# Model-theoretic syntax

Christopher Potts

Department of Linguistics, UMass Amherst

May 4, 2004
Guest lecture, CMPSCI 585,
Spring 2004 (Andrew McCallum)

# The plan

i. Have a quick look at what the goals of linguistics are and how linguists work.

ii. Contrast two ways of theorizing: one based in proof theory (formal languages) and the other based in model theory.

iii. Try to motivate a model-theoretic perspective.

## The plan

i. Have a quick look at what the goals of linguistics are and how linguists work.
ii. Contrast two ways of theorizing: one based in proof theory (formal languages) and the other based in model theory.
iii. Try to motivate a model-theoretic perspective.

My proof-theoretic example is a context-free grammar. It is non-probabilistic, because this is the dominant mode in linguistics right now (though the current is shifting).

## The plan

i. Have a quick look at what the goals of linguistics are and how linguists work.

ii. Contrast two ways of theorizing: one based in proof theory (formal languages) and the other based in model theory.

iii. Try to motivate a model-theoretic perspective.

My proof-theoretic example is a context-free grammar. It is non-probabilistic, because this is the dominant mode in linguistics right now (though the current is shifting).

**What I would like from you** I am curious to learn the extent to which my criticisms of the proof-theoretic approaches hold for the kinds of grammars that you have looked at in this class.

## Pullum and Scholz and others

The ideas in these slides are mainly extracted from the work of Geoffrey K. Pullum and Barbara Scholz. No one has done more than Pullum and Scholz to clarify the issues surrounding model-theoretic syntax and its competitors.

If you are not convinced by some or all of my arguments, write to Pullum and Scholz for copies of their papers and manuscripts before passing a final judgment on the enterprise.

The specific grammars that I offer below are inspired by the work of Patrick Blackburn and his coauthors (cited in the references).

# A note on linguistics–CS links

Syntax is probably the area of theoretical linguistics that has the *fewest* connections with theoretical computer science.

- I mean this in the sense of sharing innovations, not foundational assumptions.

Results and techniques from theoretical computer science have often been incorporated into proposals in semantics and phonology.

# Linguistic syntacticians at work

# Linguistic syntacticians at work

- **The enterprise** Broadly speaking, natural language syntacticians work to find precise, concise specifications of natural languages.

# Linguistic syntacticians at work

- **The enterprise** Broadly speaking, natural language syntacticians work to find precise, concise specifications of natural languages.
- **Universal Grammar** When we compare these language-specific descriptions, we should find a lot of overlap — recurrent patterns, similar categories, common general constraints.

## Linguistic syntacticians at work

- **The enterprise** Broadly speaking, natural language syntacticians work to find precise, concise specifications of natural languages.
- **Universal Grammar** When we compare these language-specific descriptions, we should find a lot of overlap — recurrent patterns, similar categories, common general constraints.
- **Complexity in the features, not the principles** At present, most people assume that grammars should take the form of a small number of cross-linguistically valid principles plus a large number of stipulations about the lexicon.

# Linguistic syntacticians at work

**For example** Baker (1988, 1996) describes Mohawk in great depth. Mohawk is a strikingly agglutinative language. A single complex verb form can do the work of a full English sentence, and discontinuous constituents are common:

(1) Ka-nuhs-rakv [nehneh a-ak-ahninuʔ].
3N-**house**-white [that INDEF-3F-buy].
'House-white that she would buy.'
('That house that she would buy is white.')

Baker shows that familiar principles are at work here. The rich class of bound morphemes conspires (along with some variation in the general principles) to produce outputs that differ greatly from, e.g., English.

