Computational Psycholinguistics Lecture 2: Parsing Mechanisms

Matthew W. Crocker

Marty Mayberry

Human Language Processing

- We understand language incrementally, word-by-word
  - How do people construct interpretations
- We must resolve local and global ambiguity
  - How do people decide upon a particular interpretation
- Decisions are sometimes wrong!
  - What information is used to identify we made a mistake
  - How do we search for an alternative
- Answers can reveal important details about the underlying mechanisms

Experimental Methods

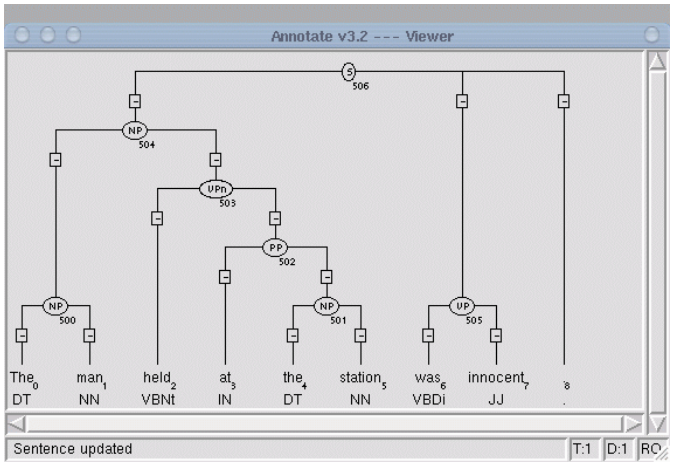
Reading times

Neuroscientific methods

Situated spoken sentence comprehesion

The Problem

- How do people recover the meaning of an utterance in real-time?



"The man held at the station was innocent"

Crocker & Brants, *Journal of Psycholinguistic Research*, 2000.

Reading time studies

- We can use controlled experiments of reading times to investigate local ambiguity resolution
- (a) The man held at the station was innocent (LA)
- (b) The man who was held at the station was innocent (UA)
- Compare the reading times of (b) where there is no ambiguity, with (a) to see if and when the ambiguity causes reading difficulty.
  - Need a “linking hypothesis” from theory to measures
  - Can then manipulate other linguistic factors to determine their influence on on RTs in a controlled manner

Reading Methods

- Whole sentence reading times:

```
The man held at the station was innocent
```

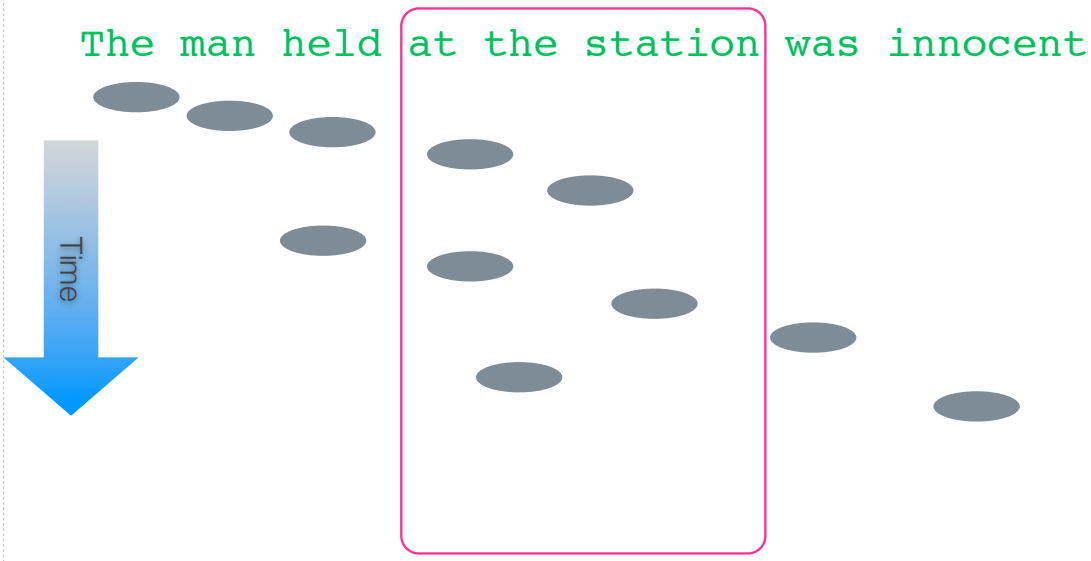
- Self-paced reading, central presentation:

```
                    is innocent
```

