Today’s class

• Review, more on PS rules / constituency
• Quiz 2
• Lexicon and syntax
• Beyond PS rules
Process for determining phrase structure rules

1. use constituency tests to determine what the constituents are
2. draw trees
3. formulate rules based on trees

rules reflect generalizations across trees
Draw trees & determine PS Rules

1. Pat kissed Robin.
2. Pat pushed the stubborn horse into the barn.
3. Robin talked to the manager over the phone.
4. Robin yelled at the manager from Ohio.
5. Lee bought a nice picture of the unicorn from Robin.
Pat pushed the stubborn horse into the barn.
A set of (preliminary!) Phrase Structure Rules

S -> NP VP
NP -> (Det) (Adj) N (PP)
VP -> V (NP) (PP)
PP -> P NP

What other sentences can be generated from these rules?
So now, what does syntax look like?
Syntax: a model

PS RULES

\[ S \rightarrow NP \ VP \]
\[ NP \rightarrow (Det) \ N \]
\[ VP \rightarrow V \ (NP) \]

Phrase Structure Trees

S

\[ NP \rightarrow (Det) \ N \]
\[ VP \ \\
\[ V \ (NP) \]
\[ Det \ N \ \\
\[ V \ \\
\[ NP \ \\
\[ Det \ N \]

Lexicon

- scare, V
- the, Det
- chicken, N
- lion, N

Morphology

- V + pst = V
- N + pl = N
Testing the Model

Make sure our model of the grammar makes correct predictions

The model should generate all and only the grammatical sentences

• the model should not *undergenerate* or *overgenerate*
Phrase Structure Rules

What other sentences can be generated from these rules?

Are there grammatical sentences that these rules can’t generate?

Does this set of rules account for the properties of language that we set out to explain?
PS rule exercise 1

S -> NP VP
NP -> (Det) (AdjP) N (PP)
VP -> V (NP) (PP)
VP -> VP PP
PP -> P NP

Try to generate new sentences yourself!
A formal language with the following rules:
\[ S \rightarrow A \ (B), \ A \rightarrow a \ (c) \ e, \ B \rightarrow b \ d \ (f) \]

With this grammar, can you generate:
\[ ae, \ ace, \ bd, \ aebdf, \ acebdf, \ aebd, \ acbd \]
Ambiguity

ambiguous
= having more than one meaning
Ambiguity

Fred jumped from the bank.

We are selling an antique desk for a lady with thick legs.
Ambiguity

Lexical ambiguity: due to a single word having more than one meaning

Structural ambiguity: different meanings due to different structures
Ambiguity

Fred jumped from the bank.

an antique desk for a lady

[with thick legs]
Structural Ambiguity

Kim smashed the statue on the rock.
Kim smashed the statue on the rock

Paraphrase: The statue (that was) on the rock was smashed by Kim.
(Location unspecified)
Meaning 2

*Kim smashed the statue on the rock*

Paraphrase: The statue was smashed by Kim and this took place on the rock.
Meaning 1: apply constit tests

✓  a) What did Kim smash?
  the statue on the rock

NO b) What did Kim smash on the rock?
  the statue

Test (b) doesn’t have meaning 1
Meaning 2: apply constit tests

a) What did Kim smash?
the statue on the rock

b) What did Kim smash on the rock?
the statue

Test (a) doesn’t have meaning 2
Meaning 1

Kim smashed the statue on the rock
Kim smashed the statue on the rock
Structural Ambiguity

*Kim smashed the statue on the rock.*

Do our phrase structure rules generate both structures?
To think about

Cases of ambiguity provide evidence for syntactic structure

How does a child learn about this structure?
PS Rules and Recursion
Recursion

A rule is *recursive* if it can apply to its own output. Did you see a recursive property in our rules?

\[
\begin{align*}
S & \rightarrow NP \ VP \\
NP & \rightarrow (Det)\ (AdjP)\ N\ (PP) \\
VP & \rightarrow V\ (NP)\ (PP) \\
VP & \rightarrow VP\ PP \\
PP & \rightarrow P\ NP
\end{align*}
\]
Recursion

VP -> VP PP
Recursion

NP → (Det) (AdjP) N (PP)
PP → P NP
Recursion

Kim lives in the house on the corner of the longest street in the town near the big lake …

NP → (Det) (AdjP) N (PP)
PP → P NP
Time for Quiz #2!
Role of lexical information in syntax
Who is chasing who?

The child is chasing the rabbits.
The rabbits are chasing the child.
*The child is chasing.
S
   /\  \
  NP VP
     \  
      V   NP
         |
        object

subject
Structural definition of grammatical functions:

- **Subject**: (agent)
- **Object**: (Patient/theme)
sleep

The mother cat slept.

*The mother cat slept the kittens.
Transitive Verbs

S
  /\  \
NP  VP
   /\   /
  V  NP

chase
scare
love
kill
Intransitive Verbs

S

NP VP

V

cry
sleep
gallop
Syntax

PS RULES

\[ S \rightarrow NP \ VP \]
\[ NP \rightarrow (\text{Det}) \ N \]
\[ VP \rightarrow V (NP) \]

Phrase Structure Trees

\[ S \]
\[ \quad NP \]
\[ \quad \quad \quad \quad \quad \text{Det} \ N \]
\[ \quad \quad \quad \quad \quad V \]
\[ \quad \quad \quad \quad \quad NP \]
\[ \quad \quad \quad \quad \quad \quad \text{Det} \ N \]

Lexicon

- scare, V
- the, Det
- chicken, N
- lion, N

Morphology

\[ V + \text{pst} = V \]
\[ N + \text{pl} = N \]
Kim put a valentine in Chris’s mailbox.

