ACL Tutorial: Unification in the Syntax/Semantics Interface
June, 1990

Unification

in the

Syntax/Semantics Interface

Per-Kristian Halvorsen
XEROX PARC and CSLI

John Nerbonne
Universität des Saarlandes

ACL Tutorial
June 6, 1990

Unification In Semantics

- Novel enabling features:
  - Algorithmic:
    - Unification
    - Constraint programming
  - Semantic:
    - Situation semantics
    - Partial (many-valued) logics
    - Many-sorted logics

Unification In Semantics: Issues

- New means of composition: Implications for semantic representation and the nature of semantic rules of unification and constraints
- Systematicity without a requirement for compositionality
- New views of modularity (coordination of syntax and semantics without homomorphisms)
- The nature of semantic rules
- Computing with partial semantic information
- New possibilities for linguistic description (ellipsis, quantification etc.)

Outline

- What's the problem: How semantic analysis differs from syntactic descriptions in its goals
- Past approaches: Comparison with Montagovian semantics
- Descriptive apparatus:

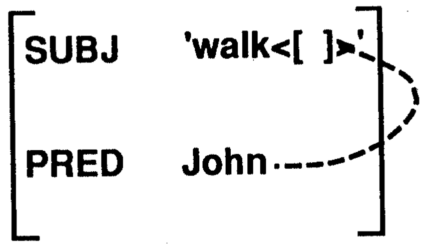
  The logic (syntax and semantics)

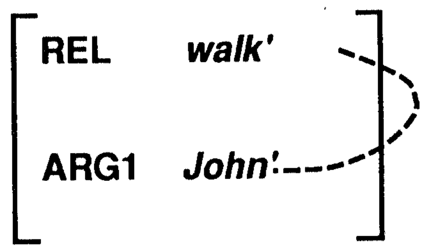
  The language for describing logical forms (the description language)
- The nature of semantic rules: Their effects and how to write them.

Outline (continued)

- Modularity: New ways of structuring of the relationship between linguistic form and semantic interpretation
- Example analyses
- Extensions and new possibilities for analyses:
  - Quantifier scope and functional uncertainty
  - Ellipsis and higher order unification

It is all DAGs anyway!




REL walk'
ARG1 John'

What Makes Semantics Different

- Semantic representations must serve as a basis for a theory of truth and entailment
  - Model theory
  - Proof-procedure (maybe)
  - Operations on the representations correspond to operations on "real" semantic objects
- Information flow: Questions of control arise from the introduction of multiple levels

Architectures for Semantic Interpretation

Computational Approaches

- "Blackboard" architectures

  Coordinated, but arbitrarily complex, operations in the interpretation step

  from several levels

- Cascaded ATNs [Woods 1980]

  Modularity in the specification combined with information sharing at run time

- Rule-by-rule interpretation

  Tightly coupled, but ad hoc, correspondence between syntax and semantics [Thompson 1963]

  Logic grammars, CHAT-80: Horn clauses for the logic

Architectures for Semantic Interpretation

Linguistic Approaches

- (Early) Transformational Grammar
  - Singulary and generalized transformations
  - Projection rules
- Generative Semantics
  - From logical form to surface structure through transformations and piecemeal restructuring restructuring of the tree
- GPSG/HPSG/LFG
  - Rule-by-rule interpretation

Logical Approaches

- Montague Grammar
  - Syntax/Semantics coordinated through homomorphisms
  - Direct Interpretation

Tools for Semantic Composition In Montague Grammar

- Function application
- Meanings providing complete, exhaustively parameterized information about interpretation

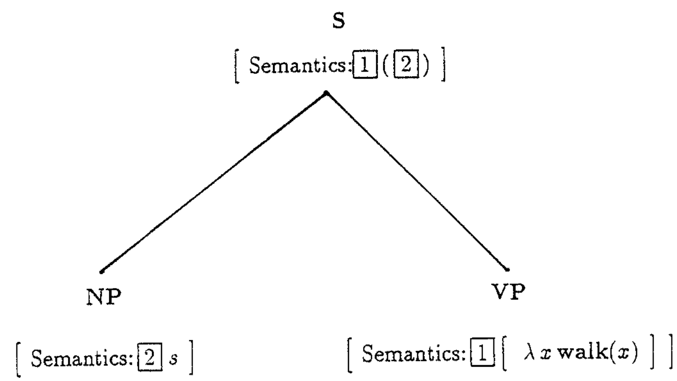
Unification-Based Semantics

Summary of Introductory Illustration

- analogy: lambda-binding :: unification
- descriptions of logical forms are unified
- semantic inference is separate
- extensive structure sharing between syntax and semantics

Semantic Interpretation using Lambda Calculus

“Sam walks”



β-reduction:

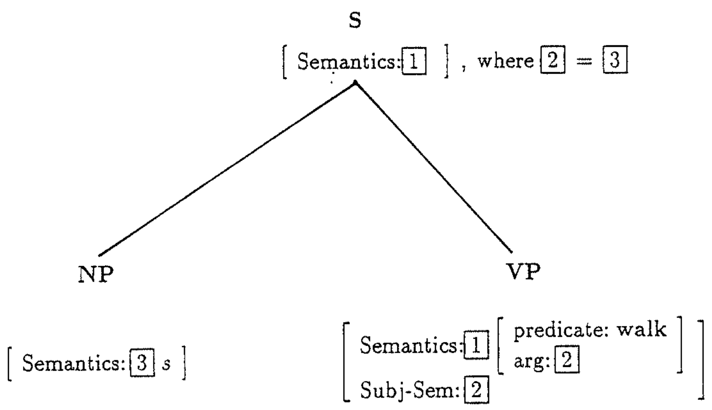
$$[\lambda x \, \mathrm{walk}(x)](s) \leftrightarrow \mathrm{walk}(s)$$

Note:

- arguments bound to argument positions via function application
- build complex structures, then simplify
- accommodate higher-order elements with higher order abstraction

Unification-Based Semantic Interpretation

“Sam walks”



Directly:

walk(s)

Cf. Shieber, 1986; Pollard and Sag, 1987; Fenstad *et al*, 1987

- arguments bound to argument positions via unification
- no complex structures which need simplifying

Higher-Order Information—Quantification

Problem:

- standard model theory for attribute-value languages is quantifier-free predicate calculus (Johnson, 88)
- standard desideratum is decidability

Solution Here:

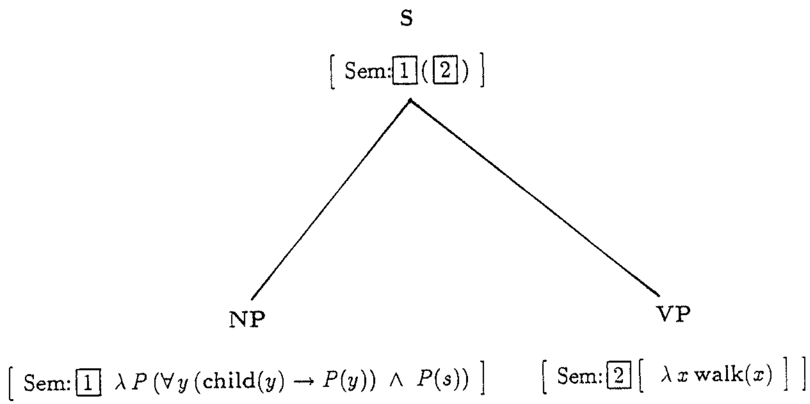
Description of Logical Forms, not Models

- retain standard model theory for avm languages
- retain decidability
- divide "consequence", inference into object language, metalanguage parts

Limited Expressiveness (Moore, 89)

Lambda Calculus Treatment of Type-Raising

"Sam and every child walks"

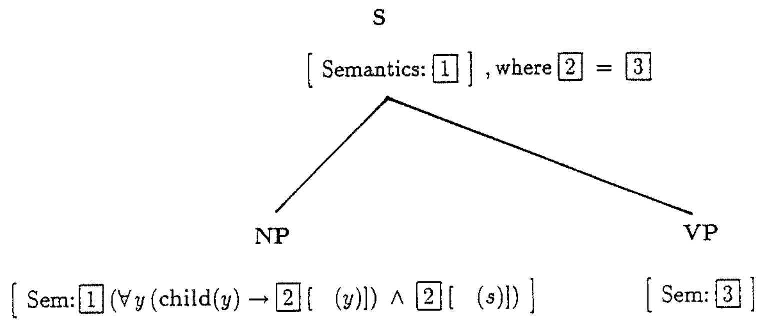


β-reduction (twice):

$$[\lambda P\,(\forall y\,(\text{child}(y) \rightarrow P(y)) \wedge P(s))]\,(\lambda x\,\text{walk}(x))$$

$$\leftrightarrow \forall y\,(\text{child}(y) \rightarrow \text{walk}(y)) \wedge \text{walk}(s)$$

An Impossible Description



Problem:

- Impossible NP Semantics—No Structure [2] with argument = y AND argument = s.
- "Patch" possible here—take only predicate from VP semantics, ignore arguments. No generalization to arbitrarily complex VP semantics.

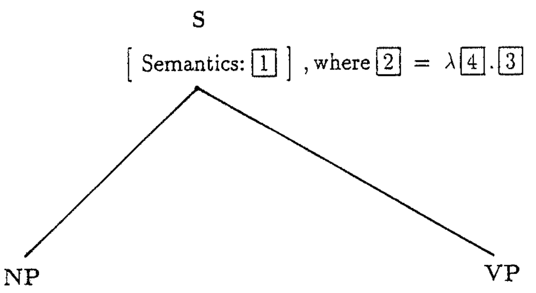
Solution Here:

Description of Logical Forms, not Models

Type-Raising Solution Using LF Description

"Sam and every child walks"

[Semantics: [1]] , where [2] $= \lambda$[4].[3]



[Sem: [1] [$\lambda P\,(\forall y\,(\text{child}(y) \rightarrow P(y)) \wedge P(s))$] ([2])]

[Sem: [3] [predicate: *walk*, arg: [4]], Subjval: [4]]

note:

λ[4].[3] $\Rightarrow$ $\lambda x\,.\,\text{walk}(x)$

β-reduction still needed:

$$[\,\lambda P\,(\forall y\,(\text{child}(y) \rightarrow P(y)) \wedge P(s))]\,(\lambda x\,\text{walk}(x))$$
$$\leftrightarrow \forall y\,(\text{child}(y) \rightarrow \text{walk}(y)) \wedge \text{walk}\,(s)$$

Distinguishing Logical Form and Meaning

Semantic Ambiguity vs. Underspecification

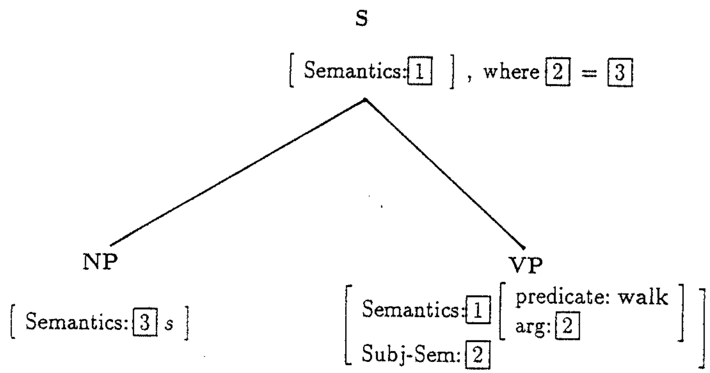
- "bank" ambiguous—home for \$, DM, £ OR riverside
- "glove" underspecified—left- or right-hand glove
- differences in quantification: *three banks* vs. *three gloves*
- differences in anaphor:
  Sam examined a ______, and Dave bought one.
- syntactic/lexical solution: *bank1*, *bank2*
- semantic solution must distinguish:
  bank $\neq$ **\$-home** $\vee$ **riverside**
  glove $=$ **left-glove** $\vee$ **right-glove**
  —but direct interpretation precludes this.

