# Plans and the Computational Structure of Language

Mark Steedman$^\star$ and Matthew Stone$^\dagger$

U. Edinburgh$^\star$ and Rutgers U.$^\dagger$

`http://www.inf.ed.ac.uk/~steedman`

`http://www.cs.rutgers.edu/~mdstone/`

> *Not only speech, but all skilled acts seem to involve the same problems of serial ordering, even down to the temporal coordination of muscular contractions in such a movement as reaching and grasping. Analysis of the nervous mechanisms underlying order in the more primitive acts may contribute ultimately to the solution of even the physiology of logic.*
>
> Karl Lashley 1951:122

**Edinburgh Computational Thinking Seminar, December 2005**

# Plans and the Structure of Language

- It’s rather odd that the dominant tradition in formal grammar has ignored the active, situation-changing, aspects of meaning in favour of truth conditions.
- Language as action:
  - I name this ship the *Nice Work If You Can Get It*.
  - Do you take this woman to be your lawful wedded wife? I do.
  - Everybody who has a face mask wears it. (*Economist*, 5 Apr 03, re SARS in Hong Kong)
- Language as Computation. All of the above utterances:
  - Access the current context (“this ship”; “take this woman to be your lawful wedded wife”; dependent “a face mask”);
  - Produce a value;
  - Update the context for subsequent computation.

# Is Natural Language Computational?

- There is abundant evidence from neurology and child development that that the language faculty is closely related evolutionarily and developmentally to planning actions in the world, particularly planning involving tools (Freud 1891; Piaget 1936; Lashley 1951).
- Computer Science and Artificial Intelligence offers interesting formalisms for planning and dynamic state-change.
- Natural language grammar exhibits some remarkable homologies to such planner formalisms
- Representing these homologies directly in the theory of language gives
  - A more explanatory theory of grammar
  - with efficient practical parsers
  - a simpler account of human language processing
  - and of child language acquisition

# Plans and the Structure of This Talk

- I: Thinking Computationally about Action
- II: How Animals and Humans Make Plans
- III: Thinking Computationally about Grammar
- IV: Thinking Computationally about Parsing
- V: Thinking Computationally about Language Development
- VI: Conclusions: for a Cognitive Informatics of Language

# I: Thinking Computationally about Action

- Basic Dynamic Logic:

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

  "If $n$ is positive, $\alpha$-ing always sets $y$ equal to $F(n)$".

- In the real world, such rules are *defaults*, but they are still *deterministic*.

- The particular dynamic logic that we are dealing with here is one that includes the following dynamic axiom (the operator ; is *sequence*, the composition of functions of type *situation* → *situation*):

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

- 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

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

# Dynamic Logic: Actions as Accessibility

- The actions $\alpha, \beta, \ldots$ can be seen as defining the accessibility relation for a modal logic with an S4 model:

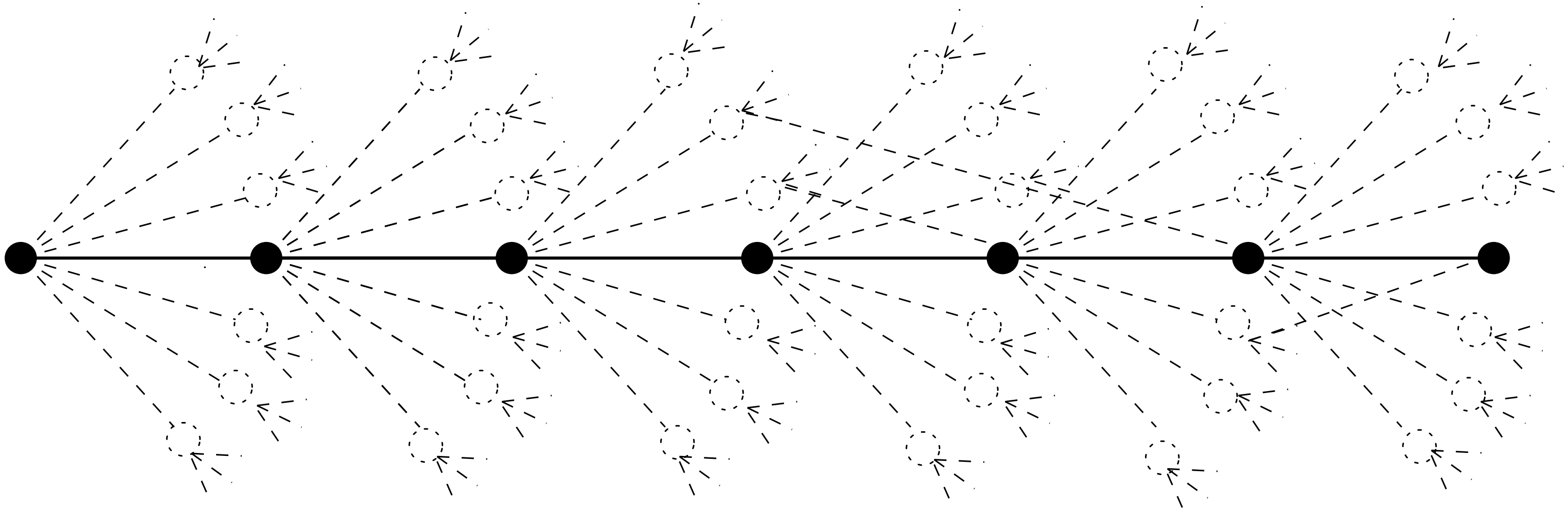


Figure 1: Kripke Model of Causal Accessibility Relation

# Situation/Event Calculi and the Frame Problem

- The Situation Calculus (McCarthy and Hayes 1970) and its descendants can be seen as versions of Dynamic Logic.
- These calculi are heir to the “Frame Problem,” which arises from the fact that humans conceptualize events in terms of very localized changes to situations.
- For example, the effects of an event of *My eating a hamburger* are confined to the hamburger and aspects of myself like hunger. The color of the walls, the day of the week, the leadership of the Conservative and Unionist party, and countless other aspects of the situation remain unchanged.
- This character of the knowledge representation raises the Frame Problem in two forms: the “Representational” and “Inferential” versions.

# The Representational Frame Problem

- Since change is local, it is cumbersome to explicitly represent the input effect of each event on each fact by innumerable rules such as

  (4) $color(wall,x) \Rightarrow [eat(hamburger)]color(wall,x)$

- Kowalski (1979) solved the representational problem using reified Frame Axioms Equivalent in the present notation to the following:

  (5) $p \wedge (p \neq hungry) \wedge (p \neq here(hamburger)) \Rightarrow [eat(hamburger)]p$

- This keeps rules defining the positive effects of eating hamburgers simple. (Note that $p$ is "overloaded," standing for both the fact that $p$ *holds* and for the term $p$ as an individual, as is standard in logic programming.)