*Kim put a valentine.

*Kim put.

*Kim put in Chris’s mailbox.
Kim put [a valentine] [in Chris’s mailbox]
Kim put a valentine in Chris’s mailbox.
complement(s) of a verb: the elements inside VP that are required by the meaning of the verb (= the head of VP)

direct object = NP complement
A diagram representing the structure of a sentence:

- **S** (Sentence)
  - **NP** (Subject)
    - **V** (Verb)
    - **NP** (Object)

The diagram indicates that the subject and object are parts of the sentence, with the verb connecting them.
Lexicon

*chase*  Category: V
Complements: NP

*sleep*  Category: V
Complements: 0

*put*  Category: V
Complements: NP  PP
say

Joan said that George likes broccoli.

S' → (that) S
Lexicon

say Category: V
Complements: S'
Beyond PS rules
But…

a. *What is the child chasing?*
b. *Who will Leo scare?*
c. *What did Kim put in Chris’s mailbox?*
Related sentences

a. the child is chasing what?

b. Leo will scare who?

c. Kim put what in Chris’s mailbox?
Phrase Structure Rules

S → NP (aux) VP
NP → (Det) (AdjP*) N (PP*)
VP → V (NP) (PP)
VP → VP PP
AdjP → Adj
PP → P NP
Additional type of rule in the grammar:

**Transformation:**
Manipulates structure built by PS Rules
Syntax

PS RULES

S → NP VP
NP → (Det) N
VP → V (NP)

Phrase Structure Trees

S
  └── NP
      └── Det N
  └── VP
      └── V (NP)

Lexicon

chase, V compl: NP
sleep, V compl: 0
the, Det
chicken, N
Who will Leo scare?
Phrase structure rules generate:

\[
S \rightarrow NP \ (aux) \ VP \\
VP \rightarrow V \ (NP)
\]

Tree representation:

```
S
  __/
 /   \                     \   
NP    aux                    V   NP
     |                         |   |
    Leo will  scare           who  scaree
          |    |         |    |
         scarer  scare  
```

The sentence generated is: Leo will scare who.
Step 1:
Phrase structure rules + lexical insertion
generate this structure:

```
S
 /    
NP   aux   VP
   /    
Leo will V
     /   
      scare NP
        /  
       who
```

NP “who” originates in complement position
Step 2: Wh-phrase and Aux move to front of sentence
Transformations

Wh-Question transformation:

- **Wh-Movement**: movement of wh-phrase (who, what, etc) to initial position
- **Subject-Aux inversion**: movement of aux to position before subject
Architecture of the Grammar

Syntax

PS Rules → Deep Structure → Transformations → Surface Structure

Lexicon

scare [x, y]
gallop [x]
Phrase structure rules create deep structure tree

Lexical meaning (e.g., argument structure) is represented in the deep structure (and is not changed by transformations)
Other types of meanings are associated with transformations:

declarative vs. question (yes-no, wh)
More on transformation
a. The rabbits are being chased by the children.
b. The trick-or-treaters were scared by Leo.
c. A valentine was put in Chris’s mailbox.
Another transformation

Passive:

Active: Chris ate the apple
Passive: The apple was eaten (by Chris)
Passive

Deep structure for active and passive:

\[
S \rightarrow \text{NP} \rightarrow V \rightarrow \text{NP} \rightarrow \text{VP}
\]

\[
\text{Chris} \rightarrow \text{ate} \rightarrow \text{the} \rightarrow \text{apple}
\]
Passive Transformation

1. a) Delete subject NP
   OR  b) Move subj to by-phrase
2. Add auxiliary verb *be*
3. Change verb form to past participle
4. Move complement NP to subject position
Passive

Passive Transformation

S

NP
Chris

aux
was

past

V

VP

NP

Det
the

N
apple
Passive

S
  NP
    Det  N
      the  apple

aux

VP
  NP
    Det  N
      the  apple
Passive

Surface structure for Passive:

```
S
  /\   /\   /\   /\
 NP aux VP
   /   /   /   /   /   /   /   /   /
 Det N V Det N V Det N V
  The apple was eaten
```
Transformations

What can transformations do?
• Move elements
• Add elements
• Delete elements
Transformations

Transformations are structure dependent

• apply to constituents
• apply to particular constituents in particular configurations
  (= context sensitive)
Transformations

Transformations are structure dependent

For example,
Passive transformation targets the complement NP & moves it to subject position
Passive

Targets complement NP

S
  NP
     aux
      past
  VP
     Det
     N
    the
   apple

Chris
ate

Transformations

Transformations apply to constituents

This is why transformations can be used as tests for constituents!!!
Transformations

Apply passive to the following sentence:

*Lee squashed the bug with the stick.*
Lee squashed the bug with the stick

=> The bug with the stick was squashed (by Lee)
Lee squashed the bug with the stick

=> The bug was squashed with the stick (by Lee)
Passive as a constituency test

a) Lee put the book on the shelf

Passive:
b) The book was put on the shelf by Lee

c) *The book on the shelf was put by Lee
Lee put the book on the shelf.
Summary

Phrase structure rules generate Deep Structure

Transformations apply to deep structure

Surface Structure is sent to (morpho)phonology

Lexical restrictions (eg subcategory restrictions) are satisfied at Deep structure
For tomorrow

• Read Jackendoff Ch6