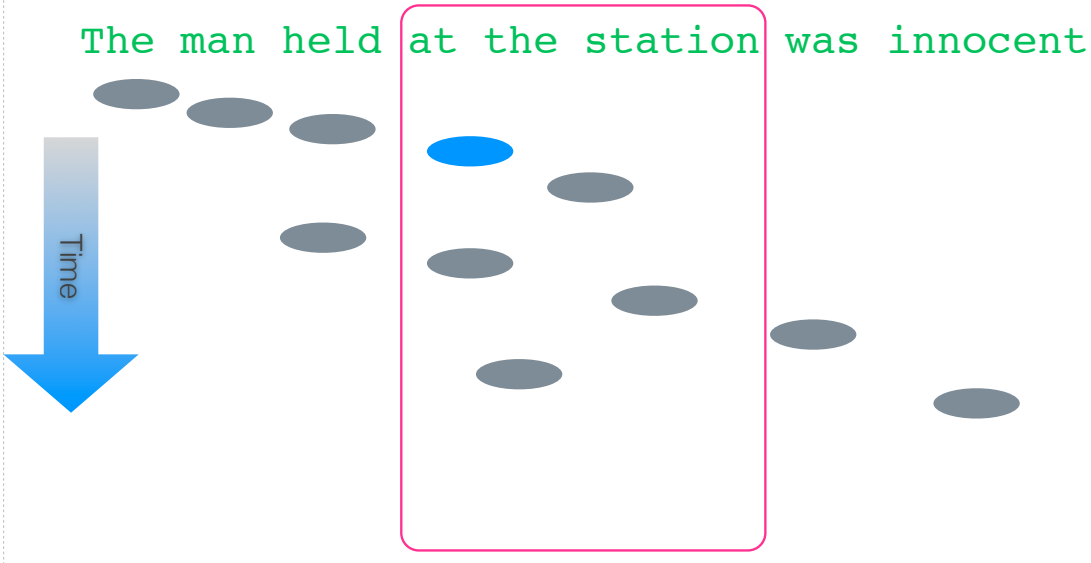
- Self-paced reading, moving window:

```
 - The man held at the station was innocent
```