- But if we ever need to know what the color of the walls is after a sequence of, say, five hamburger eating events, then we have to do costly theorem-proving search. This is the *Inferential* form of the Frame Problem.

# STRIPS and the Inferential Frame Problem

- The STRIPS program (Fikes and Nilsson 1971) solved both representational and inferential problems by representing change as sets of *preconditions* and localized *database updates*, as in the following definition of the operator *eat*:

|  |  |
|---|---|
| PRECONDITIONS: | $hamburger(x)$ |
|  | $here(x)$ |
|  | $hungry$ |
| DELETIONS: | $here(x)$ |
|  | $hungry$ |
| ADDITIONS: | $thirsty$ |

- Such representations were initially derided by logicians (because of their nonmonotonicity) …

- … but then Girard (1995) came along with Linear Logic, and update was logically respectable after all!

# The Linear Dynamic Event Calculus (LDEC)

- We can represent events involving boxes in this notation.
- The preconditions of putting something on something else can be defined as follows using standard implication and an *affords* predicate:

(6) $box(x) \wedge box(y) \wedge \neg on(z,x) \wedge \neg on(w,y) \wedge (x \neq y) \Rightarrow affords(puton(x,y))$

- A situation *affords* an action (in the sense of Gibson 1966 discussed below) if it satisfies its preconditions.
- To define the update consequences of putting something *on* something else in a situtaion that affords that action we need a different, linear implication $\multimap$ :

(7) $\{affords(puton(x,y))\} \wedge on(x,z) \multimap [puton(x,y)]on(x,y)$

- Linear implication, $\multimap$ , treats positive ground literals or “facts” in the antecedent as consumable resources, removing them from database and replacing them by the consequent.

# STRIPS updates as Linear Implication (Contd.)

- The braces in marks $\{affords(puton(x,y))\}$ mark the affordance as a nonconsumable precondition: the truth of this condition after a *puton* event is not defined by the linear implication, and is a matter for further inference, via rules like (6).
- It is related to Girards ! exponential (“Of course!”).
- Thus we use the $\{affords(\ldots)\}$ notation to “fibre” the intuitionistic and linear components of the logic.

# STRIPS Planning in LDEC

- The transitivity axiom of the affordance relation is defined as follows:

  (8) $affords(\alpha) \wedge [\alpha]affords(\beta) \Rightarrow affords(\alpha;\beta)$

- Consider the following initial situation:

  (9) $block(a) \wedge block(b) \wedge block(c) \wedge on(a,table) \wedge on(b,table) \wedge on(c,table)$

- The following conjunctive goal (10), given a search control, can be made to deliver a constructive proof that (11) is one such plan:

  (10) $goal(affords(\alpha) \wedge [\alpha](on(a,b) \wedge on(b,c)))$

  (11) $\alpha = puton(b,c);puton(a,b)$

- The result of executing this plan in situation (9) is that the following conjunction of facts is directly represented by the database:

  (12) $block(a) \wedge block(b) \wedge block(c) \wedge on(a,b) \wedge on(b,c) \wedge on(c,table)$

# LDEC Avoids a Ramification Problem

- If durative events like the agent *moving* are represented as instantaneous transitions to and from a progressive state represented as a fluent $in\_progress(move(me, there))$, LDEC is well behaved with respect to standard examples of the ramification problem such as the one that arises from moving through a paint-spray.

- In event calculi in which intervals are primitive, it is hard to specify frame axioms that capture the common-sense knowledge that if you move, your color is unaffected, and if someone sprays you with paint your color is affected, and that if you move through a paint-spray, your color is affected.

- Because in LDEC durative events are represented in terms of initiating and terminating instants and intervening states, such knowledge is easy to represent. Suppose the situation is $at(me, here) \wedge color(me, green)$:

# LDEC Avoids a Ramification Problem

- Axioms for events of spraying someone some color:

(13) $affords(start(spray(y,c)))$

(14) $\{affords(start(spray(y,c)))\} \wedge color(x)$
$\multimap [start(spray(y,c))]in\_progress(spray(y,c))$

(15) $in\_progress(spray(y,c)) \Rightarrow affords(stop(spray(y,c)))$

(16) $\{affords(stop(spray(y,c)))\} \wedge in\_progress(spray(y,c)))$
$\multimap [stop(spray(y,c))]color(y,c)$

# LDEC Avoids a Ramification Problem

- For a situation in which $at(me, here) \wedge color(me, green)$, we correctly prove the following without encountering inconsistency:

(17) $[start(move(me, there)); start(spray(me, pink));$
$stop(spray(me, pink)); stop(move(me, there))]color(me, pink)$

(18) $[start(spray(me, pink)); start(move(me, there));$
$stop(move(me, there)); stop(spray(me, pink))]at(me, there)$

(19) $[start(spray(me, pink)); start(move(me, there));$
$stop(spray(me, pink)); stop(move(me, there))]color(me, pink)$

(20) $[start(move(me, there)); start(spray(me, pink));$
$stop(move(me, there)); stop(spray(me, pink))]at(me, there)$

# II: How Animals and Humans Make Plans

- Some animals can make plans of this kind, involving tools (Köhler 1925).

Figure 2: From Köhler 1925

Figure 3: From Köhler 1925

# How Animals and Humans Make Plans (contd.)

- Such search seems to be *reactive* to the presence of the tool and *forward-chaining*, rather than backward-chaining (working from goal to tool). That is, the animal can make a plan in the presence of the tool, but has difficulty with plans that require subgoals of finding tools.
- It implies that actions are accessed via perception of the objects that mediate them—in other words that actions are represented as the *affordances* of objects, in Gibson's (1966) terms.
- This seems a good way for an animal to plan. If there *is* a short plan using available resources, forward chaining will find it.
- Backward chaining requires the evolution of tools with very general affordances, like credit cards and mobile phones.

# Formalizing Affordance in LDEC

- We can define the affordances of objects directly in terms of LDEC preconditions like (6)
- Thus the affordances of doors are *pushing* and *going through*:

(21) $$\mathit{affordances}(\mathit{door}) = \left\{ \begin{array}{l} \mathit{push} \\ \mathit{go\text{-}through} \end{array} \right\}$$

- This provides the basis for Reactive, Affordance-based, Forward-Chaining plan construction that is characteristic of primate planning.