# A bit of common ground

# A bit of common ground

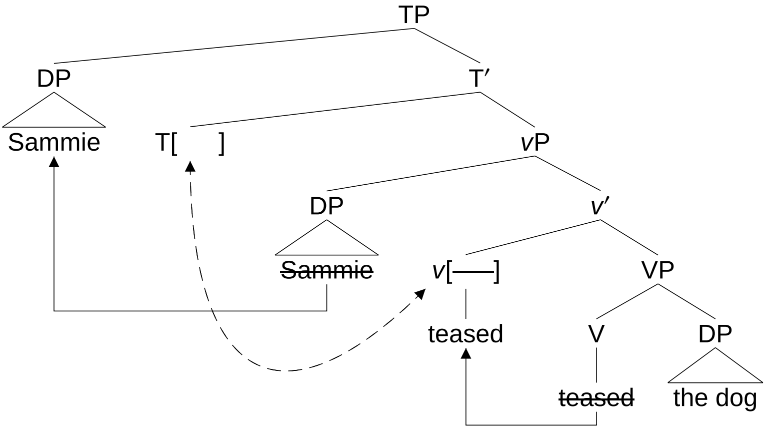
- The proper nature of syntactic theory is widely debated. Nearly everything is controversial. (This is less true for natural language semantics and phonology.)

# A bit of common ground

- The proper nature of syntactic theory is widely debated. Nearly everything is controversial. (This is less true for natural language semantics and phonology.)
- One commonality: **Everyone is talking about relational structures**. In general, these are singly-rooted, acyclic (multi-)graphs.

# The Minimalist Program

(2) Sammie teased the dog. [Simplified analysis!]



# Head-Driven Phrase Structure Grammar (HPSG)

(2) Sammie teased the dog. [Simplified analysis!]

⟨sami, tizd, ðə, dɔg⟩
S[*fin*]

- ⟨tizd, ðə, dɔg⟩
  VP[*fin*]
  - ⟨ðə, dɔg⟩
    NP[*acc*]
  - ⟨tizd⟩
    V[*fin*]
- ⟨sami⟩
  NP[*nom*]

## Categorial Grammar

(2) Sammie teased the dog. [Simplified analysis!]

$$\dfrac{\mathrm{NP} \qquad \dfrac{(\mathrm{S}\backslash\mathrm{NP})/\mathrm{NP} \qquad \dfrac{\mathrm{NP}/\mathrm{N} \qquad \mathrm{N}}{\mathrm{NP}}}{\mathrm{S}\backslash\mathrm{NP}}}{\mathrm{S}}$$

# Categorial Grammar

(2) Sammie teased the dog. [Simplified analysis!]

S
**teased**(**the**(**dog**))(**sammie**)

---

NP — **sammie**

S\NP — **teased**(**the**(**dog**))

---

(S\NP)/NP — **teased**

NP — **the**(**dog**)

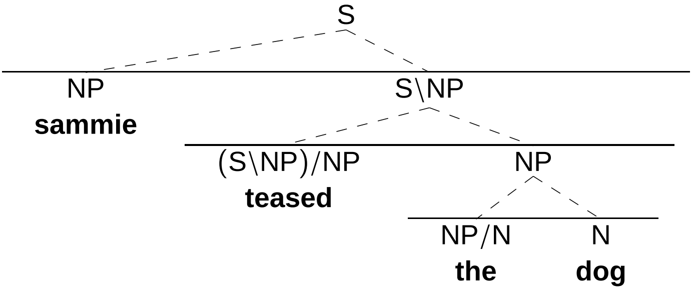
---

NP/N — **the**

N — **dog**

# Categorial Grammar

(2) Sammie teased the dog. [Simplified analysis!]



# Three modes of theorizing

## Proof-theoretic

# Three modes of theorizing

## Proof-theoretic

- Chomsky's (1964, 1965) original approach, heavily influenced by the nascent field of computer science and machine translation

# Three modes of theorizing

## Proof-theoretic

- Chomsky's (1964, 1965) original approach, heavily influenced by the nascent field of computer science and machine translation
- A grammar for a language *L* is a recursive specification of the full set of well-formed objects for *L*.

# Three modes of theorizing

## Proof-theoretic

- Chomsky's (1964, 1965) original approach, heavily influenced by the nascent field of computer science and machine translation
- A grammar for a language $L$ is a recursive specification of the full set of well-formed objects for $L$.
- A sentence $S$ is grammatical according to a grammar $G$ iff $S$ is derivable in $G$.

# Three modes of theorizing

## Proof-theoretic: Examples

i. all the phrase-structure grammars in the Chomsky Hierarchy
ii. categorial grammars, which are generally context-free equivalent
iii. tree-adjoining grammars (intermediate between context-free and context-sensitive on the Chomsky Hierarchy)

# Three modes of theorizing

## Model-theoretic

Syntacticians are talking about structures. Let's reason directly in terms of them.

