Historical Background: Incremental Processing and the Strict Competence Hypothesis

Vera Demberg

Universität des Saarlandes

– SS 2011 –

April 21st, 2011

Organization: Topics and Time Slots

- Oral exams: yes
- Time slots, see sheets
- more presenters than slots
- I suggest to have some double sessions which begin at 10:30.
- Alternative: meet for one triple session on a Saturday.

Today: more like a lecture – but I want to try a new thing:
after each section: pause 5 min, take notes, ask questions.

Organization: Topics and Time Slots

- Oral exams: yes
- Time slots, see sheets
- more presenters than slots
- I suggest to have some double sessions which begin at 10:30.
- Alternative: meet for one triple session on a Saturday.

Today: more like a lecture – but I want to try a new thing:
after each section: pause 5 min, take notes, ask questions.

Introduction

The Historic Discussion (’80s / ’90s).

1. Introduction
2. The Paradox (Steedman)
3. Suggested Solution 1: Change notion of constituents (Steedman)
4. Suggested Solution 2: Less Strict Competence Hypothesis (Stabler)
5. Comparison of Parsing Strategies (Abney & Johnson)
6. Suggested Solution 3: Asynchronous Processing (Shieber & Johnson)

Table of Contents

Introduction

- Main players in the debate:
  Mark Steedman, Ed Stabler, Steven Abney, Mark Johnson, Stuart Shieber
- When? 1988-1993
- Question: what's a psycholinguistically plausible parsing mechanism?
- Criteria:
  - should explain linguistic constraints on language
  - processing difficulties (e.g. center embedding)
  - account for incremental interpretation (e.g. fast ambiguity resolution)
  - simple relationship between grammar and processor (Occam's razor)

Background: Definitions (1)

Constituents = “grammatical entities”

A constituent is a word or a group of words that functions as a single unit within a hierarchical structure.

Some constituency tests:

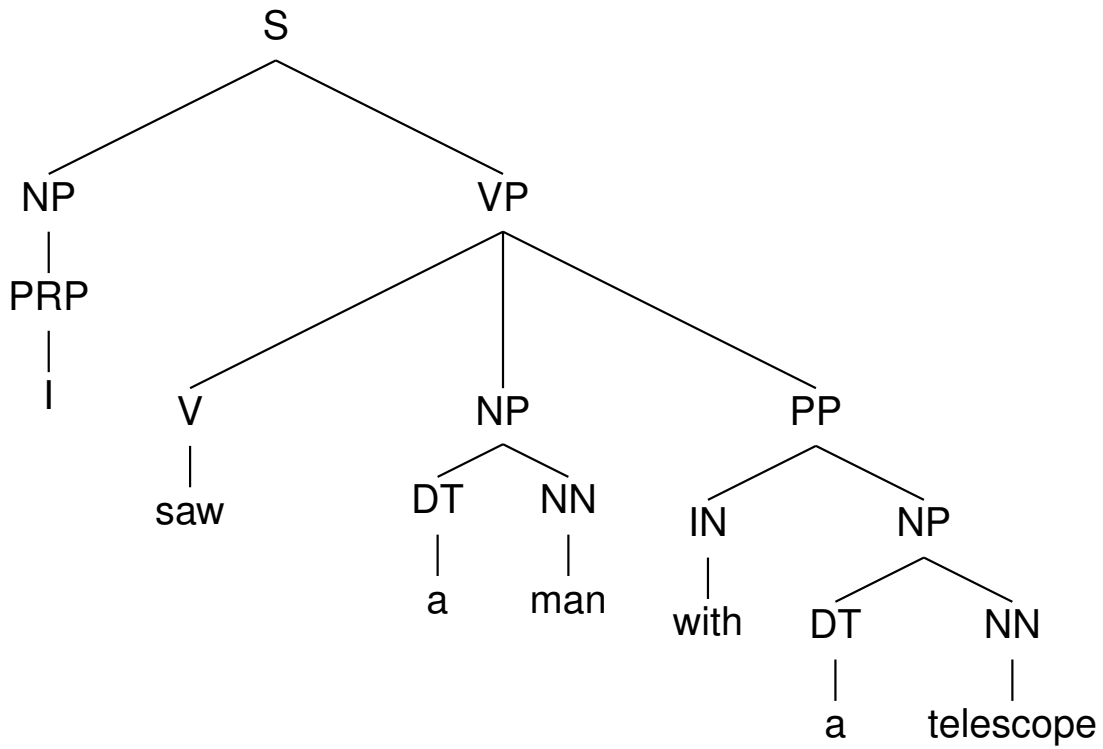
1. Substitution ([The man] knows [the dog that barks].)
2. Deletion (He knows the dog [that barks].)
3. Movement (He attends a course [to improve his German].)
4. Coordination (He [cooked dinner] and [went to bed].)
5. Question-test ([Who] cooked dinner? Maria)

Background: Definitions (1)

Constituents = “grammatical entities”

A constituent is a word or a group of words that functions as a single unit within a hierarchical structure.

Constituents are reflected directly in the syntactic structure we assign to a sentence:



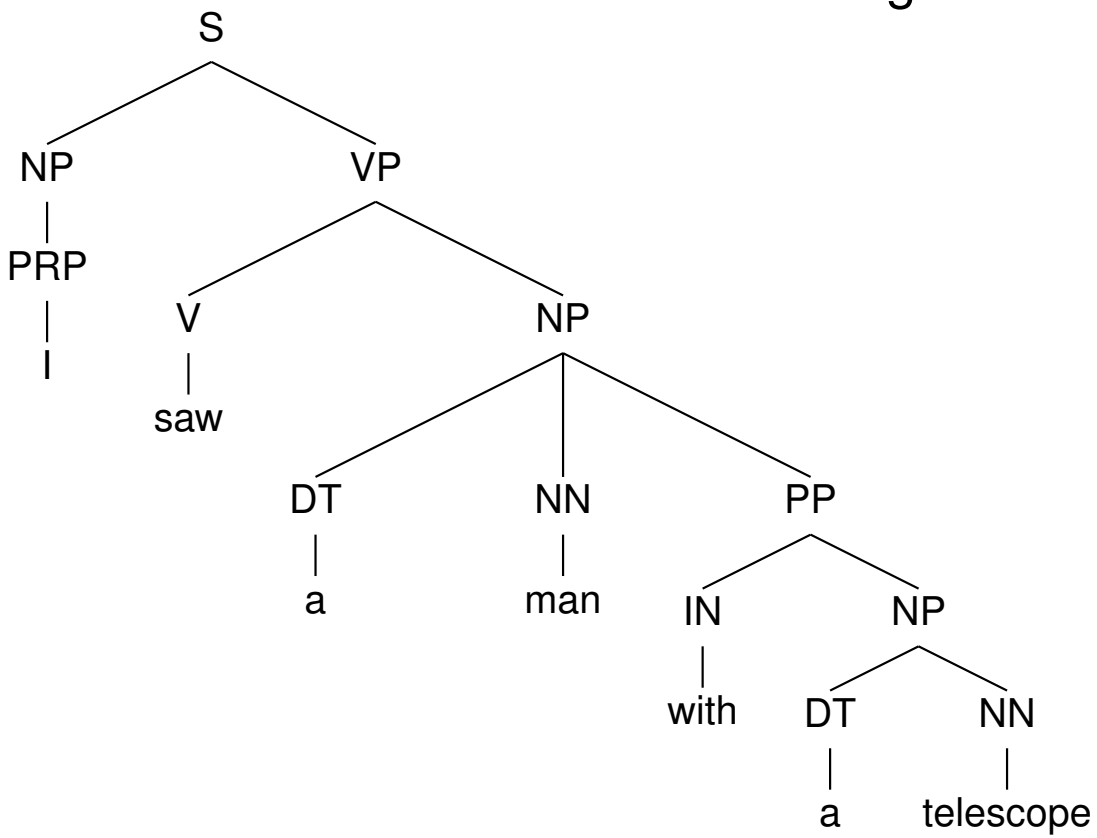
I [saw [a man] [with [a telescope]]]

Background: Definitions (1)

Constituents = “grammatical entities”

A constituent is a word or a group of words that functions as a single unit within a hierarchical structure.

Ambiguous sentences can have alternative constituent assignments:



Background: Definitions (2)

Incremental Interpretation

Semantic interpretation in human sentence processing can occur before sentence boundaries, and even before constituent boundaries. Some psycholinguistic evidence indicates that semantic interpretation in fact occurs on a word-by-word basis.

Background: Definitions (3)

Competence

Competence is the **'ideal' language system** that makes it possible for speakers to produce and understand an infinite number of sentences in their language, and to distinguish grammatical sentences from ungrammatical sentences.

Performance

Linguistic performance is governed by **principles of cognitive structure** such as memory limitations, distractions, shifts of attention and interest, and (random or characteristic) errors.

Distinction first introduced by Noam Chomsky (1965)

- **Linguistics** is concerned primarily with **competence**
- **Psychology** is concerned primarily with **performance**

What's the relationship between the two?

Background: Definitions (4)

Strong Competence Hypothesis (Bresnan and Kaplan, 1982)

The Strong Competence Hypothesis of asserts that there exists a direct correspondence between the rules of a grammar and the operations performed by the human language processor.

+

Rule-to-Rule Assumption (Bach, 1976)

Each syntactic rule corresponds to a rule of semantic interpretation.
($\Rightarrow$ entities combined by syntactic rules must be semantically interpretable)

=

Strict Competence Hypothesis (Steedman, 1992)

Structures manipulated by the processor are isomorphic to the constituents listed in the grammar.

Background: Definitions (4)

Strong Competence Hypothesis (Bresnan and Kaplan, 1982)

The Strong Competence Hypothesis of asserts that there exists a direct correspondence between the rules of a grammar and the operations performed by the human language processor.