# Formalizing Affordance in LDEC (Contd.)

- The Gibsonian affordance-based door-schema can then in turn be defined as a function mapping doors into (second-order) functions from their affordances like pushing and going-through to their results:

(22) $door' = \lambda x_{door}.\lambda p_{affordances(door)}.px$

- The operation of turning an object of a given type into a function over those functions that apply to objects of that type is another primitive combinator called **T** or *type raising*, so (22) can be rewritten $door' = \lambda x_{door}.\mathbf{T}x$, where

(23) $\mathbf{T}a \equiv \lambda p.p(a)$

# LDEC and Human Cognition

- The dynamic axioms of LDEC can be viewed as a representation of Miller et al's **TOTE units**, Piaget (1936)'s **Circular Reactions** , or of the Behaviorists' notion of **operant**.
- The "Test-Operate/Test-Exit" loop of TOTE units is necessary for the execution of the plan in the world, and is also represented in the dynamic logic.
- For example the following LDEC rules represent what a 1-4 month infant has learned about the breast (simplifying somewhat). First, a breast "affords" sucking, in Gibson's sense, where $\Rightarrow$ is standard implication:

  (24) $\mathit{breast} \Rightarrow \mathit{affords}(\mathit{suck})$

  And the following rule represents the effects of sucking using Kleene $+$ iteration of a test and an elementary action:

  (25) $\{\mathit{affords}(\mathit{suck})\} \wedge \mathit{hungry} \multimap [(\mathit{hungry}?;\mathit{suck})^{+}]\neg \mathit{hungry}$

# Languages that Lexicalize Affordance

- Many North American Indian languages, such as the Athabascan group that includes Navaho, are comparatively poorly off for nouns. Many nouns for artefacts are morphological derivatives of verbs.
- For example, “towel” is *bee ’ádít’oodí*, glossed as “one wipes oneself with it”, and “towelrack” is *bee ’ádít’oodí bąąh dah náhidiiltsos*—roughly “one wipes oneself with it is repeatedly hung on it”.
- Such languages appear to lexicalize nouns as a default affordance.

☡ *Of course*, we should avoid crassly Whorfean inferences about Navaho-speakers abilities to reason about objects. Though productive, these lexicalizations are as conventional as our own.

- Navaho-speakers probably think English is totally weird in allowing denominal verbs, like “shelve” and “pocket” with equal productivity. We shall return to this question.

# III: Thinking Computationally about Grammar

- Categorial Grammar replaces PS rules by lexical categories and general combinatory rules (**Lexicalization**):

(26)
$$\xcancel{\begin{array}{lll} S & \rightarrow & NP \quad VP \\ VP & \rightarrow & TV \quad NP \\ TV & \rightarrow & \{proved,\ finds,\ \ldots\} \end{array}}$$

- Categories:

(27) proved := $(S\backslash NP)/NP$ : $prove'$

# Combinatory Categorial Grammar (CCG)

- Combinatory Rules:

$$\frac{X/_{\star}Y : f \quad Y : g}{X : f(g)} > \qquad \frac{Y : g \quad X\backslash_{\star}Y : f}{X : f(g)} <$$

$$\frac{X/_{\diamond}Y : f \quad Y/_{\diamond}Z : g}{X/_{\diamond}Z : \lambda z.f(g(z))} >\mathbf{B} \qquad \frac{Y\backslash_{\diamond}Z : g \quad X\backslash_{\diamond}Y : f}{X\backslash_{\diamond}Z : \lambda z.f(g(z))} <\mathbf{B}$$

$$\frac{X/_{\times}Y : f \quad Y\backslash_{\times}Z : g}{X\backslash_{\times}Z : \lambda z.f(g(z))} >\mathbf{B}_{\times} \qquad \frac{Y/_{\times}Z : g \quad X\backslash_{\times}Y : f}{X/_{\times}Z : \lambda z.f(g(z))} <\mathbf{B}_{\times}$$

- All arguments are type-raised via the lexicon:

$$\frac{X : x}{\mathrm{T}/(\mathrm{T}\backslash X) : \lambda f.f(x)} >\mathbf{T} \qquad \frac{X : x}{\mathrm{T}\backslash(\mathrm{T}/X) : \lambda f.f(x)} <\mathbf{T}$$

# Combinatory Derivation

(28)

$$\begin{array}{cccl}
\text{Marcel} & \text{proved} & \text{completeness} & \\
\cline{1-1}\cline{2-2}\cline{3-3}
NP : marcel' & (S\backslash NP)/NP : prove' & S\backslash(S/NP) : \lambda p.p\ completeness' & \\
\cline{1-1}
S/(S\backslash NP) : \lambda f.f\ marcel' & & & >\mathbf{T} \\
\cline{1-2}
\multicolumn{2}{c}{S/NP : \lambda x.prove'x\ marcel'} & & >\mathbf{B} \\
\cline{1-3}
\multicolumn{3}{c}{S : prove'completeness'marcel'} & < 
\end{array}$$

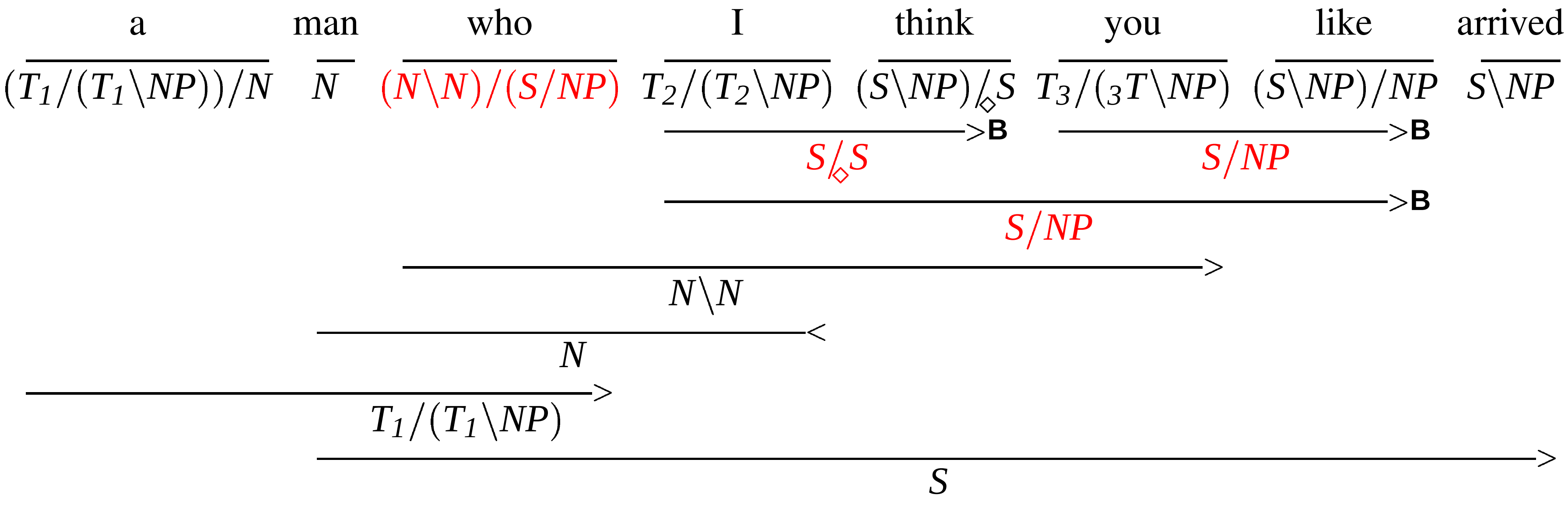
(29)