# Three modes of theorizing

## Model-theoretic

Syntacticians are talking about structures. Let's reason directly in terms of them.

- The linguist proposes a set of logical conditions $\gamma_1, \ldots, \gamma_n$ based on investigation of a language $L$.

# Three modes of theorizing

## Model-theoretic

Syntacticians are talking about structures. Let's reason directly in terms of them.

- The linguist proposes a set of logical conditions $\gamma_1, \ldots, \gamma_n$ based on investigation of a language $L$.
- In the simplest case, the grammar for $L$ could be their logical conjunction: $\mathcal{G} = \gamma_1 \wedge \cdots \wedge \gamma_n$.

# Three modes of theorizing

## Model-theoretic

Syntacticians are talking about structures. Let's reason directly in terms of them.

- The linguist proposes a set of logical conditions $\gamma_1, \ldots, \gamma_n$ based on investigation of a language $L$.
- In the simplest case, the grammar for $L$ could be their logical conjunction: $\mathcal{G} = \gamma_1 \wedge \cdots \wedge \gamma_n$.
- A sentence $S$ is grammatical in language $L$ iff $\mathcal{S}$, the structure for $S$, is such that

$$\mathcal{S} \models \mathcal{G}$$

  That is, $S$ is grammatical iff $\mathcal{S}$ is a model of $\mathcal{G}$.

# Three modes of theorizing

## Model-theoretic: Examples

i. Arc-Pair Grammar (Johnson and Postal 1981)
ii. The earliest versions of Lexical-Functional Grammar (Kaplan and Bresnan 1982)
iii. Rogers (1998) (re)conceptualization of Government and Binding theory

# Three modes of theorizing

## Hybrids

The two modes can coexist and interact, just as they do in logic.

Hybrids of the proof-theoretic and model-theoretic approaches have an underlying phrase-structure system that generates lots of garbage that the constraints filter off.

i. HPSG

ii. the constraint-based Chomskian theories of the 1980s

# A context-free grammar (CFG)

i. $G = \langle T, N, S, R \rangle$

ii. $T = \left\{ \textit{Sally}, \textit{believes}, \textit{pigs}, \textit{fly} \right\}$

iii. $N = \left\{ S, NP, VP, V \right\}$

iv. Start symbol $= S$

v. $$R = \left\{ \begin{array}{rcl} S & \Rightarrow & NP\ VP \\ VP & \Rightarrow & V\ S \\ VP & \Rightarrow & V \\ NP & \Rightarrow & \textit{Sally} \mid \textit{pigs} \\ V & \Rightarrow & \textit{believes} \mid \textit{fly} \end{array} \right\}$$

## A context-free grammar (CFG) and some derivations

i. $G = \langle T, N, S, R \rangle$

ii. $T = \{\textit{Sally}, \textit{believes}, \textit{pigs}, \textit{fly}\}$

iii. $N = \{S, NP, VP, V\}$

iv. Start symbol $= S$

v. $$R = \left\{ \begin{array}{rcl} S & \Rightarrow & NP\ VP \\ VP & \Rightarrow & V\ S \\ VP & \Rightarrow & V \\ NP & \Rightarrow & \textit{Sally} \mid \textit{pigs} \\ V & \Rightarrow & \textit{believes} \mid \textit{fly} \end{array} \right\}$$

$$\begin{array}{cc} \multicolumn{2}{c}{S} \\ NP & VP \\ \textit{pigs} & VP \\ \textit{pigs} & V \\ \textit{pigs} & \textit{fly} \end{array} \qquad \begin{array}{cc} \multicolumn{2}{c}{S} \\ NP & VP \\ NP & V \\ NP & \textit{fly} \\ \textit{pigs} & \textit{fly} \end{array} \qquad \ldots$$