+

Rule-to-Rule Assumption (Bach, 1976)

Each syntactic rule corresponds to a rule of semantic interpretation.
($\Rightarrow$ entities combined by syntactic rules must be semantically interpretable)

=

Strict Competence Hypothesis (Steedman, 1992)

Structures manipulated by the processor are isomorphic to the constituents listed in the grammar.

Background: Definitions (4)

Strong Competence Hypothesis (Bresnan and Kaplan, 1982)

The Strong Competence Hypothesis of asserts that there exists a direct correspondence between the rules of a grammar and the operations performed by the human language processor.

$+$

Rule-to-Rule Assumption (Bach, 1976)

Each syntactic rule corresponds to a rule of semantic interpretation.
($\Rightarrow$ entities combined by syntactic rules must be semantically interpretable)

$=$

Strict Competence Hypothesis (Steedman, 1992)

Structures manipulated by the processor are isomorphic to the constituents listed in the grammar.

Time for notes

Take notes on the following concepts:

- a constituent
- incremental interpretation
- competence
- performance
- strong / strict competence hypothesis

If you've trouble with any of them, please don't feel shy to ask!
We want to discuss these before proceeding to the rest of the lecture.

Timeline of debate

1989 Mark Steedman brings up “Paradox” of three assumptions

- Incremental Interpretation
- Syntax Theories with Right-Branching Structures
- Strict Competence Hypothesis

and argues for bottom-up parsing and new notion of constituents.

1991 Edward Stabler replicates that paradox is not really valid

1991 Steven Abney and Mark Johnson argue against both top-down and bottom-up parsing strategies

1992 Mark Steedman replies to criticisms by Stabler

1993 Alternative proposal by Shieber and Johnson (opposing both Stabler and Steedman)

Table of Contents

In detail: What's the Paradox?

Mark Steedman brings up "Paradox" of three assumptions

- Incremental Interpretation
- Syntax Theories with Right-Branching Structures
- Strict Competence Hypothesis

Example

a) The doctor sent for the patients arrived. (more difficult)
b) The flowers sent for the patients arrived. (less difficult)

- If b) is easier, this indicates that the processor has figured out at the point of "sent" that *flowers* cannot be the agent of a sending action.
  $\Rightarrow$ **Incremental interpretation at "the flowers sent"**
- But: "the flowers sent" is **not a constituent**!
- Non-constituents cannot have a semantic interpretation according to strict competence hypothesis!

In detail: What's the Paradox?

Mark Steedman brings up "Paradox" of three assumptions

- Incremental Interpretation
- Syntax Theories with Right-Branching Structures
- Strict Competence Hypothesis

Example

a) The doctor sent for the patients arrived. (more difficult)
b) The flowers sent for the patients arrived. (less difficult)

- If b) is easier, this indicates that the processor has figured out at the point of "sent" that *flowers* cannot be the agent of a sending action.
  $\Rightarrow$ **Incremental interpretation at "the flowers sent"**
- But: "the flowers sent" is **not a constituent**!
- Non-constituents cannot have a semantic interpretation according to strict competence hypothesis!

Ways out of the paradox

Paradox – Can't simultaneously have:

- Incremental Interpretation
- Syntax Theories with Right-Branching Structures
- Strict Competence Hypothesis

Way out: Drop or weaken at least one of the assumptions!

Ways out of the paradox

Paradox – Can't simultaneously have:

- Incremental Interpretation
- Syntax Theories with Right-Branching Structures
- Strict Competence Hypothesis

Way out: Drop or weaken at least one of the assumptions!

Ways out of the paradox

Paradox – Can't simultaneously have:

- Incremental Interpretation
- Syntax Theories with Right-Branching Structures
- Strict Competence Hypothesis

Way out: Drop or weaken at least one of the assumptions!

Steedman

- Incremental Interpretation
- ~~Syntax Theories with Right-Branching Structures~~ **CCG**
- Strict Competence Hypothesis

Ways out of the paradox

Paradox – Can't simultaneously have:

- Incremental Interpretation
- Syntax Theories with Right-Branching Structures
- Strict Competence Hypothesis

Way out: Drop or weaken at least one of the assumptions!

Stabler

- Incremental Interpretation
- Syntax Theories with Right-Branching Structures
- ~~Strict~~ Competence Hypothesis

Time for notes

What is the paradox here?

Table of Contents

What is Steedman's solution?

For parsing process (performance), what do we require minimally?

1. a competence grammar
2. an algorithm
3. a stack (memory)
4. an oracle (to resolve non-determinism)

How little else can we get away with? top-down vs. bottom-up processing:

Top-down processing

Step 1:

What is Steedman's solution?

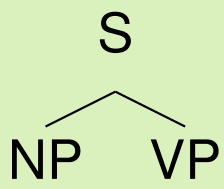
For parsing process (performance), what do we require minimally?

1. a competence grammar
2. an algorithm
3. a stack (memory)
4. an oracle (to resolve non-determinism)

How little else can we get away with? top-down vs. bottom-up processing:

Top-down processing

Step 1:



What is Steedman's solution?

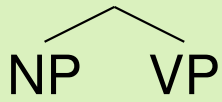
For parsing process (performance), what do we require minimally?

1. a competence grammar
2. an algorithm
3. a stack (memory)
4. an oracle (to resolve non-determinism)

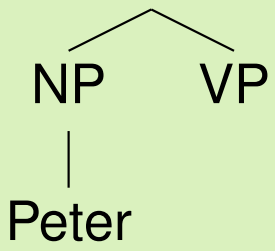
How little else can we get away with? top-down vs. bottom-up processing:

Top-down processing

Step 1:



Step 2:



What is Steedman's solution?

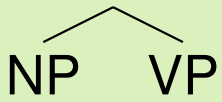
For parsing process (performance), what do we require minimally?

1. a competence grammar
2. an algorithm
3. a stack (memory)
4. an oracle (to resolve non-determinism)

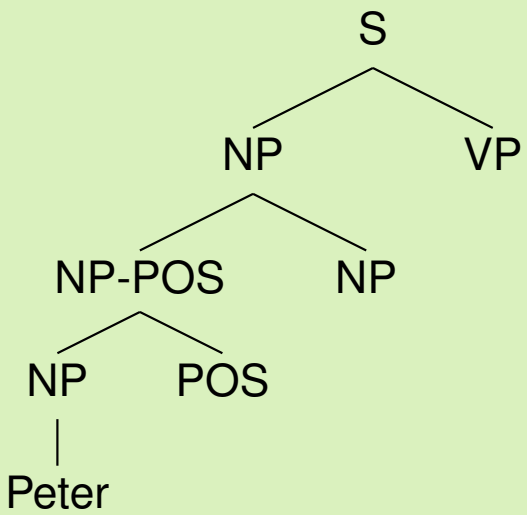
How little else can we get away with? top-down vs. bottom-up processing:

Top-down processing

Step 1:



Step 2:



What is Steedman's solution?

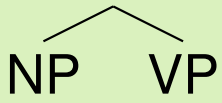
For parsing process (performance), what do we require minimally?

1. a competence grammar
2. an algorithm
3. a stack (memory)
4. an oracle (to resolve non-determinism)

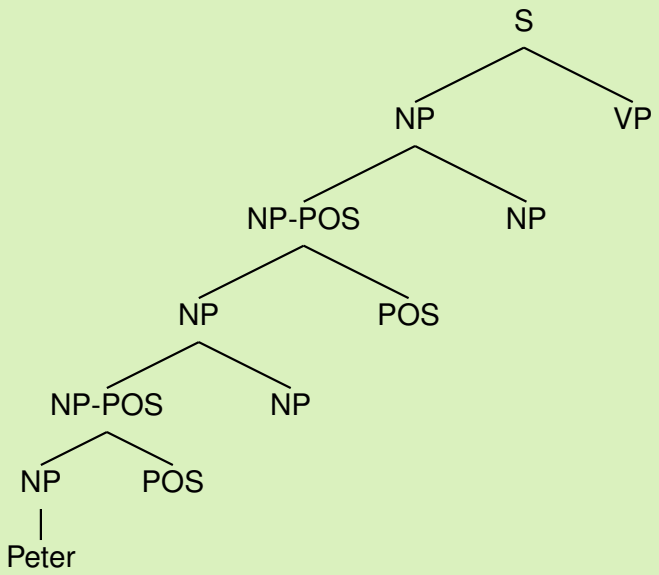
How little else can we get away with? top-down vs. bottom-up processing:

Top-down processing

Step 1:



Step 2:



What is Steedman's solution?

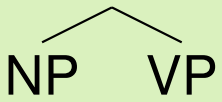
For parsing process (performance), what do we require minimally?

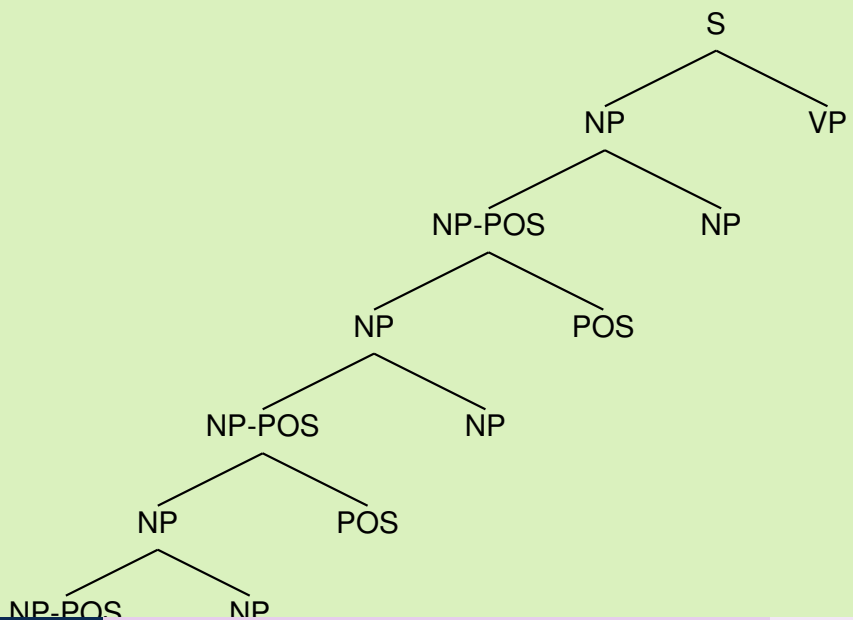
1. a competence grammar
2. an algorithm
3. a stack (memory)
4. an oracle (to resolve non-determinism)

How little else can we get away with? top-down vs. bottom-up processing:

Top-down processing

Step 1: S





What is Steedman's solution?