Ambiguity "bank" [predicate: **\$-home**, arg: x] $\sqcup$ [predicate: **riverside**, arg: x]

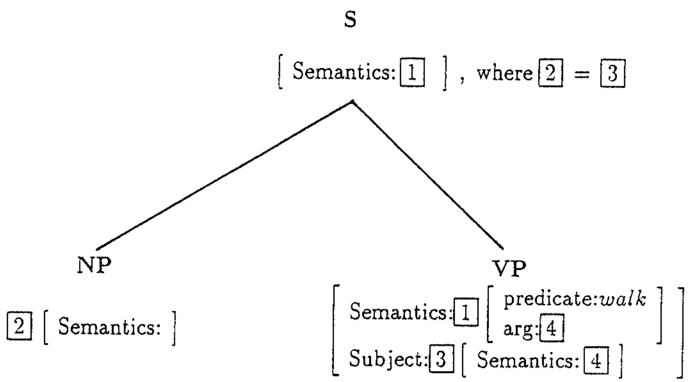
Underspecified "glove" [predicate: **left-glove** $\vee$ **right-glove**, arg: x]

Unification-Based Semantic Interpretation

Semantic Formulation:

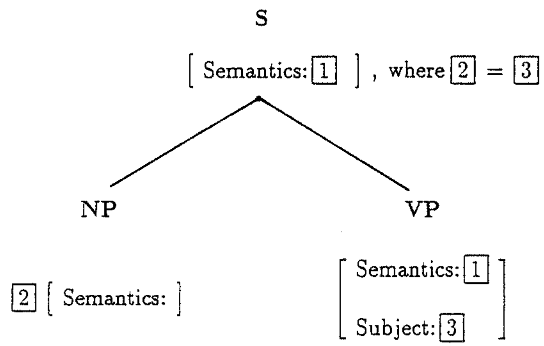


A Revision—Integrating Syntax and Semantics



ACL Tutorial, 5.Jun.90

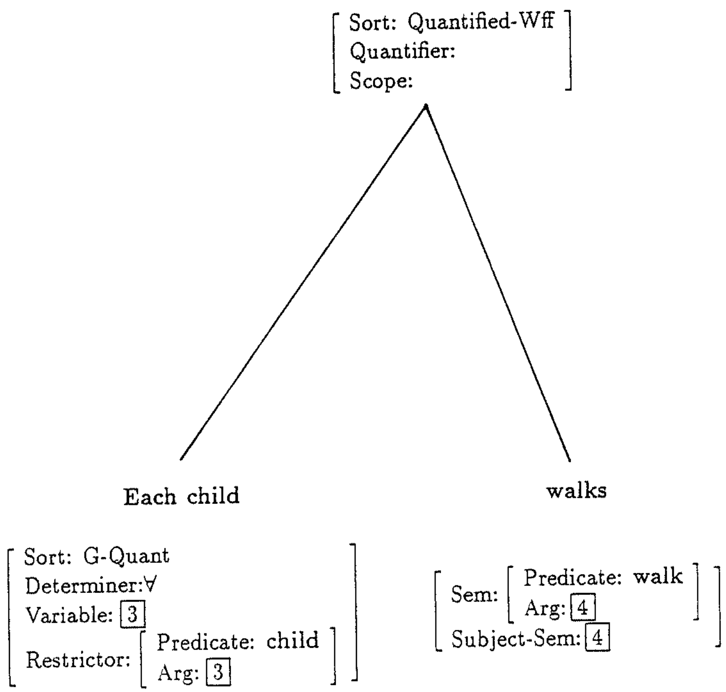
A More Schematic Rule



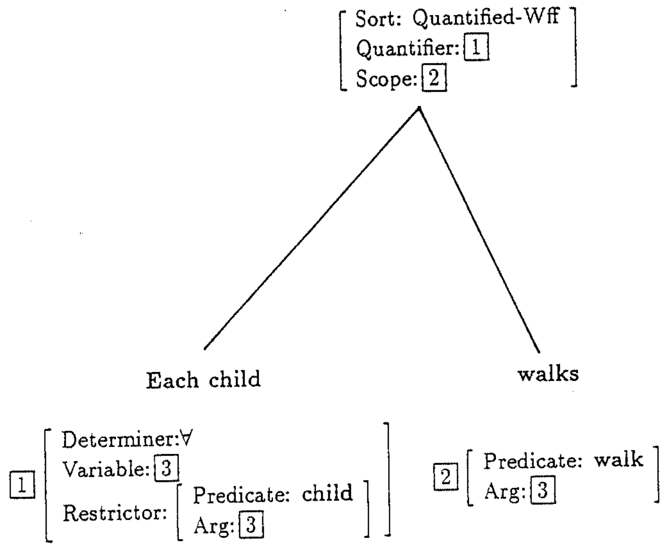
Structure Sharing

- syntax and semantics share structure
- shared structure constitutes interface
- no need for separation in processing

A Quantified Example



Quantified Example Realized



(∀ y child(y) walk(y))

Exercise: Specify the rule needed.

Summary of Introductory Illustration

- analogy: lambda-binding :: unification
- descriptions of logical forms are unified
- semantic inference is separate
- extensive structure sharing between syntax and semantics

3.2 Language of Generalized Quantifiers

A BNF specification of LGQ:

$\langle\text{var}\rangle \Rightarrow x_0, x_1, \ldots$

$\langle\text{const}\rangle \Rightarrow c_0, c_1, \ldots$

$\langle\text{pred}\rangle \Rightarrow P_0, P_1, \ldots$

$\langle\text{role}\rangle \Rightarrow r_0, r_1, \ldots$

$\langle\text{term}\rangle \Rightarrow \langle\text{var}\rangle \mid \langle\text{const}\rangle$

$\langle\text{pair}\rangle \Rightarrow \langle\text{role}\rangle\langle\text{term}\rangle$

$\langle\text{atomic-wff}\rangle \Rightarrow (\langle\text{pred}\rangle\langle\text{pair}\rangle*)$

$\langle\text{conn}\rangle \Rightarrow \wedge, \vee, \leftrightarrow$

$\langle\text{det}\rangle \Rightarrow \forall, \exists, MOST, \ldots$

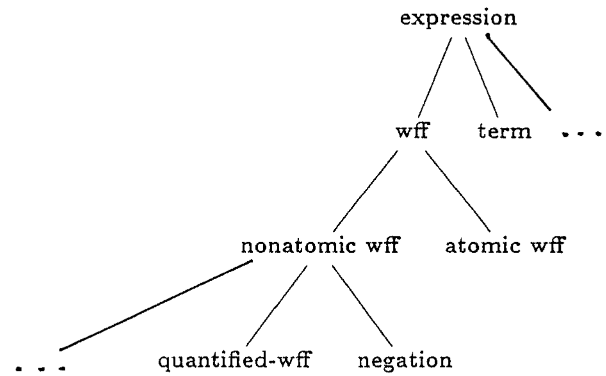
$\langle\text{g-quant}\rangle \Rightarrow \langle\text{det}\rangle\langle\text{var}\rangle\langle\text{wff}\rangle$

$$\begin{aligned}\langle\text{wff}\rangle \Rightarrow\ & \langle\text{atomic-wff}\rangle \\ & \mid (\langle\text{conn}\rangle\langle\text{wff}\rangle+) \\ & \mid (\neg\langle\text{wff}\rangle) \\ & \mid (\langle\text{wff}\rangle \rightarrow \langle\text{wff}\rangle) \\ & \mid (\langle\text{g-quant}\rangle\langle\text{wff}\rangle)\end{aligned}$$

N - 14

Logic Defintion

Sort Signature (Smolka, 1988)



N - 15

3.3 AVM Specifications for LGQ

$$\langle\text{var}\rangle \Rightarrow x_0, x_1, \ldots \qquad \begin{bmatrix} \text{Sort: Var} \\ \text{Index:} \end{bmatrix}$$

$$\langle\text{const}\rangle \Rightarrow c_0, c_1, \ldots \qquad \begin{bmatrix} \text{Sort: Const} \\ \text{Index:} \end{bmatrix}$$

$$\langle\text{atomic-wff}\rangle \Rightarrow (\langle\text{pred}\rangle\langle\text{pair}\rangle*) \qquad \begin{bmatrix} \text{Sort: Atomic-Wff} \\ \text{Predicate:} \\ \text{Arguments:} \begin{bmatrix} \text{Sort: List} \\ \text{r}_i : \\ \vdots \\ \text{r}_n : \end{bmatrix} \end{bmatrix}$$

AVM Specifications for LGQ

$$\langle\text{g-quant}\rangle \Rightarrow \langle\text{det}\rangle\langle\text{var}\rangle\langle\text{wff}\rangle \qquad \begin{bmatrix} \text{Sort: Gen-Quant} \\ \text{Determiner:} \\ \text{Variable:} \begin{bmatrix} \text{Sort: Var} \\ \text{Index:} \end{bmatrix} \\ \text{Restrictor:} \begin{bmatrix} \text{Sort: Wff} \end{bmatrix} \end{bmatrix}$$

AVM Specifications for LGQ

$$\langle \text{wff} \rangle \Rightarrow \ldots \mid (\langle \text{conn} \rangle \langle \text{wff} \rangle^{+}) \qquad \begin{bmatrix} \text{Sort: Complex-Wff} \\ \text{Connective:} \\ \text{Component-Wffs:} \begin{bmatrix} \text{Sort: List} \\ \text{wff}_1 : \\ \vdots \\ \text{wff}_n : \end{bmatrix} \end{bmatrix}$$

$$\ldots \mid (\neg \langle \text{wff} \rangle) \qquad \begin{bmatrix} \text{Sort: Negation} \\ \text{Connective:} \neg \\ \text{Scope-Wff:} \begin{bmatrix} \text{Sort: Wff} \end{bmatrix} \end{bmatrix}$$

$$\ldots \mid (\langle \text{wff} \rangle \rightarrow \langle \text{wff} \rangle) \qquad \begin{bmatrix} \text{Sort: Implication} \\ \text{Connective:} \rightarrow \\ \text{Antecedent:} \begin{bmatrix} \text{Sort: Wff} \end{bmatrix} \\ \text{Consequent:} \begin{bmatrix} \text{Sort: Wff} \end{bmatrix} \end{bmatrix}$$

$$\ldots \mid (\langle \text{g-quant} \rangle \langle \text{wff} \rangle) \qquad \begin{bmatrix} \text{Sort: Quantified-Wff} \\ \text{Gen-Quant:} \begin{bmatrix} \text{Sort: G-Quant} \end{bmatrix} \\ \text{Scope:} \begin{bmatrix} \text{Sort: Wff} \end{bmatrix} \end{bmatrix}$$

