Navajo, one might expect that they reflect a universal ontology of entities. The advantage of such ontologies is that they allow an agent to generalize the notion of affordances of doors to other actions applying to objects of that class. The extension to a system of case allows even further generalization to the full range of transitive actions. Second, the type-raising nature of case shows up very directly in the theory of grammar, considered next.

## 9 B, T, and the Combinatory Projection Principle

Besides supporting the basic operations of seriation and object-orientation that planning depends upon, syntactic versions of combinators **B**, **T** support a rebracketing and reordering calculus of exactly the kind that is needed to capture natural language syntax, and provide the basis of Combinatory Categorial Grammar (CCG, Ades and Steedman (1982)—see Steedman 2000b for references)

CCG eschews language-specific syntactic rules like (20) for English. Instead, all language-specific syntactic information is *lexicalized*, via lexical entries like (21) for the English transitive

---

<sup>13</sup>I once read a transcript of a Navajo radio broadcast that is revealing in this connection. The participants were discussing how to translate the name of the band called Hootie and the Blowfish. They had no trouble with “Hootie” and “fish”, but thought “blow” deplorably vague, demanding to know exactly *who* was blowing exactly *what* and with *what result*, in order to come up with the correct translation—roughly, “fish which inflates itself”.

verb:

~~(20)  $S \rightarrow NP VP$~~   
 ~~$VP \rightarrow TV NP$~~   
 $TV \rightarrow \{proved, finds, \dots\}$

(21)  $\text{proved} := (S \setminus NP) / NP$

This syntactic “category” identifies the transitive verb as a function, and specifies the type and directionality of its arguments and the type of its result,  $/NP$  indicating an NP argument to the right,  $\setminus NP$  indicating an NP argument to the left, and the brackets indicating that the rightward argument is the first argument to combine.

The category (21) also reflects its semantic type ( $e \rightarrow (e \rightarrow t)$ ), expressed in (22a) below as a lambda term paired with it via a colon operator, in which primes mark constants, non-primes are variables, and concatenation denotes function application under a “left associative” convention, so that the expression  $\text{prove}'xy$  is equivalent to  $(\text{prove}'x)y$ .

We follow Baldridge (2002) in generalizing this notation to freer word order languages as follows, where brackets {} enclose one or more sets of arguments that can combine in any order, and the preceding slash /, \, or | indicates that all members of the set must be found to the right, left or either direction respectively. We also generalize the semantic notation using a parallel argument set notation for lambda terms and a convention that pairs the unordered syntactic arguments with the unordered semantic arguments in the left-to-right order in which they appear on the page. Typical transitive verb categories then appear as follows:<sup>14</sup>

- (22) a. English:  $(S \setminus NP) / NP : \lambda x \lambda y. \text{prove}'xy$   
 b. Latin:  $S | \{NP_{nom}, NP_{acc}\} : \lambda \{y, x\}. \text{prove}'xy$   
 c. Tagalog:  $S / \{NP_{nom}, NP_{acc}\} : \lambda \{y, x\}. \text{prove}'xy$   
 d. Japanese:  $S \setminus \{NP_{nom}, NP_{acc}\} : \lambda \{y, x\}. \text{prove}'xy$

Such categories should be thought of as schemata covering a finite number of deterministic categories like (22a).

Some very general syntactic rules, corresponding to function application, and the combinator **B** and **T**, together with a third combinator **S** which we will pass over here, but which is parallel in every respect to **B**, then constitute the universal mechanism of syntactic derivation or projection onto the set of all and only the sentences of the language specified by its CCG lexicon. This Universal set of rules is the following:

(23) *The functional application rules*

- |   |     |
|---|-----|
| a. $X /_* Y : f \quad Y : a \Rightarrow X : fa$         | (>) |
| b. $Y : a \quad X \setminus_* Y : f \Rightarrow X : fa$ | (<) |

---

<sup>14</sup>These categories are deliberately simplified for expository purposes, and certainly overstate the degree to which alternative constituent orders are semantically equivalent in these languages.

(24) *The functional composition rules*

- a.  $X/\diamond Y:f \quad Y/Z:g \Rightarrow_B X/\diamond Z:\lambda x.f(gx)$  ( $>\mathbf{B}$ )
- b.  $Y\backslash_{\diamond} Z:g \quad X\backslash_{\diamond} Y:f \Rightarrow_B X\backslash_{\diamond} Z:\lambda x.f(gx)$  ( $<\mathbf{B}$ )
- c.  $X/\times Y:f \quad Y\backslash_{\times} Z:g \Rightarrow_B X\backslash_{\times} Z:\lambda x.f(gx)$  ( $>\mathbf{B}_{\times}$ )
- d.  $Y/\times Z:g \quad X\backslash_{\times} Y:f \Rightarrow_B X/\times Z:\lambda x.f(gx)$  ( $<\mathbf{B}_{\times}$ )

(25) *The order-preserving type-raising rules*

- a.  $X:a \Rightarrow_T T/i(T\backslash_i X):\lambda f.f a$  ( $>\mathbf{T}$ )
- b.  $X:a \Rightarrow_T T\backslash_i(T/i X):\lambda f.f a$  ( $<\mathbf{T}$ )

The types  $\star$ ,  $\diamond$ , and  $\times$  on the slashes in rules (23) restrict the categories that may combine by them. While all categories seen so far have the unadorned slash types  $/$ ,  $\backslash$ , or  $|$ , which can combine by any rule, the language-specific lexicon can restrict the combinatory potential of lexical function categories using these slash-types. Thus, coordinators like *and* are restricted via the  $\star$  type to *only* combine by the application rules:

$$(26) \text{ and } := (X\backslash_{\star} X)/_{\star} X$$

The  $\diamond$  slash-type on a function category means that it can combine *either* by the application rules (23), *or* by the rules  $>\mathbf{B}$  and  $<\mathbf{B}$  bearing that slash-type in (24), but *not* by the rules  $>\mathbf{B}_{\times}$  or  $<\mathbf{B}_{\times}$ . In English (as opposed to, say, Latin), adjectives are restricted using this slash-type, because they are entirely fixed in terms of linear order with respect to the head, unlike adverbs, and it is the latter rules that allow reordering:

$$(27) \text{ young } := N/\diamond N$$

The variable  $i$  type on the type-raising rules (25) means that the raised category inherits the slash-type of its argument.

The composition rules are all generalized to cover cases where the “lower” function  $Y|Z$  and the result  $X|Z$  are of higher valency  $(Y|Z)|W$  and  $(X|Z)|W$ , etc., up to some low value such as 4  $((Y|Z)|W)|V|U$  and  $((X|Z)|W)|V|U$ , which appears to be the highest valency in the universal inventory of lexical types (Newmeyer 2005, citing Pesetsky 1995). It is the combination of crossed composition, as in  $>\mathbf{B}_{\times}$  and  $<\mathbf{B}_{\times}$ , and this generalization that increases the expressive power of the formalism to the lowest known trans-context-free level of the “mildly context-sensitive” class identified by Joshi, Vijay-Shanker and Weir (1991), weakly equivalent to basic Lexicalized Tree-Adjoining Grammars (LTAG) and Linear Indexed Grammars (LIG). The theory thus embodies a very strong claim about a Formal Universal, namely that all natural languages fall into this low-power class.

