Syntax:
Further Fundamental Concepts
Supplementary Readings

The following readings have been posted to the Moodle course site:

- Language Files: Chapter 5 (pp. 194-198, 204-215)
- Language Instinct: Chapter 4 (pp. 74-99)
Review of What We Know

- **Syntax** = (the study of) the rules of sentence formation
  - These rules give a general recipe for making sentences.
  - They don’t mention specific words (‘cat’, ‘jump’, ‘dog’)
  - Instead, they mention *categories* of words
Review of What We Know

- **Syntax** = (the study of) the rules of sentence formation
  - These rules give a general recipe for making sentences.
  - They don’t mention specific words (‘cat’, ‘jump’, ‘dog’)
  - Instead, they mention *categories* of words

- **Syntactic category** =
  the word-categories that the rules of syntax refer to.
  - Noun (N) dog, cat, table, happiness...
  - Verb (V) jump, sleep, love, think...
  - Adjective (A) tall, ugly, dead, uncool...
  - Determiner (D) the, a, this, many, most...
  - Preposition (P) in, on, to, with, for...
Review of What We Know

- Sentences are not just strings of words.
- They have a complex internal structure:
  - Sentences are made out of *phrases*: NPs and VPs.
  - NPs are made out of Ds, As, Ns, and PPs
  - VPs are made out of Vs, NPs and PPs
  - PPs are made out of Ps and NPs
Review of What We Know

- Sentences are not just strings of words.
- They have a complex internal structure:
  - Sentences are made out of *phrases*: NPs and VPs.
  - NPs are made out of Ds, As, Ns, and PPs
  - VPs are made out of Vs, NPs and PPs
  - PPs are made out of Ps and NPs

- We can express these groupings with ‘Phrase Structure (PS) Rules’:
  - S → NP VP
  - NP → (D) (A) N (PP)
  - VP → V (NP) (PP)
  - PP → P (NP)

- Some notes on the PS Rule notation:
  - As before, ‘→’ means ‘can be made from’
  - An item in parentheses is *optional*; the phrase doesn’t *need* to have it.
Review of What We Know

Our Four Phrase Structure Rules:

\[ S \rightarrow \text{NP VP} \]
\[ \text{NP} \rightarrow (\text{D}) (\text{A}) \text{ N} (\text{PP}) \]
\[ \text{VP} \rightarrow \text{V} (\text{NP}) (\text{PP}) \]
\[ \text{PP} \rightarrow \text{P} (\text{NP}) \]
Review of What We Know

Our Four Phrase Structure Rules:

S → NP VP
NP → (D) (A) N (PP)
VP → V (NP) (PP)
PP → P (NP)

Our linguistic systems make sentences by using these PS rules and our ‘mental lexicon’:
Review of What We Know

Our Four Phrase Structure Rules:

S → NP VP
NP → (D) (A) N (PP)
VP → V (NP) (PP)
PP → P (NP)

Our linguistic systems make sentences by using these PS rules and our ‘mental lexicon’:

- **Step One:** Use PS rules to make a tree structure.

```
  S
   /\  /
   NP VP
   / \ /  
  A N V  PP
     / \   /
    P   NP
     / \  
    D   N
```
Review of What We Know

Our Four Phrase Structure Rules:

\[
\begin{align*}
S &\to NP \; VP \\
NP &\to (D) \; (A) \; N \; (PP) \\
VP &\to V \; (NP) \; (PP) \\
PP &\to P \; (NP)
\end{align*}
\]

Our linguistic systems make sentences by using these PS rules and our ‘mental lexicon’:

- **Step Two:** Find words in mental lexicon that match categories in tree.
Review of What We Know

Our Four Phrase Structure Rules:

\[
S \rightarrow NP \ VP \\
NP \rightarrow (D) \ (A) \ N \ (PP) \\
VP \rightarrow V \ (NP) \ (PP) \\
PP \rightarrow P \ (NP)
\]

Our linguistic systems make sentences by using these PS rules and our ‘mental lexicon’:

- **Step Two:**
  Find words in mental lexicon that match categories in tree.
  - A = ‘angry’
  - N = ‘bees’
  - V = ‘sting’
  - P = ‘on’
  - D = ‘the’
  - N = ‘face’
Review of What We Know

Our Four Phrase Structure Rules:

\[ S \rightarrow \text{NP VP} \]
\[ \text{NP} \rightarrow (\text{D}) (\text{A}) \text{N} (\text{PP}) \]
\[ \text{VP} \rightarrow \text{V} (\text{NP}) (\text{PP}) \]
\[ \text{PP} \rightarrow \text{P} (\text{NP}) \]

Our linguistic systems make sentences by using these PS rules and our ‘mental lexicon’:

- **Step Three:**
  - Insert those words into the tree structure, under the appropriate category labels.