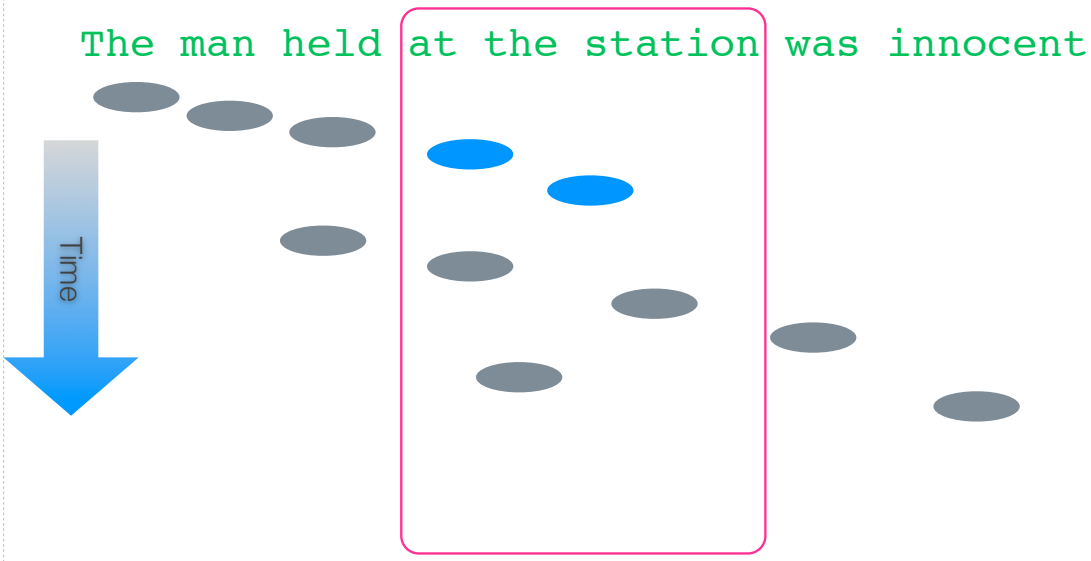
Eye-tracking: Difference Measures



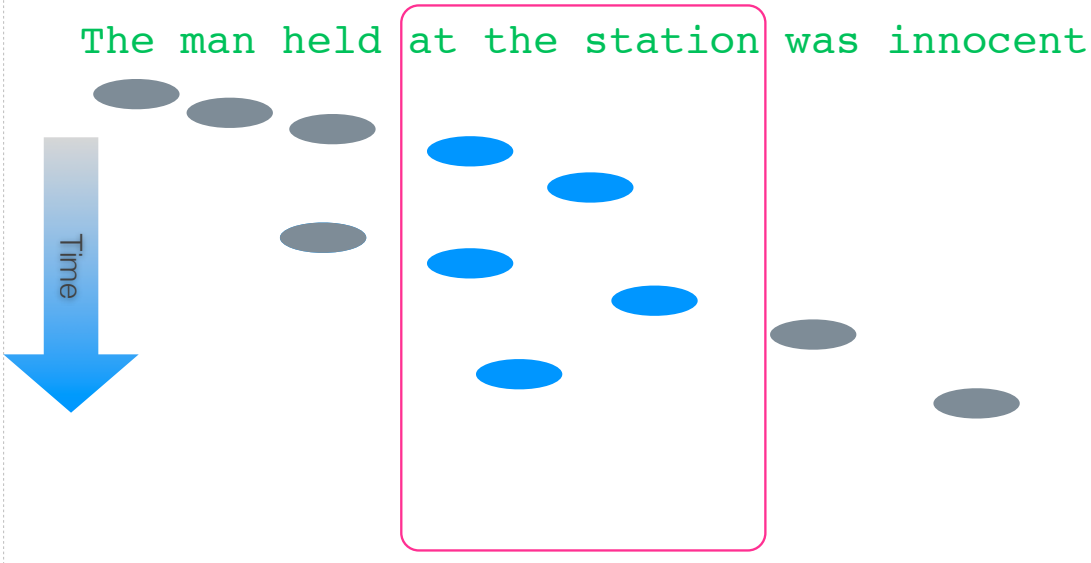
Eye-tracking: First Fixation



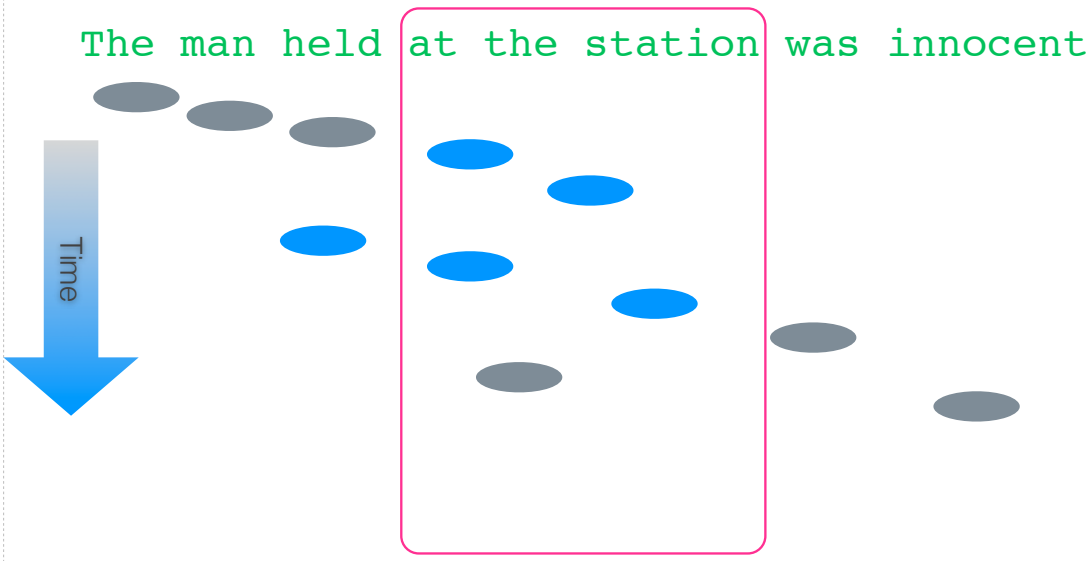
Eye-tracking: First Pass



Eye-tracking: Total time

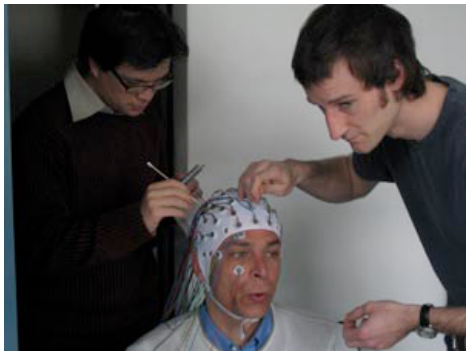


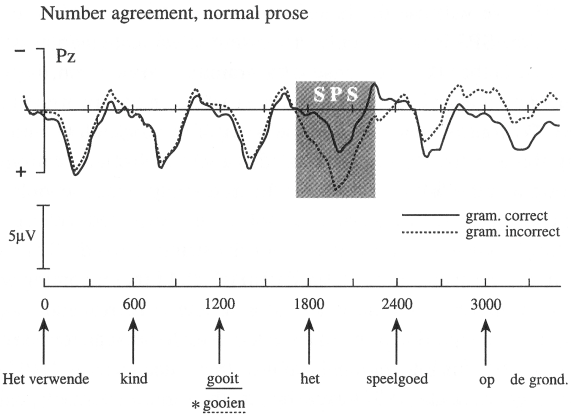