A number of Principles which amount to the following statement mean that these are the *only* combinatory rules that are available to Universal Grammar:

(28) *The Strict Lexicalization Principle*

The universal combinatory rules must project, and may not override, the directionality and slash-type specified in the language-specific lexicon

This theory has been applied to the linguistic analysis of coordination, relativization, and intonational structure in English and many other languages (Steedman 1996, 2000a; Hoffman 1995; Bozsahin 1998; Komagata 1999; Baldridge 1998, 2002). For example, we can define relativization without syntactic movement or empty categories, as in (30), via the following

category for the relative pronoun:

$$(29) \text{ that} := (N \setminus_{\diamond} N) / (S/NP)$$

This category yields the following derivation:

$$(30) \begin{array}{ccccccc} \text{(The woman)} & \text{that} & \text{Thelma} & \text{met} \\ \overline{(N \setminus_{\diamond} N) / (S/NP)} & \overline{S / (S \setminus NP_{3SG})} & \overline{(S \setminus NP_{3SG}) / NP} & \\ & \overline{\overline{S / (S \setminus NP_{3SG})} >^T} & \overline{(S \setminus NP_{3SG}) / NP} & \\ & & \overline{S / NP} & \\ & & \overline{\overline{S / NP} >^B} & \\ & & \overline{N \setminus_{\diamond} N} & \\ & & \overline{\overline{N \setminus_{\diamond} N} >} & \end{array}$$

Such “extractions” are correctly predicted to be unbounded, since composition can operate across clause boundaries:

$$(31) \begin{array}{ccccccccc} \text{(The woman)} & \text{that} & \text{Thelma} & \text{says} & \text{she} & \text{met} \\ \overline{(N \setminus_{\diamond} N) / (S/NP)} & \overline{S / (S \setminus NP_{3SG})} & \overline{(S \setminus NP_{3SG}) / S} & \overline{S / (S \setminus NP_{3SG})} & \overline{(S \setminus NP_{3SG}) / NP} & \\ & & \overline{\overline{S / (S \setminus NP_{3SG})} >^T} & \overline{(S \setminus NP_{3SG}) / NP} & & \\ & & \overline{S / S} & & & \\ & & \overline{\overline{S / S} >^B} & & & \\ & & & \overline{S / NP} & & \\ & & & \overline{\overline{S / NP} >^B} & & \\ & & & & \overline{N \setminus_{\diamond} N} & \\ & & & & \overline{\overline{N \setminus_{\diamond} N} >} & \end{array}$$

It is the lexical category (29) of the relative pronoun that establishes the long-range dependency between noun and verb (via the semantics defined in the lexicon via the logical form (not shown here): syntactic derivation merely projects it onto the phrasal logical form, with composition and type-raising, as well as application, doing the work of Merge rather than Move, in the terms of the Minimalist Program.

The conjunction category (26) allows a related movement- and deletion- free account of right node raising, as in (32):

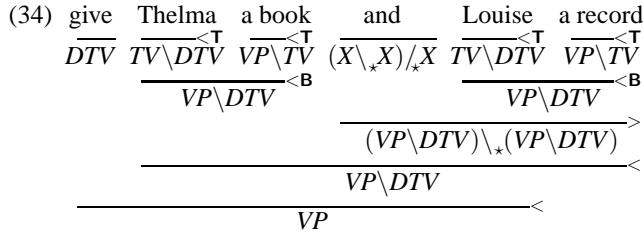
$$(32) \begin{array}{c} [\text{Thelma met}] \quad \text{and} \quad [\text{Fred says he likes}] \quad \text{Louise} \\ \overline{S / NP} >^B \quad \overline{(X \setminus_{\star} X) / X} \quad \overline{S / NP} >^B \quad \overline{S \setminus (S / NP)} <^T \\ \overline{\overline{(X \setminus_{\star} X) / X} >} \quad \overline{S / NP} >^B \\ \overline{\overline{(S / NP) \setminus_{\star} (S / NP)} >} \\ \overline{\overline{\overline{(S / NP) \setminus_{\star} (S / NP)} <} <} \\ \overline{\overline{\overline{S} <}} \end{array}$$

The  $\star$  modality on the conjunction category (26) means that it can *only* combine like types by the application rules (23). Hence, the across-the-board condition (ATB) on extractions from coordinate structures (including the “same case” condition) is captured:

- (33) a. A woman [that<sub>(N \setminus\_{\diamond} N) / (S/NP)</sub> [[Thelma met]<sub>S / NP</sub> and [Louise likes]<sub>S / NP</sub>]<sub>S / NP</sub>]<sub>N \setminus\_{\diamond} N</sub>
- b. A woman [that<sub>(N \setminus\_{\diamond} N) / (S/NP)</sub> \*[[Thelma met]<sub>S / NP</sub> and [likes Louise]<sub>S / NP</sub>]<sub>S / NP</sub>]<sub>N \setminus\_{\diamond} N</sub>
- c. A woman that<sub>(N \setminus\_{\diamond} N) / (S/NP)</sub> \*[[Thelma met]<sub>S / NP</sub> and [Louise likes her]<sub>S</sub>]]
- d. A woman that<sub>(N \setminus\_{\diamond} N) / (S/NP)</sub> \*[[Thelma met her]<sub>S</sub> and [Louise likes]<sub>S / NP</sub>]

CCG offers startlingly simple analyses of a wide variety of further coordination phenomena, including English “argument-cluster coordination”, “backward gapping” and “verb-raising” constructions in Germanic languages, and English gapping. The first of these is illustrated by the following analysis, from Dowty (1988—cf. Steedman 1985), in which the ditransitive verb

category  $(VP/NP)/NP$  is abbreviated as  $DTV$ , and the transitive verb category  $VP/NP$  is abbreviated as  $TV$ :<sup>15</sup>



The universal set of combinatory rules does not allow any derivation for word orders like the following, given the lexicon of English:

- (35) \*Thelma a book and give Louise a record.

Thus, the universal noted by Ross (1970) concerning the direction of gapping and the base order of constituents in constructions is a theorem of the theory of extraction without movement based on combinatory projection with rules based on **B** and **T**.

It should be evident from the fact that the type raising operation in (30) turns the NP *Thelma* into a function over predicates  $S \setminus NP$ , while in (34) it turns the same word into a function over ditransitive verbs  $(VP/NP) \setminus ((VP/NP)/NP)$  and the NP *a book* into a function over transitive verbs  $VP \setminus (VP/NP)$  that type-raising, even in English, is simply (respectively: nominative, dative and accusative) grammatical *case*, albeit marked “structurally” by position with respect to the verb, rather than morphologically, as in Latin *Thelma*, *Thelmae*, *Thelمام*. We have seen that notions of case and affordance are highly related. Thus sentence (34) can be seen as composing pairs of functions over affordances and conjoining the result.

It is likely that a number of other universals concerning possible word orders can be base-generated on similar assumptions of a universal projection principle based on the combinators **B** and **T**. Universal 20 of Greenberg (1963) concerning the possible base orders of Dem, Num, A and N, as expanded by Hawkins (1983) and Cinque (2005), is particularly promising in this respect, as Hawkins 1983:121-122 points out.