- \( A = \) ‘angry’
- \( N = \) ‘bees’
- \( V = \) ‘sting’
- \( P = \) ‘on’
- \( D = \) ‘the’
- \( N = \) ‘face’
Review of What We Know

Our Four Phrase Structure Rules:

\[
S \rightarrow \text{NP VP} \\
\text{NP} \rightarrow (\text{D}) (\text{A}) \text{ N (PP)} \\
\text{VP} \rightarrow \text{V (NP) (PP)} \\
\text{PP} \rightarrow \text{P (NP)}
\]

Our linguistic systems make sentences by using these PS rules and our ‘mental lexicon’:

▶ Step Three:

\[
\begin{align*}
S & \quad \text{(Sentence Root)} \\
\text{NP} & \quad \text{(Subject)} \\
\text{VP} & \quad \text{(Verb Phrase)} \\
\text{A} & \quad \text{Angry} \\
\text{N} & \quad \text{bees} \\
\text{V} & \quad \text{sting} \\
\text{PP} & \quad \text{(Prepositional Phrase)} \\
\text{P} & \quad \text{on} \\
\text{NP} & \quad \text{(Possessive Phrase)} \\
\text{D} & \quad \text{the} \\
\text{N} & \quad \text{face}
\end{align*}
\]
In this sub-unit, we will push farther into the basics of syntax:

- We will learn a new key concept (‘syntactic ambiguity’)  
- We will learn some new formal notations for PS rules (stars and brackets)  
- We will learn a few more PS rules central to English (complementizer phrases)
Multiple Ways to Make One Sentence

Our Four Phrase Structure Rules:

\[
S \rightarrow NP \ VP \\
NP \rightarrow (D) (A) \ N \ (PP) \\
VP \rightarrow V \ (NP) \ (PP) \\
PP \rightarrow P \ (NP)
\]
Multiple Ways to Make One Sentence

Our Four Phrase Structure Rules:

\[
S \rightarrow \text{NP } \text{VP} \\
\text{NP} \rightarrow (\text{D}) (\text{A}) \text{ N (PP)} \\
\text{VP} \rightarrow \text{V (NP) (PP)} \\
\text{PP} \rightarrow \text{P (NP)}
\]

Key Observation:
Our rules often make the same sentence in *multiple* ways.

- Consider: ‘The boy saw the man with the telescope’
- Our rules can make this in *two* different ways.
The First Structure: “PP in VP”

First, the rule ‘VP → V (NP) (PP)’ forms a VP containing the PP ‘with the telescope’
The Second Structure: “PP in NP”

- Secondly, the rule ‘NP → (D) (A) N (PP)’ forms an NP containing the PP ‘with the telescope’
Multiple Structures, Multiple Meanings

Fun Fact:
As in morphology, there’s a connection between:

1. The number of ways our syntax rules make a sentence.
2. The number of meanings a sentence has.

Illustration:

- 2 ways to make ‘the boy saw the man with the telescope’.
- Also, 2 meanings for the sentence:
  (cf. Wreck-it-Ralph “You hit a guy... with glasses.”)
  1. ‘The boy saw the man by using a telescope.
  2. ‘The boy saw the man who was holding a telescope

- Each meaning relates naturally to one tree structure.
The First Structure: “PP in VP”

- In this structure, ‘with the telescope’ is part of the VP.
- Since it’s part of VP, it’s describing ‘the action’.
- So, this tree would correspond to the meaning: ‘The boy saw the man by using a telescope.’
  (the seeing of the man was ‘with a telescope’).
The Second Structure: “PP in NP”

In this structure, ‘with the telescope’ is part of the NP.

Since it’s part of NP, it’s describing ‘the man’.

So, this tree would correspond to the meaning:
‘The boy saw the man who was holding a telescope.’
(‘the man’ was ‘with a telescope’)
Syntactic Ambiguity

The Main Point:

- Since our rules can make the same sentence in multiple ways...
- A particular sentence can (sometimes) have multiple meanings...