$$\begin{array}{cccl}
\text{Marcel} & \text{proved} & \text{completeness} & \\
\cline{1-1}\cline{2-2}\cline{3-3}
NP : marcel' & (S\backslash NP)/NP : prove' & (S\backslash NP)\backslash((S\backslash NP)/NP) & \\
 & & : \lambda p.p\ completeness' & \\
\cline{1-1}
S/(S\backslash NP) : \lambda f.f\ marcel' & & & >\mathbf{T} \\
\cline{2-3}
 & \multicolumn{2}{c}{S\backslash NP : \lambda y.prove'completeness'y} & < \\
\cline{1-3}
\multicolumn{3}{c}{S : prove'completeness'marcel'} & < 
\end{array}$$

# Linguistic Predictions: Unbounded "Movement"

- The combination of type-raising and composition allows derivation to project lexical function-argument relations onto "unbounded" constructions such as relative clauses and coordinate structures, without transformational rules:

(30)



a man who I think you like arrived

$(T_1/(T_1\backslash NP))/N$ $N$ $(N\backslash N)/(S/NP)$ $T_2/(T_2\backslash NP)$ $(S\backslash NP)/_{\diamond}S$ $T_3/({}_3T\backslash NP)$ $(S\backslash NP)/NP$ $S\backslash NP$

$\xrightarrow{>\mathbf{B}}$ $S/_{\diamond}S$

$\xrightarrow{>\mathbf{B}}$ $S/NP$

$\xrightarrow{>\mathbf{B}}$ $S/NP$

$\xrightarrow{>}$ $N\backslash N$

$\xleftarrow{<}$ $N$

$\xrightarrow{>}$ $T_1/(T_1\backslash NP)$

$\xrightarrow{>}$ $S$

# Predictions: Argument-Cluster Coordination

- The following construction is predicted on arguments of symmetry.

(31)

$$\begin{array}{ccccccccc}
\text{give} & \text{a teacher} & & \text{an apple} & & \text{and} & \text{a policeman} & & \text{a flower} \\
\cline{1-1}\cline{2-2}\cline{4-4}\cline{6-6}\cline{7-7}\cline{9-9}
(VP/NP)/NP & T_1\backslash(T_1/NP) & {<}\mathbf{T} & T_2\backslash(T_2/NP) & {<}\mathbf{T} & CONJ & T_3\backslash(T_3/NP) & {<}\mathbf{T} & T_4\backslash(T_4/NP)\ {<}\mathbf{T} \\
\cline{2-4}\cline{7-9}
 & \multicolumn{3}{c}{T_2\backslash((T_2/NP)/NP)} & {<}\mathbf{B} & & \multicolumn{3}{c}{T_4\backslash((T_4/NP)/NP)\ {<}\mathbf{B}} \\
\cline{2-9}
 & \multicolumn{8}{c}{T_6\backslash((T_6/NP)/NP)\ {<}\Phi{>}} \\
\cline{1-9}
\multicolumn{9}{c}{VP\ {<}}
\end{array}$$

- The derivation of utterance(31) is isomorphic to the process of composing a plan for another's action from the affordances of teachers, apples, (etc.), in a situation affording the plan by a speaker who desires its side-effects.
- The parallel involvement of type-raising **T** and composition **B** in planning and grammar suggest that the latter is evolutionarily and developmentally a transparent attachment to the former.

# These Things are Out There in the Treebank

- Full Object Relatives ( 570 in WSJ treebank)
- Reduced Object Relatives ( 1070 in WSJ treebank)
- Argument Cluster Coordination ( 230 in WSJ treebank):

```
(S (NP-SBJ It)
   (VP (MD could)
       (VP (VP (VB cost)
               (NP-1 taxpayers)
               (NP-2 $ 15 million))
           (CC and)
           (VP (NP=1 BPC residents)
               (NP=2 $ 1 million)))))
```

# These Things are Out There (contd.)

- Parasitic Gaps (at least 6 in WSJ treebank):

```
(S (NP-SBJ Hong Kong's uneasy relationship with China)
   (VP (MD will)
       (VP (VP (VB constrain)
               (NP (-NONE- *RNR*-1)))
           (PRN (: --)
                (IN though)
                (VP (RB not)
                    (VB inhibit)
                    (NP (-NONE- *RNR*-1)))
                (: --))
           (NP-1 long-term economic growth))))
```

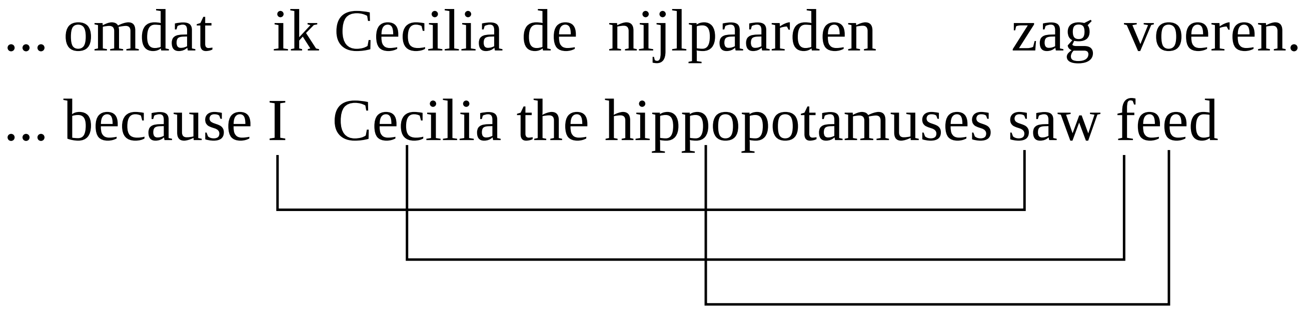
# CCG is Just Trans-Context Free

- CCG is provably weakly equivalent to Linear Indexed Grammar (LIG) Joshi et al. (1991).
- Hence it is not merely “Mildly Context Sensitive” (Joshi 1988) but rather just Trans-Context Free, or “Type $1.\dot{9}$” in the Extended Chomsky Hierarchy.