# A context-free grammar (CFG) and some derivations

i. $G = \langle T, N, S, R \rangle$

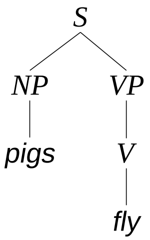
ii. $T = \{\textit{Sally}, \textit{believes}, \textit{pigs}, \textit{fly}\}$

iii. $N = \{S, NP, VP, V\}$

iv. Start symbol $= S$

v. $R = \left\{ \begin{array}{rcl} S & \Rightarrow & NP\ VP \\ VP & \Rightarrow & V\ S \\ VP & \Rightarrow & V \\ NP & \Rightarrow & \textit{Sally} \mid \textit{pigs} \\ V & \Rightarrow & \textit{believes} \mid \textit{fly} \end{array} \right\}$

| | | | | |
|---|---|---|---|---|
| $S$ | | $S$ | | |
| $NP$ | $VP$ | $NP$ | $VP$ | |
| *pigs* | $VP$ | $NP$ | $V$ | ... |
| *pigs* | $V$ | $NP$ | *fly* | |
| *pigs* | *fly* | *pigs* | *fly* | |



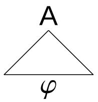
# James McCawley's insight

The late, truly great linguist James McCawley had the following insight:

## James McCawley's insight

The late, truly great linguist James McCawley had the following insight:

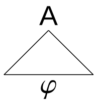
We needn't view $A \Rightarrow \varphi$ as a production. We can view it as the specification of a local tree:




## James McCawley's insight

The late, truly great linguist James McCawley had the following insight:

We needn't view $A \Rightarrow \varphi$ as a production. We can view it as the specification of a local tree:




This view (re)emerged (also) in logicians' study of proofs as model-theoretic objects.

# A model-theoretic framework

## Ingredients

i. Trees (the models)
ii. A logic (for defining grammars)
iii. An interpretation scheme (so that we can use the logic to talk about the trees)

## Trees (our models)

A *tree* $\mathcal{T}$ is a relational structure $(N, D, r, V)$ where:

i. $N$ is a set of atomic points called nodes.
ii. $D$ is a binary relation on $T$ (the dominance relation).
iii. $r \in N$ is a distinguished *root* node: $\forall n \in N, \langle r, n \rangle \in D^*$.
iv. $D^*$ is acyclic.
v. Every $n \in N$ that is not the root has exactly one mother.
vi. $V$ is a valuation (labelling) function, taking formulae of our logic to subsets of $N$.

## A logic (our tool for grammar writing)

The logic $\mathcal{L}$ is defined as follows:

i. $T$ is a set of atomic propositions (terminal labels). If $\varphi \in T$, then $\varphi$ is a WFF.

ii. $N$ is a set of atomic propositions (nonterminal labels). If $\varphi \in N$, then $\varphi$ is a WFF.

iii. If $\varphi$ and $\psi$ are WFFs, then $\neg\varphi$ and $\varphi \rightarrow \psi$ are WFFs.

iv. If $\varphi$ and $\psi$ are WFFs, then $\triangle(\varphi)$ and $\triangle(\varphi\ \psi)$ are WFFs.

v. Nothing else is a WFF.

## Interpretation

$\mathcal{T}, n \models \varphi$ iff $n \in V(\varphi)$ for $\varphi \in (T \cup N)$

$\mathcal{T}, n \models \neg\varphi$ iff $\mathcal{T}, n \not\models \varphi$

$\mathcal{T}, n \models \varphi \rightarrow \psi$ iff $\mathcal{T}, n \not\models \varphi$ or $\mathcal{T}, n \models \psi$

$\mathcal{T}, n \models \triangle(\varphi)$ iff $n$ has exactly one daughter $n'$ and $\mathcal{T}, n' \models \varphi$

$\mathcal{T}, n \models \triangle(\varphi\, \psi)$ iff $n$ has exactly two daughters, $n'$ and $n''$, and $n'$ is to the left of $n''$ and $\mathcal{T}, n' \models \varphi$ and $\mathcal{T}, n'' \models \psi$