Eye-tracking: Regression Path



Neuroscientific Measures: ERPs

- Syntactic and semantic processes are partially revealed by signature patterns in EEGs: Event-Related Potentials (ERPs)
- Syntactic Anomaly: P600 or SPS

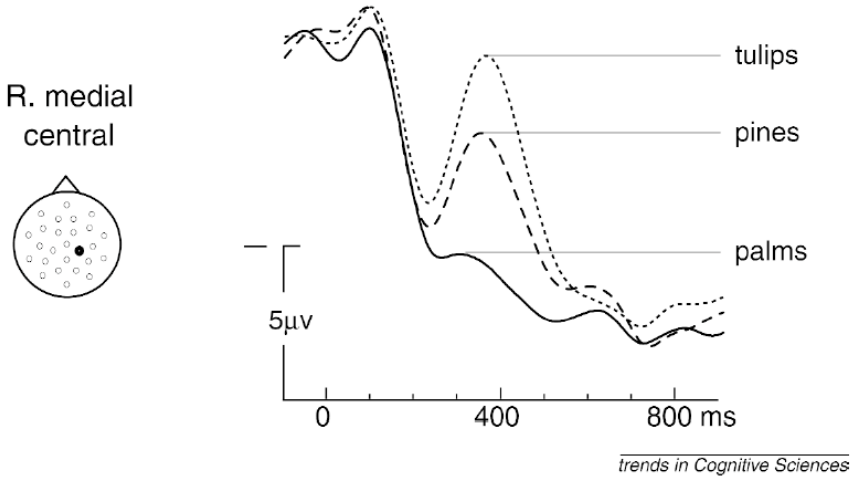




"The spoilt child throw(s) the toy on the ground"