| **Language Type** | **Automaton** | **Rule-types** | | | **Exemplar** |
|---|---|---|---|---|---|
| Type 0: RE | Universal Turing Machine | $\alpha$ | $\rightarrow$ | $\beta$ | |
| Type 1: CS | Linear Bound Automaton (LBA) | $\phi A \psi$ | $\rightarrow$ | $\phi \alpha \psi$ | $a^{2^n}$ |
| “Type $1.9\dot{9}$: LI” | Embedded PDA (EPDA) | $A_{[(i),...]}$ | $\rightarrow$ | $\phi B_{[(i),...]} \psi$ | $a^n b^n c^n$ |
| Type 2: CF | Push-Down Automaton (PDA) | $A$ | $\rightarrow$ | $\alpha$ | $a^n b^n$ |
| Type 3: FS | Finite-state Automaton (FSA) | $A$ | $\rightarrow$ | $\begin{cases} \mathrm{a\ B} \\ \mathrm{a} \end{cases}$ | $a^n$ |

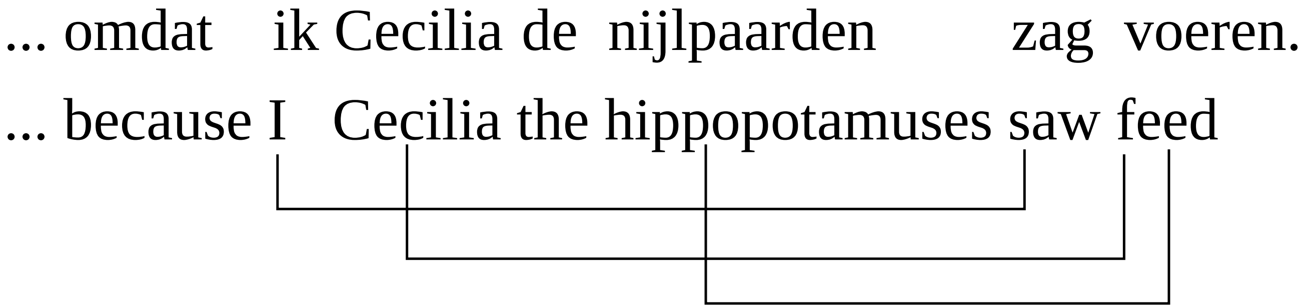
# A Trans-Context Free Natural Language

- CCG can capture unboundedly crossed dependencies in Dutch:



... omdat ik Cecilia de nijlpaarden zag voeren.
... because I Cecilia the hippopotamuses saw feed

'... because I saw Cecilia feed the hippopotamuses.'



... omdat ik Cecilia de nijlpaarden zag voeren.
... because I Cecilia the hippopotamuses saw feed

'... because I saw Cecilia feed the hippopotamuses.'

# CCG is Just Trans-Context Free (contd.)

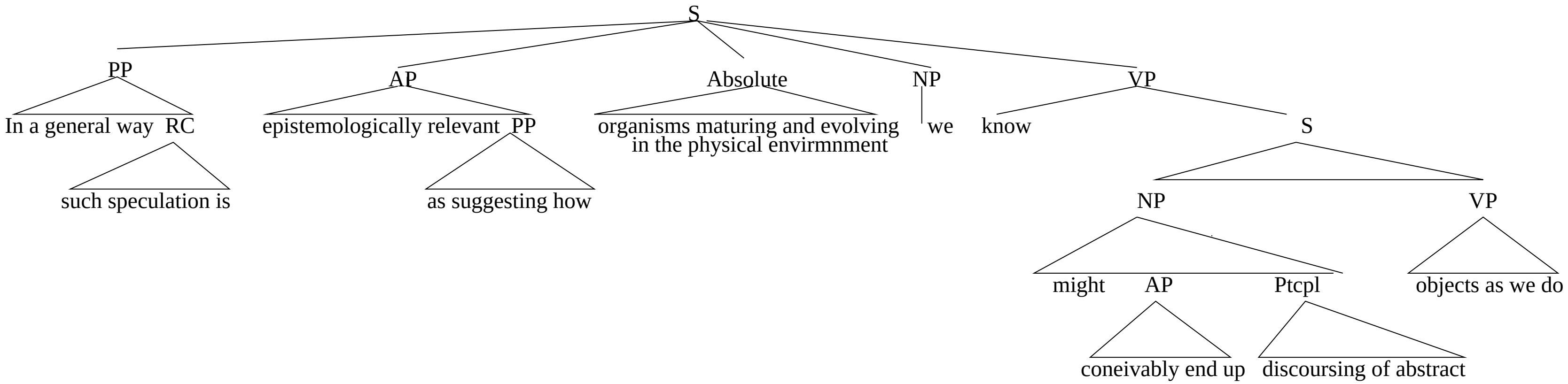
- It has polynomial parsing complexity (Vijay-Shanker and Weir 1990)
- Hence it has nice “Divide and Conquer” algorithms, like CKY, and Dynamic Programming.
- For real-life sized examples like parsing the newspaper, such algorithms must be statistically optimized.

# IV: Thinking Computationally about Parsing

- No handwritten grammar ever has the coverage that is needed to read the daily newspaper.
- Language is syntactically highly ambiguous and it is hard to pick the best parse. Quite ordinary sentences of the kind you read every day routinely turn out to have hundreds and on occasion thousands of parses, albeit mostly semantically wildly implausible ones.
- High ambiguity and long sentences break exhaustive parsers.

# For Example:

- “In a general way such speculation is epistemologically relevant, as suggesting how organisms maturing and evolving in the physical environment we know might conceivably end up discoursing of abstract objects as we do.” (Quine 1960. 123).
- —yields the following (from Abney 1996), among many other horrors:



# Wide Coverage Parsing: the State of the Art

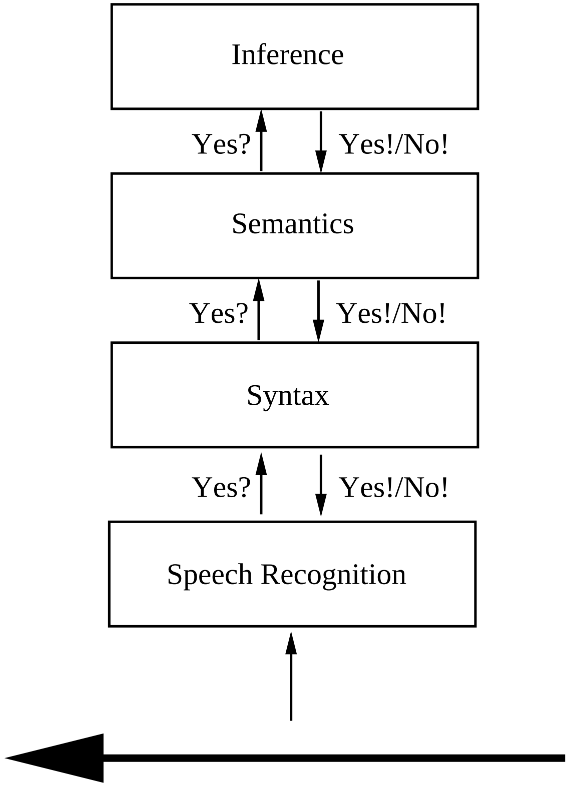
- Early attempts to model parse probability by attaching probabilities to rules of CFG performed poorly.
- Great progress as measured by the ParsEval measure has been made by combining statistical models of headword dependencies with CF grammar-based parsing (Collins 1999; Charniak 2000; Bod 2001)
- However, the ParsEval measure is very forgiving. Such parsers have until now been based on highly overgenerating context-free covering grammars. Analyses depart in important respects from interpretable structures.
- In particular, they fail to represent the long-range "deep" semantic dependencies that are involved in relative and coordinate constructions, as in $A\ company_i\ that_i\ I\ think\ IBM\ bought_i$, and $IBM_i\ bought_{i,j}\ and\ sold_{i,j}\ Lotus_j$.

# The Anatomy of a Parser

- Every parser can be identified by three elements:
  - A Grammar (Regular, Context Free, Linear Indexed, etc.) and an associated automaton (Finite state, Push-Down, Embedded Push-Down, etc.);
  - A search Algorithm characterized as left-to-right (etc.), bottom-up (etc.), and the associated working memories (etc.);
  - An Oracle, to resolve ambiguity.
- The oracle can be used in two ways, either to actively limit the search space, or in the case of an “all paths” parser, to rank the results.
- In wide coverage parsing, we have to use it in the former way.

# The Architecture of the Human Sentence Processor

- “Garden path” effects are sensitive to semantic content (Bever 1970) and context (Altmann and Steedman 1988) requiring a “cascade”:



Inference

Yes? Yes!/No!

Semantics

Yes? Yes!/No!

Syntax

Yes? Yes!/No!

Speech Recognition

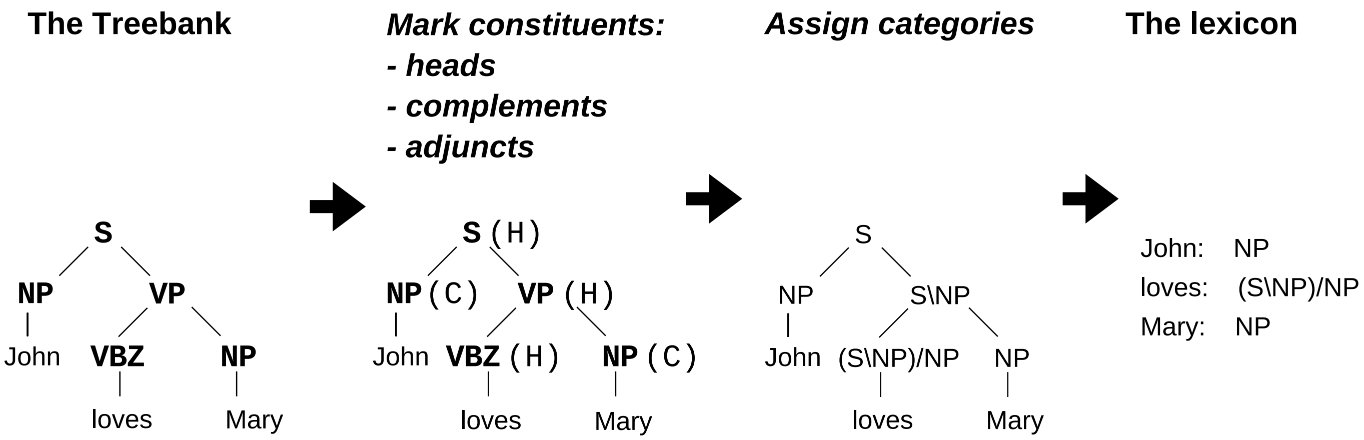
*The* $\left\{\begin{array}{l}\textit{flowers}\\ \textit{doctor}\end{array}\right\}$ *sent for the patient died*

# Head-dependencies as Oracle

- Head-dependency-Based Statistical Parser Optimization works because it approximates an oracle using semantics and real world inference.
- Its probably as close as we will get to the real thing for the foreseeable future.
- In fact, the knowledge- and inference- based psychological oracle may be much more like a probabilistic relational model than like traditional logicist representations, especially if embedded in associative knowledge representations, augmented by ontologies and integrated with a dynamic context model.
- Many context-free processing techniques generalize to the mildly context sensitive class.
- The “nearly context free” grammars such as LTAG and CCG—the least expressive generalization of CFG known—have been treated by Xia (1999), Hockenmaier and Steedman (2002), and Clark and Curran (2004).

# Supervised CCG Induction by Machine

- Extract a CCG lexicon from the Penn Treebank: Hockenmaier and Steedman (2002), Hockenmaier (2003) (cf. Buszkowski and Penn 1990; Xia 1999).



**The Treebank**

**S**
**NP** **VP**
John **VBZ** **NP**
loves Mary

***Mark constituents:***
***- heads***
***- complements***
***- adjuncts***

**S** (H)
**NP** (C) **VP** (H)
John **VBZ** (H) **NP** (C)
loves Mary

***Assign categories***

S
NP S\NP
John (S\NP)/NP NP
loves Mary

**The lexicon**

John: NP
loves: (S\NP)/NP
Mary: NP

- This trades lexical types (500 against 48) for rules (around 3000 instantiated binary combinatory rule types against around 12000 PS rule types) with standard Treebank grammars.