The close relation between the combinatory syntactic primitives and those involved in planned action should not come as a surprise. If we turn to those aspects of language which presumably reflect its origin most directly, namely its use to manipulate the actions of others to our own advantage, then it is clear that this is quintessentially a planning problem, rather than a distinctively linguistic one. For example, the problem of identifying the fact that the utterance most likely to effect the manipulation of getting the window shut is not the imperative “Shut the window” but the declarative “It’s cold in here” can be captured in essentially the same terms of affordance and change in knowledge state that are used to plan with doors and locations, the main difference lying in the fact that representation of the state of other minds is required, as discussed in Steedman 2002, 2006.

<sup>15</sup>In more recent work, Dowty has disowned this analysis, because of the implicit “intrinsic” use of logical form that it entails.

## 10 Conclusion

This paper has sketched a theory of the way in which experience shapes object- and action-concepts, how they are used to plan purposive actions in dynamic worlds, and how this system forms a basis for language, to which the latter is almost entirely transparent. This account is highly speculative. In particular, it remains to be shown that the particularly simple form of associative memory assumed here is capable in practice of the kind of learning required, or whether some other form is needed, such as that proposed by Plate (1991), and whether such mechanisms scale to realistically-sized problems. Many details will undoubtedly have to be changed, and many more filled in.

Nevertheless, it seems likely that a proper theory of action representation will have to embody the ideas of object-orientation and dynamism, embodied in the associative memory mechanisms that have long been associated with the hippocampus, that are assumed here. The fact that the language faculty, whose syntactic aspects have long been thought to be quite mysterious and unique, appears to reflect these properties so directly may lend conviction to this expectation.

## References

- Ades, Anthony, and Mark Steedman. 1982. "On the Order of Words." *Linguistics and Philosophy*, 4, 517–558.
- Agre, Phillip, and David Chapman. 1987. "Pengi: An Implementation of a Theory of Activity." In *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*. Los Altos, CA: Morgan Kaufmann.
- Aizerman, M.A., E.M. Braverman, and L.I. Rozonoer. 1964. "Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning." *Automation and Remote Control*, 25, 821–837.
- Allen, James. 1984. "Towards a General Theory of Action and Time." *Artificial Intelligence*, 23, 123–154.
- Allen, James, and Patrick Hayes. 1989. "Moments and Points in an Interval-Based Temporal Logic." *Computational Intelligence*, 5, 225–238.
- Anderson, John. 1968. "A Memory Storage Model Utilizing Spatial Correlation Functions." *Kybernetik*, 5, 113–119.
- Baldridge, Jason. 1998. *Local Scrambling and Syntactic Asymmetries in Tagalog*. Master's thesis, University of Pennsylvania.
- Baldridge, Jason. 2002. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh.
- Bertsekas, Dimitri, and John Tsitsiklis. 1996. *Neuro-dynamic Programming*. Belmont, MA: Athena Scientific.

- Bibel, Wolfgang, Luis Farinas del Cerro, B. Fronhfer, and A. Herzig. 1989. “Plan generation by linear proofs: on semantics.” In *German Workshop on Artificial Intelligence - GWAI’89*, vol. 216 of *Informatik-Fachberichte*. Berlin: Springer Verlag.
- Blake, Barry. 2001. *Case*. Cambridge: Cambridge University Press, second edn.
- Bozsahin, Cem. 1998. “Deriving Predicate-Argument Structure for a Free Word Order Language.” In *Proceedings of COLING-ACL ’98, Montreal*, 167–173. Cambridge, MA: MIT Press.
- Brooks, Rodney A. 1986. “A Robust Layered Control System for a Mobile Robot.” *IEEE Journal of Robotics and Automation*, 2, 14–23.
- Bryson, Joanna. 2001. “Intelligent Control Requires More Structure than the Theory of Event Coding Provides.” *Behavioral and Brain Sciences*, 24, 878–879. Open peer commentary on Hommel et al. (2001).
- Bryson, Joanna, and Lynn Andrea Stein. 2001. “Modularity and Design in Reactive Intelligence.” In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*. AAAI.
- Buszkowski, Wojciech, and Gerald Penn. 1990. “Categorial Grammars Determined from Linguistic Data by Unification.” *Studia Logica*, 49, 431–454.
- Chomsky, Noam. 1957. *Syntactic Structures*. The Hague: Mouton.
- Chomsky, Noam. 1965. *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press.
- Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.
- Cinque, Guglielmo. 2005. “Deriving Greenberg’s Universal 20 and its Exceptions.” *Linguistic Inquiry*, 36, 315–332.
- Curry, Haskell B., and Robert Feys. 1958. *Combinatory Logic: Vol. I*. Amsterdam: North Holland.
- Damasio, Antonio. 1999. *The Feeling of What Happens*. New York: Harcourt.
- Dowty, David. 1979. *Word Meaning in Montague Grammar*. Dordrecht: Reidel.
- Dowty, David. 1988. “Type-Raising, Functional Composition, and Nonconstituent Coordination.” In Richard T. Oehrle, Emmon Bach, and Deirdre Wheeler, eds., *Categorial Grammars and Natural Language Structures*, 153–198. Dordrecht: Reidel.
- Dryer, Matthew. 1992. “The Greenbergian Word Order Correlations.” *Language*, 68, 81–138.
- Fikes, Richard, and Nils Nilsson. 1971. “STRIPS: a new approach to the application of theorem proving to problem solving.” *AI Journal*, 2, 189–208.
- Finney, Sarah, Natalia H. Gardiol, Leslie Pack Kaelbling, and Tim Oates. 2002a. “Learning with Deictic Representation.” AI Laboratory Memo AIM-2002-006, MIT.

- Finney, Sarah, Natalia H. Gardiol, Leslie Pack Kaelbling, and Tim Oates. 2002b. “The Thing That We Tried Didn’t Work Very Well: Deictic Representation in Reinforcement Learning.” In *Proceedings of the 18th International Conference on Uncertainty in Artificial Intelligence*. Edmonton, AAAI.
- Freund, Yoav, and Robert Shapire. 1999. “Large Margin Classification Using the Perceptron Algorithm.” *Machine Learning*, 37, 277–296.
- Girard, Jean-Yves. 1987. “Linear Logic.” *Theoretical Computer Science*, 50, 1–102.
- Gluck, Mark, and Catherine Myers. 2000. *Gateway to Memory*. Cambridge MA: MIT Press.
- Gold, E. M. 1967. “Language Identification in the Limit.” *Information and Control*, 16, 447–474.
- Greenberg, Joseph. 1963. “Some Universals of Grammar with Particular Reference to the Order of Meaningful Elements.” In Joseph Greenberg, ed., *Universals of Language*, XXX–XXX. Cambridge MA: MIT Press.
- Hanks, Steven, and Drew McDermott. 1986. “Default Reasoning, Nonmonotonic Logics, and the Frame Problem.” In *Proceedings of the 8th National Conference on Artificial Intelligence (AAAI-86)*, 328–333.
- Harel, David. 1984. “Dynamic Logic.” In Dov Gabbay and F. Guenthner, eds., *Handbook of Philosophical Logic*, vol. II, 497–604. Dordrecht: Reidel.
- Hauser, Marc, Noam Chomsky, and W. Tecumseh Fitch. 2002. “The Faculty of Language: What Is It, Who Has It, and How did it Evolve?” *Science*, 298, 1569–1579.
- Hawkins, John. 1983. *Word Order Universals*. New York: Academic.
- Hayes, Patrick. 1979. “The Naive Physics Manifesto.” In Donald Michie, ed., *Expert Systems in the Microelectronics Age*, 242–270. Edinburgh: Edinburgh University Press.
- Hinton, Geoffrey, ed. 1990. *Connectionist Symbol Processing*. Cambridge, MA: MIT Press/Elsevier. Reprint of *Artificial Intelligence*, 46:1-2.
- Hinton, Geoffrey, and Terrence Sejnowski. 1986. “Learning and Relearning in Boltzmann Machines.” In James McClelland, David Rumelhart, and the PDP Research Group, eds., *Parallel Distributed Processing*, vol. 1, 282–317. Cambridge, MA: MIT Press.
- Hoffman, Beryl. 1995. *Computational Analysis of the Syntax and Interpretation of “Free” Word-Order in Turkish*. Ph.D. thesis, University of Pennsylvania. IRCS Report 95-17.
- Hoffman, Jörg, and Bernhard Nebel. 2001. “The FF Planning System: Fast Plan Generation Through Heuristic Search.” *Journal of Artificial Intelligence Research*, 14, 253–302.
- Hommel, Bernhard, Jochen Müsseler, Gisa Aschersleben, and Wolfgang Prinz. 2001. “The Theory of Event Coding (TEC): A Framework for Perception and Action Planning.” *Behavioral and Brain Sciences*, 24, 849–878.