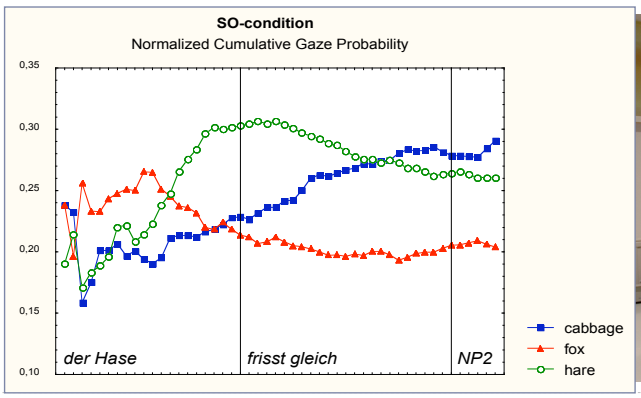
Semantic Anomaly: N400

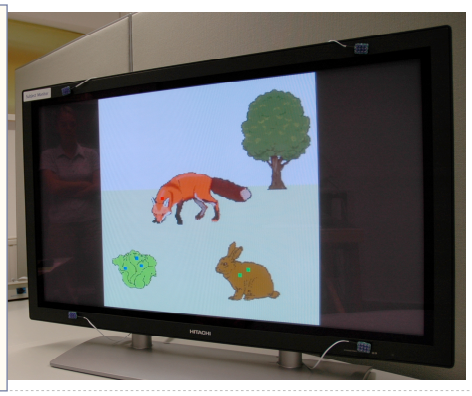
'They wanted to make the hotel look more like a tropical resort.
So along the driveway they planted rows of ...'



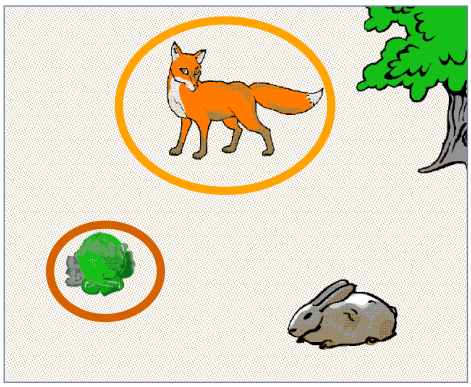
Spoken comprehension in visual

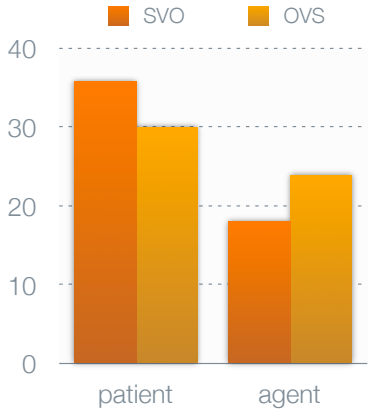
- Monitor gaze in the scene as people hear a spoken utterance
  - Listeners fixate objects which are mentioned (180ms)
  - Anticipatory eye-movements reflect interpretation





Anticipation in Visual Worlds





SVO: *Der Hase frisst gleich* ***den Kohl***
"The rabbit eats soon the cabbage"
OVS: *Den Hasen frisst gleich* ***der Fuchs***
"The rabbit is eaten soon by the fox"

Kamide, Scheepers & Altmann, *JPR*, 2003

Summary

- People construct an interpretations, word-by-word
  - People must resolve ambiguity
  - Sometimes we must revise our interpretation of the sentence so far
- On-line measures can tell us about how/when this occurs
  - Reading times, ERPs, gaze in visual scene
- We can design experiments which exploit these methods (and others!) to investigate the underlying processing architectures and mechanisms

Linking Hypotheses

- Reading times: relative processing difficulty
  - correlated with processing complexity and reanalysis
- Visual attention: reference and anticipation
  - correlated with interpretation and inference
- N-400: semantic anomaly
  - correlated with semantic integration
- P-600/SPS: syntactic anomaly
  - correlated with disambiguation and reanalysis

Parsing Mechanisms

- Syntactic processing requires a solution to the problem of:
  - How structures are incrementally constructed
  - How local and global ambiguity
- Incremental Parsing
  - Top-down; Bottom-up; Mixed strategies
- Ambiguity and parsing:
  - Serial (deterministic/non-deterministic)
  - Parallel (bounded/unbounded)