Allowable Specifications

$$\begin{bmatrix} \text{Sort: Atomic-Wff} \\ \text{Predicate: Walk} \\ \text{Arguments:} \begin{bmatrix} \text{Sort: List} \\ \text{agent:} \begin{bmatrix} \text{Sort: Ind-Const} \\ \text{Index: j} \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \mathit{Walk}(\mathit{agt}\ j)$$

$$\begin{bmatrix} \text{Sort: Quantified-Wff} \\ \text{Quantifier:} \begin{bmatrix} \text{Sort: G-Quant} \\ \text{Determiner:} \forall \\ \text{Variable:} \boxed{1} \begin{bmatrix} \text{Sort: Var} \\ \text{Index: y} \end{bmatrix} \\ \text{Restrictor:} \begin{bmatrix} \text{Sort: Atomic-Wff} \\ \text{Predicate: Child} \\ \text{Arguments:} \begin{bmatrix} \text{Sort: List} \\ \text{instance:} \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{Scope:} \begin{bmatrix} \text{Sort: Atomic-Wff} \\ \text{Predicate: Walk} \\ \text{Arguments:} \begin{bmatrix} \text{Sort: List} \\ \text{agent:} \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$(\forall y \quad (\mathit{child\ inst}\ y) \quad (\mathit{walk\ agent}\ y))$$

$$\begin{bmatrix} \text{Sort: Quantified-Wff} \\ \text{Quantifier:} \begin{bmatrix} \text{Sort: G-Quant} \\ \text{Determiner:} \forall \\ \text{Variable:} \boxed{1} \begin{bmatrix} \text{Sort: Var} \\ \text{Index: y} \end{bmatrix} \\ \text{Restrictor:} \begin{bmatrix} \text{Sort: Atomic-Wff} \\ \text{Predicate: Child} \\ \text{Arguments:} \begin{bmatrix} \text{Sort: List} \\ \text{instance:} \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{Scope:} \begin{bmatrix} \text{Sort: Wff} \end{bmatrix} \end{bmatrix}$$

$$(\forall y \quad (\mathit{child\ inst}\ y) \quad (\mathit{walk\ agent}\ y))$$

$$(\forall y \quad (\mathit{child\ inst}\ y) \quad (P))$$

$$(\forall y \quad (\mathit{child\ inst}\ y) \quad (\wedge (\mathit{walk\ agent}\ y) \quad (\mathit{talk\ agent}\ y)))$$

6 Consistent Overspecifications

6.1 Semantic Theory of Agreement

- Pollard and Sag ([8]), agreement of pronouns and reflexive pronouns with antecedents
- Dowty and Jacobson ([6]), agreement of controllers and controlled complements

Agreement Information

$$\begin{bmatrix} \text{Sort: Var} \\ \text{Index:} \\ \text{Agreement:} \begin{bmatrix} \text{Person:} \\ \text{Number:} \\ \text{(Gender:)} \end{bmatrix} \end{bmatrix}$$

LGQ Descriptions—Projections of Meaning Representations

Meaning Representation	Logic Description	Satisfying Instance
$\begin{bmatrix} \text{Sort: Var} \\ \text{Index: } z \\ \text{Agreement:} \begin{bmatrix} \text{Person:3} \\ \text{Number:} \\ \text{(Gender: Masc)} \end{bmatrix} \end{bmatrix}$	$\begin{bmatrix} \text{Sort: Var} \\ \text{Index: } z \end{bmatrix}$	
$\begin{bmatrix} \text{Sort: Var} \\ \text{Index: } z \\ \text{Agreement:} \begin{bmatrix} \text{Person:3} \\ \text{Number:} \\ \text{(Gender: Fem)} \end{bmatrix} \end{bmatrix}$	$\begin{bmatrix} \text{Sort: Var} \\ \text{Index: } z \end{bmatrix}$	

1 The Unification-Based Approach

1.1 Complexes of Information

1.2 Structure-Sharing

From Shieber ([10], p.15):

$$\begin{bmatrix} \text{agreement:} \boxed{1} \begin{bmatrix} \text{number: singular} \end{bmatrix} \\ \text{subject:} \begin{bmatrix} \text{agreement:} \boxed{1} \end{bmatrix} \end{bmatrix} \sqcup \begin{bmatrix} \text{subject:} \begin{bmatrix} \text{agreement:} \begin{bmatrix} \text{person: third} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} \text{agreement:} \boxed{1} \begin{bmatrix} \text{number: singular} \\ \text{person: third} \end{bmatrix} \\ \text{subject:} \begin{bmatrix} \text{agreement:} \boxed{1} \end{bmatrix} \end{bmatrix}$$

2 Structure-sharing in Semantics—Motivation

2.1 Bach and Partee's "Anaphora and Semantic Structure"

Bach and Partee's "Anaphora and Semantic Structure" provide a good example of the degree to which structure is assumed to be shared in semantic representation ([1], p.21):

> I. The objects generated by this grammar are 6-tuples [...]:
>
> 1. an expression of English;
> 2. its syntactic category;
> 3. its translation into IL;
> 4. a set of pairs (maybe empty) $\langle \alpha, i \rangle$, where α is an NP meaning, WH, SELF1, or SELF2 (this is like Cooper's store;
> 5. a set of natural numbers (the indices of the pronoun meanings locally free in (3);
> 6. a set of natural numbers (the indices of pronoun meanings free in (3).

2.2 Later Semantics Frameworks

- Barwise ([2], pp.5ff)
- Pollard and Sag ([9], p.109)
- Carpenter ([3], p.129)

3 A Formalization that Allows Sharing

$$(1)\quad \begin{bmatrix} \text{Phonology:} \ldots & \\ \text{Syntax:} \ldots & \\ \text{Semantics:} & \begin{bmatrix} \text{Logic:} \ldots \\ \text{Obligatory:} \ldots \\ \text{Binding-Variables:} \ldots \\ \text{Reflexives:} \ldots \\ \text{Pronouns:} \ldots \\ \text{Quantifier-Store:} \ldots \\ \tilde{\text{N}}\text{-Anaphors:} \ldots \\ \text{Reference-Markers:} \ldots \end{bmatrix} \end{bmatrix}$$

3.1 Some Examples

$$(2)\quad \begin{bmatrix} \text{Phonology:} & \text{``himself''} \\ \text{Syntax:} \ldots & \\ \text{Semantics:} & \begin{bmatrix} \text{Logic:} r_i \\ \text{Obligatory:} \emptyset \\ \text{Binding-Variables:} \emptyset \\ \text{Reflexives:} \{r_i\} \\ \text{Pronouns:} \emptyset \\ \text{Quantifier-Store:} \emptyset \\ \tilde{\text{N}}\text{-Anaphors:} \emptyset \\ \text{Reference-Markers:} \emptyset \end{bmatrix} \end{bmatrix}$$

$$(3)\quad \begin{bmatrix} \text{Phonology:} & \text{``himself''} \\ \text{Syntax:} \ldots & \\ \text{Semantics:} & \begin{bmatrix} \text{Logic:} r_i \\ \text{Reflexives:} \{r_i\} \end{bmatrix} \end{bmatrix}$$

Structure-Sharing

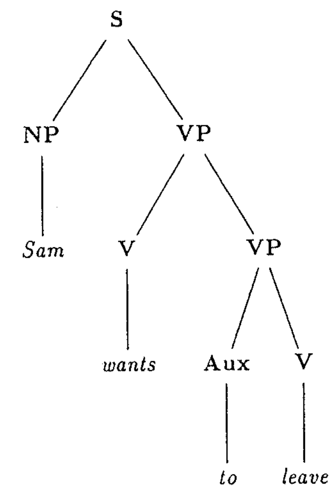
Control

- Control arises when a head constrains the semantics not just of its own dependents, but also that of the dependents' dependents. This typically results in a many-to-one relation between grammatical dependents and semantic argument positions.

Sam wants to leave

want(s, ˆ(leave(s)))

Specifying Control



Phrasal structure isn't remarkable; lexical specification is therefore preferred.

Specifying Control

We can't simply specify control using lexical semantics alone; the property may be arbitrarily complex.

$$\text{want}(s, \hat{}(\text{leave}(s)))$$

$$\text{want}(x, \hat{}(P(x)))$$

Sam wants to invite everyone over if it rains

$$\text{want}(s, \hat{}(\text{rain} \rightarrow \forall x\,(\text{invite-over}(s, x))))$$

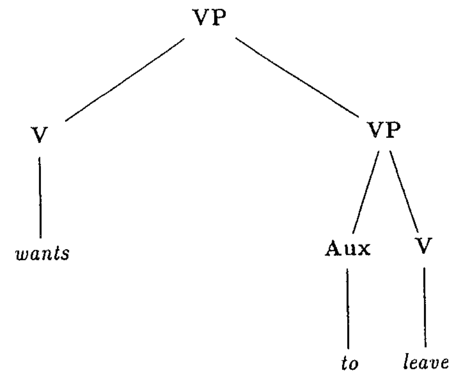
Specifying Control—A Structure-Sharing Solution

Lexical Entry

$$\begin{bmatrix} \text{stem: } \textit{want} \\ \text{sem:want}(x, \hat{}e) \\ \text{syn: } \begin{bmatrix} \text{subject: } \begin{bmatrix} \text{sem:}x \end{bmatrix} \\ \text{complement: } \begin{bmatrix} \text{subject: } \begin{bmatrix} \text{sem:}x \end{bmatrix} \\ \text{sem:}e \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \text{stem: } \textit{want} \\ \text{sem:want}(\boxed{1}, \hat{}\boxed{2}) \\ \text{syn: } \begin{bmatrix} \text{subject: } \begin{bmatrix} \text{sem: } \boxed{1} \end{bmatrix} \\ \text{complement: } \begin{bmatrix} \text{subject: } \begin{bmatrix} \text{sem: } \boxed{1} \end{bmatrix} \\ \text{sem: } \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Effects of Lexical Structure