**Validity** Where $\varphi$ is true at all points in $\mathcal{T}$, we write $\mathcal{T} \models \varphi$.

# Defining the CFG in $\mathcal{L}$

The grammar $\mathcal{G}$:

i. $T = \{\mathit{Sally}, \mathit{believes}, \mathit{pigs}, \mathit{fly}\}$

ii. $N = \{S, NP, VP, V\}$

iii. $r \in V(S)$

iv. $\mathcal{G} =$

$$\left\{\begin{array}{l} S \rightarrow \triangle(NP\ VP) \\ VP \rightarrow \big(\triangle(V\ S) \vee \triangle(V)\big) \\ NP \rightarrow \big(\triangle(\mathit{Sally}) \vee \triangle(\mathit{pigs})\big) \\ V \rightarrow \big(\triangle(\mathit{believes}) \vee \triangle(\mathit{fly})\big) \end{array}\right\}$$

# Defining the CFG in $\mathcal{L}$

The grammar $\mathcal{G}$:

i. $T = \{\textit{Sally}, \textit{believes}, \textit{pigs}, \textit{fly}\}$

ii. $N = \{S, NP, VP, V\}$

iii. $r \in V(S)$

iv. $\mathcal{G} =$

$$\left\{ \begin{array}{l} \varphi \rightarrow \neg\Big(\triangle(\top) \vee \triangle(\top\ \top)\Big) \text{ for } \varphi \in T \\ \varphi \rightarrow \triangle(\top) \vee \triangle(\top\ \top) \text{ for } \varphi \in N \\ S \rightarrow \triangle(NP\ VP) \\ VP \rightarrow \Big(\triangle(V\ S) \vee \triangle(V)\Big) \\ NP \rightarrow \Big(\triangle(\textit{Sally}) \vee \triangle(\textit{pigs})\Big) \\ V \rightarrow \Big(\triangle(\textit{believes}) \vee \triangle(\textit{fly})\Big) \end{array} \right\}$$

# Defining the CFG in $\mathcal{L}$

The grammar $\mathcal{G}$:

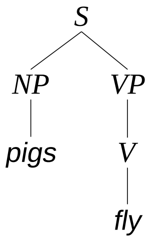
i. $T = \{\textit{Sally}, \textit{believes}, \textit{pigs}, \textit{fly}\}$

ii. $N = \{S, NP, VP, V\}$

iii. $r \in V(S)$

iv. $\mathcal{G} =$

$$\left\{ \begin{array}{l} S \rightarrow \triangle(NP\ VP) \\ VP \rightarrow \left(\triangle(V\ S) \vee \triangle(V)\right) \\ NP \rightarrow \left(\triangle(\textit{Sally}) \vee \triangle(\textit{pigs})\right) \\ V \rightarrow \left(\triangle(\textit{believes}) \vee \triangle(\textit{fly})\right) \end{array} \right\}$$




$\models \mathcal{G}$

# A translation scheme

The mapping from our CFG to $\mathcal{L}$ is transparent:

$$\begin{array}{rcl} A \Rightarrow B & \text{becomes} & A \rightarrow \triangle(B) \\ A \Rightarrow B\ C & \text{becomes} & A \rightarrow \triangle(B\ C) \\ A \mid B & \text{becomes} & A \vee B \end{array}$$

But the $\mathcal{L}$ treatment does not conflate the syntax of syntax (the proofs) with the semantics of syntax (the trees, or whatever relational structures we want to talk about).

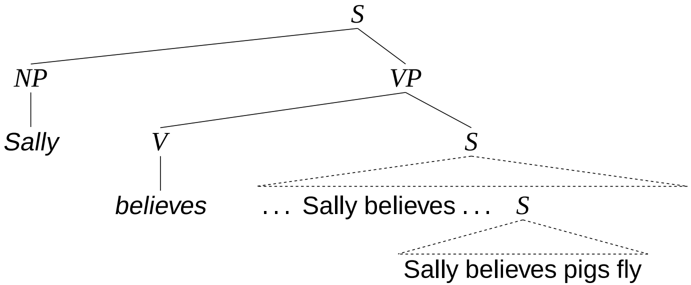
So we should look to the models to find differences.

# Phenomena that might favor a model-theoretic approach

i. Language size (is not a linguistic issue)
ii. Degrees of ungrammaticality (are an undeniable fact of language)
iii. Grammatical quandaries (when *nothing* works)
iv. Fragments

# Language size

Consider this **infinitely sized** tree:




- This cannot be generated by any CFG.
- It can be a model of $\mathcal{L}$.