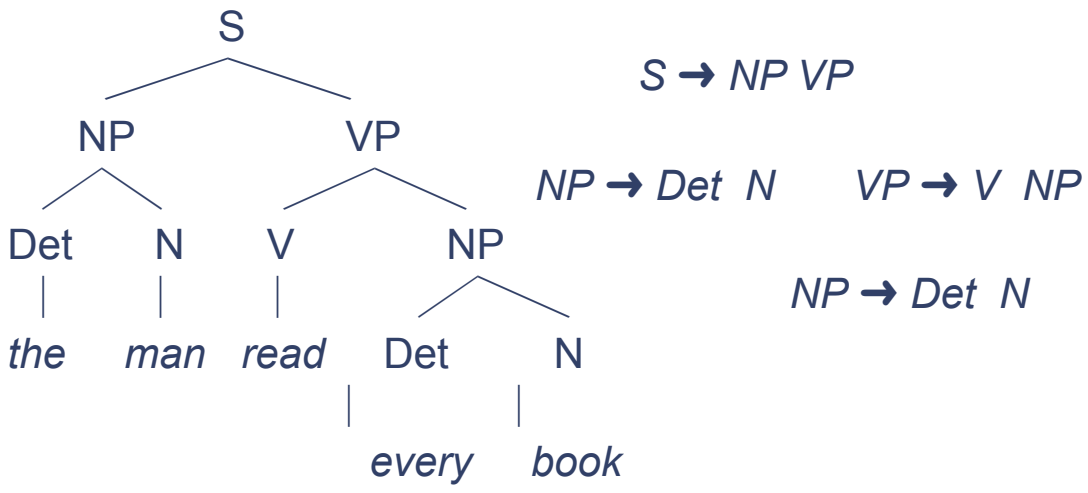
Context Free Grammars

- Context-free grammar rules:

S → NP VP	Det → the
PP → P NP	Det → every
VP → V NP	N → man, woman
VP → V	N → book
NP → NP PP	P → with
NP → Det N	V → read, reads

- Node admissibility criterion:
  - A tree is admitted by the grammar, if for each non-terminal node, N, with daughters Ds, there is a rule in the grammar of the form: N → Ds.

Simple example

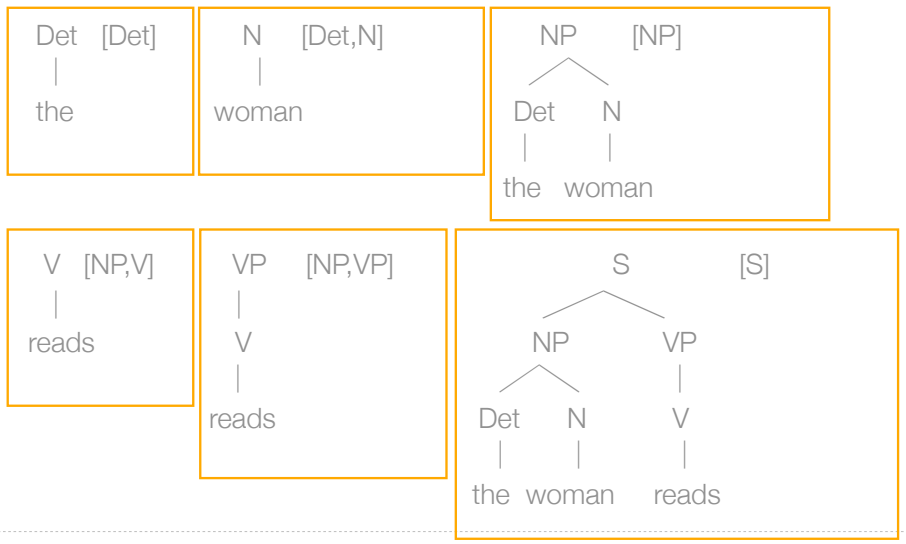


Parsing Algorithms for PSGs

- Algorithms to recover the parse tree for an utterance vary ...
  - left-to-right, head-driven, right to left
  - top-down, bottom-up, mixed
  - deterministic, serial, parallel
- Processing complexity:
  - Time: what time is required to parse a sentence as a function of sentence length, grammar size?
  - Space: how much memory does the parser require?