- Hopfield, John. 1982. “Neurons with Graded Response have Collective Computational Properties Like Those of Two-State Neurons.” *Proceedings of the National Academy of Sciences of the United States of America*, 79, 2554–2558.
- Horning, James. 1969. *A Study of Grammatical Inference*. Ph.D. thesis, Stanford.
- Huybregts, Riny. 1984. “The Weak Inadequacy of Context-free Phrase-structure Grammars.” In Ger de Haan, Mieke Trommelen, and Wim Zonneveld, eds., *Van Periferie naar Kern*. Dordrecht: Foris.
- Joshi, Aravind, K. Vijay-Shanker, and David Weir. 1991. “The Convergence of Mildly Context-Sensitive Formalisms.” In Peter Sells, Stuart Shieber, and Tom Wasow, eds., *Processing of Linguistic Structure*, 31–81. Cambridge, MA: MIT Press.
- Kaplan, Ron, and Lauri Karttunen. 1994. “Regular Models of Phonological Rule Systems.” *Computational Linguistics*, 20, 331–378.
- Komagata, Nobo. 1999. *Information Structure in Texts: A Computational Analysis of Contextual Appropriateness in English and Japanese*. Ph.D. thesis, University of Pennsylvania.
- Kowalski, Robert. 1979. *Logic for Problem Solving*. Amsterdam: North Holland.
- Kowalski, Robert, and Maurice Sergot. 1986. “A Logic-Based Calculus of Events.” *New Generation Computing*, 4, 67–95.
- Lasnik, Howard, and Juan Uriagereka. 2005. *A Course in Minimalist Syntax*. Oxford: Blackwell.
- Luck, Steven, and Nancy Beach. 1998. “Visual attention and the binding problem: A neurophysiological perspective.” In Richard Wright, ed., *Visual Attention*, 455–478. Oxford: Oxford University Press.
- von der Malsberg, Christoph. 1995. “Binding in Models of Perception and Brain Function.” *Current Opinion in Neurobiology*, 5, 520–526.
- Marr, David. 1969. “A Theory of Cerebellar Cortex.” *Journal of Physiology*, 202, 437–470. Reprinted in Vaina 1991.
- Marr, David. 1970. “A Theory for Cerebral Neocortex.” *Proceedings of the Royal Society of London*, 176, 161–234. Reprinted in Vaina 1991.
- Marr, David. 1971. “Simple Memory: A Theory for Archicortex.” *Philosophical Transactions of the Royal Society of London*, 262, 23–81. Reprinted in Vaina 1991.
- Martí-Oliet, Narciso, and José Meseguer. 1999. “Action and Change in Rewriting Logic.” In Remo Pareschi and B. Fronhöfer, eds., *Dynamic Worlds*, 1–53. Dordrecht: Kluwer.
- McCarthy, John, and Patrick Hayes. 1969. “Some Philosophical Problems from the Standpoint of Artificial Intelligence.” In Bernard Meltzer and Donald Michie, eds., *Machine Intelligence*, vol. 4, 473–502. Edinburgh: Edinburgh University Press.

- McNaughton, B.L. 1989. "Neuronal Mechanisms for Spatial Computation and Information Storage." In Lynn Nadel, Lynn Cooper, Peter Culicover, and Michael Harnish, eds., *Neural Connections, Mental Computation*, 285–350. Cambridge, MA: MIT Press.
- Minsky, Marvin, and Seymour Papert. 1969. *Perceptrons*. Cambridge, MA: MIT Press.
- Minsky, Marvin, and Seymour Papert. 1988a. *Epilogue*, 247–280. In Minsky and Papert (1988b). Expanded second edition of Minsky and Papert 1969.
- Minsky, Marvin, and Seymour Papert. 1988b. *Perceptrons: Expanded Edition*. Cambridge, MA: MIT Press. Expanded second edition of Minsky and Papert 1969.
- Morris, Richard, Paul Garrud, J Rawlins, and John O'Keefe. 1982. "Place navigation impaired in rats with hippocampal lesions." *Nature*, 297, 681–683.
- Newmeyer, Frederick. 2005. *Possible and Probable Languages*. Oxford: Oxford University Press.
- O'Keefe, John. 1989. "Computations the Hippocampus Might Perform." In Lynn Nadel, Lynn Cooper, Peter Culicover, and Michael Harnish, eds., *Neural Connections, Mental Computation*, 225–284. Cambridge, MA: MIT Press.
- O'Keefe, John, and Lynn Nadel. 1978. *The Hippocampus as a Cognitive Map*. Oxford: Oxford University Press.
- Pesetsky, David. 1995. *Zero Syntax*. Cambridge, MA: MIT Press.
- Petrick, Ronald P. A., and Fahiem Bacchus. 2002. "A Knowledge-Based Approach to Planning with Incomplete Information and Sensing." In Malik Ghallab, Joachim Hertzberg, and Paolo Traverso, eds., *Proceedings of the Sixth International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2002)*, 212–221. Menlo Park, CA: AAAI Press.
- Petrick, Ronald P. A., and Fahiem Bacchus. 2004. "Extending the Knowledge-Based Approach to Planning with Incomplete Information and Sensing." In Shlomo Zilberstein, Jana Koehler, and Sven Koenig, eds., *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS-04)*, 2–11. Menlo Park, CA: AAAI Press.
- Pinker, Steven. 1979. "Formal Models of Language Learning." *Cognition*, 7, 217–283.
- Plate, Tony. 1991. "Holographic Reduced Representations: Convolution Algebra for Compositional Distributed Representations." In *Proceedings of the 12th International Joint Conference on Artificial Intelligence, San Mateo CA*, 30–35. San Francisco, CA: Morgan Kaufmann.
- Pollack, Jordan. 1990. "Recursive Distributed Representations." *Artificial Intelligence*, 46, 77–105. Reprinted in Hinton (1990).
- Poole, David. 1993. "Probabilistic Horn Abduction and Bayesian Networks." *Artificial Intelligence*, 64, 81–129.