# Language size

- It is taken for granted in theoretical linguistics that every natural language is infinite.

# Language size

- It is taken for granted in theoretical linguistics that every natural language is infinite.
- **But** Linguists determine whether something is grammatical based on speakers' intuitions.

## Language size

- It is taken for granted in theoretical linguistics that every natural language is infinite.
- **But** Linguists determine whether something is grammatical based on speakers' intuitions.
- Language size is not something that speakers can have intuitions about.

# Language size

- It is taken for granted in theoretical linguistics that every natural language is infinite.
- **But** Linguists determine whether something is grammatical based on speakers' intuitions.
- Language size is not something that speakers can have intuitions about.
- Thus, the question of language size is different in kind from the usual questions in linguistics.

# Language size

**The proof-theoretic view gives a specific answer:**

Languages are countably infinite.

# Language size

**The proof-theoretic view gives a specific answer:**

Languages are countably infinite.

**Langendoen and Postal object:**

There are English sentences of infinite size. Therefore, the class of all sentences is nondenumerably infinite.

Langendoen and Postal's argument is based on the notion that limiting sentences to finite size is stipulative.

# Language size

## Model-theoretic views can remain silent on the matter

The bulk of the work for a model-theoretic syntactician involves finding constraints for the grammar. This work can be done independently of a specific class of models. One might know that one wants tree models, but one need not specify whether they are finite models.

The model-theoretic syntactician can work as easily with finite corpora as with infinite classes. She can give a concise, precise description of exactly the class in question. No stipulations about language size appear in the grammar or as a consequence of the formalism. (Leave this question to the philosophers.)

# Degrees of ungrammaticality

The examples in (3) grow increasingly bad from top to bottom:

(3) a. *They probably did wanted you to leave.
b. **They probably does wanted you to leave.
c. $*^{500}$ Leave to you wanted probably they.

## Degrees of ungrammaticality

The examples in (3) grow increasingly bad from top to bottom:

(3) a. $^{*}$They probably did wanted you to leave.
b. $^{**}$They probably does wanted you to leave.
c. $^{*^{500}}$Leave to you wanted probably they.

No proof-theoretic grammar can capture gradient ungrammaticality, at least not without extra-special machinery. The notion *generates* is not gradable; an object is either in the generated set or it is not.

## Degrees of ungrammaticality

The examples in (3) grow increasingly bad from top to bottom:

(3) a. $^{*}$They probably did wanted you to leave.
b. $^{**}$They probably does wanted you to leave.
c. $*^{500}$ Leave to you wanted probably they.

No proof-theoretic grammar can capture gradient ungrammaticality, at least not without extra-special machinery. The notion *generates* is not gradable; an object is either in the generated set or it is not.

A model-theoretic approach does much better. We simply assume that a grammar is not the logical conjunction of all its individual constraints, but rather a set of constraints. Then it is easy enough to keep track of which constraints are violated by a given structure, assign weights to the constraints, etc.

## Grammatical quandaries

(4)
a. My sister is happy.
b. I am happy.
c. ??My sister or I am happy.
d. ??My sister or I is happy.

## Grammatical quandaries

(4) a. My sister is happy.
  b. I am happy.
  c. ??My sister or I am happy.
  d. ??My sister or I is happy.

i. The verb *am* requires the feature [1 ] on its subject.

ii. The NP *I* has a [1 ] feature.

## Grammatical quandaries

(4)
a. My sister is happy.
b. I am happy.
c. ??My sister or I am happy.
d. ??My sister or I is happy.

i. The verb *am* requires the feature [1 ] on its subject.
ii. The NP *I* has a [1 ] feature.
iii. The verb *is* requires the feature [3 ] on its subject.
iv. The NP *my sister* has a [3 ] feature.

# Grammatical quandaries

(4) a. My sister is happy.
b. I am happy.
c. ??My sister or I am happy.
d. ??My sister or I is happy.

i. The verb *am* requires the feature [1 ] on its subject.
ii. The NP *I* has a [1 ] feature.
iii. The verb *is* requires the feature [3 ] on its subject.
iv. The NP *my sister* has a [3 ] feature.
v. A disjunctive NP is not marked with any person feature if its disjuncts do not agree on this point.