Bottom-up Parsing

"The woman reads"

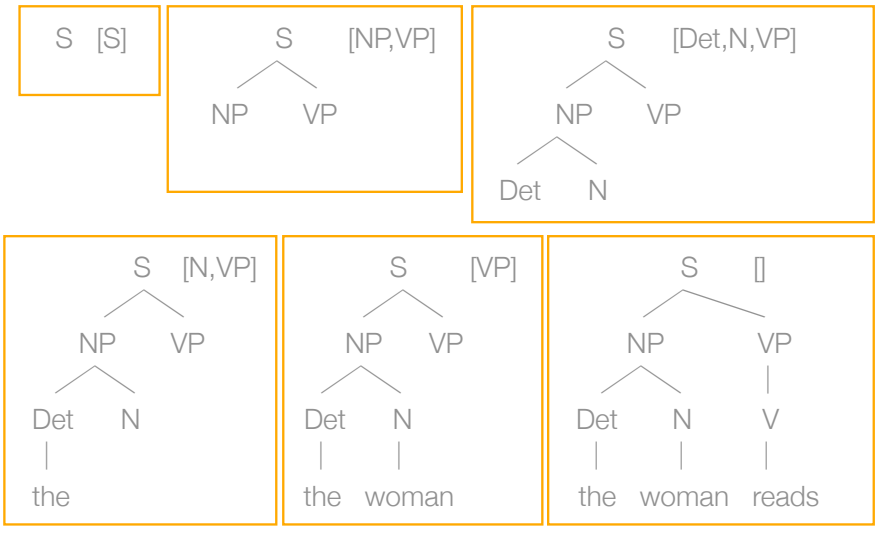


Shift-reduce Algorithm

1. Initialise *Stack* = []
2. loop: Either *shift*:
   - Determine category, *C*, for next word in sentence;
   - Push *C* onto the stack;
3. Or *reduce*:
   - If categories on the *Stack* match the RHS of a rule:
     - Remove those categories from the *Stack*;
     - Push the LHS category onto the *Stack*;
4. No more words to process?
   - If *Stack* = [S], then done;
5. Goto loop

Top-down Parsing

"The woman reads"



Top-down Algorithm

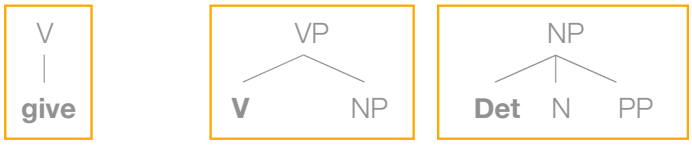
1. Initialise *Stack* = [S]
2. If top(*Stack*) is a non-terminal, *N:*
   - Select rule *N* → *RHS*;
   - pop(*N)* off the stack and push(*RHS*) on the stack;
3. If top(*Stack*) is a pre-terminal, *P:*
   - Get next word, W, from the input;
   - If *P* → *W*, then pop(*P*) from the stack;
   - Else fail;
4. No more words to process?
   - If *Stack* = [], then done;
5. Goto ②

Evaluating top-down & bottom-up

- Are these parsers psychologically plausible?
- Incrementality:
  - Bottom-up: no
  - Top-down: yes
- Input-driven:
  - Bottom-up: yes
  - Top-down: no + problems with left-recursion