**Syntactic Ambiguity**

When an expression has multiple meanings because there are multiple ways it can be syntactically constructed.
Some Fun Real-Life Examples

Here are some cute real-life examples of syntactic ambiguity (Pinker 1995):

“Tonight’s program discusses sex with Dick Cavitt”:

- First structure: [ discusses [ sex ] [ with Dick Cavitt ] ]
  (The discussion is with Dick Cavitt)
- Second structure: [ discusses [ sex with Dick Cavitt ] ]
  (The sex is with Dick Cavitt)
Some Fun Real-Life Examples

Here are some cute real-life examples of syntactic ambiguity (Pinker 1995):

“Tonight’s program discusses sex with Dick Cavitt”:

► First structure: [discusses [sex] [with Dick Cavitt]]
  (The discussion is with Dick Cavitt)
► Second structure: [discusses [sex with Dick Cavitt]]
  (The sex is with Dick Cavitt)

“Two cars were reported stolen by police”:

► First structure: [reported [stolen] [by police]]
  (The reporting was by the police)
► Second structure: [reported [stolen by police]]
  (The stealing was by the police)
Expanding Our Rule Notation

Our Four Phrase Structure Rules:

\[
S \rightarrow \text{NP} \ \text{VP} \\
\text{NP} \rightarrow (D) \ (A) \ N \ (\text{PP}) \\
\text{VP} \rightarrow V \ (\text{NP}) \ (\text{PP}) \\
\text{PP} \rightarrow P \ (\text{NP})
\]

So far, our PS rules have the following ingredients:

- Category labels: S, NP, VP, PP, N, V, P, D, A
- ‘can be made from’
- Parentheses: (X) = ‘X is optional’

Now we’re going to add a fourth ingredient:

- The Star: X* = ‘as many Xs as you want’
The Problem of Multiple Adjectives

Our Four Phrase Structure Rules:

S → NP VP
NP → (D) (A) N (PP)
VP → V (NP) (PP)
PP → P (NP)

A Problem for Our NP Rule:

- Our NP rule says it may contain an A (but need not)
- But, NPs in English can have multiple As:
  - The nice cat.
  - The nice old cat.
  - The nice old fluffy cat.
  - The nice old fluffy orange cat....
- In fact, English lets you have as many As as you want (as long as you can keep track of them all, there’s no limit)
The Star Notation

Our Four Phrase Structure Rules:

S → NP VP
NP → (D) (A*) N (PP)
VP → V (NP) (PP)
PP → P (NP)

Question:
How do we fix our NP rule, so we can have unlimited As?
The Star Notation

Our Four Phrase Structure Rules:

S → NP VP
NP → (D) (A) N (PP)
VP → V (NP) (PP)
PP → P (NP)

Question:
How do we fix our NP rule, so we can have unlimited As?

Answer: **The Star Notation!**
In a phrase structure rule, X* means ‘as many Xs as you want’.

Illustration: ‘NP → (D) (A*) N (PP)’

▶ An NP must contain an N, but it can contain:
  ▶ a (single) determiner
  ▶ **as many As as you want**
  ▶ a (single) PP
Our Updated Phrase Structure Rules:

\[
S \rightarrow NP \ VP \\
NP \rightarrow (D) (A^*) N (PP) \\
VP \rightarrow V (NP) (PP) \\
PP \rightarrow P (NP)
\]

Advantage:
We can now easily create NPs with multiple adjectives:

```
NP
   /\          
  /   \        
D     A   A   A   A   A   N
     |       |     |     |     |
   the nice old fluffy orange cat
```
Multiple PPs within an NP

Our Updated Phrase Structure Rules:

\[
S \rightarrow NP \ VP \\
NP \rightarrow (D) \ (A^*) \ N \ (PP) \\
VP \rightarrow V \ (NP) \ (PP) \\
PP \rightarrow P \ (NP)
\]

Problem:

- This rule still limits NPs to just one PP.
- But, an NP can have as many PPs as you like:
  - The boy \textit{in the yard}
  - The boy \textit{from New York in the yard}
  - The boy \textit{from New York in the yard under a tree}
Multiple PPs within an NP