# Grammatical quandaries

(4) a. My sister is happy.
b. I am happy.
c. ??My sister or I am happy.
d. ??My sister or I is happy.

i. The verb *am* requires the feature [1 ] on its subject.
ii. The NP *I* has a [1 ] feature.
iii. The verb *is* requires the feature [3 ] on its subject.
iv. The NP *my sister* has a [3 ] feature.
v. A disjunctive NP is not marked with any person feature if its disjuncts do not agree on this point.

**The conjunction of these principles creates a quandary.**

# Grammatical quandaries

(5) a. ??This is my and Ali's office.
b. ?This is mine and Ali's office.
c. ?This is Ali's and my office.
d. ??This is Ali's and mine office.

**For more quandaries** McCawley 1998

# Structured fragments

(6) a. Ahoy!
    b. Amazing.
    c. Coffee to go.

## Structured fragments

(6) a. Ahoy!
    b. Amazing.
    c. Coffee to go.

If we analyzed these with a CFG, we would have to start with out start symbol, $S$. We are basically committed to the view that nothing is truly a fragment — everything begins from the beginning.

## Structured fragments

(6) a. Ahoy!
 b. Amazing.
 c. Coffee to go.

If we analyzed these with a CFG, we would have to start with out start symbol, $S$. We are basically committed to the view that nothing is truly a fragment — everything begins from the beginning.

In a $\mathcal{L}$-style approach, it is easy to define these as models of the full set of constraints. We could, for instance, relax the requirement that the root is labelled $S$.

# Transformations, model-theoretically

# Transformations, model-theoretically

- Chomskian theories are typically based largely in **transformations**, which linguists conceive of as mapping trees into trees.

## Transformations, model-theoretically

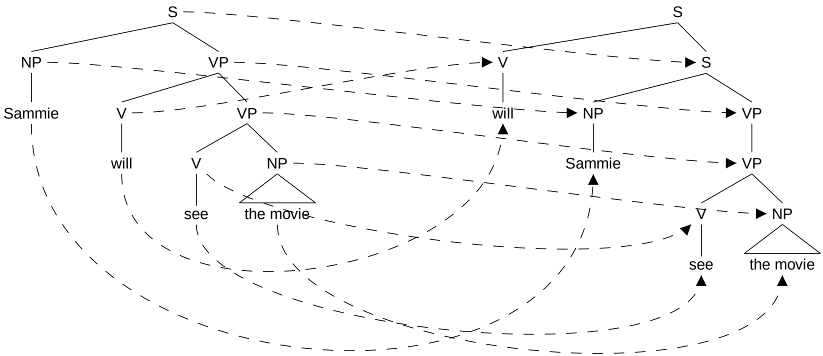
- Chomskian theories are typically based largely in **transformations**, which linguists conceive of as mapping trees into trees.
- A model-theoretic approach can make sense of this, by defining the objects of the theory as tuples of trees.

# Transformations, model-theoretically

- Chomskian theories are typically based largely in **transformations**, which linguists conceive of as mapping trees into trees.
- A model-theoretic approach can make sense of this, by defining the objects of the theory as tuples of trees.
- The insight that this could be done dates to Lakoff 1971, and it was first given a solid formal foundation in the earliest work on Lexical-Functional Grammar (Kaplan and Bresnan 1982).

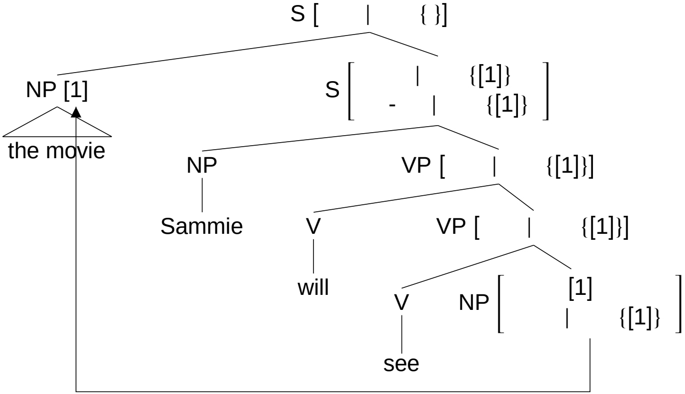
# Transformations, model-theoretically

The *transformation relation*, represented with dotted lines, is another binary relation nodes, formally like dominance.