A Psychologically Plausible Parser

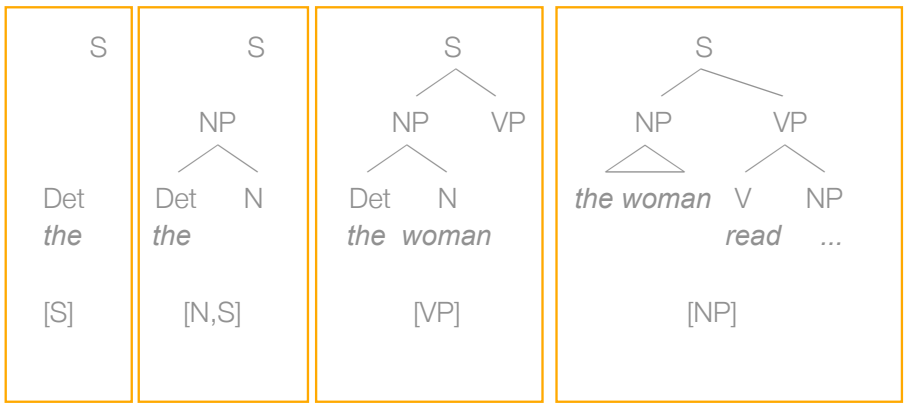
- Left-Corner Parsing
- Rules are 'activated' by their 'left-corner'



- Combines input-driven with top-down
- There is a 'class' of LC parsers

An example LC parse

"The woman read the book"



Is this incremental?

Evaluating the LC Parser

- Not necessarily incremental:
  - Variations: Arc-standard *versus* arc-eager

- Affect on ambiguity resolution for arc-eager:
  - Commitment to attachments is early, before daughters are completely built
  - Top-down use of syntactic context and possible left-recursion problems

Incrementality and Memory

- It wasn't incrementality that led to the LC algorithm, but memory load:
  - *"The mouse died"*
  - *"The mouse the cat chased died"*
  - *"The mouse the cat the dog bit chased died"*

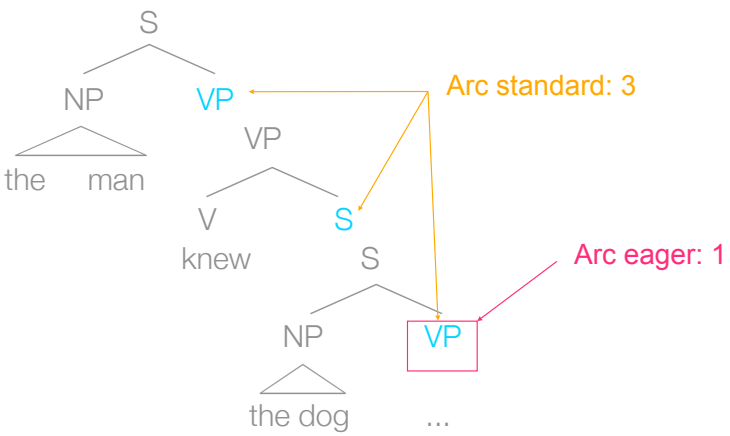
    (Or: "The mouse that the cat that the dog bit chased died")
- Grammatical, not ambiguous, what's the problem?
- Memory load: too high for centre embedding
  - *"[The mouse [the cat [the dog bit] chased] died]"*

Memory Load in Parsing

- Left-embedding is easy: *[[[John's brother]'s car door]'s handle] broke off.*
- Right-embedding too: *John believes [Bill knows [Mary said [she likes cats]]]*
- Centre-embedding is hard: *[The mouse [the cat [the dog bit] chased] died]*
- Memory load for parsers:
  - Top-down: LE: hard CE: hard RE: easy
  - Bottom-up: LE: easy CE: hard RE: hard
  - Left-corner: LE: easy CE: hard RE: easy

Evaluating the LC Parser

- Variations: Arc-standard versus Arc-eager



Summary of Behaviour

Node	Arcs	Left	Centre	Right
Top-down	Either	O(n)	O(n)	O(1)
Shift-reduce	Either	O(1)	O(n)	O(n)
Left-corner	Standard	O(1)	O(n)	O(n)
Left-corner	Eager	O(1)	O(n)	O(1)
People		O(1)	O(n)	O(1)

Comments on Left-Corner

- Mixed data-driven and hypothesis driven approaches
  - Eager corresponds to composition of partial structures
- Arc Standard: less ambiguity
  - attach when constituents are complete: safer
  - delayed attachment means more is kept on the stack
- Arc Eager: less memory
  - early composition reduces stack growth
  - eager attachments are less bottom-up