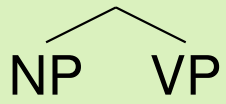
For parsing process (performance), what do we require minimally?

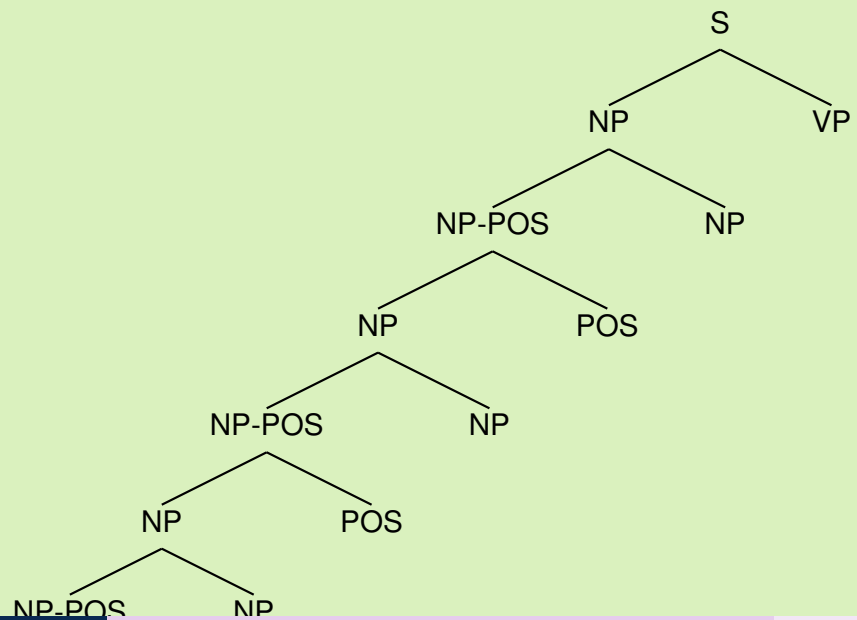
1. a competence grammar
2. an algorithm
3. a stack (memory)
4. an oracle (to resolve non-determinism)

How little else can we get away with? top-down vs. bottom-up processing:

Top-down processing

Step 1:





Top-down algorithm needs a special watchdog to deal with left-recursive rules.

What is Steedman's solution?

For parsing process (performance), what do we require minimally?

1. a competence grammar
2. an algorithm
3. a stack (memory)
4. an oracle (to resolve non-determinism)

How little else can we get away with? top-down vs. bottom-up processing:

Bottom-up processing

Step 1:

What is Steedman's solution?

For parsing process (performance), what do we require minimally?

1. a competence grammar
2. an algorithm
3. a stack (memory)
4. an oracle (to resolve non-determinism)

How little else can we get away with? top-down vs. bottom-up processing:

Bottom-up processing

Step 2:

What is Steedman's solution?

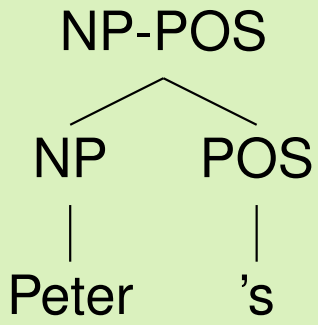
For parsing process (performance), what do we require minimally?

1. a competence grammar
2. an algorithm
3. a stack (memory)
4. an oracle (to resolve non-determinism)

How little else can we get away with? top-down vs. bottom-up processing:

Bottom-up processing

Step 2:



Don't have the infinity problem...

What is Steedman's solution?

For parsing process (performance), what do we require minimally?

1. a competence grammar
2. an algorithm
3. a stack (memory)
4. an oracle (to resolve non-determinism)

How little else can we get away with? top-down vs. bottom-up processing:

Bottom-up processing

... but can't do incremental interpretation if syntactic structures unconnected.

So how does Steedman solve paradox?

Remember our problem:

a) The doctor sent for the patients arrived. (more difficult)
b) The flowers sent for the patients arrived. (less difficult)

- If b) is easier, this indicates that the processor has figured out at the point of "sent" that *flowers* cannot be the agent of a sending action.
  $\Rightarrow$ **Incremental interpretation at "the flowers sent"**
- But: "the flowers sent" is **not a constituent**!
- Non-constituents cannot have a semantic interpretation according to strict competence hypothesis!

Steedman's solution:

1. Do bottom-up parsing
2. Use grammar formalism in which "the flower sent" is a constituent
   $\Rightarrow$ more generally: where constituent structure is left branching

So how does Steedman solve paradox?

Remember our problem:

a) The doctor sent for the patients arrived. (more difficult)
b) The flowers sent for the patients arrived. (less difficult)

- If b) is easier, this indicates that the processor has figured out at the point of "sent" that *flowers* cannot be the agent of a sending action.
  $\Rightarrow$ **Incremental interpretation at "the flowers sent"**
- "the flowers sent" is a constituent
- Non-constituents cannot have a semantic interpretation according to strict competence hypothesis!

Steedman's solution:

1. Do bottom-up parsing
2. Use grammar formalism in which "the flower sent" is a constituent
   $\Rightarrow$ more generally: where constituent structure is left branching

So how does that work?

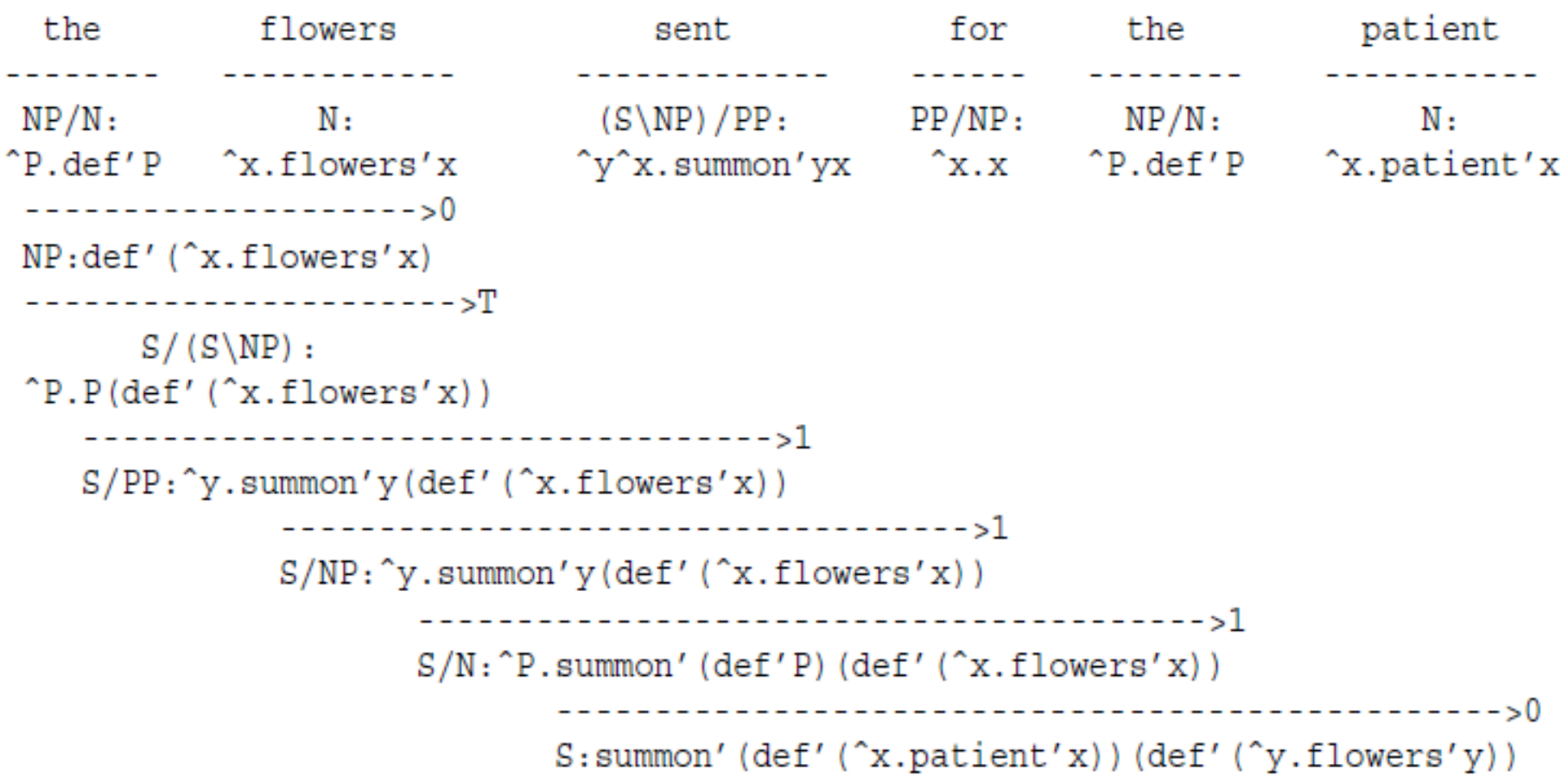


Figure: Incremental CCG derivation (Figure taken from McConville's PhD thesis.)