# Movement or reentrancy?

Kracht (2001) show that, for a broad class of structures, the HPSG notion of reentrancy is identical to the movement conception favored by most transformationalists.



# References: Origins of model-theoretic syntax

Kaplan, Ronald and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan, ed., *The Mental Representation of Grammatical Relations*, 282–390. Cambridge, MA: MIT Press.

Johnson, David E. and Paul M. Postal. 1980. *Arc-Pair Grammar*. Princeton, NJ: Princeton University Press.

Lakoff, George. 1971. On Generative Semantics. In Danny D. Steinberg and Leon A. Jakobovitz, eds., *Semantics: An Interdisciplinary Reader in Philosophy, Linguistics and Psychology*, 232–296. Cambridge: Cambridge University Press.

McCawley, James D. 1968. Concerning the base component of a transformational grammar. *Foundations of Language* 4(1):55–81. Reprinted in *Meaning and Grammar*, 35–58. New York, NY: Academic Press (1976).

# References: Current model-theoretic syntax

Blackburn, Patrick and Claire Gardent. 1995. A specification language for Lexical Functional Grammars. In *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics*, 39–44. San Francisco: Morgan Kaufmann

Blackburn, Patrick, Claire Gardent, and Wilfried Meyer-Viol. 1993. Talking about trees. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*, 21–29. San Francisco: Morgan Kaufmann.

Blackburn, Patrick and Wilfried Meyer-Viol. 1997. Modal logic and model-theoretic syntax. In Maarten de Rijke, ed., *Advances in Intensional Logic*, 29–60. Dordrecht: Kluwer.

# References: Current model-theoretic syntax (con't)

Kracht, Marcus. 2001. Syntax in chains. *Linguistics and Philosophy* 24(4):467–529.

Pullum, Geoffrey K. and Barbara C. Scholz. 2001. On the distinction between model-theoretic and generative-enumerative syntactic frameworks. In Philippe de Groote, Glyn Morrill, and Christian Retoré, *Logical Aspects of Computational Linguistics: 4th International Conference, LACL 2001*, 17–43. Berlin: Springer.

Rogers, James. 1997. "Grammarless" phrase structure grammar. *Linguistics and Philosophy* 20(6):721–746.

Rogers, James. 1998. *A Descriptive Approach to Language-Theoretic Complexity*. Stanford, CA: CSLI/FoLLI.

# References: Proof-theoretic approaches

Carpenter, Bob. 1997. *Type-Logical Semantics*. Cambridge, MA: MIT Press.

Chomsky, Noam. 1964. *Syntactic Structures*. The Hague: Mouton.

Chomsky, Noam. 1965. *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press.

Moortgat, Michael. 1997. Categorial type logics. In Johan van Benthem and Alice ter Meulen, eds., *Handbook of Logic and Language*, 93–177. Cambridge, MA and Amsterdam: MIT Press and North-Holland.

# References: Hybrid approaches

Adger, David. 2003. *Core Syntax*. Oxford: Oxford University Press.

Chomsky, Noam. 1986. *Barriers*. Cambridge, MA: MIT Press.

Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.

Pollard, Carl and IvanA. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Chicago: The University of Chicago Press.

Sag, Ivan A. and Thomas Wasow. 1999. *Syntactic theory: A Formal Introduction*. Stanford, CA: CSLI.

# References: Others

Baker, Mark. 1988. *Incorporation: A Theory of Grammatical Function Changing*. Chicago: University of Chicago Press.

Baker, Mark. 1996. *The Polysynthesis Parameter*. Oxford: Oxford University Press.

Langendoen, D. Terence and Paul M. Postal. 1984. *The Vastness of Natural Languages*. Cambridge, MA: Basil Blackwell.

McCawley, James D. 1998. *The Syntactic Phenomena of English*, 2nd ed. Chicago: Chicago University Press.