- Rosenbaum, Harvey. 1977. “Zapotec Gapping as Counterevidence to Some Universal Propositions.” *Linguistic Inquiry*, 8, 379–395.
- Rosenblatt, Frank. 1962. *Principles of Neurodynamics*. New York: Spartan Books.
- Ross, John Robert. 1970. “Gapping and the Order of Constituents.” In Manfred Bierwisch and Karl Heidolph, eds., *Progress in Linguistics*, 249–259. The Hague: Mouton.
- Russell, Stuart, and Peter Norvig. 2003. *Artificial Intelligence: a Modern Approach*. Upper Saddle River NJ: Prentice Hall, second edn.
- Samuel, Arthur. 1959. “Some Studies in Machine Learning Using the Game of Checkers.” *IBM Journal of Research and Development*, 3, 211–229.
- Sandewall, Eric. 1994. *Features and Fluents*. Oxford: Oxford University Press.
- Shanahan, Murray. 1997. *Solving the Frame Problem*. Cambridge, MA: MIT Press.
- Shastri, Lokendra, and Ajjanagadde. 1993. “From Simple Associations to Systematic Reasoning.” *Behavioral and Brain Sciences*, 16, 417–494.
- Shieber, Stuart. 1985. “Evidence against the Context-Freeness of Natural Language.” *Linguistics and Philosophy*, 8, 333–343.
- Siskind, Jeffrey. 1996. “A Computational Study of Cross-Situational Techniques for Learning Word-to-Meaning Mappings.” *Cognition*, 61, 39–91.
- Sommer, Friedrich T., and Günther Palm. 1998. “Bidirectional Retrieval from Associative Memory.” In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, eds., *Advances in Neural Information Processing Systems*, vol. 10. The MIT Press.
- Steedman, Mark. 1985. “Dependency and Coordination in the Grammar of Dutch and English.” *Language*, 61, 523–568.
- Steedman, Mark. 1993. “Categorial Grammar.” *Lingua*, 90, 221–258.
- Steedman, Mark. 1996. *Surface Structure and Interpretation*. Cambridge, MA: MIT Press.
- Steedman, Mark. 1997. “Temporality.” In Johan van Benthem and Alice ter Meulen, eds., *Handbook of Logic and Language*, 895–938. Amsterdam: North Holland/Elsevier.
- Steedman, Mark. 2000a. “Information Structure and the Syntax-Phonology Interface.” *Linguistic Inquiry*, 34, 649–689.
- Steedman, Mark. 2000b. *The Syntactic Process*. Cambridge, MA: MIT Press.
- Steedman, Mark. 2002. “Plans, Affordances, and Combinatory Grammar.” *Linguistics and Philosophy*, 25, 723–753.
- Steedman, Mark. 2006. “Surface Compositional Semantics of Intonation.” *Submitted*.

- Steinbuch, K. 1961. “Die Lernmatrix.” *Kybernetik*, 1, 36–45.
- Thielscher, Michael. 1999. “From Situation Calculus to Fluent Calculus: State Update Axioms as a Solution to the Inferential Frame Problem.” *Artificial Intelligence*, 111, 277–299.
- Tomasello, Michael. 1999. *The Cultural Origins of Human Cognition*. Cambridge, MA: Harvard University Press.
- Triesman, Anne. 1982. “Perceptual Grouping and Attention in Visual Search for Features and for Objects.” *Journal of Experimental Psychology: Human Perception and Performance*, 8, 194–214.
- Vaina, Lucia, ed. 1991. *From Retina to Neocortex: Selected Papers of David Marr*. Boston, MA: Birkhauser.
- Villavicencio, Aline. 2002. *The Acquisition of a Unification-Based Generalised Categorial Grammar*. Ph.D. thesis, University of Cambridge.
- Willshaw, David. 1981. “Holography, Association and Induction.” In Geoffrey Hinton and James Anderson, eds., *Parallel Models of Associative Memory*, 83–104. Hillsdale, NJ: Erlbaum.
- Willshaw, David, Peter Buneman, and Christopher Longuet-Higgins. 1969. “Non-Holographic Associative Memory.” *Nature*, 222, 960–962.
- Young, Robert, and William Morgan. 1987. *The Navaho Language*. Albuquerque, NM: University of New Mexico Press.
- Zettlemoyer, Luke, and Michael Collins. 2005. “Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars.” In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. Held in conjunction with IJCAI 2005, Edinburgh.



## Planning Dialog Actions

### Annex B

**Mark Steedman and Ron Petrick**

School of Informatics

University of Edinburgh

2 Buccleuch Place, Edinburgh EH8 9LW

steedman/rpetrick@inf.ed.ac.uk

### Abstract

The problem of planning dialog moves can be viewed as an instance of a more general AI problem of planning with sensing actions. Planning with sensing actions is complicated by the fact that such actions engender potentially infinite state-spaces. We adapt the PKS planner and the Linear Dynamic Event Calculus (LDEC) to the representation of dialog acts, and show beneficial consequences for planning mixed-initiative collaborative discourse.

### 1 Introduction

Successful planning in dynamic domains often requires reasoning about sensing acts, which when executed, update the planner's knowledge state, without necessarily changing the world state. For instance, reading a piece of paper with a telephone number printed on it may provide the reader with the prerequisite information needed to successfully complete a phone call. Such actions typically have very large, even infinite, sets of possible outcomes in terms of the actual valuefound, and threaten to make search impracticable. There have been several suggestions in the AI literature for how to handle this problem, including Moore 1985; Morgenstern 1988; Etzioni et al. 1992; Stone 1998; and Petrick & Bacchus 2002, 2004.

Stone (2000) points out that the problem of planning effective conversational moves is also a problem of planning with sensing or knowledge-producing actions, a view that is also implicit in

early "beliefs, desires and intentions" (BDI) -based approaches (e.g. Litman & Allen 1987; Bratman et al. 1988; Cohen & Levesque 1990; Grosz & Sidner 1990). Nevertheless, most work on dialog planning has in practice tended to segregate domain planning and discourse planning, treating the former as an AI black box, and capturing the latter in large state-transition machines mediated or controlled via a blackboard or "information state" representing mutual belief, updated by specialized rules more or less directly embodying some form of speech-act theory or theory of textual coherence (e.g. Lambert & Carberry 1991; Traum & Allen 1992; Green & Carberry 1994; Young & Moore 1994; Chu-Carroll & Carberry 1995; Matheson et al. 2000; Asher & Lascarides 2003). Such accounts lend themselves readily to optimization using statistical models (e.g. Singh et al. 2002).

One of the ostensible reasons for making this separation is that *indirect* speech acts, achieving coherence via conversational implicatures, abound in conversation. (Green and Carberry cite studies showing around 13% of answers to Yes/No questions to be indirect.) Moreover, not all implicatures are conversational, and the ontology of the ones that are is complex.