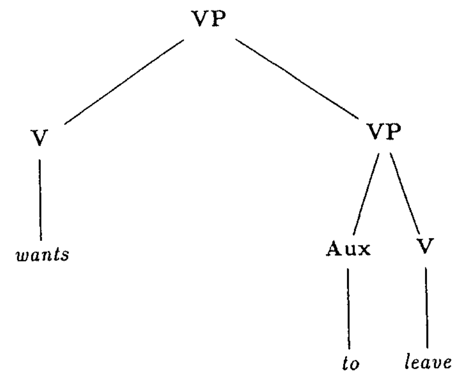



```
[ stem: want
  sem:want( [1] , ^ [2] )
  syn: [ subject: [ sem: [1] ]
         complement: [ subject: [ sem: [1] ]
                       sem: [2] ] ] ]
```

```
[ subject: [ sem: [3] ]
  sem:leave( [3] ) ]
```


Contribution of Phrasal Grammar

[sem: [4]] , where [5] = [6]



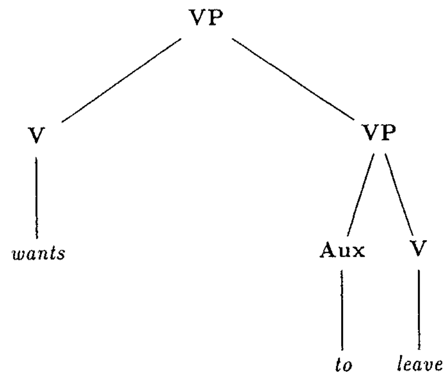

```
[ stem: want
  sem: [4] want( [1] , ^ [2] )
  syn: [ subject: [ sem: [1] ]
         complement: [5] [ subject: [ sem: [1] ]
                           sem: [2] ] ] ]
```

```
[6] [ subject: [ sem: [3] ]
      sem:leave( [3] ) ]
```

Control

Results

$$\left[\text{sem:want}(\boxed{1} \sqcup \boxed{3}, \hat{}\,\text{leave}(\boxed{1} \sqcup \boxed{3}))\right]$$



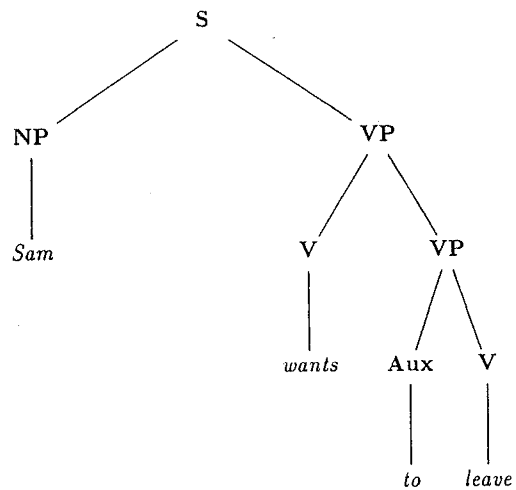
$$\begin{bmatrix} \text{stem: } \textit{want} \\ \text{sem: } \boxed{4}\ \text{want}(\boxed{1}, \hat{}\ \boxed{2}) \\ \text{syn: } \begin{bmatrix} \text{subject: } [\text{sem: } \boxed{1}] \\ \text{complement: } \boxed{5} \begin{bmatrix} \text{subject: } [\text{sem: } \boxed{1}] \\ \text{sem: } \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\boxed{6} \begin{bmatrix} \text{subject: } [\text{sem: } \boxed{3}] \\ \text{sem:leave}(\boxed{3}) \end{bmatrix}$$

Control

Sentence Semantics

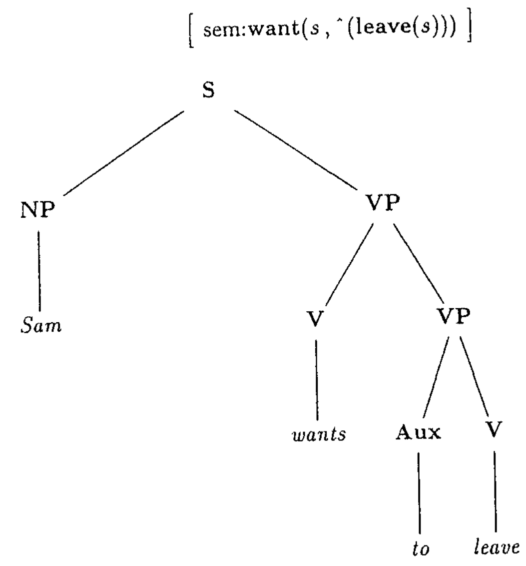
$$[\text{sem: } \boxed{4}]\text{, where } \boxed{5} = \boxed{6}$$



$$\boxed{5}\ [\text{sem: } s]$$

$$\begin{bmatrix} \text{sem: } \boxed{4}\ \text{want}(\boxed{1} \sqcup \boxed{3}, \hat{}\,\text{leave}(\boxed{1} \sqcup \boxed{3})) \\ \text{syn: } [\text{subject: } \boxed{6}\ [\text{sem: } \boxed{1} \sqcup \boxed{3}]] \end{bmatrix}$$

Sentence Semantics—Results



Other Examples

Sam will leave

$$\text{will}\ \begin{bmatrix} \text{stem: } \textit{will} \\ \text{sem:} F(\boxed{2}) \\ \text{syn: } \begin{bmatrix} \text{subject: } \begin{bmatrix} \text{sem: } \boxed{1} \end{bmatrix} \\ \text{complement: } \begin{bmatrix} \text{subject: } \begin{bmatrix} \text{sem: } \boxed{1} \end{bmatrix} \\ \text{sem: } \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

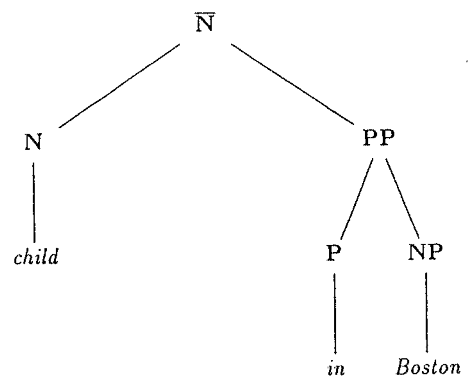
Sam does leave

$$\text{do}\ \begin{bmatrix} \text{stem: } \textit{do} \\ \text{sem: } \boxed{2} \\ \text{syn: } \begin{bmatrix} \text{subject: } \begin{bmatrix} \text{sem: } \boxed{1} \end{bmatrix} \\ \text{complement: } \begin{bmatrix} \text{subject: } \begin{bmatrix} \text{sem: } \boxed{1} \end{bmatrix} \\ \text{sem: } \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Sam did not leave

$$\text{not}\ \begin{bmatrix} \text{stem: } \textit{not} \\ \text{sem:} \neg(\boxed{2}) \\ \text{syn: } \begin{bmatrix} \text{subject: } \begin{bmatrix} \text{sem: } \boxed{1} \end{bmatrix} \\ \text{complement: } \begin{bmatrix} \text{subject: } \begin{bmatrix} \text{sem: } \boxed{1} \end{bmatrix} \\ \text{sem: } \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

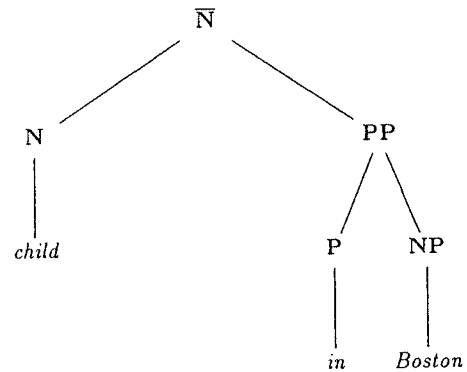
Other Applications of Control-Like Analyses



$$\begin{bmatrix} \text{stem: } \textit{child} \\ \text{sem: } \begin{bmatrix} \text{logic: child}(\boxed{1}) \\ \text{var: } \boxed{1} \end{bmatrix} \\ \text{syn: } \begin{bmatrix} \text{PP Adjunct: } \begin{bmatrix} \text{subj.sem: } \begin{bmatrix} \text{logic: } \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \quad \begin{bmatrix} \text{subj: } \begin{bmatrix} \text{sem: } \begin{bmatrix} \text{logic: } \boxed{2} \end{bmatrix} \end{bmatrix} \\ \text{sem: } \begin{bmatrix} \text{logic: in}(\boxed{2}, \text{Boston}) \end{bmatrix} \end{bmatrix}$$

Adnominal PP's

$$\begin{bmatrix} \text{sem: } \begin{bmatrix} \text{logic: } \wedge \{\boxed{3}, \boxed{4}\} \\ \text{var: } \boxed{1} \sqcup \boxed{2} \end{bmatrix} \end{bmatrix}$$



$$\begin{bmatrix} \text{stem: } \textit{child} \\ \text{sem: } \begin{bmatrix} \text{logic: } \boxed{3}\ \text{child}(\boxed{1}) \\ \text{var: } \boxed{1} \end{bmatrix} \\ \text{syn: } \begin{bmatrix} \text{PP Adjunct: } \begin{bmatrix} \text{subj.sem: } \begin{bmatrix} \text{logic: } \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \quad \begin{bmatrix} \text{subj: } \begin{bmatrix} \text{sem: } \begin{bmatrix} \text{logic: } \boxed{2} \end{bmatrix} \end{bmatrix} \\ \text{sem: } \begin{bmatrix} \text{logic: } \boxed{4}\ \text{in}(\boxed{2}, \text{Boston}) \end{bmatrix} \end{bmatrix}$$

Control

Other Adnominals

child swimming away — participial VP

$$\begin{bmatrix} \text{stem: } \textit{child} \\ \text{sem: } \begin{bmatrix} \text{logic: child} (\boxed{1}) \\ \text{var: } \boxed{1} \end{bmatrix} \\ \text{syn: } \begin{bmatrix} \begin{matrix} \text{VP} \\ [+\text{part}] \end{matrix} \text{: } \begin{bmatrix} \text{subj.sem: } \begin{bmatrix} \text{logic: } \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Dutch child — Adjective

$$\begin{bmatrix} \text{stem: } \textit{child} \\ \text{sem: } \begin{bmatrix} \text{logic: child} (\boxed{1}) \\ \text{var: } \boxed{1} \end{bmatrix} \\ \text{syn: } \begin{bmatrix} \text{Adjective: } \begin{bmatrix} \text{subj.sem: } \begin{bmatrix} \text{logic: } \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

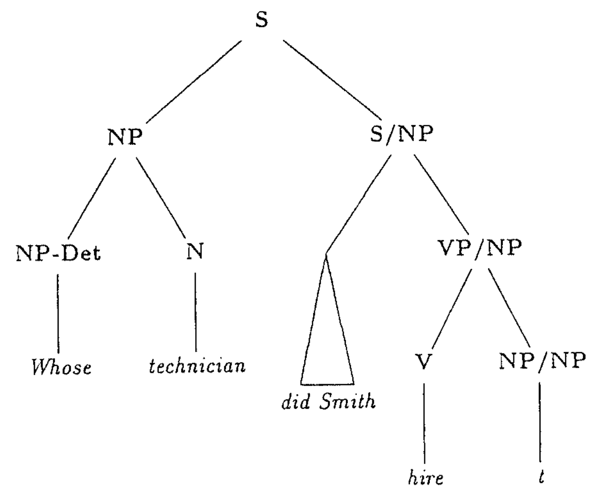
Structure-Sharing

Long-Distance Dependence

- questions
  –Which part was reported to have been defective?
- relative clauses
  –Did Smith review the orders where parts had been reported as defective?
- comparatives
  –Which customers wanted more software than they had ordered the previous year?