So how does that work?

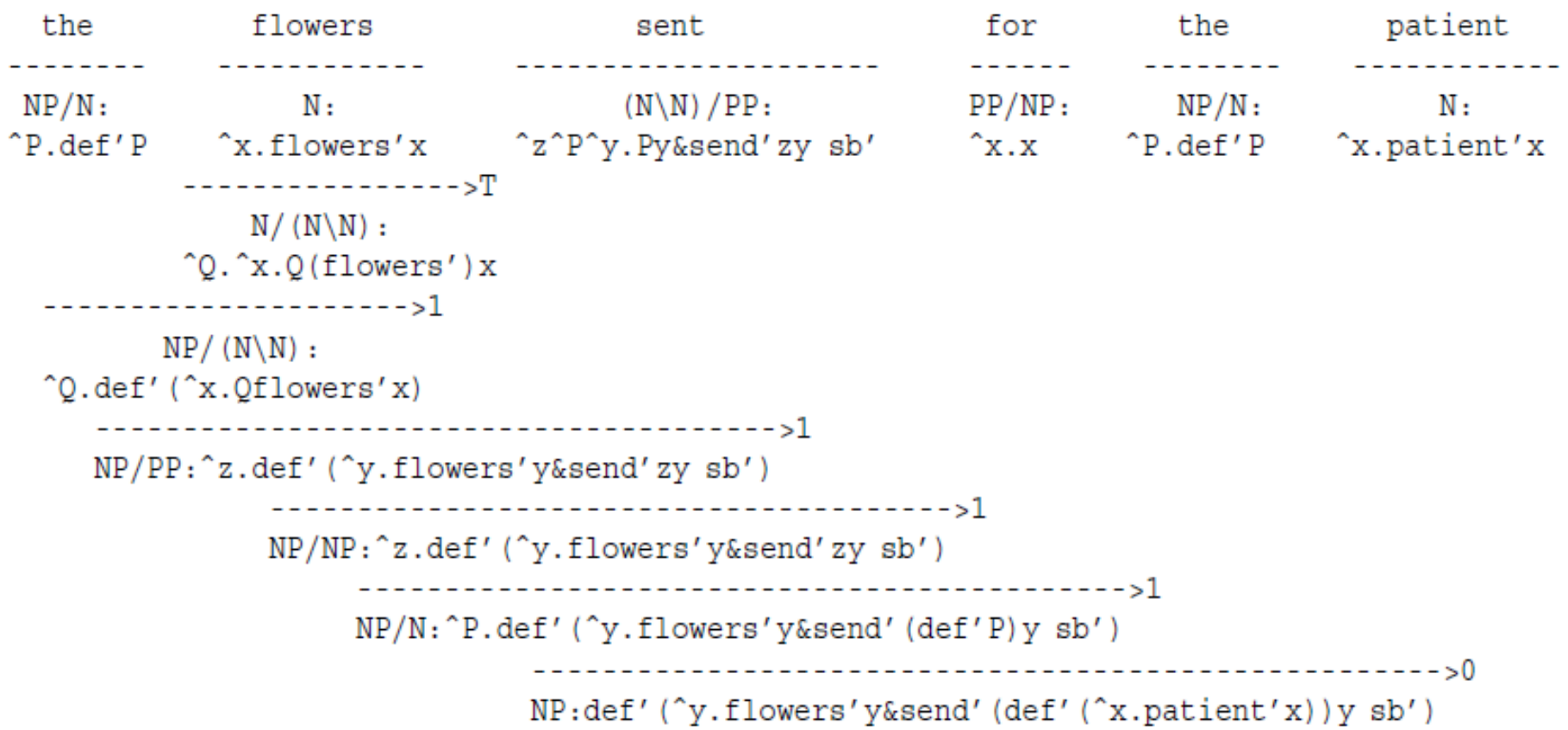


Figure: Incremental CCG derivation (Figure taken from McConville's PhD thesis.)

Time for notes

According to Steedman

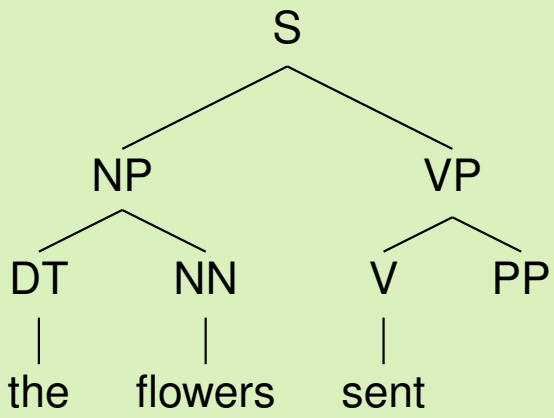
- What's the trouble with CFG top-down processing?
- What's the trouble with CFG bottom-up processing?
- How does he suggest to solve the paradox?
- Why/How does his suggestion satisfy
  1. Incremental Interpretation
  2. Being a sensible competence grammar
  3. the Strict Competence Hypothesis

Table of Contents

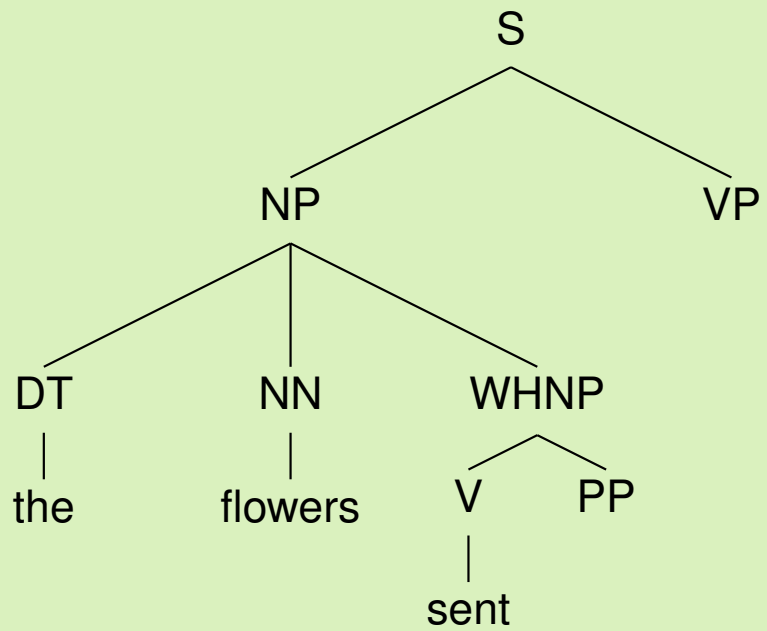
What are Stabler's arguments against the paradox?

- Strict competence hypothesis is unnecessarily strict.
- Parser can be constructed to combine semantics for non-constituents / incomplete constituents

Top-down parsing



sent'(x, (the' (flowers')))



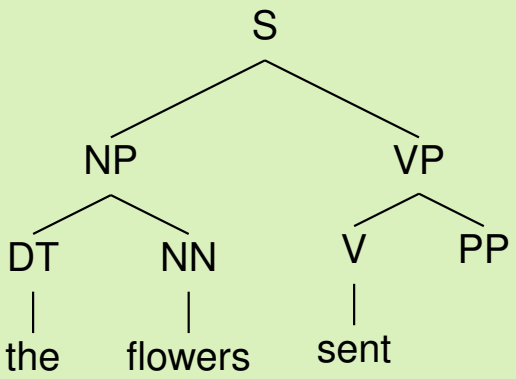
x((the' (flowers')) & sent'(y, flowers', somebody'))

why need to wait till x and y have been instantiated??

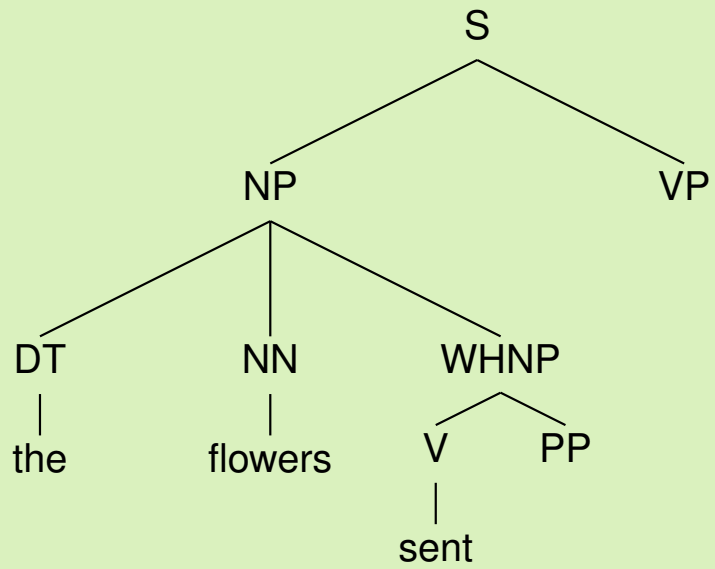
What are Stabler's arguments against the paradox?

- Strict competence hypothesis is unnecessarily strict.
- Parser can be constructed to combine semantics for non-constituents / incomplete constituents

Top-down parsing



sent'(x, (the' (flowers')))



x((the' (flowers')) & sent'(y, flowers', somebody'))

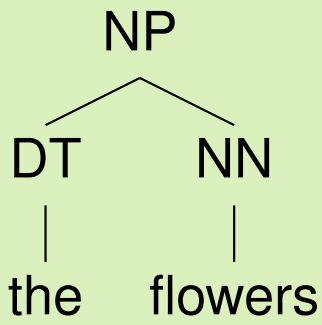
why need to wait till x and y have been instantiated?? $\Rightarrow$ Less complex architecture, don't need mechanism for dealing with partial semantic structures.