Nevertheless, that very same ubiquity of the phenomenon suggests that it is a manifestation of the same planning apparatus as the domain planner, and that it should not be necessary to construct a completely separate specialized planner to handle dialog acts. This paper considers the problem of applying techniques developed in the AI planning literature for handling sensing and incomplete information to

the problem of dialog planning.

## 2 Introduction to PKS/LDEC

In this paper we work with a particular planning system, called PKS, and a formal axiomatization of planning domains in the language of the Linear Dynamic Event Calculus (LDEC).

PKS (Planning with Knowledge and Sensing) is a “knowledge-level” planner that is able to construct conditional plans in the presence of incomplete knowledge and sensing Petrick & Bacchus (2002, 2004). The key idea of this approach is to represent the planner’s knowledge using a first-order language, and to represent actions by their effects on the agent’s knowledge state, rather than their effects on the world state. Since general reasoning in such a rich language is impractical, PKS employs a restricted subset of a first-order language and a limited amount of inference in that subset. By doing so, PKS is able to make use of non-propositional features, such as functions and variables.

The knowledge-level approach to planning differs from those approaches that concentrate on propositional representations over which complete reasoning is feasible. Such works often focus on representing the set of all possible worlds (i.e., the set of all states compatible with the planner’s incomplete knowledge) using a variety of techniques (e.g., BDDs, Graphplan-like structures, or clausal representations). By representing problems at the knowledge level, PKS can often “abstract” away from the irrelevant distinctions that occur at the world level. Furthermore, the resulting plans are often quite “natural” and have a simple structure. Compared to the possible-worlds approaches, PKS’s higher-level representation is richer, but the inferences it supports are weaker. Nevertheless, PKS is able to solve problems that are often difficult for world-level planners.

PKS is based on a generalization of STRIPS Fikes & Nilsson (1971). In STRIPS, the world state is modelled by a single database; actions update this database and, by doing so, update the planner’s model of the world. In PKS, the planner’s knowledge state (rather than the world state) is represented by a set of four databases, the contents of which have a fixed, formal interpretation into a modal logic of knowledge. Given any configuration of the

databases, this interpretation precisely characterizes the planner’s knowledge state. To ensure an efficient inference mechanism, we restrict the types of knowledge (especially disjunctive knowledge) that can be contained in each of the databases.

$K_f$ : The first database is like a standard STRIPS database except that both positive and negative facts are allowed and the closed world assumption is not applied.  $K_f$  can include any ground literal,  $\ell$ , where  $\ell \in K_f$  means that  $\ell$  is known.  $K_f$  can also contain knowledge of function values.

$K_v$ : The second database stores information about function values that will become known to PKS at execution time.  $K_v$  can contain any unnested function term; such terms model the plan-time effects of sensing actions that return numeric values. PKS can use  $K_v$  knowledge of finite-range functions to insert multi-way branches into a plan, or use function terms as a form of “run-time variable”.

$K_w$ : The third database models the plan-time effects of “binary” sensing actions.  $\phi \in K_w$  means that at plan time the planner either knows  $\phi$  or knows  $\neg\phi$ , and that at execution time this disjunction will be resolved. PKS uses such “know-whether” facts to construct conditional branches in a plan.

$K_x$ : The fourth database contains “exclusive-or” knowledge of literals. Entries in  $K_x$  have the form  $(\ell_1|\ell_2|\dots|\ell_n)$ , where each  $\ell_i$  is a ground literal. Such a formula represents knowledge of the fact that “exactly one of the  $\ell_i$  is true.” Such knowledge is common in many planning scenarios.

Actions in PKS are modelled by sets of queries and updates to the databases. Action preconditions are specified as a list of primitive queries that invoke an inference algorithm to answer simple questions about the databases: (i)  $Kp$ , is  $p$  known to be true?, (ii)  $K_v t$ , is the value of  $t$  known?, (iii)  $K_w p$ , is  $p$  known to be true or known to be false (i.e., does the planner know-whether  $p$ ?), or (iv) the negation of queries (i)–(iii).

Action effects are described by a set of updates to the databases, i.e., collections of formulae that should be added to and deleted from the databases. Since updates are made directly to the databases, they reflect changes to the planner’s knowledge state, rather than changes to the world state.

Using this representation, PKS can construct plans by applying actions in a simple forward-

chaining manner: provided the preconditions of an action are satisfied by the planner’s knowledge state, an action can be applied; applying an action updates the planner’s knowledge state to form a new knowledge state, allowing planning to continue. A conditional branch can be added to a plan if the planner has  $K_w$  information about a formula  $p$ . Along one branch,  $p$  is assumed to be known while along the other branch  $\neg p$  is assumed to be known. The planning process then continues along each branch until each branch satisfied the goal.

The Linear Dynamic Event Calculus (LDEC) Steedman (1997, 2002) is a logical formalism that combines the insights of the Event Calculus of Kowalski & Sergot (1986), itself a descendant of the Situation Calculus McCarthy & Hayes (1969), and the STRIPS planner of Fikes & Nilsson (1971), together with the Dynamic and Linear Logics developed by Girard (1987), Harel (1984), and others.

The particular dynamic logic that we work with here exclusively uses the deterministic “necessity” modality  $[\alpha]$ . For instance, if a program  $\alpha$  computes a function  $f$  over the integers, then an expression like “ $n \geq 0 \Rightarrow [\alpha](y = f(n))$ ” indicates that “in any situation in which  $n \geq 0$ , after every execution of  $\alpha$  that terminates,  $y = f(n)$ .” We can think of this modality as defining a logic whose models are Kripke diagrams. Accessibility between situations is represented by events defined in terms of the conditions which must hold before an event can occur (e.g., “ $n \geq 0$ ”), and the consequences of the event that hold as a result (e.g., “ $y = f(n)$ ”).

This logic also defines the *sequence* operator “;” as a composition operation over events. Like other dynamic logics, LDEC does not use explicit situation terms to denote the state-dependent values of domain properties. Instead, it uses the sequence operator to chain together finite sequences of actions. For instance,  $[\alpha_1, \alpha_2, \dots, \alpha_n]$  denotes a sequence of  $n$  actions and  $[\alpha_1, \alpha_2, \dots, \alpha_n]\phi$  indicates that  $\phi$  must necessarily hold after every execution of this action sequence.

LDEC also mixes two forms of logical implication, which contributes to its representational power. Besides standard (or intuitionistic) implication  $\Rightarrow$ , LDEC follows Bibel et al. (1989) and others in using *linear* logical implication, denoted by the symbol  $\multimap$ . This second form of implication provides

a solution to the *frame problem* McCarthy & Hayes (1969).

An LDEC *domain* is formally described by a collection of axioms. *Actions* (or *events*) provide the sole means of change in the world, and affect the *fluents* (i.e., properties) of the domain. For each action  $\alpha$ , an LDEC domain includes an *action precondition axiom* of the form:

$$L_1 \wedge L_2 \wedge \dots \wedge L_k \Rightarrow \text{affords}(\alpha),$$

where each  $L_i$  is a fluent or its negation, and an *effect axiom* of the form:

$$\{\text{affords}(\alpha)\} \wedge \phi \multimap [\alpha]\psi,$$

where  $\phi$  and  $\psi$  are conjunctions of fluents. An LDEC domain can also include a collection of *initial situation axioms* of the form:

$$L_1 \wedge L_2 \wedge \dots \wedge L_p,$$

where each  $L_i$  is a ground fluent literal.