Our Updated Phrase Structure Rules:

\[
S \rightarrow NP \ VP \\
NP \rightarrow (D) (A^*) N (PP) \\
VP \rightarrow V (NP) (PP) \\
PP \rightarrow P (NP)
\]

Problem:

- This rule still limits NPs to just one PP.
- But, an NP can have as many PPs as you like:
  - The boy \textit{in the yard}
  - The boy \textit{from New York in the yard}
  - The boy \textit{from New York in the yard under a tree}

The Solution: \ ‘NP \rightarrow (D) (A^*) N (PP^*)’

- An NP must contain an N, but it can contain:
  - a (single) determiner
  - as many As as you want
  - as many PPs as you want
Multiple PPs within an NP

Our Updated Phrase Structure Rules:

\[ S \rightarrow \text{NP VP} \]
\[ \text{NP} \rightarrow (D) (A^*) \text{N} (PP^*) \]
\[ \text{VP} \rightarrow \text{V} (\text{NP}) (\text{PP}) \]
\[ \text{PP} \rightarrow \text{P} (\text{NP}) \]

**Advantage:**
We can now easily create NPs with multiple PPs:

```
NP
  /\   /\   /\   /\    /\   /\   /\   /\   /\    /\   /\   /\    /\   /\   /\   /\
|   |   |   |   |     |   |   |   |   |     |   |   |   |   |     |   |   |   |   |     |   |   |   |   |     |   |   |   |   |     |   |   |   |   |     |   |   |   |   |
the | the | from | in | D | N | in | D | N | under | D | N | from | in | D | N | in | D | N | a | tree
  |   |   |   |   |     |   |   |   |   |     |   |   |   |   |     |   |   |   |   |     |   |   |   |   |     |   |   |   |   |     |   |   |   |   |     |   |   |   |   |
Only One Determiner, Though

Our Updated Phrase Structure Rules:

\[
S \rightarrow \text{NP} \ \text{VP} \\
\text{NP} \rightarrow (D) \ (A^*) \ N \ (PP^*) \\
\text{VP} \rightarrow V \ (\text{NP}) \ (PP) \\
\text{PP} \rightarrow P \ (\text{NP})
\]

Advantage:

We correctly limit the NP to just a single D.

- * the a dog walked in
- * every the cat was sleeping
- * the every many a boys are nice.
Multiple PPs within a VP

Our Updated Phrase Structure Rules:

S → NP VP
NP → (D) (A*) N (PP*)
VP → V (NP) (PP)
PP → P (NP)

Problem:

- Our rule for VPs limits them to just one PP.
- But, a VP can also have as many PPs as you like:
  - The boy danced in the yard
  - The boy danced in the yard under a tree
  - The boy danced in the yard under a tree near a rock.
Multiple PPs within a VP

Our Updated Phrase Structure Rules:

\[
\begin{align*}
S & \rightarrow \text{NP} \; \text{VP} \\
\text{NP} & \rightarrow (D) \; (A^*) \; N \; (PP^*) \\
\text{VP} & \rightarrow V \; (NP) \; (PP) \\
\text{PP} & \rightarrow P \; (NP)
\end{align*}
\]

Problem:

- Our rule for VPs limits them to just one PP.
- But, a VP can also have as many PPs as you like:
  - The boy danced *in the yard*
  - The boy danced *in the yard under a tree*
  - The boy danced *in the yard under a tree near a rock.*

The Solution: \[‘\text{VP} \rightarrow V \; (NP) \; (PP^*)’\]

- A VP must contain a V, but it can contain:
  - a (single) NP
  - as many PPs as you want
Multiple PPs within an VP

Our Updated Phrase Structure Rules:

\[
\begin{align*}
S & \rightarrow \text{NP \ VP} \\
\text{NP} & \rightarrow (D) (A^*) \text{ N (PP*)} \\