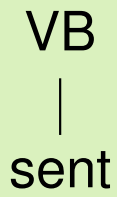
What are Stabler's arguments against the paradox?

- Strict competence hypothesis is unnecessarily strict.
- Parser can be constructed to combine semantics for non-constituents / incomplete constituents

Bottom-up parsing

Stabler suggests to change notation for semantics from prefix to postfix notation, but this still **requires establishing semantic relation where no syntactic relation has been determined**.





Several people have argued that his approach does not logically work out, so we're going to skip it here.

Time for notes

What's Stabler's argument against the paradox?

In how far is his interpretation of the competence hypothesis less strict?

Table of Contents

What are Abney and Johnson's arguments against top-down and bottom-up parsing?

Motivation #1: Which parsing strategy can explain processing difficulty phenomena such as **center embedding**?

Center embedding

a) The rat that the cat that the dog chased bit ate the cheese.
b) The dog chased the cat that bit the rat that ate the cheese.

Observations:

- a) is much more difficult to understand than b).
- a) requires holding too many incomplete substructures in memory.

Motivation #2: There's a trade-off between space requirements and amount of local ambiguity for parsing strategies – Is there an optimum?

Space requirements for parsing strategies

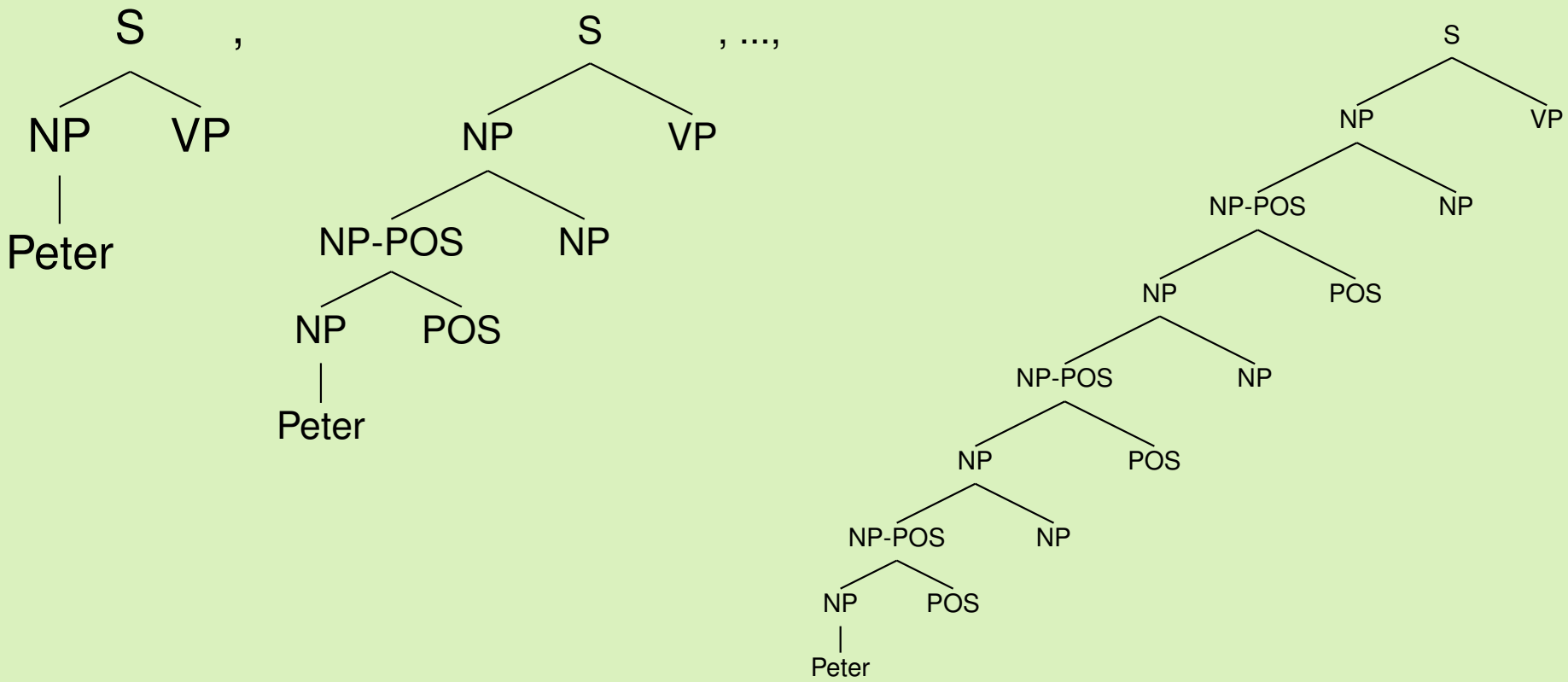
Structure	left	center	right
top-down	n	n/2 + 1	2
bottom-up	3	n/2 + 2	n+1
left-corner	2	n/2 + 1	3

Only left-corner strategy correctly predicts that center embedding structures are most difficult.

Amount of local ambiguity

Top-down strategy incurs much higher local ambiguity than either left-corner or bottom-up strategy.

Remember the left-recursion problem



Time for notes

Take notes:

- What is center embedding?
- Can you explain why which parsing strategy takes how much space for left branching / center embedded / right branching structures?
- What does this have to do with ambiguity?

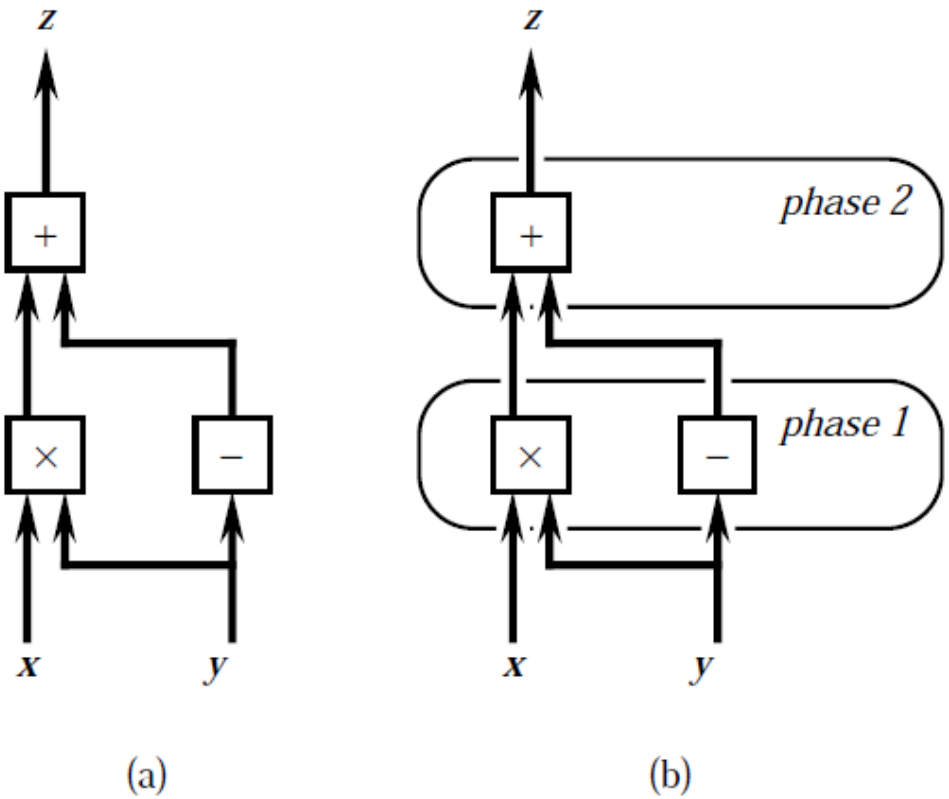
Table of Contents

What is Shieber and Johnson's alternative?

How Shieber and Johnson propose to **solve the Paradox**

- similar idea as Stabler: ~~strict~~ competence hypothesis
- but very different realization of the idea
- key idea: **asynchronous processing**
- strict competence requires synchronous processing
- **asynchronous processing simpler** since doesn't require synchronization mechanism
- using Synchronous Tree-Adjoining Grammar (S-TAG)

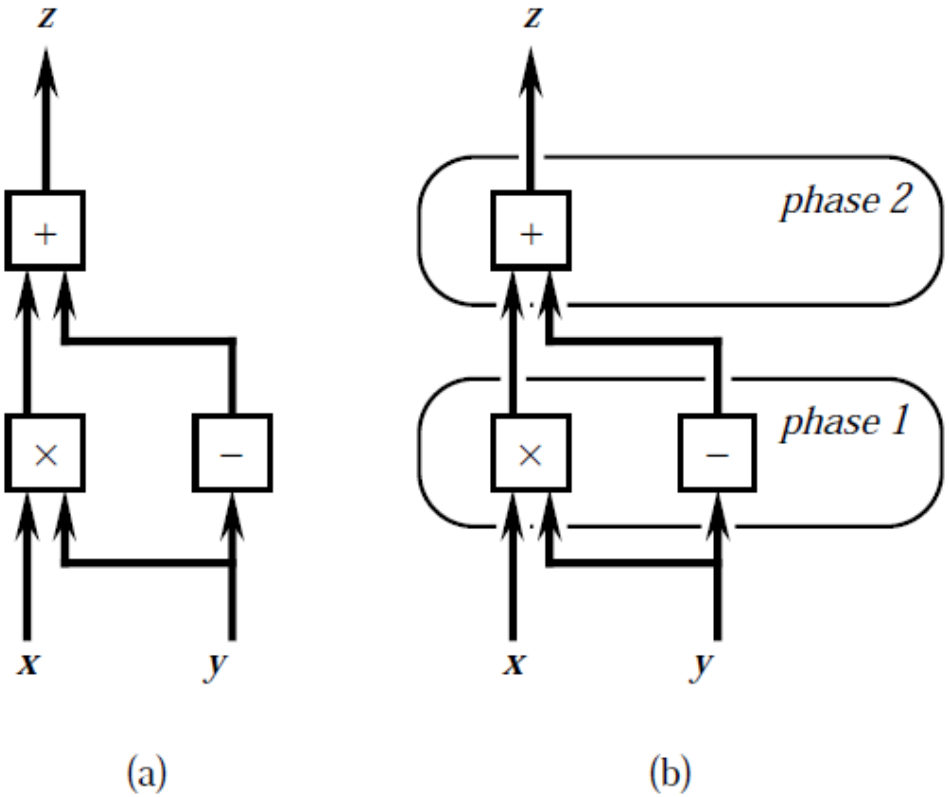
Asynchronous Processing



- (a) asynchronous process
- (b) synchronous process
- in some conditions, (a) can calculate z with only one of the inputs
- Examples: y=0 or x=1