# Overall Dependency Recovery

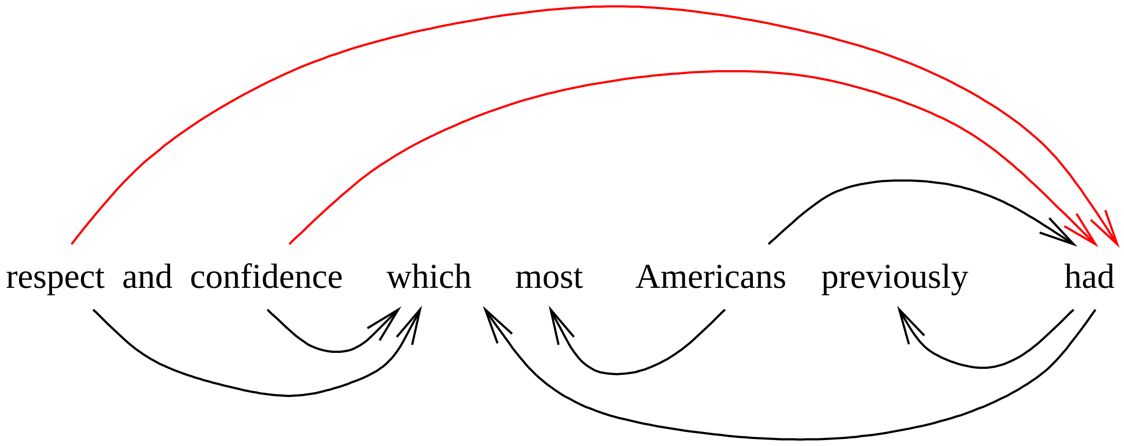
| | LP | LR | UP | UR | cat |
|---|---|---|---|---|---|
| Clark et al. 2002 | 81.9 | 81.8 | 90.1 | 89.9 | 90.3 |
| Hockenmaier 2003 | 84.3 | 84.6 | 91.8 | 92.2 | 92.2 |
| **Log-linear** | **86.6** | **86.3** | **92.5** | **92.1** | **93.6** |
| Hockenmaier (POS) | 83.1 | 83.5 | 91.1 | 91.5 | 91.5 |
| **Log-linear (POS)** | **84.8** | **84.5** | **91.4** | **91.0** | **92.5** |

Table 1: Dependency evaluation on Section 00 of the Penn Treebank

- To maintain comparability to Collins, Hockenmaier (2003) did not use a Supertagger, and was forced to use beam-search. With a Supertagger front-end, the Generative model might well do as well as the Log-Linear model. We have yet to try this experiment.

# Recovering Deep or Semantic Dependencies

Clark et al. (2002)



respect and confidence which most Americans previously had

| lexical_item | category | slot | head_of_arg |
|---|---|---|---|
| *which* | $(NP_X\backslash NP_{X,1})/(S[dcl]_2/NP_X)$ | 2 | *had* |
| *which* | $(NP_X\backslash NP_{X,1})/(S[dcl]_2/NP_X)$ | 1 | *confidence* |
| *which* | $(NP_X\backslash NP_{X,1})/(S[dcl]_2/NP_X)$ | 1 | *respect* |
| *had* | $(S[dcl]_{had}\backslash NP_1)/NP_2)$ | 2 | *confidence* |
| *had* | $(S[dcl]_{had}\backslash NP_1)/NP_2)$ | 2 | *respect* |

# Full Object Relatives in Section 00

- 431 sentences in WSJ 2-21, 20 sentences (24 object dependencies) in Section 00. 1. Commonwealth Edison now faces an additional court-ordered *refund* on its summerwinter rate differential collections *that* the Illinois Appellate Court has *estimated* at DOLLARS.

  2. Mrs. Hills said many of the 25 *countries that* she *placed* under varying degrees of scrutiny have made genuine progress on this touchy issue.

  √ 3. It's the petulant complaint of an impudent *American whom* Sony *hosted* for a year while he was on a Luce Fellowship in Tokyo – to the regret of both parties.

  √ 4. It said the *man*, *whom* it did not *name*, had been found to have the disease after hospital tests.

  5. Democratic Lt. Gov. Douglas Wilder opened his gubernatorial battle with Republican Marshall Coleman with an abortion *commercial* produced by Frank Greer *that* analysts of every political persuasion *agree* was a tour de force.

  6. Against a shot of Monticello superimposed on an American flag, an announcer talks about the strong *tradition* of freedom and individual liberty *that* Virginians have *nurtured* for generations.

  √ 7. Interviews with analysts and business people in the U.S. suggest that Japanese capital may produce the economic *cooperation that* Southeast Asian politicians have *pursued* in fits and starts for decades.

  8. Another was Nancy Yeargin, who came to Greenville in 1985, full of the *energy* and *ambitions that* reformers wanted to *reward*.

  9. Mostly, she says, she wanted to prevent the *damage* to self-esteem *that* her low-ability students would *suffer* from doing badly on the test.

  √ 10. Mrs. Ward says that when the cheating was discovered, she wanted to avoid the morale-damaging public *disclosure that* a trial would *bring*.

  √ 11. In CAT sections where students' knowledge of two-letter consonant sounds is tested, the authors noted that

Scoring High concentrated on the same *sounds that* the test *does* – to the exclusion of other *sounds that* fifth graders should *know*.

√ 12. Interpublic Group said its television programming *operations* – *which* it *expanded* earlier this year – agreed to supply more than 4,000 hours of original programming across Europe in 1990.

13. Interpublic is providing the programming in return for advertising *time*, *which* it *said* will be valued at more than DOLLARS in 1990 and DOLLARS in 1991.

√ 14. Mr. Sherwood speculated that the *leeway that* Sea Containers *has* means that Temple would have to substantially increase their bid if they're going to top us.

√ 15. The Japanese companies bankroll many small U.S. companies with promising products or ideas, frequently putting their money behind *projects that* commercial banks won't *touch*.

√ 16. In investing on the basis of future transactions, a role often performed by merchant banks, trading companies can cut through the *logjam that* small-company owners often *face* with their local commercial banks.

17. A high-balance *customer that* banks *pine for*, she didn't give much thought to the rates she was receiving, nor to the fees she was paying.

√ 18. The events of April through June damaged the *respect* and *confidence which* most Americans previously *had* for the leaders of China.

√ 19. He described the situation as an escrow *problem*, a timing *issue*, *which* he *said* was rapidly rectified, with no losses to customers.

√ 20. But Rep. Marge Roukema (R., N.J.) instead praised the House's acceptance of a new youth training wage, a *subminimum that* GOP administrations have *sought* for many years.

Cases of object extraction from a relative clause in 00; the extracted object, relative pronoun and verb are in italics; sentences marked with a √ are cases where the parser correctly recovers all object dependencies

# V: Thinking Computationally about Acquisition

- The child's problem is similar but a little harder.
  - They have unordered logical forms, not language-specific ordered derivation trees.
  - So they have to work out which word(s) go with which element(s) of logical form, as well as the directionality of the syntactic categories (which are otherwise universally determined by the semantic types of the latter).
- They do not seem to have to deal with a greater amount of error than the Penn WSJ treebank has (McWhinnie 2005).
  - But they may need to deal with situations which support a number of logical forms.
  - And they need to be able to recover from temporary wrong lexical assignments.
  - And they need to be able to handle lexical ambiguity.

# Example

- The Stage VI child has encountered a dog. Then she encounters more dogs.