Action precondition axioms specify the conditions that *afford* a particular action. Effect axioms use linear implication to build certain “update rules” directly into the LDEC representation. In particular, when an effect axiom is applied, the fluents in the antecedent (i.e.,  $\phi$ ) are treated as consumable resources that are “replaced” by the fluents in the consequent (i.e.,  $\psi$ ). (We treat consumed fluents as being made false.) A formula contained in  $\{\cdot\}$  indicates that it is a non-consumable resource. All other fluents remain unchanged. Thus, the LDEC treatment of action is very similar to STRIPS; in particular, LDEC’s use of linear implication is similar to the STRIPS assumption, and lets us avoid having to include explicit frame axioms in our LDEC domains.

Recent work Petrick & Steedman (2007) has established a preliminary link between PKS and LDEC, in particular for the representation of simple sensing actions. We do not go into detail about this work here, however, we adapt this approach so that we can include PKS-style queries directly in our LDEC axioms, as a form of knowledge fluent. Moreover, we extend these fluents to include speaker-hearer modalities. Thus, we can write LDEC axioms that include fluent expressions like  $[X]Kp$  (“X knows  $p$ ”),  $[X]K_{v,t}$  (“X knows the

value of  $t$ ), or  $[X]K_w p$  (“X knows-whether  $p$ ). We can also nest such modal expressions to form more complex representations of multi-agent knowledge, e.g.,  $[X]K\neg[Y]Kp$  (“X knows that Y does not know  $p$ ”).

We will also assume that our LDEC domains include the following standard axioms of knowledge:

- (1)  $[X]Kp \Rightarrow p$  Veridicality
- (2)  $\neg[X]p \Rightarrow [X]K\neg[X]p$  Negative Introspection

### 3 Planning Speech Acts with PKS/LDEC

#### 3.1 Facts

- (3) a. “I suppose Bonnie doesn’t know what train I will catch”
- b.  $[S]K\neg[B]K_v time$
- (4) a. “If I know what time it is, I know what train I will catch.”
- b.  $[S]K_v time \Rightarrow [S]K_v train$
- (5) a. “I don’t know what time it is.”
- b.  $\neg[S]K_v time$
- (6) a. “I suppose you know what time it is.”
- b.  $[S]K[H]K_v time$

#### 3.2 Rules

- (7) a. “If X supposes  $p$ , and X supposes  $p$  is not common ground, X can tell Y  $p$ ”
- b.  $[X]p \wedge [X]K\neg[C]p \Rightarrow affords(tell(X, Y, p))$
- (8) a. “If X tells Y  $p$ , Y stops not knowing it and starts to know it.”
- b.  $affords(tell(X, Y, p)) \wedge \neg[Y]p$   
 $\quad \circ [tell(X, Y, p)][Y]p$
- (9) a. “If X doesn’t know  $p$  and X supposes Y does, X can ask Y about it.”
- b.  $\neg[X]p \wedge [X]K[Y]p \Rightarrow affords(ask(X, Y, p))$
- (10) a. “If X asks Y about  $p$ , it makes it common ground X doesn’t know it”
- b.  $affords(ask(X, Y, p))$   
 $\quad \circ [ask(X, Y, p)][C]K\neg[X]p$

#### 3.3 Planning a Direct Speech Act

- (11) Goal: I need Bonnie to know which train I will catch

Lemma: By speaker supposition, the hearer

knows what time it is:

$$(12) \Rightarrow [H]K_v time \quad (6b); (1)$$

Lemma: By speaker supposition, Bonnie doesn’t know what train the speaker will catch:

$$(13) \Rightarrow \neg[B]K_v train \quad (3b); (K_s p \Rightarrow p)$$

The situation affords  $ask(S, H, time)$ :

$$(14) \Rightarrow affords(ask(S, H, time)) \quad (5b); (6b); (9b)$$

After applying  $ask(S, H, time)$ :

$$(15) \Rightarrow [H]K\neg[S]K_v time \quad (14); (10b)$$

The situation now affords  $tell(H, S, time)$ :

$$(16) \Rightarrow affords(tell(H, S, time)) \quad (12); (15); (7b)$$

After applying  $tell(H, S, time)$ :

$$(17) \Rightarrow [S]K_v time \quad (16); (5b); (8b)$$

—which means I know what train I will catch:

$$(18) \Rightarrow [S]K_v train \quad (17); (4b)$$

The situation now affords  $tell(S, B, train)$

$$(19) \Rightarrow affords(tell(S, B, train)) \quad (18); (3b); (7b)$$

After applying  $tell(S, B, train)$

$$(20) \Rightarrow [B]K_v train \quad (19); (13); (8b)$$

#### 3.4 Planning an Indirect Speech Act

The situation in section 3.1 also affords  $tell(S, H, \neg[S]K_v time)$ , telling the hearer that I don’t know the time:

$$(21) \Rightarrow [S]K\neg[C]K\neg[S]K_v time \quad (2)$$

$$(22) \Rightarrow [S]K\neg[S]K_v time \quad (5b); (2)$$

$$(23) \Rightarrow affords(tell(S, H, \neg[S]K_v time))(22); (21); (7b)$$

After applying  $tell(S, H, \neg[S]K_v time)$ —that is, saying “I don’t know what time it is”:

$$(24) \Rightarrow [C]K\neg[S]K_v time \quad (10b)$$

Since (24) is identical to (15), the situation now again affords  $tell(H, S, time)$ , and the rest of the plan is as before.

Asking the time by saying “I don’t know what time it is” would usually be regarded as an indirect speech act. However, under the present account, both “direct” and “indirect” speech acts have their effects by changing the same set of facts about the

knowledge states of the participants. Both involve inference. In some sense, there is no such thing as a “direct” speech act. In that sense, it is not surprising that indirect speech acts are so widespread: *all* speech acts are indirect in the sense of involving inference.

Crucially, at no point does the plan depend upon the hearer identifying the fact that the speakers utterance “I don’t know what time it is” had the illocutionary force of a request or question such as “What time is it?”.

### 3.5 On So-called Conversational Implicature

The fact that we distinguish speaker suppositions about common ground from the hearer suppositions themselves means that we can include the following rules parallel to (7) and (8) without inconsistency:

- (25) a. “If X supposes the value of p is common ground, X can say to Y that the value of Y is something else”
- b.  $[X]K[C]KF = V \wedge V \neq W$   
 $\Rightarrow affords(say(X, Y, F = W))$
- (26) a. “If X says to Y a value of f, and Y supposes a different value of f, then Y continues to suppose that value, and supposes that it is not common ground.”
- b.  $affords(say(X, Y, F = W)) \wedge [Y]KF = V \wedge V \neq W$   
 $\nego[say(X, Y, F = W)]$   
 $[Y]F = V \wedge [Y]K\neg[C]KF = V$

Speakers’ calculations about what will follow from making claims about hearers’ knowledge states extend to what will follow from making *false* claims of this kind. To take a famous example from Grice, suppose that we both know that you have done me an unfriendly turn:

- (27) a. “I know that as a friend you are bad”
- b.  $[S]Kfriend(h) = bad$
- (28) a. “You know that as a friend you are bad”
- b.  $[H]Kfriend(h) = bad$

After applying  $tell(S, H, friend(h) = good)$ , say by uttering the following:

(29) You’re a fine friend!

the following holds:

$$(30) \Rightarrow [H]Kfriend(h) = bad$$

$$\wedge [H]K\neg[C]friend(h) = bad$$

$$(27); (28); (26b)$$

One might not think that getting the hearer to infer something they already know is very useful. However, if we assume a mechanism of attention, whereby things that are inferred become salient, then we have drawn their attention to their trespass. Moreover, the information state that we have brought them to is one that would normally suggest, via rules like (7) and (8), that the hearer should *correct* the original speaker. Of course, further reflection (via similar rules that we will pass over here) is likely to make the hearer unwilling to do so, leaving them few conversational gambits other than to slink silently away. This of course is what the original speaker really intended.