Figure: Circuit for computing $z = xy + (-y)$

Asynchronous Processing

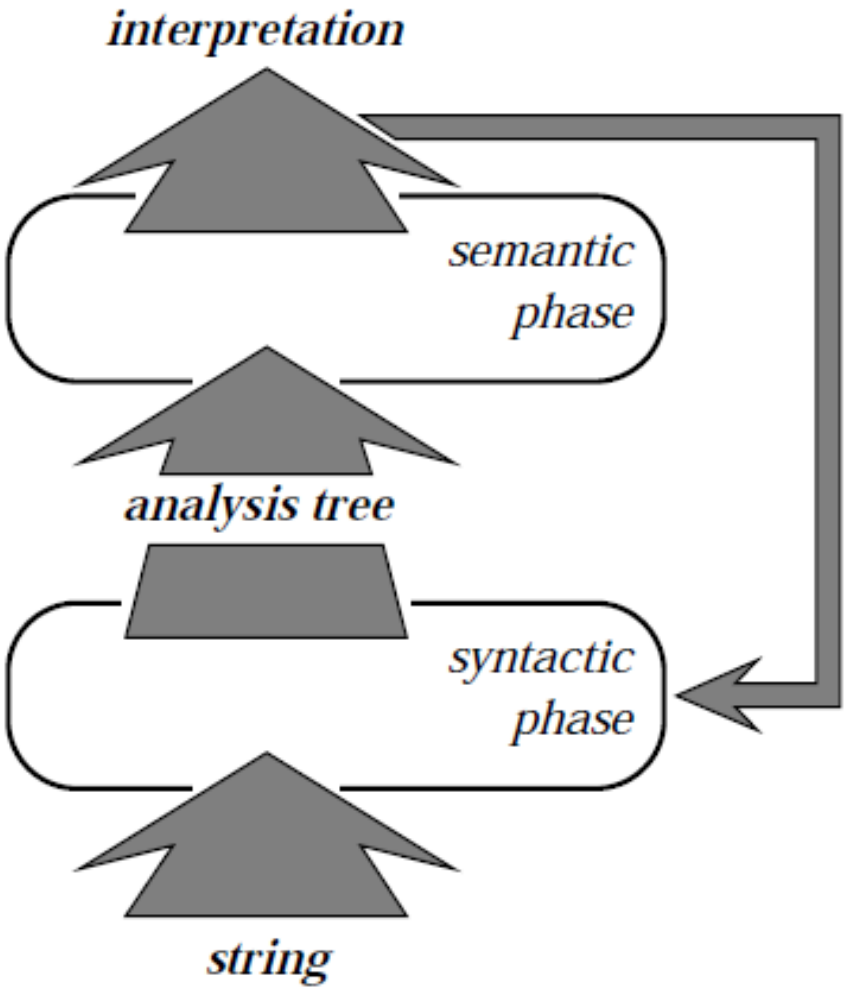


- (a) asynchronous process
- (b) synchronous process
- in some conditions, (a) can calculate z with only one of the inputs
- Examples: y=0 or x=1

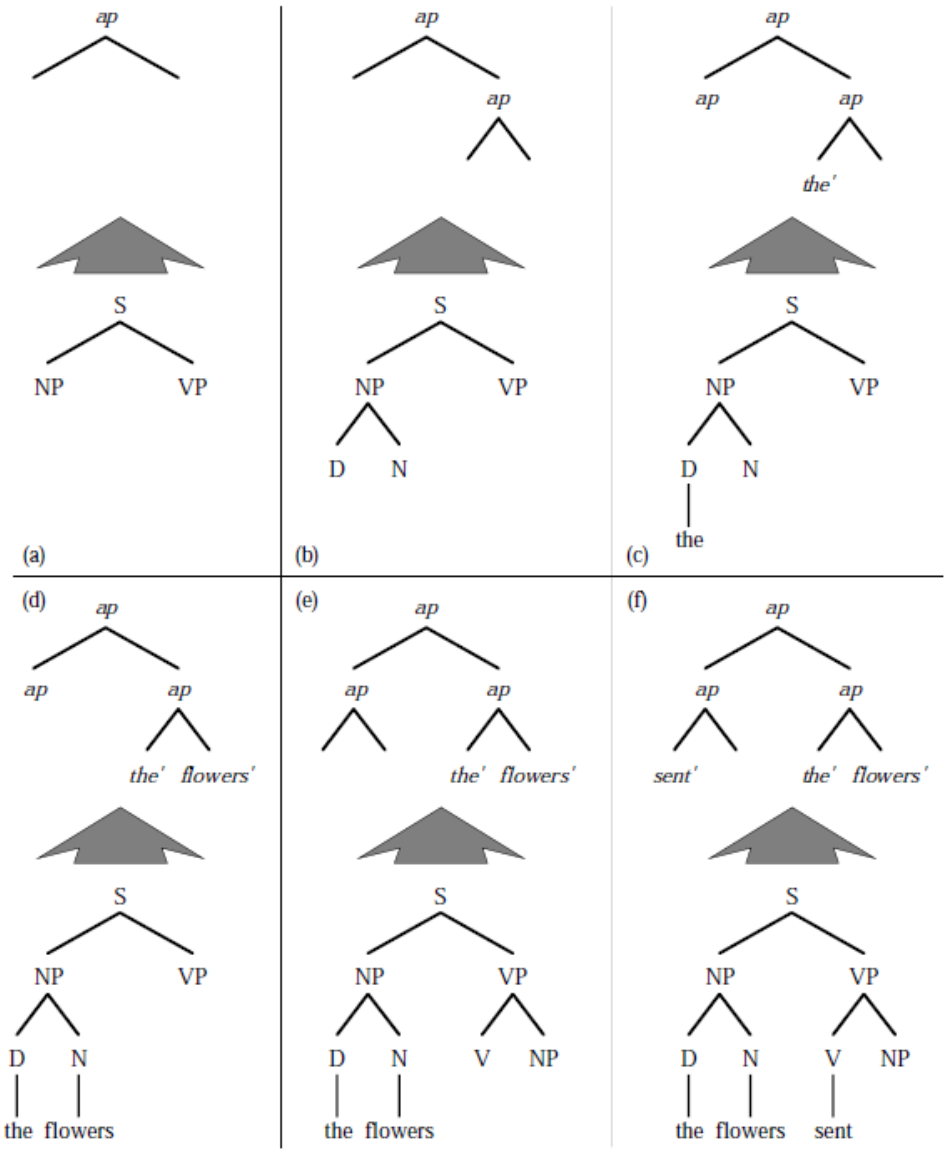
Figure: Circuit for computing $z = xy + (-y)$

But: How (non-)trivial is the abstraction of calculating with variables?