(32) a. Child: (thinks:) $more'dog'$

b. Adult: “More doggies!”

c. Child’s lexical candidates:

more $:= NP/NP : \lambda x.x$ doggies $:= NP/NP : \lambda x.x$
more $:= NP\backslash NP : \lambda x.x$ doggies $:= NP\backslash NP : \lambda x.x$
more $:= NP/N : more'$ doggies $:= NP/N : more'$
more $:= NP\backslash N : more'$ doggies $:= NP\backslash N : more'$
more $:= N : dog'$ doggies $:= N : dog'$
more $:= NP : more'dog'$ doggies $:= NP : more'dog'$
more doggies $:= more'dog'$

- She might get it wrong, starting to use “doggies” to mean “more”. But she soon corrects in the light of further evidence.
- Where $more'dog'$ came from is a different question—see Quine (1960).

# Computational Accounts

- Siskind (1995, 1996), Villavicencio (2002), and Zettlemoyer and Collins (2005) offer computational models of this process, the latter two explicitly using CCG.
- All of these models depend on availability to the learner of short sentences paired with logical forms, since complexity is determined by a cross-product of powersets both of which are exponential in sentence length.
- A number of techniques are available to make search efficient including association of incrementally adjusted Bayesian priors with category-types.
- No notion of "triggers" distinct from reasonably short string-meaning pairs is necessary.
- It is possible to use the statistics of the lexicon itself to implicitly represent "parameters" such as verb-finality, via incrementally adjusted prior probabilities on the members of the set of universally available category types.

# Conclusions: for a Cognitive Informatics

Since the grammar describes language as action to start with:

- Language production is planning (and planning is derivation in the grammar)
- Language understanding is plan recognition (this also is just derivation in the grammar)
- Dialogue management is plan-based collaboration (applying directly to the representations delivered by NLG and NLU)
- Competence grammar = syntax, denotational semantics, dynamic semantics (but all processing integrates context and pragmatics)

# Conclusions (contd.)

- It’s not surprising that the language faculty is grounded in this way in planning, tool use, and action as a group. These skills have been evolved over a long period, and are what distinguishes primate evolution, and among primates, our own. There is evidence of this at the level of:
    - Representation: The existence of “mirror neurons” in macaques in areas homologous to Broca’s in humans shows the lineage of the ability to represent own and other’s actions identically, and infer from action to goal.
    - Inference: Mechanisms that take account of object-oriented information when planning and recognizing plans, including such information about others’ abilities in this regard (tool concepts, including potentially recursive propositional attitude concepts)
    - Learning: Reward mechanisms for successful knowledge coordination (“peekaboo” games)

# References

Altmann, G. and Steedman, M. (1988). Interaction with context during human sentence processing. *Cognition*, 30:191–238.

Bever, T. (1970). The cognitive basis for linguistic structures. In Hayes, J., editor, *Cognition and the Development of Language*, pages 279–362. Wiley, New York.

Bod, R. (2001). What is the minimal set of fragments that achieves maximal par se accuracy? In *Proceedings of the 39th Meeting of the ACL*, Toulouse, France.

Buszkowski, W. and Penn, G. (1990). Categorial grammars determined from linguistic data by unification. *Studia Logica*, 49:431–454.

Charniak, E. (2000). A maximum-entropy-inspired parser. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 132–139, Seattle, WA.

Clark, S. and Curran, J. R. (2004). Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd Meeting of the ACL*, pages 104–111, Barcelona, Spain.

Clark, S., Hockenmaier, J., and Steedman, M. (2002). Building deep dependency structures with a wide-coverage ccg parser. In *Proceedings of the 40th Meeting of the ACL*, pages 327–334, Philadelphia, PA.

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania.

Curry, H. B. and Feys, R. (1958). *Combinatory Logic: Vol. I*. North Holland, Amsterdam.

Freud, S. (1891). *Zur Auffassung der Aphasien*. Franz Deuticke, Leipzig and Wien. English Translation 1953, *On Aphasia*, Imago.

Gibson, J. (1966). *The Senses Considered as Perceptual Systems*. Houghton-Mifflin Co., Boston, MA.

Girard, J.-Y. (1995). Linear logic: its syntax and semantics. In Grirard, J.-Y., Lafont, Y., and Regnier, L., editors, *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Notes*, pages 1–42. Cambridge University Press, Cambridge.

Hockenmaier, J. (2003). *Data and models for statistical parsing with CCG.* PhD thesis, School of Informatics, University of Edinburgh.

Hockenmaier, J. and Steedman, M. (2002). Acquiring compact lexicalized grammars from a cleaner treebank. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, pages 1974–1981, Las Palmas, Spain.

Joshi, A. (1988). Tree adjoining grammars. In Dowty, D., Karttunen, L., and Zwicky, A., editors, *Natural Language Parsing*, pages 206–250. Cambridge University Press, Cambridge.

Joshi, A., Vijay-Shanker, K., and Weir, D. (1991). The convergence of mildly context-sensitive formalisms. In Sells, P., Shieber, S., and Wasow, T., editors, *Processing of Linguistic Structure*, pages 31–81. MIT Press, Cambridge, MA.

Köhler, W. (1925). *The Mentality of Apes.* Harcourt Brace and World, New York.

Kowalski, R. (1979). *Logic for Problem Solving.* North Holland, Amsterdam.

Lashley, K. (1951). The problem of serial order in behavior. In Jeffress, L., editor, *Cerebral Mechanisms in Behavior*, pages 112–136. Wiley, New York. reprinted in Saporta (1961).

McWhinnie, B. (2005). Item based constructions and the logical problem. In *Proceedings of the Workshop on Psychocomputational Models of Human Language Acquisition. CoNNL-9*, pages 53–68, New Brunswick. ACL.

Piaget, J. (1936). *La naissance de l'intelligence chez l'enfant*. Delachaux et Niestle, Paris. translated 1953 as *The origin of Intelligence in the Child*, Routledge and Kegan Paul.

Quine, W. v. O. (1960). *Word and Object*. MIT Press, Cambridge MA.

Saporta, S., editor (1961). *Psycholinguistics: A Book of Readings*. Holt Rinehart Winston, New York.

Siskind, J. (1995). Grounding language in perception. *Artificial Intelligence Review*, 8:371–391.

Siskind, J. (1996). A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, 61:39–91.

Steedman, M. (2000). *The Syntactic Process*. MIT Press, Cambridge, MA.

Vijay-Shanker, K. and Weir, D. (1990). Polynomial time parsing of combinatory categorial grammars. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics, Pittsburgh*, pages 1–8, San Francisco, CA. Morgan Kaufmann.

Villavicencio, A. (2002). *The Acquisition of a Unification-Based Generalised Categorial Grammar*. PhD thesis, University of Cambridge.

Xia, F. (1999). Extracting tree adjoining grammars from bracketed corpora. In *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium(NLPRS-99)*.

Zettlemoyer, L. and Collins, M. (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.