\text{VP} & \rightarrow \text{V (NP) (PP*)} \\
\text{PP} & \rightarrow \text{P (NP)}
\end{align*}
\]

Advantage:
We can now easily create VPs with multiple PPs:

\[
\text{VP} \\
\text{V} \quad \text{PP} \quad \text{PP} \quad \text{PP} \\
\text{danced} \quad \text{P} \quad \text{NP} \quad \text{P} \quad \text{NP} \quad \text{P} \quad \text{NP} \\
\text{in} \quad \text{D} \quad \text{N} \quad \text{under} \quad \text{D} \quad \text{N} \quad \text{near} \quad \text{D} \quad \text{N} \\
\text{the} \quad \text{yard} \quad \text{a} \quad \text{tree} \quad \text{a} \quad \text{rock}
\]
Multiple NPs within a VP

Our Updated Phrase Structure Rules:

\[
S \rightarrow NP \ VP \\
NP \rightarrow (D) \ (A^*) \ N \ (PP^*) \\
VP \rightarrow V \ (NP) \ (PP^*) \\
PP \rightarrow P \ (NP)
\]

Problem:

- Our rule for VPs limits them to just one NP.
- But, a VP can also have more than one NP:
  - The boy gave **Bill a book**.
  - The boy baked **Bill a cake**.
  - The boy told **Bill the story**.
Multiple NPs within a VP

Our Updated Phrase Structure Rules:

\[
S \rightarrow \text{NP VP} \\
\text{NP} \rightarrow (D) (A^*) \text{ N} (\text{PP}^*) \\
\text{VP} \rightarrow \text{V (NP) (PP)*} \\
\text{PP} \rightarrow \text{P (NP)}
\]

Problem:

- Our rule for VPs limits them to just one NP.
- But, a VP can also have more than one NP:
  - The boy gave Bill a book.
  - The boy baked Bill a cake.
  - The boy told Bill the story.

Maybe A Solution?: \(\text{‘VP} \rightarrow \text{V (NP*) (PP*)’}\)

- A VP must contain a V, but it can contain:
  - as many NPs as you want
  - as many PPs as you want
Multiple NPs within a VP

Our Updated Phrase Structure Rules:

- \( S \rightarrow \text{NP VP} \)
- \( \text{NP} \rightarrow (D) (A^*) \text{N} (\text{PP}^*) \)
- \( \text{VP} \rightarrow V (\text{NP}^*) (\text{PP}^*) \)
- \( \text{PP} \rightarrow P (\text{NP}) \)

Problem:

- Our new rule lets VPs have unlimited NPs.
- But, it seems a VP can never have more than two NPs:
  - * The boy gave \textbf{Bill a book the dog}.
  - * The boy baked \textbf{Bill a cake a book}.
  - * The boy told \textbf{Bill the story a cake}.
Multiple NPs within a VP

Our Updated Phrase Structure Rules:

\[
\begin{align*}
S & \rightarrow \text{NP VP} \\
\text{NP} & \rightarrow (D) (A^*) N (PP^*) \\
\text{VP} & \rightarrow V (NP^*) (PP^*) \\
\text{PP} & \rightarrow P (NP)
\end{align*}
\]

Problem:

- Our new rule lets VPs have unlimited NPs.
- But, it seems a VP can *never* have more than *two* NPs:
  - * The boy gave Bill a book the dog.
  - * The boy baked Bill a cake a book.
  - * The boy told Bill the story a cake.

The Real Solution!:

\[
\text{‘VP } \rightarrow \text{ V (NP) (NP) (PP*)’}
\]

- A VP must contain a V, but it can contain:
  - * up to two NPs, next to each other
  - * as many PPs as you want
Multiple NPs within a VP

Our Updated Phrase Structure Rules:

\[
S \rightarrow NP \ VP \\
NP \rightarrow (D) \ (A^*) \ N \ (PP^*) \\
VP \rightarrow V \ (NP) \ (NP) \ (PP^*) \\
PP \rightarrow P \ (NP)
\]

Advantage:
We can now easily create VPs with (the right number of) multiple NPs:

```
S
  NP
    N  VP
      V  NP  NP
      Dave baked N D N
      Bill a cake
```
Sentences within Sentences

Our Current Phrase Structure Rules:

\[ S \rightarrow NP \ VP \]
\[ NP \rightarrow (D) \ (A^*) \ N \ (PP^*) \]
\[ VP \rightarrow V \ (NP) \ (NP) \ (PP^*) \]
\[ PP \rightarrow P \ (NP) \]

Problem:

Our rules are not enough to make these sentences:

- Dave thinks **that Mary danced**.
- Dave wonders **if Mary danced**.
- Dave knows **whether Mary danced**.
- Dave told Bill **that Mary danced**.
Sentences within Sentences

Our Current Phrase Structure Rules:

\[
\begin{align*}
S & \rightarrow \text{NP VP} \\
\text{NP} & \rightarrow (D) (A^*) \text{ N (PP*)} \\
\text{VP} & \rightarrow V (\text{NP}) (\text{NP}) (\text{PP*}) \\
\text{PP} & \rightarrow P (\text{NP})
\end{align*}
\]

Problem:
Our rules are not enough to make these sentences:

- Dave thinks **that Mary danced**.
- Dave wonders **if Mary danced**.
- Dave knows **whether Mary danced**.
- Dave told Bill **that Mary danced**.

The Key Feature of These Sentences:

- The V is followed by an S (‘Mary danced’)
- Just before the S, there is a very short word (‘that’, ‘if’, ‘whether’)
- These ‘short words’ are called **Complementizers (C)**
Sentences within Sentences

Our Current Phrase Structure Rules:

\[
S \rightarrow NP\ VP \\
NP \rightarrow (D)\ (A^*)\ N\ (PP^*) \\
VP \rightarrow V\ (NP)\ (NP)\ (PP^*) \\
PP \rightarrow P\ (NP)
\]

Problem:
Our rules are not enough to make these sentences:

- Dave thinks that Mary danced.
- Dave wonders if Mary danced.
- Dave knows whether Mary danced.
- Dave told Bill that Mary danced.

A Quick Side-Note:

- In some sentences, you can drop the complementizer ‘that’
  - Dave thinks Mary danced.
- For this class, let’s forget about those...
Sentences within Sentences

Our Current Phrase Structure Rules:

S → NP VP
NP → (D) (A*) N (PP*)
VP → V (NP) (NP) (PP*)
PP → P (NP)

Problem:
Our rules are not enough to make these sentences:

- Dave thinks that Mary danced.
- Dave wonders if Mary danced.
- Dave knows whether Mary danced.
- Dave told Bill that Mary danced.

Maybe A Solution?

- VP → V (NP) (NP) (C S) (PP*)
  (A VP can contain... a C followed by an S)
Sentences within Sentences

Our Current Phrase Structure Rules:

\[ S \rightarrow NP \ VP \]
\[ NP \rightarrow (D) (A^*) N (PP^*) \]
\[ VP \rightarrow V (NP) (NP) (C S) (PP^*) \]
\[ PP \rightarrow P (NP) \]

Problem:

- Our new rule will make VPs with two NPs, a C and an S
- But, such VPs are (almost) never possible:
  - Dave told Bill that Mary danced.
  - * Dave told Bill John that Mary danced.
Sentences within Sentences

Our Current Phrase Structure Rules:

\[
\begin{align*}
S & \rightarrow \text{NP VP} \\
\text{NP} & \rightarrow (D) (A^*) \text{ N (PP*)} \\
\text{VP} & \rightarrow V (\text{NP}) (\text{NP}) (C \text{ S}) (\text{PP*}) \\
\text{PP} & \rightarrow P (\text{NP})
\end{align*}
\]

The Real Pattern:
If a V is followed by an NP, then...

- It can be followed by one more NP:  
  (Dave gave Bill a cake)

- It can be followed by a C and an S:  
  (Dave told Bill that he stinks)

- But not both an NP and an S!  
  *(Dave told Bill Dave that he stinks)*
Introducing Bracket Notation

Question:
How can we capture this pattern:

- \( V \ NP \ NP \) = GOOD VP
- \( V \ NP \ C \ S \) = GOOD VP
- \( V \ NP \ NP \ C \ S \) = BAD VP

The Solution:
Bracket Notation!
In a PS rule, \( f \ X, Y \ g \) means 'either X or Y, but not both'.
Illustration:
\[ VP ! V (NP) \ f (NP) (C S) \ g (PP*) \]