Applying the idea to human sentence processing



Asynchronous Processing Top-Down

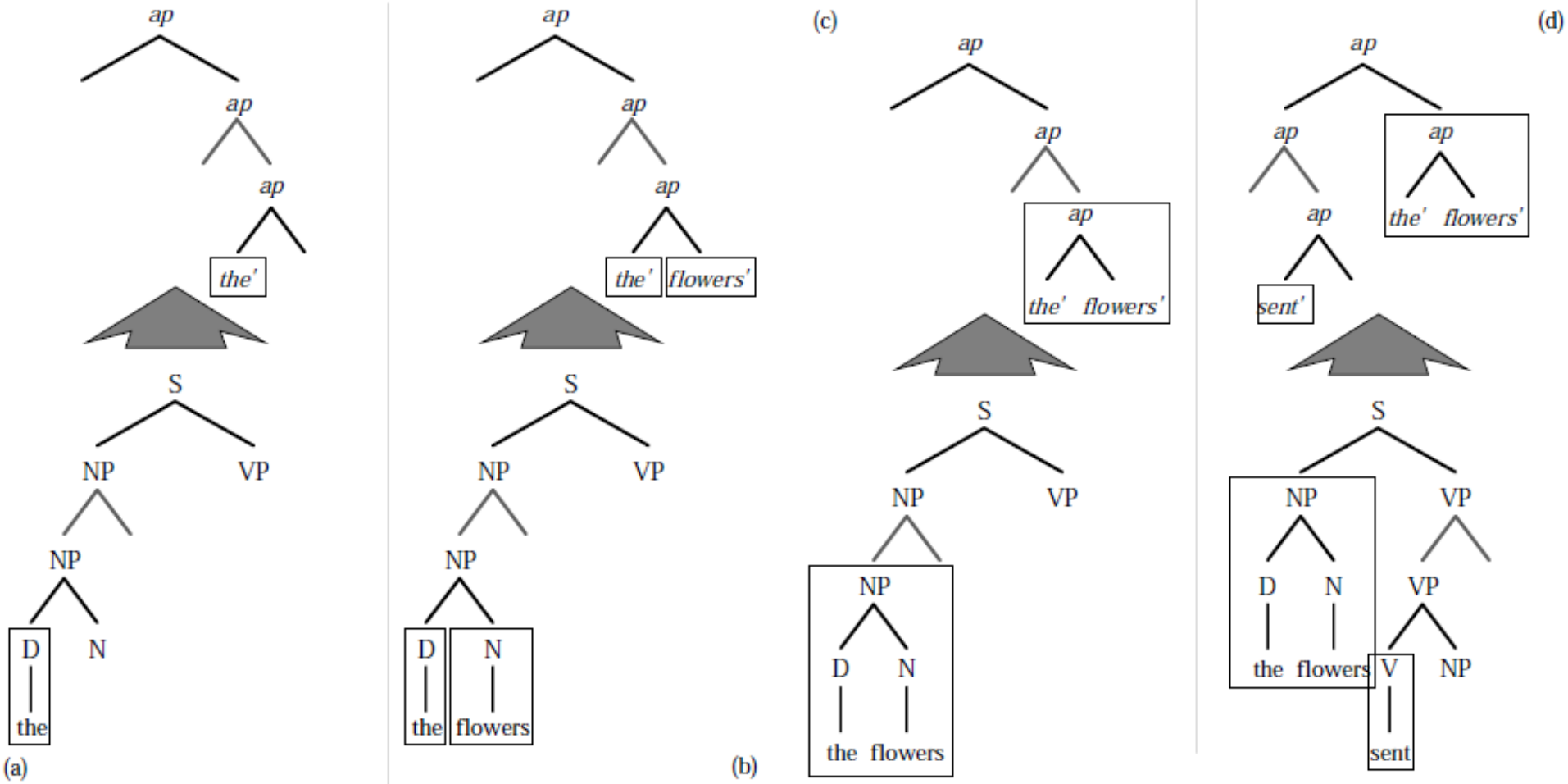


We don't need to see the complete constituent which "sent" is part of. **Hence we achieve Incremental Interpretation!**

But note:

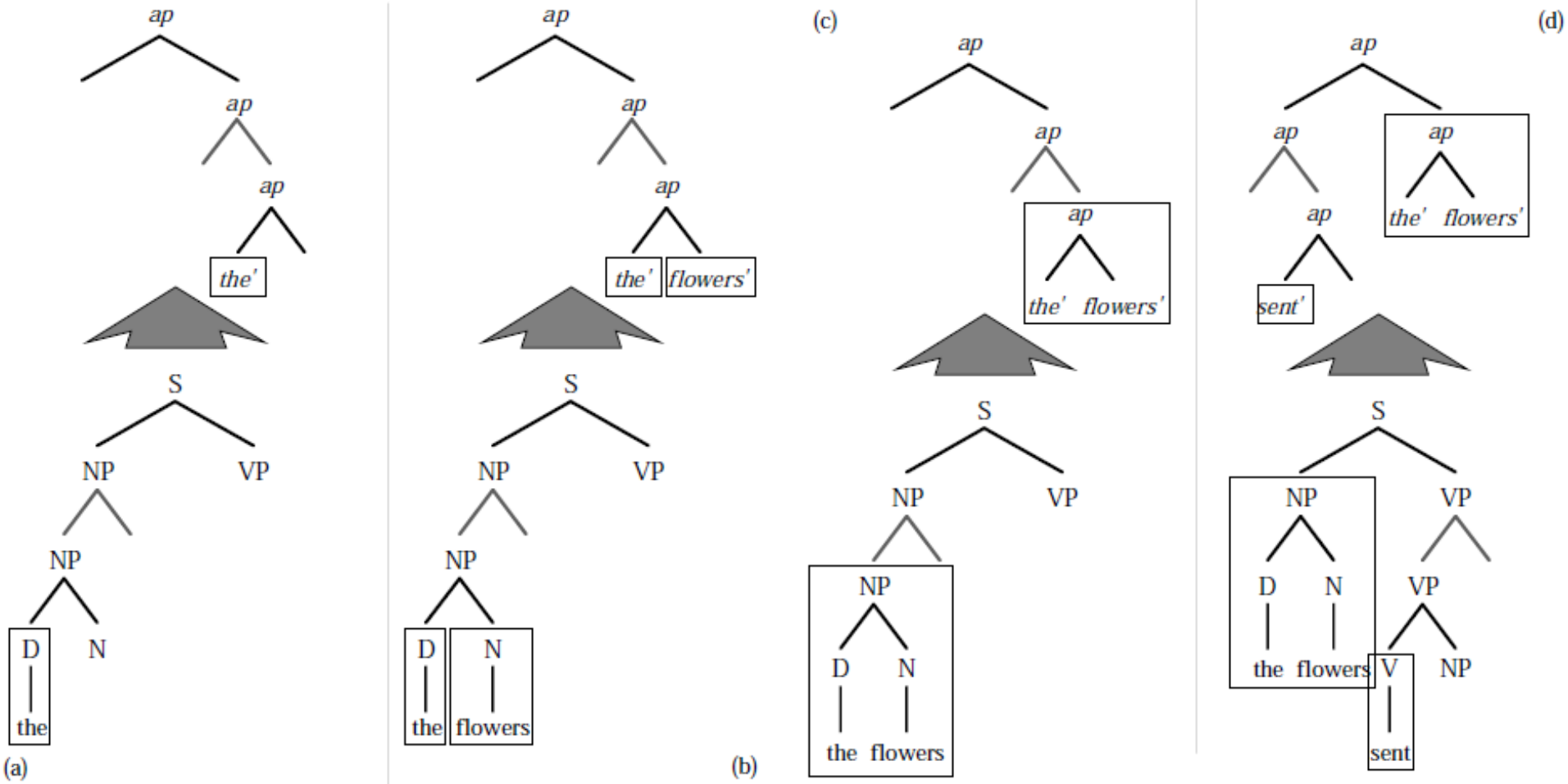
- this does not resolve left recursion
- or center embedding problem
- or large ambiguity
  (early commitment to structure)

Asynchronous Processing Bottom-Up



- boxed structures is what the parser has actually built
- unboxed structures are defined by equivalence classes

Asynchronous Processing Bottom-Up



- boxed structures is what the parser has actually built
- unboxed structures are defined by equivalence classes