Long-Distance Dependence—à la GPSG/HPSG

- Whose technician did Smith hire?



A Logical Type for Questions—Lambda Expressions

BNF Clause:

$\langle \lambda\text{-questioner} \rangle \Rightarrow ?\lambda \langle \text{var} \rangle \langle \text{wff} \rangle$

$\langle \lambda\text{-wff} \rangle \Rightarrow (\langle \lambda\text{-questioner} \rangle \langle \text{wff} \rangle)$

AVM Specification:

$\langle \lambda\text{-questioner} \rangle \Rightarrow ?\lambda \langle \text{var} \rangle \langle \text{wff} \rangle$

$$\begin{bmatrix} \text{Sort: } \lambda\text{-Questioner} \\ \text{Variable: } [\text{ Sort: Var }] \\ \text{Restrictor: } [\text{ Sort: Wff }] \end{bmatrix}$$

$\langle \lambda\text{-wff} \rangle \Rightarrow (\langle \lambda\text{-questioner} \rangle \langle \text{wff} \rangle)$

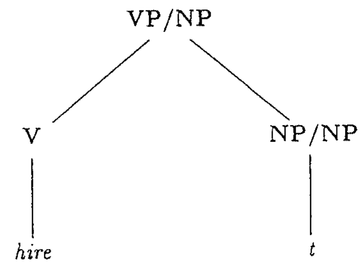
$$\begin{bmatrix} \text{Sort: } \lambda\text{-Wff} \\ \lambda\text{-Questioner: } [\text{ Sort:}\lambda\text{-Questioner }] \\ \text{Scope: } [\text{ Sort: Wff }] \end{bmatrix}$$

- Whose technician did Smith hire?

$$\begin{array}{l} (?\lambda\, x \text{ person}(x) \\ \quad !y \ \wedge \{ \text{possess}(x, y), \text{tech}(y) \} \ ) \\ \qquad \text{hire}(s, y) \end{array}$$

Structure-Sharing

Semantics of Gaps

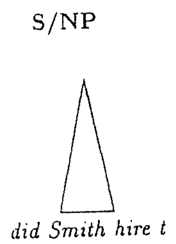


$$\begin{bmatrix} \text{stem: } \textit{hire} \\ \text{sem.logic:hire}(\boxed{1}, \boxed{2}) \\ \text{syn: } \begin{bmatrix} \text{subject: } [\text{ sem: } \boxed{1}] \\ \text{object: } [\text{ sem: } \boxed{2}] \end{bmatrix} \end{bmatrix} \qquad \begin{bmatrix} \text{sem: } \begin{bmatrix} \text{logic: } \boxed{3} \\ \text{slash: } \boxed{3} \end{bmatrix} \end{bmatrix}$$

Long-Distance Dependence

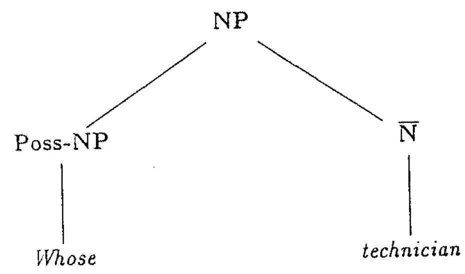
Results of Passing Slash

$$\left[\text{sem: }\left[\begin{array}{l}\text{logic: hire}(s, \boxed{1})\\ \text{slash: }\boxed{1}\end{array}\right]\right]$$



Structure-Sharing

Semantics of Question Words



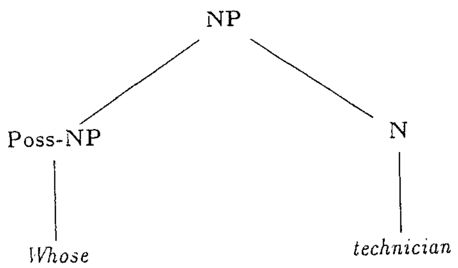
$$\left[\text{sem: }\left[\begin{array}{l}\text{logic: }!\boxed{2} \wedge \{\text{possess}(\boxed{1},\boxed{2}), \boxed{4}\}\\ \text{var: }\boxed{2}\\ \text{que: }\left[\ ?\lambda\boxed{1}\,\text{person}(\boxed{1})\ \right]\end{array}\right]\right]$$

$$\left[\text{sem: }\left[\begin{array}{l}\text{logic: tech}(\boxed{3})\\ \text{var: }\boxed{3}\end{array}\right]\right]$$

Structure-Sharing

Rule for Possessive NP + $\overline{\text{N}}$

$$\left[\text{sem:}\boxed{6}\right]\text{, where } \begin{array}{l}\boxed{4}=\boxed{5}\\ \boxed{2}=\boxed{3}\end{array}$$

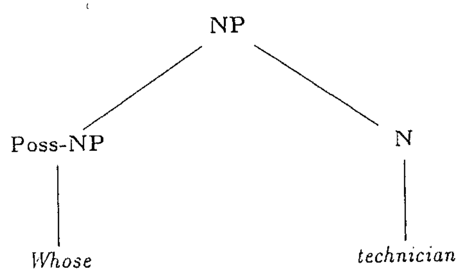


$$\left[\text{sem:}\boxed{6}\begin{bmatrix}\text{logic: } !\boxed{2}\wedge\{\text{possess}(\boxed{1},\boxed{2}),\boxed{4}\}\\ \text{var: }\boxed{2}\\ \text{que: }\left[\ ?\lambda\boxed{1}\,\text{person}(\boxed{1})\right]\end{bmatrix}\right] \qquad \left[\text{sem: }\begin{bmatrix}\text{logic: }\boxed{5}\,\text{tech}(\boxed{3})\\ \text{var: }\boxed{3}\end{bmatrix}\right]$$

Structure-Sharing

Possessive NP Semantics

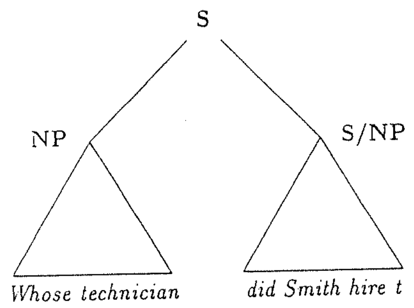
$$\left[\text{sem: }\begin{bmatrix}\text{logic: } !\boxed{2}\wedge\{\text{possess}(\boxed{1},\boxed{2}),\ \text{tech}(\boxed{2}\sqcup\boxed{3})\}\\ \text{var: }\boxed{2}\sqcup\boxed{3}\\ \text{que: }\left[\ ?\lambda\boxed{1}\,\text{person}(\boxed{1})\right]\end{bmatrix}\right]$$



$$\left[\text{sem:}\boxed{6}\begin{bmatrix}\text{logic: } !\boxed{2}\wedge\{\text{possess}(\boxed{1},\boxed{2}),\boxed{4}\}\\ \text{var: }\boxed{2}\\ \text{que: }\left[\ ?\lambda\boxed{1}\,\text{person}(\boxed{1})\right]\end{bmatrix}\right] \qquad \left[\text{sem: }\begin{bmatrix}\text{logic: }\boxed{5}\,\text{tech}(\boxed{3})\\ \text{var: }\boxed{3}\end{bmatrix}\right]$$

Linking Filler + Gappy Constituent

[sem: [logic: [Sort: Quantified-Wff, Quantifier: [4], Scope: [5]], slash: ∅]], where [2] = [3]



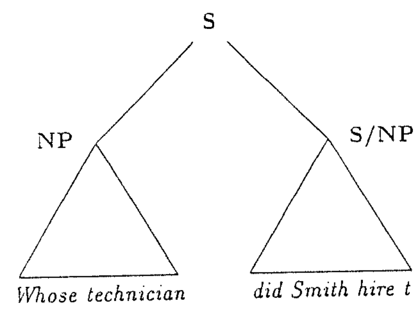
[sem: [logic: [4] ![2] ∧ { possess ([1],[2]), tech ([2])}, var: [2], que: [?λ[1] person([1])]]]

[sem: [logic: [5] hire (s , [3]), slash: [3]]]

Cf. exercise on semantics of quantified complements.

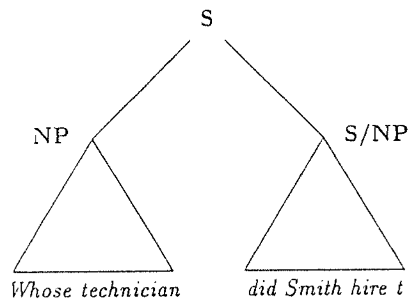
Results of Linking

[sem: [logic: ![2] ∧ { possess ([1],[2] ⊔ [3]), tech ([2] ⊔ [3])} hire (s , [2] ⊔ [3]), slash: ∅]]



Inheriting QUE

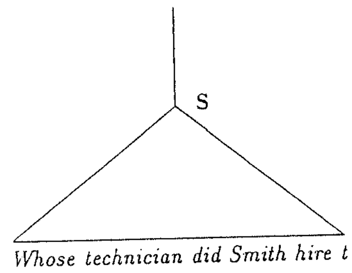
```
[ sem: [ logic: ![2] ∧ { possess([1],[2]),  tech([2]) }
                         hire(s, [2])
         slash: ∅
         que: [6] ] ]
```




```
[ sem: [ logic: ![2] ∧  { possess([1],[2]),
                          tech([2]) }
         var: [2]
         que: [6] [ ?λ[1] person([1]) ] ] ]
```


Retrieving QUE

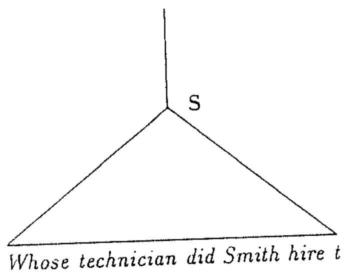
```
[ sem.logic: [ Sort: λ-Wff
               λ-Questioner: [3]
               Scope: [4] ] ]
```




```
[ sem: [ logic: [4] ![2] ∧ { possess([1],[2]),  tech([2]) }
                             hire(s, [2])
         que: [ [3] ?λ[1] person([1]) ] ] ]
```

Results

$$\left[\text{sem.logic: } ?\lambda\boxed{1}\ \text{person}(\boxed{1})\ \ !\boxed{2}\ \wedge \{\text{possess}(\boxed{1},\boxed{2}),\ \text{tech}(\boxed{2})\}\ \ \text{hire}(s,\boxed{2})\right]$$



$$?\lambda x\ \text{person}(x)\ \ !y\ \wedge\{\text{possess}(x,y),\text{tech}(y)\}\ \ \text{hire}(s,y)$$

Structure-Sharing

Long-Distance Dependencies involve structure-sharing in semantic representation.