I A VP must contain a V.
But, a VP can contain:
- An NP
- Either a second NP, or a C and an S, but not both
- As many PPs as you like
Introducing Bracket Notation

Question:
How can we capture this pattern:

V NP NP = GOOD VP
V NP C S = GOOD VP
V NP NP C S = BAD VP

The Solution: Bracket Notation!
In a PS rule, \{ X, Y \} means ‘either X or Y, but not both’.
Introducing Bracket Notation

Question:
How can we capture this pattern:

\[ \text{V NP NP} = \text{GOOD VP} \]
\[ \text{V NP C S} = \text{GOOD VP} \]
\[ \text{V NP NP C S} = \text{BAD VP} \]

The Solution: Bracket Notation!
In a PS rule, \{ X, Y \} means ‘either X or Y, but not both’.

Illustration:
\[ \text{VP} \rightarrow \text{V (NP) \{ (NP) (C S) \} (PP^*)} \]

- A VP \textit{must} contain a V.
- But, a VP \textit{can} contain:
  - An NP
  - \textit{Either a second NP, or a C and an S, but not both}
  - As many PPs as you like
Cs and Ss Elsewhere

Our Updated PS Rules:

S → NP VP
NP → (D) (A*) N (PP*)
VP → V (NP) { (NP) (C S) } (PP*)
PP → P (NP)

Problem:
VPs aren’t the only place where you find ‘(C S)’
Cs and Ss Elsewhere

Our Updated PS Rules:

\[
S \rightarrow NP \ VP \\
NP \rightarrow (D) \ (A^*) \ N \ (PP^*) \\
VP \rightarrow V \ (NP) \ \{ \ (NP) \ (C \ S) \ \} \ (PP^*) \\
PP \rightarrow P \ (NP)
\]

Problem:
VPs aren't the only place where you find ‘(C S)’

Sentence | Rule Needed
---|---
the rumor that Bill stinks upset Mary

Key Observations:
I.
Inside a sentence, you never see an S without a C.
II.
You also never see a C without an S.
Cs and Ss Elsewhere

Our Updated PS Rules:

\[
\begin{align*}
S & \rightarrow NP \ VP \\
NP & \rightarrow (D) (A^*) N (PP^*) \\
VP & \rightarrow V (NP) \{ (NP) (C\ S) \} (PP^*) \\
PP & \rightarrow P (NP)
\end{align*}
\]

Problem:

VPs aren’t the only place where you find ‘(C S)’

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Rule Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>the rumor that Bill stinks upset Mary</td>
<td>[NP \rightarrow D (A^<em>) N (C\ S) (PP^</em>)]</td>
</tr>
</tbody>
</table>
Cs and Ss Elsewhere

Our Updated PS Rules:

\[
S \rightarrow NP \ VP \\
NP \rightarrow (D) \ (A^*) \ N \ (PP^*) \\
VP \rightarrow V \ (NP) \ \{ (NP) \ (C \ S) \} \ (PP^*) \\
PP \rightarrow P \ (NP)
\]

Problem:
VPs aren’t the only place where you find ‘(C S)’

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Rule Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>the rumor that Bill stinks upset Mary</td>
<td>NP \rightarrow D \ (A^<em>) \ N \ (C \ S) \ (PP^</em>)</td>
</tr>
<tr>
<td>that Bill stinks surprised Mary</td>
<td></td>
</tr>
</tbody>
</table>
Cs and Ss Elsewhere

Our Updated PS Rules:

\[
\begin{align*}
S & \rightarrow \text{NP } \text{VP} \\
\text{NP} & \rightarrow (\text{D}) \ (\text{A}^*) \ \text{N} \ (\text{PP}^*) \\
\text{VP} & \rightarrow \text{V} \ (\text{NP}) \ \{ \ (\text{NP}) \ (\text{C} \ \text{S}) \ \} \ (\text{PP}^*) \\
\text{PP} & \rightarrow \text{P} \ (\text{NP})
\end{align*}
\]

Problem:

VPs aren’t the only place where you find ‘(C S)’

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Rule Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>the rumor that Bill stinks upset Mary</td>
<td>NP → D (A*) N (C S) (PP*)</td>
</tr>
<tr>
<td>that Bill stinks surprised Mary</td>
<td>S → { NP , C S } VP</td>
</tr>
</tbody>
</table>
Cs and Ss Elsewhere

Our Updated PS Rules:

\[
\begin{align*}
S & \rightarrow NP \ VP \\
NP & \rightarrow (D) (A^*) N (PP^*) \\
VP & \rightarrow V (NP) \{ (NP) (C\ S) \} (PP^*) \\