Asynchronous Processing Bottom-Up

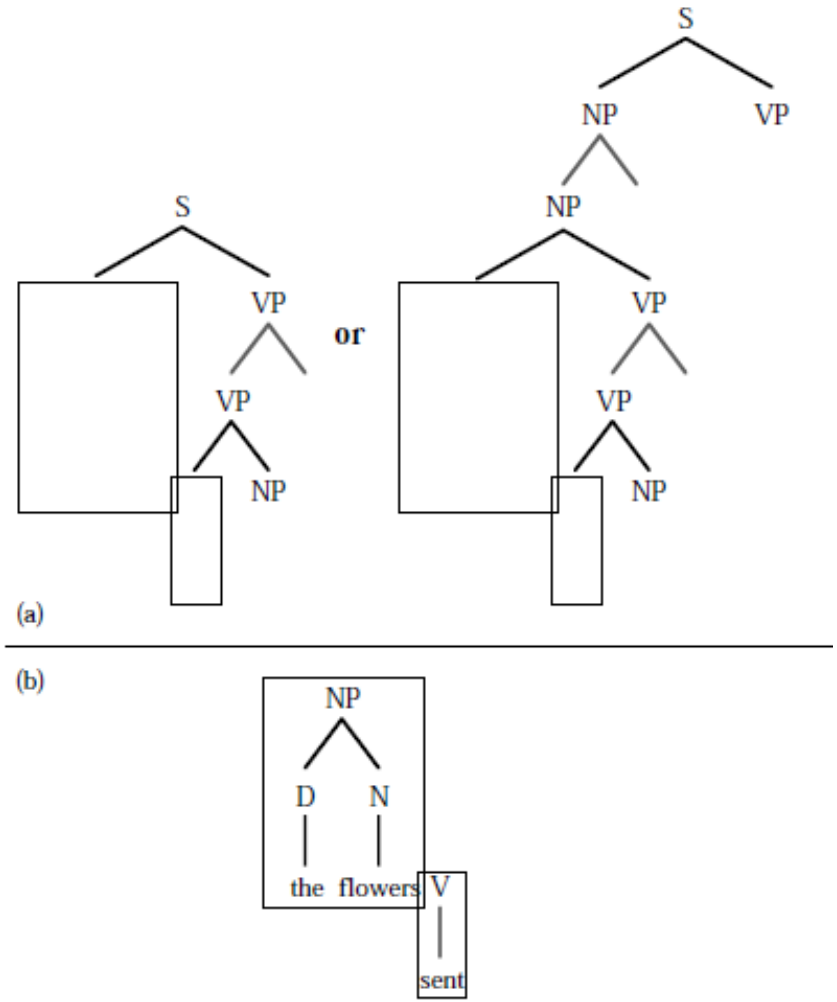


Figure: Syntax

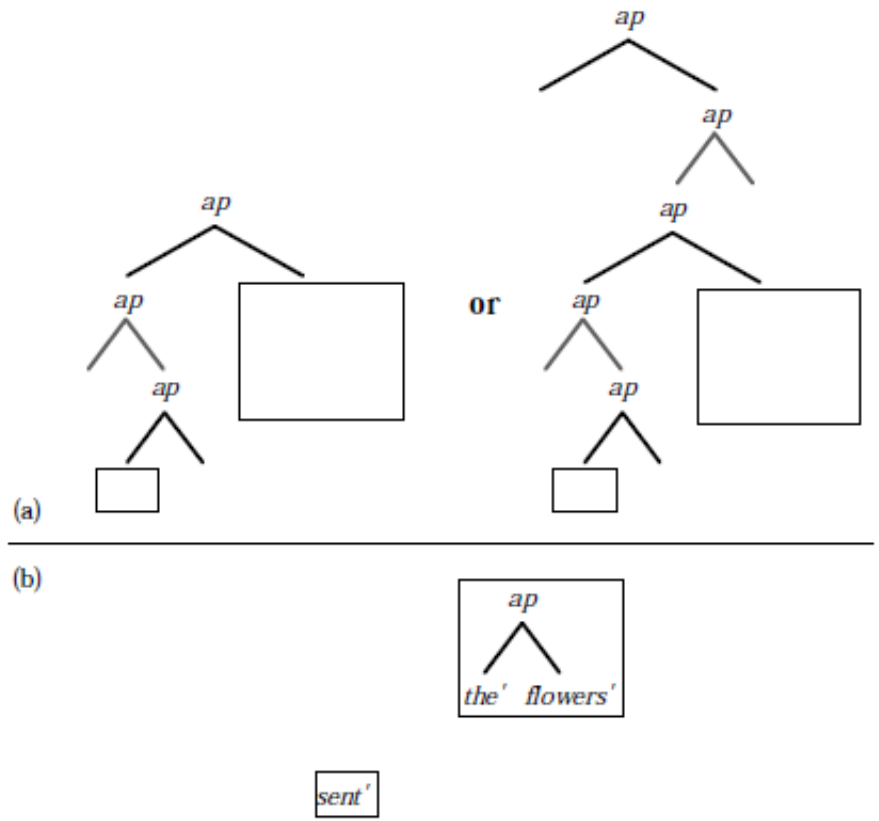
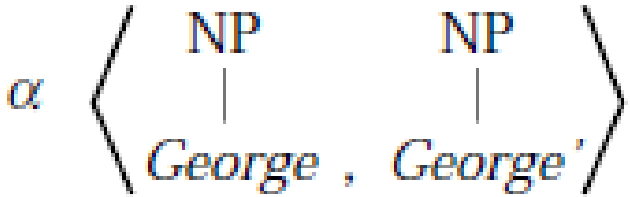
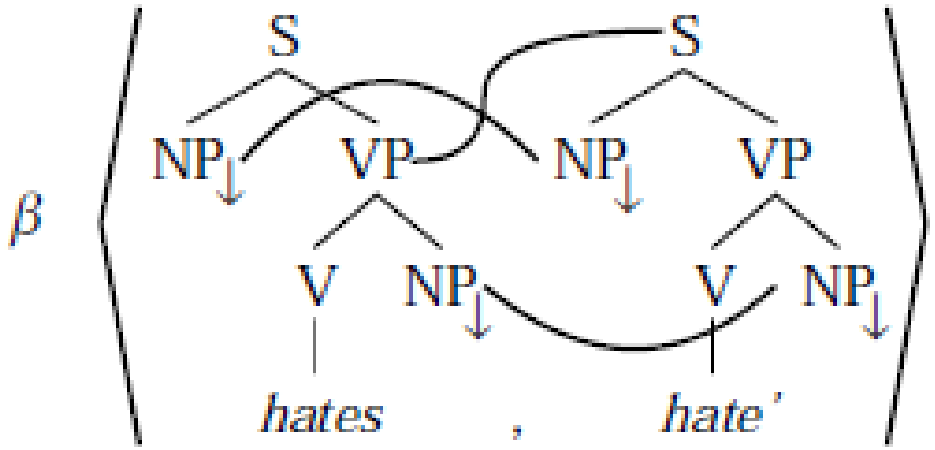


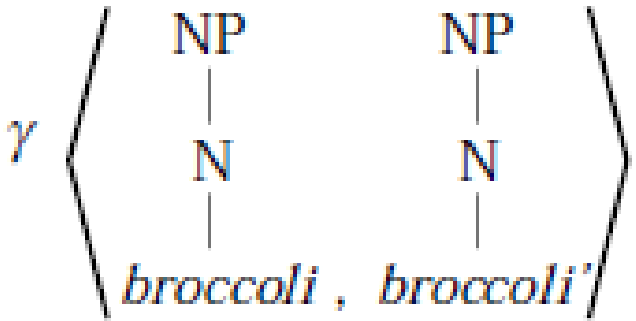
Figure: Semantics – Now parser can decide to discard the first interpretation.
How can we reasonably calculate all those equivalence structures (outside boxes)?

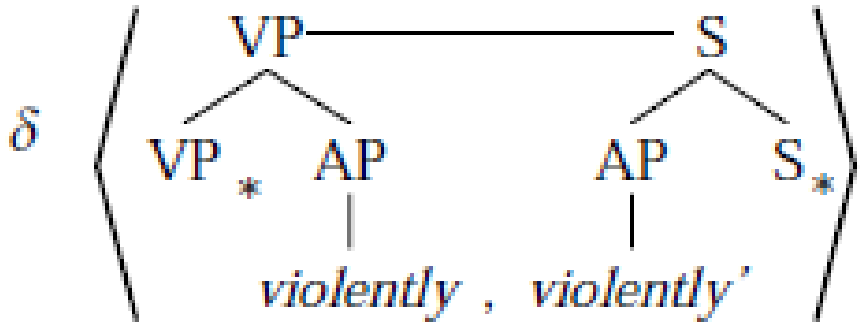
Synchronous TAG

synchronous = syntactic and semantic trees are paired









Synchronous TAG

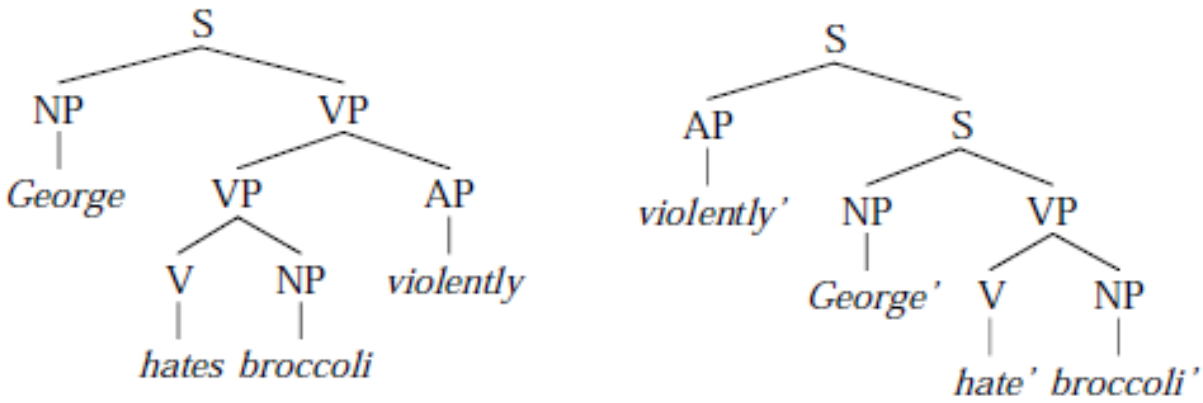


Figure 12: Adjunction to VP at s-structure and corresponding adjunction to S at LF

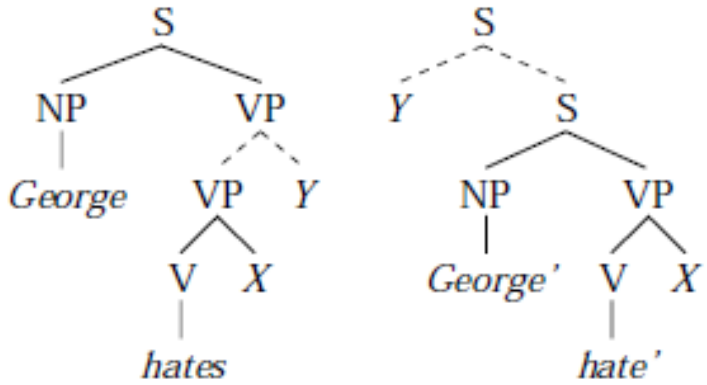


Figure 13: "Partial trees" constructed at the word *hates*

Critique on Shieber and Johnson's approach

Critique: extra machinery to inspect internal state of parser

syntax-semantic interface much simpler if syntactic relations are explicit and not implicit ($\Rightarrow$ need compiled version of competence grammar so we know instantly what structures are possible)

Time for notes

Take notes:

- What is asynchronous processing?
- How is incremental interpretation achieved in top-down processing?
- How is incremental interpretation achieved in bottom-up processing?
- What about the strict competence hypothesis in this approach?
- What's the advantage of using S-TAG as opposed to a normal context-free grammar?

Summary

- **Incremental Interpretation and Rule-to-Rule assumption**
  - need syntactic structure before we can achieve semantic interpretation
  - otherwise need additional machinery to inspect internal state of parser (Shieber & Johnson)
- **Strict Competence Hypothesis**
  - Why would we want it: simplest possible relationship between competence grammar and processor, and between syntax and semantics.
  - Problem: paradox of simultaneous incremental interpretation, strict competence and right-branching structures
- **Solutions to Paradox**
  - less strict version of competence hypothesis
  - or constructing semantics without syntactic relationships

Summary

Parsing Processes:

	reasons pro:	reasons contra:
top-down	– syntax connected, so can achieve semantic interpretation	– left recursion problem – can't explain center-embedding – large degree of ambiguity
bottom-up	– no recursion problem – less ambiguity	– either syntax unconnected or need left-branching constituents – can't explain center-embedding
left-corner	– explains center embedding – syntax connected	– left recursion still a problem – more ambiguous than bottom-up

Outlook

The seminar is going to address:

- Experimental evidence for incremental interpretation, incremental syntax in addition to the “center embedding” argument we’ve seen today
- Incremental algorithms, observe:
  - psycholinguistically plausible?
  - incremental syntax / incremental semantics?
  - top-down / bottom-up / left-corner / ?
  - how does it deal with the challenges (left recursion, ambiguity, missing connectedness etc.)
- Applications: practical impacts of incrementality

Finally...

Please give me feedback on this taking-notes-after-each-section method.

TODO for next week:

- Please read at least one of the papers for topic APP1
- Come up with at least 2 good questions / interesting points to discuss!
- poll in the second doodle!