- questions
  -Which part was reported to have been defective?
- relative clauses
  -Did Smith review the orders where parts had been reported as defective?
- comparatives
  -Which customers wanted more software than they had ordered the previous year?

$$\left[\begin{array}{ll} \text{Syntax:} & [\text{Sort: Syntax}] \\ \text{Semantics:} & \left[\begin{array}{l} \text{logic: } [\text{Sort: Logic-Expression}] \\ \text{var: } [\text{Sort: Var}] \\ \text{slash: } [\text{Sort: Var}] \\ \text{que: } [\text{Sort: } \lambda\text{-Questioner:}] \\ \text{rel: } [\text{Sort: Quantifier}] \\ \text{comparative-specifier: } [\text{Sort: Measure-Term}] \end{array}\right] \end{array}\right]$$

Unfication-Based Semantics

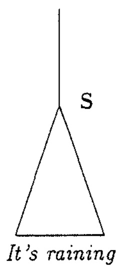
Compositionality Questions

- semantics (logic) is no longer a function purely of syntax (including semantics (logic) of component constituents) —Cf. Relational Theory of Language
- nonsyntactic dependencies also possible
  - phonology: stress, intonation
  - pragmatics: focus, theme
  - domain semantics
- underspecified forms may not correspond to semantic objects —scope
- selective use of semantic information possible (cf. section on structure sharing).

Compositionality Questions

Semantics-Phonology Interface: Intonation

$$\left[\text{sem: } \left[\text{logic: } \lambda p . \check{p} \leftrightarrow \boxed{2}\right]\right], \text{if } \boxed{1} = \uparrow$$



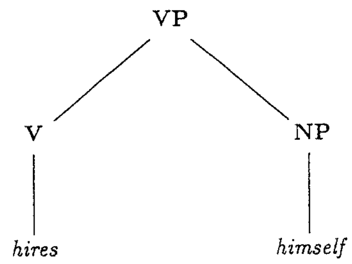
$$\left[\begin{array}{l}\text{phon.intonation: } \boxed{1} \\ \text{sem: } \left[\text{logic: } \boxed{2}\, \textbf{rain}(\,)\right]\end{array}\right]$$

- note use of conditonal in avm specification
- little work on semantics/phonology, but clear relevance
  - stress—contrastive, emphatic
  - duration, tempo

Compositionality Questions

Reflexives

- cf. Popowich, ACL '89 (also unification-based, otherwise dissimilar)



$$\begin{bmatrix} \text{stem: } \textit{hires} \\ \text{sem.logic:} \textbf{hire}(\boxed{1}, \boxed{2}) \\ \text{syn.subcat: } \begin{bmatrix} \text{subject: } [\text{sem.logic: } \boxed{1}] \\ \text{object: } [\text{sem.logic: } \boxed{2}] \end{bmatrix} \end{bmatrix} \qquad \begin{bmatrix} \text{sem: } \begin{bmatrix} \text{logic: } \boxed{3} \\ \text{refl: } \boxed{3} \end{bmatrix} \end{bmatrix}$$

(Simplified) Coindexing Condition (Pollard & Sag, '83):

—reflexives coindexed with some less oblique element governed by head

$$r \in \text{comp.refl} \;\Rightarrow\; \exists c \in \text{syn.subcat}\ (\text{c.sem} = \text{r.sem})$$

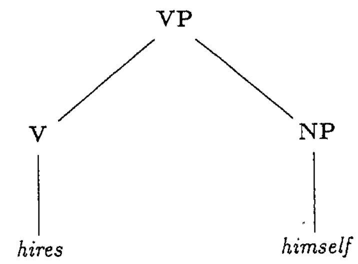
—expressible via "regular path" expressions (Kaplan & Maxwell, '87). We represent lists as: $\begin{bmatrix} \text{first:} \\ \text{rest:} \end{bmatrix}$. Then:

$$r = \text{comp.refl.}\{\text{rest}\}^*\text{.first} \;\Rightarrow\; \text{syn.subcat.}\{\text{rest}\}^*\text{.first.sem} = \text{r.sem})$$

Results of Reflexive Processing

- $\boxed{2} = \boxed{3}$ (complementation)
- $\boxed{1} = \boxed{3}$ (reflexive binding)

$$\begin{bmatrix} \text{sem.logic:} \textbf{hire}(\boxed{1} \sqcup \boxed{2} \sqcup \boxed{3}, \boxed{1} \sqcup \boxed{2} \sqcup \boxed{3}) \\ \text{syn.subcat: } [\text{subject: } [\text{sem.logic: } \boxed{1} \sqcup \boxed{2} \sqcup \boxed{3}]] \end{bmatrix}$$



$$\begin{bmatrix} \text{stem: } \textit{hires} \\ \text{sem.logic:} \textbf{hire}(\boxed{1}, \boxed{2}) \\ \text{syn.subcat: } \begin{bmatrix} \text{subject: } [\text{sem.logic: } \boxed{1}] \\ \text{object: } [\text{sem.logic: } \boxed{2}] \end{bmatrix} \end{bmatrix} \qquad \begin{bmatrix} \text{sem: } \begin{bmatrix} \text{logic: } \boxed{3} \\ \text{refl: } \boxed{3} \end{bmatrix} \end{bmatrix}$$

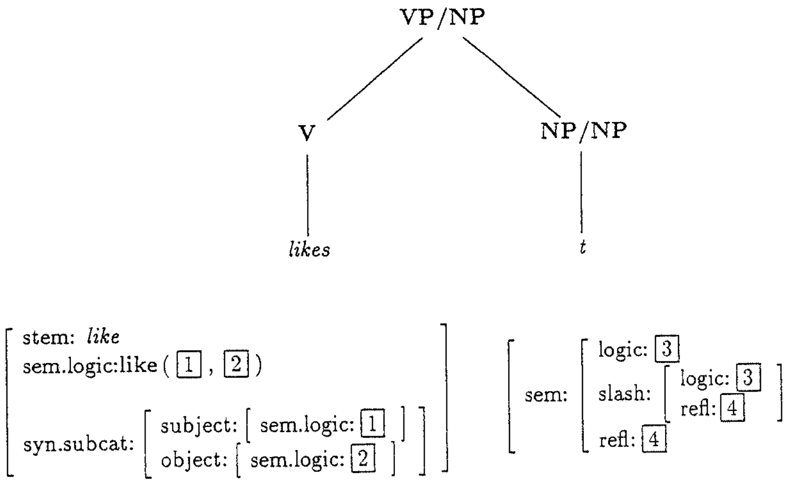
Incorporation of Subject must result in reflexive meaning.

Compositionality Questions

Apparent Exceptions to Reflexive-Binding Condition (Engdahl):

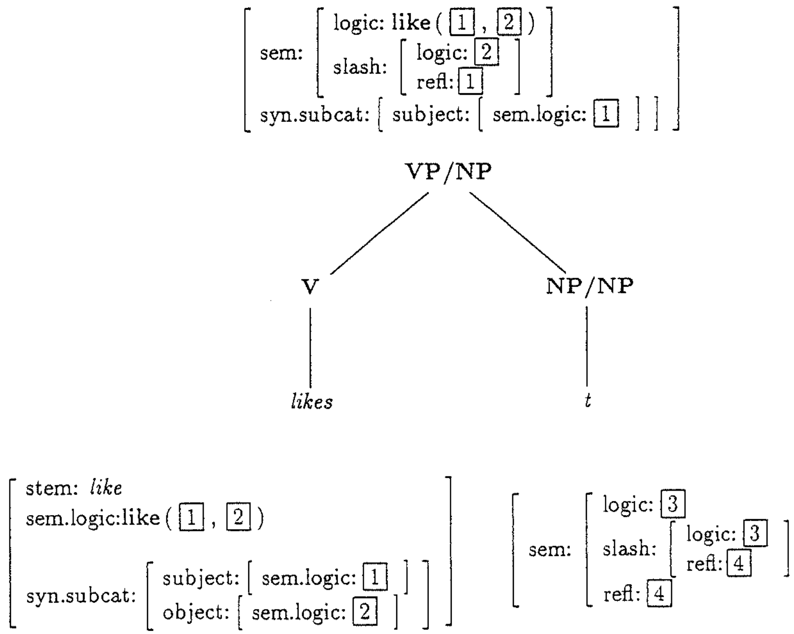
- Which picture of herself did every competitor want?
- Himself, Sam likes.

Insight: Slash may contain Reflexives



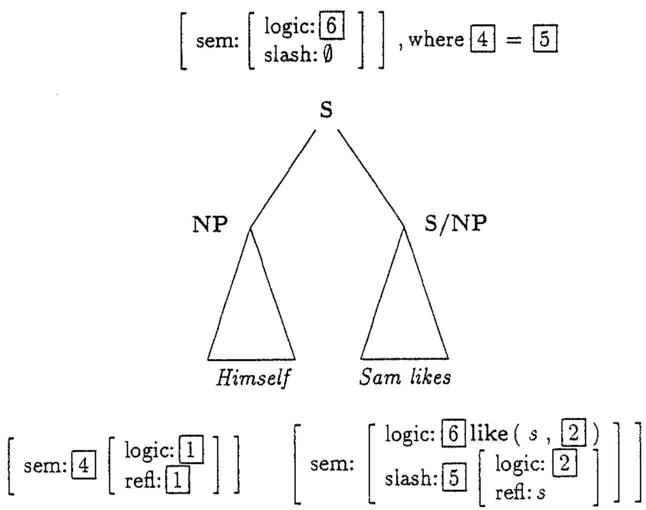
Normal Object Incorporation

- [2] = [3] (complementation)
- [1] = [4] (reflexive binding)



Since slash must unify with filler, any reflexives contained in filler must unify with subject.

Linking Filler + Gappy Constituent



$\boxed{4} = \boxed{5} \;\Rightarrow\; \boxed{2} = s.$

Thus, semantics of S = like (s , s)

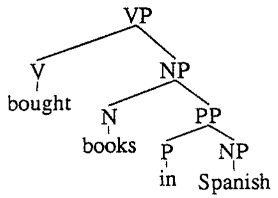
Compositionality: Semantics of phrase is function of semantics of constituents. —Weakened here.

Compositionality Questions

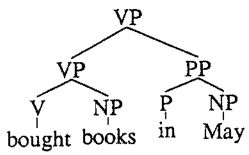
Sortal Disambiguation via Unification

- Moens, *et al.*, E-ACL '89

Who bought books in Spanish?



Who bought books in May?