### 3.6 A Prediction of the Theory

This theory explains, as Grice did not, why this trope is asymmetrical: the following is predicted to be an ineffectual way of indirectly complementing a friend on a friendly act:

- (31) #You’re a lousy friend!

Making a hearer think of the key fact for themselves does not constitute a complement at all, and this time there is no reason for them not to respond to the contradiction. Unlike (29), this utterance is therefore likely to evoke a vociferous correction to the common ground, rather than smug acquiescence to the contrary, parallel to the sheepish response evoked by (29).

## 4 Discussion

The above are toy examples: scaling to realistic domains will raise the usual problems of knowledge representation that AI is heir to. However, the update effects (and side-effects) of the discourse planner are general-purpose. They are entirely driven by the knowledge state, without recourse to specifically conversational rules, other than some very general rules of consistency maintenance in common ground. Rhetorical relations such as explanation, elaboration, and causation-to-believe, are emergent from these general rules. There is therefore some hope that that conversational planning itself is of low complexity, and that any domain that we can actu-

ally plan in, we can also plan conversations about.

## References

- Asher, N. & Lascarides, A. (2003). *Logics of Conversation*. Cambridge: Cambridge University Press.
- Bibel, W., del Cerro, L. F., Fronhfer, B., & Herzog, A. (1989). Plan generation by linear proofs: on semantics. In *German Workshop on Artificial Intelligence - GWAI'89*, vol. 216 of *Informatik-Fachberichte*, Berlin. Springer Verlag.
- Bratman, M., Israel, D., & Pollack, M. (1988). Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4, 349–355.
- Chu-Carroll, J. & Carberry, S. (1995). Response generation in collaborative negotiation. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 136–143. ACL.
- Cohen, P. & Levesque, H. (1990). Rational interaction as the basis for communication. In P. Cohen, J. Morgan, & M. Pollack (eds.), *Intentions in Communication*, pp. 221–255. Cambridge, MA: MIT Press.
- Etzioni, O., Hanks, S., Weld, D., Draper, D., Lesh, N., & Williamson, M. (1992). An approach to planning with incomplete information. In *Proceedings of the 2nd International Conference on Knowledge Representation and Reasoning (KRR-2)*, pp. 115–125.
- Fikes, R. & Nilsson, N. (1971). Strips: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2, 189–208.
- Girard, J.-Y. (1987). Linear logic. *Theoretical Computer Science*, 50, 1–102.
- Green, N. & Carberry, S. (1994). A hybrid reasoning model for indirect answers. In *Proceedings of the 2nd*, pp. 58–65. ACL.
- Grosz, B. & Sidner, C. (1990). Plans for discourse. In P. Cohen, J. Morgan, & M. Pollack (eds.), *Intentions in Communication*, pp. 417–444. Cambridge, MA: MIT Press.
- Harel, D. (1984). Dynamic logic. In D. Gabbay & F. Guenther (eds.), *Handbook of Philosophical Logic*, vol. II, pp. 497–604. Dordrecht: Reidel.
- Kowalski, R. & Sergot, M. (1986). A logic-based calculus of events. *New Generation Computing*, 4, 67–95.
- Lambert, L. & Carberry, S. (1991). A tripartite plan-based model of dialogue. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pp. 47–54. ACL.
- Litman, D. & Allen, J. (1987). A plan recognition model for subdialogues in conversation. *Cognitive Science*, 11, 163–200.
- Matheson, C., Poesio, M., & Traum, D. (2000). Modeling grounding and discourse obligations using update rules. In *Proceedings of NAACL 2000*, Seattle, San Francisco, CA. Morgan Kaufmann.
- McCarthy, J. & Hayes, P. (1969). Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer & D. Michie (eds.), *Machine Intelligence*, vol. 4, pp. 473–502. Edinburgh: Edinburgh University Press.
- Moore, R. (1985). A formal theory of knowledge and action. In J. Hobbs & R. Moore (eds.), *Formal Theories of the Commonsense World*, pp. 319–358. Norwood, NJ: Ablex. Reprinted as Ch. 3 of Moore (1995).
- Moore, R. (1995). *Logic and Representation*, vol. 39 of *CSLI Lecture Notes*. Stanford CA: CSLI/Cambridge University Press.
- Morgenstern, L. (1988). Foundations of a logic of knowledge, action, and communication. Ph.D. thesis, NYU, Courant Institute of Mathematical Sciences.
- Petrick, R. P. A. & Bacchus, F. (2002). A knowledge-based approach to planning with incomplete information and sensing. In M. Ghallab, J. Hertzberg, & P. Traverso (eds.), *Proceedings of the Sixth International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2002)*, pp. 212–221, Menlo Park, CA. AAAI Press.
- Petrick, R. P. A. & Bacchus, F. (2004). Extending the knowledge-based approach to planning with

- incomplete information and sensing. In S. Zilberstein, J. Koehler, & S. Koenig (eds.), *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS-04)*, pp. 2–11, Menlo Park, CA. AAAI Press.
- Petrick, R. P. A. & Steedman, M. (2007). Representing knowledge and sensing in the linear dynamic event calculus for knowledge-level planning. In *Submitted to: Proceedings of the 22nd National Conference on Artificial Intelligence (AAAI-07)*, Menlo Park, CA. AAAI Press.
- Singh, S., Litman, D., Kearns, M., & Walker, M. (2002). Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *Journal of Artificial Intelligence Research*, 16, 105–133.
- Steedman, M. (1997). Temporality. In J. van Benthem & A. ter Meulen (eds.), *Handbook of Logic and Language*, pp. 895–938. Amsterdam: North Holland/Elsevier.
- Steedman, M. (2002). Plans, affordances, and combinatorial grammar. *Linguistics and Philosophy*, 25, 723–753.
- Stone, M. (1998). Abductive planning with sensing. In *Proceedings of the 15th National Conference on Artificial Intelligence*, pp. 631–636, Menlo Park CA. Madison WI, AAAI.
- Stone, M. (2000). Towards a computational account of knowledge, action and inference in instructions. *Journal of Language and Computation*, 1, 231–246.
- Traum, D. & Allen, J. (1992). A speech acts approach to grounding in conversation. In *Proceedings of the 2nd International Conference on Spoken Language Processing (ICSLP-92)*, pp. 137–140.
- Young, R. M. & Moore, J. D. (1994). DPOCL: a principled approach to discourse planning. In *Proceedings of the Seventh International Workshop on Natural Language Generation*, pp. 13–20.