PP & \rightarrow P (NP)
\end{align*}
\]

Problem:

VPs aren’t the only place where you find ‘(C S)’

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Rule Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>the rumor that Bill stinks</td>
<td>NP \rightarrow D (A^<em>) N (C\ S) (PP^</em>)</td>
</tr>
<tr>
<td>upset Mary</td>
<td></td>
</tr>
<tr>
<td>that Bill stinks</td>
<td>S \rightarrow { NP , C\ S } \ VP</td>
</tr>
<tr>
<td>surprised Mary</td>
<td></td>
</tr>
</tbody>
</table>

Key Observations:

- In each of these rules, ‘C S’ keeps coming up.
- Inside a sentence, you never see an S without a C.
- You also never see a C without an S.
Introducing CPs

Our Updated PS Rules:

\[
S \rightarrow \text{NP VP} \\
\text{NP} \rightarrow (D) (A^*) N (PP^*) \\
\text{VP} \rightarrow V (\text{NP}) \{ (\text{NP}) (C \ S) \} (PP^*) \\
\text{PP} \rightarrow P (\text{NP})
\]

**Problem:**

VPs aren’t the only place where you find ‘(C S)’

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Rule Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>the rumor that Bill stinks upset Mary</td>
<td>NP \rightarrow D (A^<em>) N (C S) (PP^</em>)</td>
</tr>
<tr>
<td>that Bill stinks surprised Mary</td>
<td>S \rightarrow { NP, C S } VP</td>
</tr>
</tbody>
</table>

**Key Idea:**

English treats ‘C S’ as a kind of ‘unit’

- Let’s call this a **Complementizer Phrase (CP)** (since it always has a C in it)
Introducing CPs

Our Updated PS Rules:

\[
S \rightarrow \text{NP VP} \\
\text{NP} \rightarrow \text{(D) (A*) N (PP*)} \\
\text{VP} \rightarrow \text{V (NP) \{ (NP) (C S) \} (PP*)} \\
\text{PP} \rightarrow \text{P (NP)}
\]

Problem:

VPs aren’t the only place where you find ‘(C S)’

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Rule Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>the rumor that Bill stinks upset Mary</td>
<td>NP \rightarrow \text{D (A*) N (C S) (PP*)}</td>
</tr>
<tr>
<td>that Bill stinks surprised Mary</td>
<td>S \rightarrow { \text{NP , C S} } \text{ VP}</td>
</tr>
</tbody>
</table>

The PS Rule for CPs: \( \text{CP} \rightarrow \text{C S} \)

‘CP is formed from a C followed by an S’
Introducing CPs

Our Updated PS Rules:

\[ S \rightarrow \{ \text{NP, CP} \} \text{ VP} \]
\[ \text{NP} \rightarrow (D) (\text{A}^*) \text{ N (CP) (PP*)} \]
\[ \text{VP} \rightarrow V \text{ (NP) } \{ \text{NP) (CP) } \} \text{ (PP*)} \]
\[ \text{PP} \rightarrow P \text{ (NP)} \]
\[ \text{CP} \rightarrow C \text{ S} \]

Advantage:
We can now create sentences that have sentences in them:
Introducing CPs

Our Updated PS Rules:

\[ S \rightarrow \{ \text{NP} , \text{CP} \} \text{VP} \]
\[ \text{NP} \rightarrow (D) (A^*) \text{N} (\text{CP}) (\text{PP}^*) \]
\[ \text{VP} \rightarrow \text{V} (\text{NP}) \{ (\text{NP}) (\text{CP}) \} (\text{PP}^*) \]
\[ \text{PP} \rightarrow \text{P} (\text{NP}) \]
\[ \text{CP} \rightarrow \text{C} \text{S} \]
Introducing CPs

Our Updated PS Rules:

\[
S \rightarrow \{ \text{NP} , \text{CP} \} \text{ VP} \\
\text{NP} \rightarrow (D) (A^*) \text{ N } (\text{CP}) (\text{PP}^*) \\
\text{VP} \rightarrow V (\text{NP}) \{ (\text{NP}) (\text{CP}) \} (\text{PP}^*) \\
\text{PP} \rightarrow P (\text{NP})
\]

\[
\text{CP} \rightarrow C \text{ S}
\]
Introducing CPs

Our Updated PS Rules:

\[
S \rightarrow \{ \text{NP}, \text{CP} \} \ VP \\
NP \rightarrow (D) (A^*) \text{ N (CP) (PP*)} \\
VP \rightarrow V (NP) \{ (NP) (CP) \} (PP*) \\
PP \rightarrow P (NP) \\
\text{CP} \rightarrow C \ S
\]

The rumor that Bill sucks surprised him.