Sortal Disambiguation via Unification

Strategy:

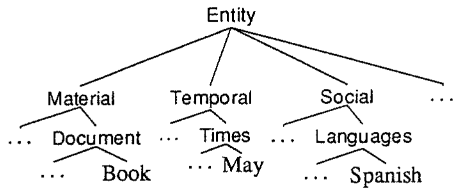
- Specify sorts of terms (ind. constants, var.)
- Specify sorts required in argument positions of relatons and functions
- Enforce compatibility

Advantage of Unification-Based Approach:

- Integration of compatibility enforcement

Sortal Disambiguation via Unification

Domain Sorts



N-63

Sortal Disambiguation via Unification

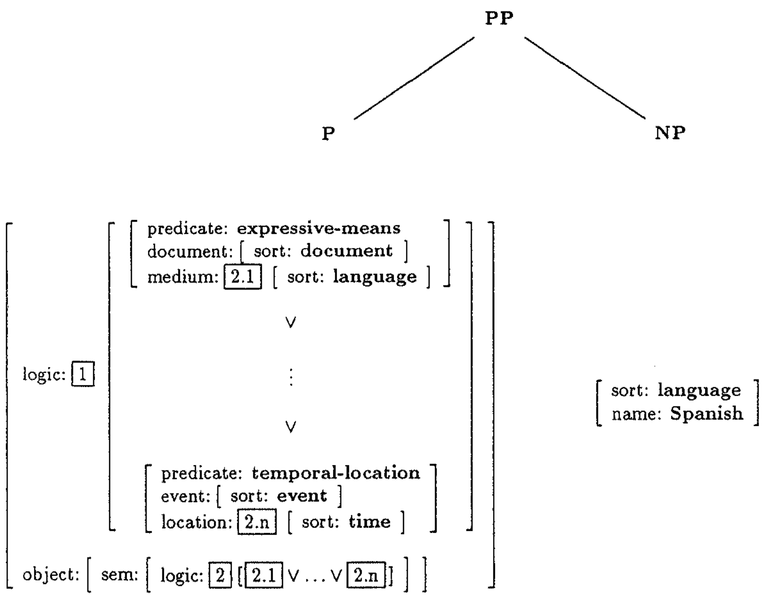
A Sortally Ambiguous ‘in’

$$\left[\text{logic:}\left[\begin{array}{c}\left[\begin{array}{l}\text{predicate: }\textbf{expressive-means}\\ \text{document: }\left[\text{ sort: }\textbf{document}\right]\\ \text{medium: }\left[\text{ sort: }\textbf{language}\right]\end{array}\right]\\ \vee\\ \vdots\\ \vee\\ \left[\begin{array}{l}\text{predicate: }\textbf{temporal-location}\\ \text{event: }\left[\text{ sort: }\textbf{event}\right]\\ \text{location: }\left[\text{ sort: }\textbf{time}\right]\end{array}\right]\end{array}\right]\right]$$

N-64

Sortal Disambiguation via Unification

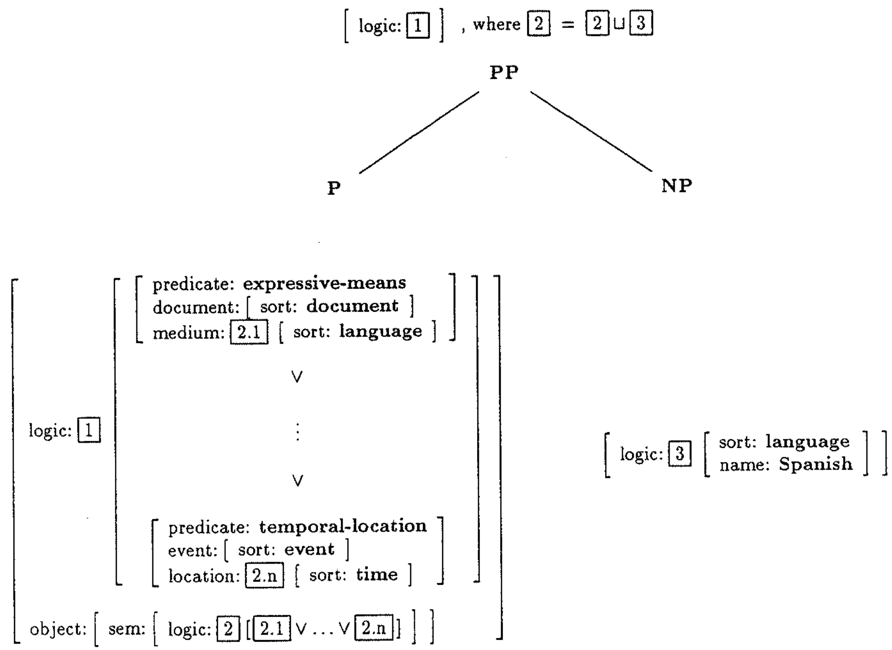
Preposition plus Prepositional Object



N-65

Sortal Disambiguation via Unification

Preposition plus Prepositional Object



N-66

Sortal Disambiguation via Unification

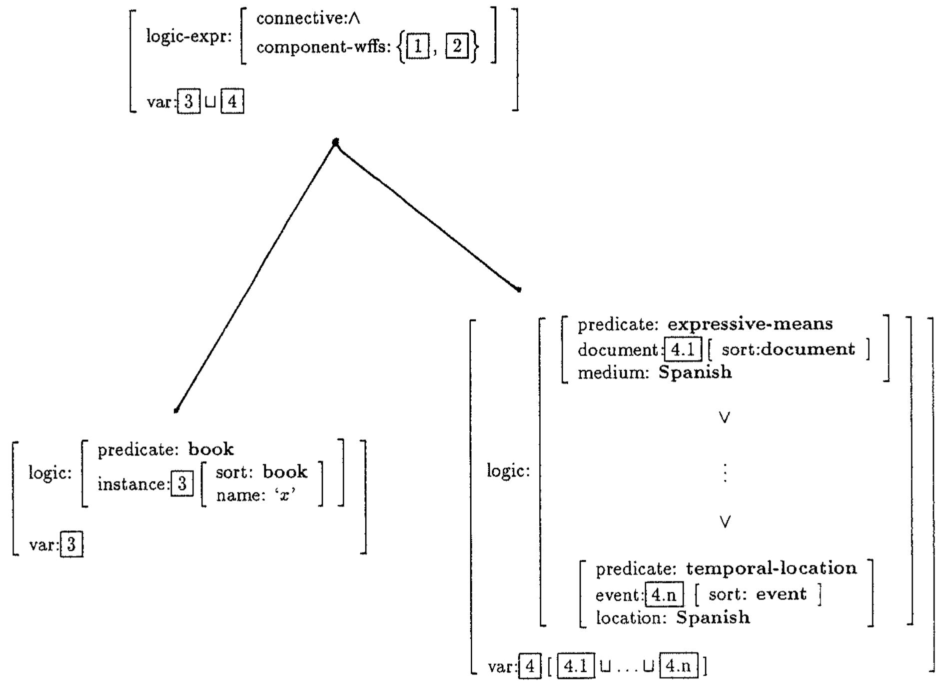
Nominal Semantics with Domain Sort Specifications

$$\begin{bmatrix} \text{predicate: book} \\ \text{instance:} \begin{bmatrix} \text{sort: book} \\ \text{name: } `x\text{'} \end{bmatrix} \end{bmatrix}$$

N-67

Sortal Disambiguation via Unification

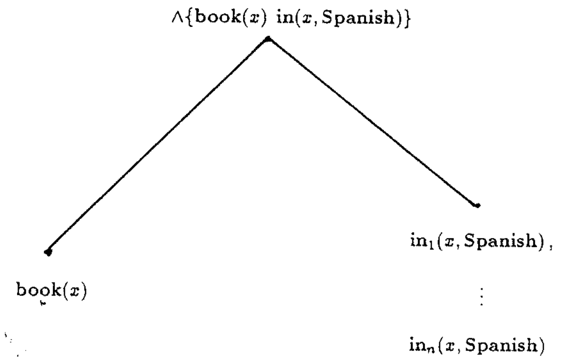
Nominal Adjunct Semantics



N-68

Sortal Disambiguation via Unification

Object Language Results



N-69

Sortal Disambiguation

Cautions

- Danger: Overworking ‘Sort’
- Pitfall: Confusing Well-Formed and Correct in Entailment
  A book can’t be in May the same way it’s in Spanish.

N-70

Enabling Features of Constraint-Based Approaches

- Partiality
  - Models of partial information (situation semantics)
  - Partial logics

- New means of composition
  - Unification of partially specified objects

- Interpretation by successive approximation
  - Monotonicity in the description space
  - Non-monotonic effects in the model space

A Model of Interpretation in Unification Grammars

- Multiple levels of semantic significance
- Constituent structure, phonological structure, functional structure....
- Discourse context
- Background information
- Physical and pragmatic constraints on the described situation

- Informational dependencies expressed through constraints:

$S \rightarrow$	NP	VP	ADVP
	$(\uparrow \text{SUBJ})=\downarrow$	$\phi\mathcal{M}*=\phi*$ $(\sigma\mathcal{M}*\ \text{PRED})=\sigma*$	$\phi* \in (\phi\mathcal{M}*\ \text{ADJUNCTS})$ $(\sigma\mathcal{M}*\ \text{MOD})=\sigma*$

- Interpretation through translation

Description of Logical Forms vs. Construction of Semantic Objects

- Prerequisites for direct interpretation
  - Established correspondence between semantic operations (in the model) and syntactic operations (in the syntax and/or in the logic)
    - MG Homomorphisms:
      Syntax ⇔ Logical Form ⇔ Model
    - Prolog
- Such correspondences are established between unification and restricted subsets of FOL, but most linguistic semantic descriptions go beyond these limits (Montague's intensional logic; situation semantics)
- New analytic possibilities arise from working with descriptions of logical forms as opposed to descriptions of semantic objects

Functional Uncertainty: Concept and Notation

- Functional uncertainty permits the characterization of classes of paths through an AVG, *f*, meeting the conditions of a regular expression, α, e.g. $(f\alpha)$.

- Example: $(f\ (P\ Q)^{+})$

$$f\begin{bmatrix} P & \begin{bmatrix} Q & \begin{bmatrix} P & \begin{bmatrix} P & A \\ Q & B \\ Z & C \end{bmatrix} \\ Q & E \\ Z & F \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Functional Uncertainty: Applications

- Long-distance syntactic (functional) dependencies

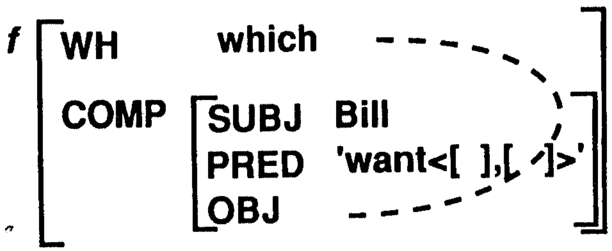
  The book which Bill wants

  The book which John thinks Bill wants

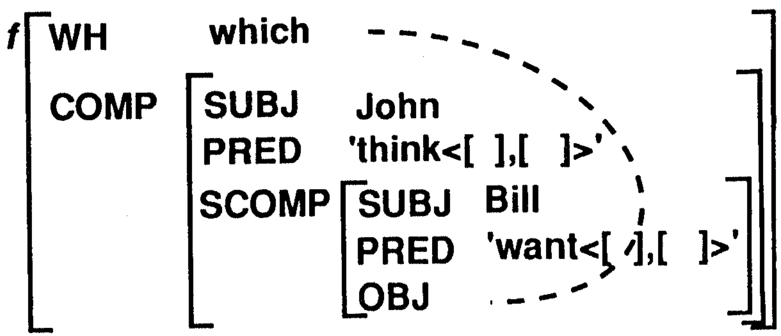
  The book which Vera said John thought Bill wanted

- Path: f OBJ, f SCOMP OBJ, f SCOMP SCOMP OBJ

The book which Bill wants

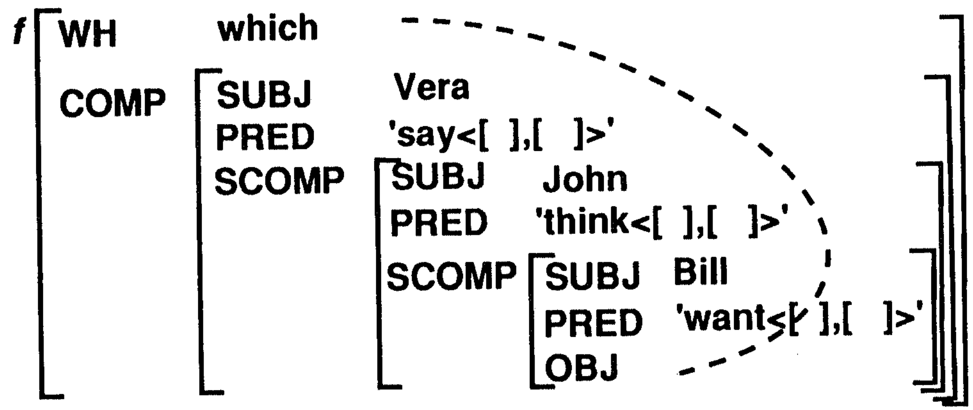


The book which John thinks Bill wants



Functional Uncertainty: Relative Clauses

The book which Vera said John thought Bill wanted



- **Regular characterization:**
  (f COMP SCOMP* OBJ)

  (f COMP SCOMP* GF)

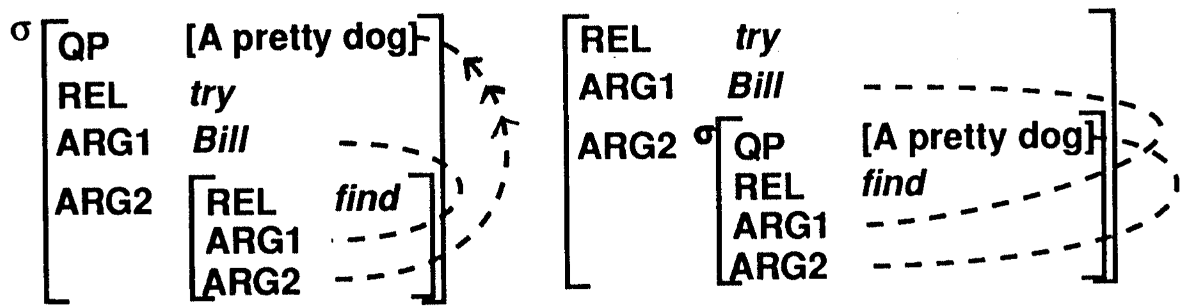
Quantification

- Quantifiers can take scope over any complement (simplified assumption)

  Bill is trying to find a pretty dog

  ∃x[Bill is trying [Bill find a pretty dog(x)]] (specific)

  Bill is trying ∃x[Bill find a pretty dog(x)]] (non-specific)

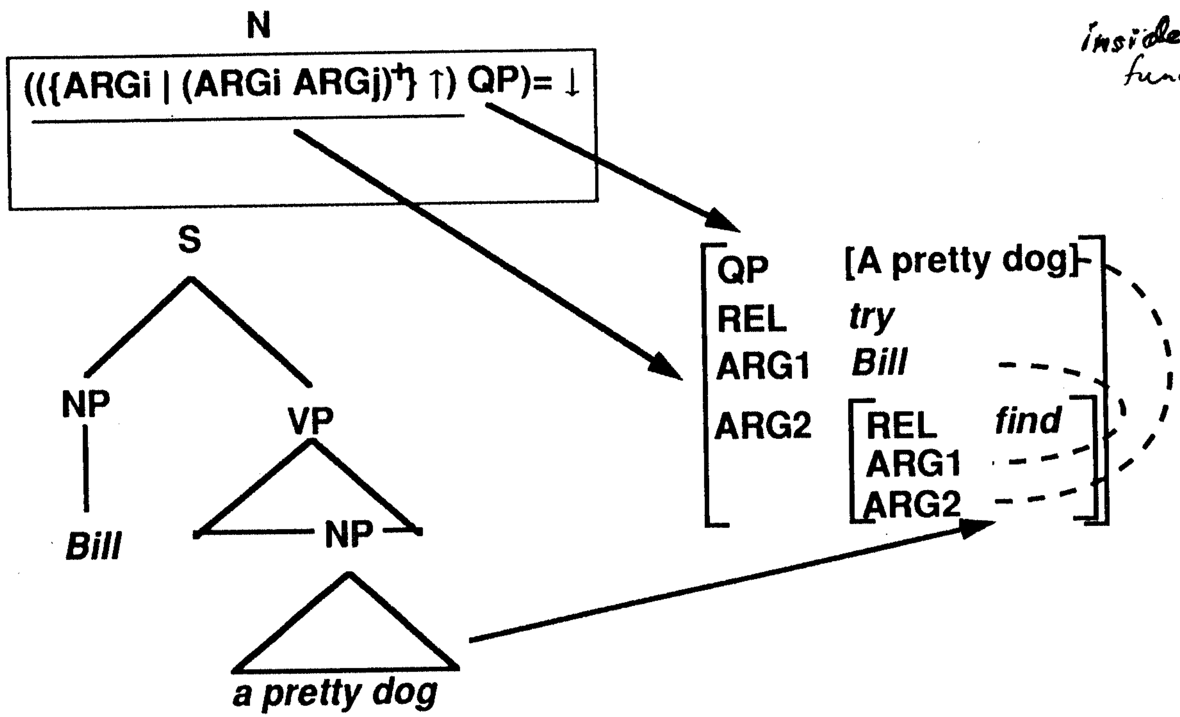


- Path: σ ARG1 ARG2 ; σ ARG1

  Generalization: σ {ARGi | (ARGi ARGj)$^+$ }

Quantification: Simplified Analysis

- All rules introducing quantified Ns receives an annotation:

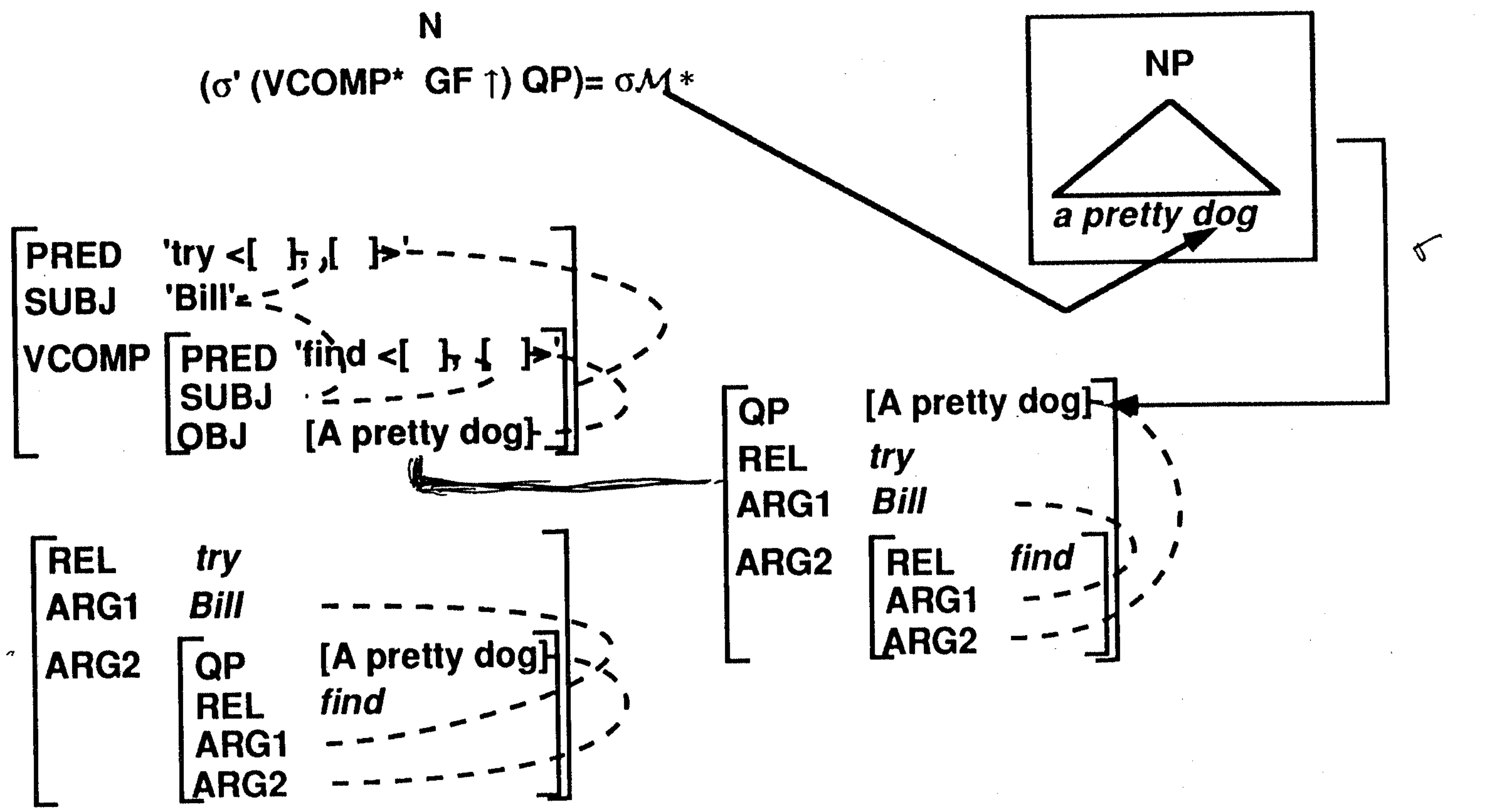


from value to attribute?
– no, path is rooted in unknown higher node.

inside out –
functional uncertainty

Quantification: Analysis with projections

- All rules introducing quantified Ns receives an annotation:



Conclusions

- Unification and unification grammars provide:

  New mechanisms for combining semantic information

  New ways of correlating linguistic form and its interpretation

  New analytic possibilities due new descriptive apparatus or extensions to the notion of unification

- New and unresolved issues arise regarding

  The significance of compositionality

  The semantic correlates of syntactic operations on semantic descriptions