Theories of Processing
Lecture 3
Introduction to Psycholinguistics

Matthew W. Crocker
Pia Knoeferle

Department of Computational Linguistics
Saarland University

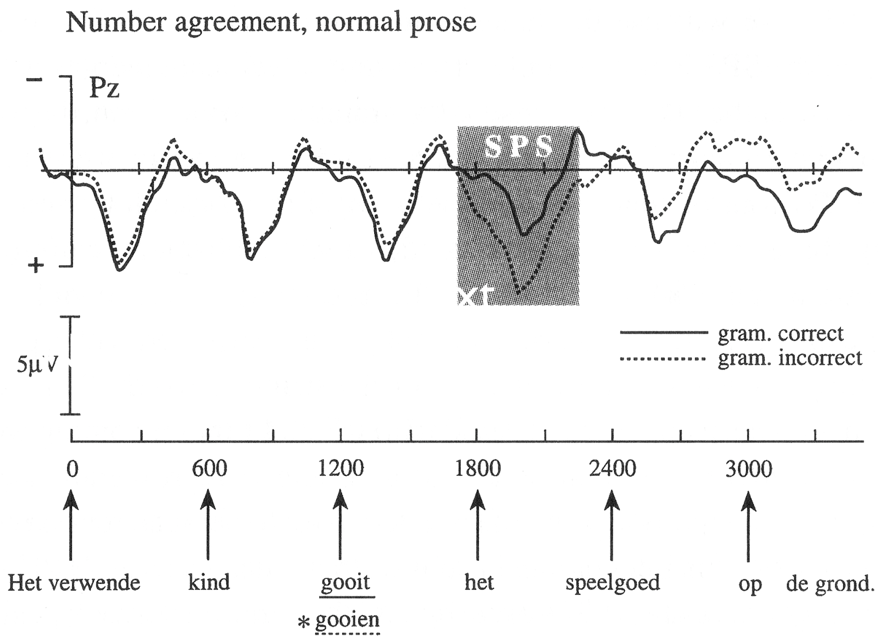
Neuroscientific Methods

In addition to examining cases of aphasia, we can also monitor on-line brain response to language in "normals"

Event-Related Potentials (ERPs)

- Voltage changes on the surface of the scalp in response to some stimulus and/or cognitive process
- Different stimuli cause different spatiotemporal potentials
  - LAN: syntactic and morphosyntactic violations
  - N400: semantic integration/anomaly
  - P600: syntactic disambiguation and reanalysis

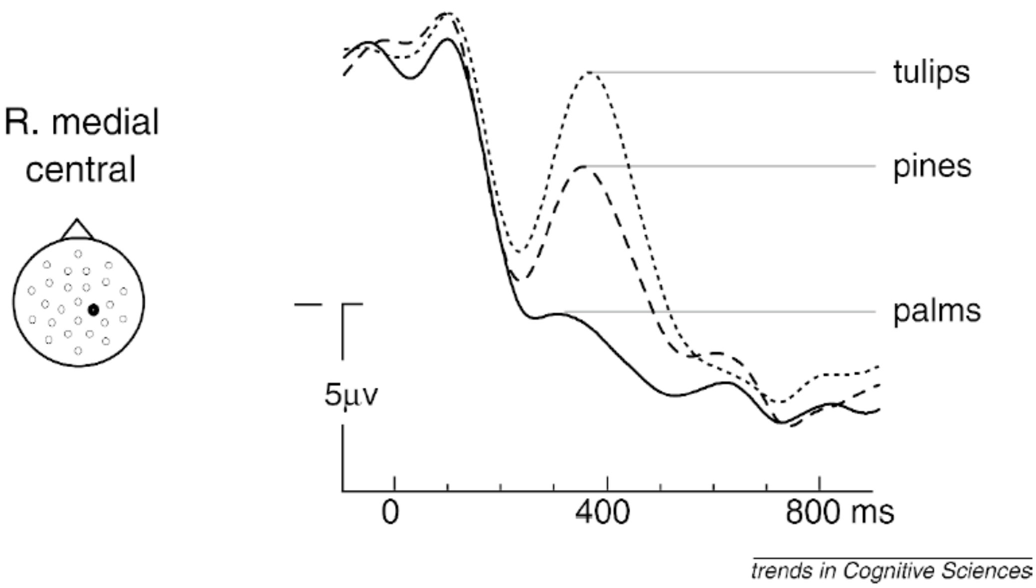
Syntactic Anomaly: P600 or SPS



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

Semantic Integration: N400

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



Summary of On-line Methods

People construct interpretations, word-by-word

- People resolve ambiguity immediately
- Sometimes we must revise our interpretation

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

Different methods reveal different aspects of the underlying architectures and mechanisms

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

Mechanisms for Language Processing

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 strategy

Ambiguity and parsing

- Serial (deterministic/non-deterministic)
- Parallel (bounded/unbounded)

Context Free Grammars

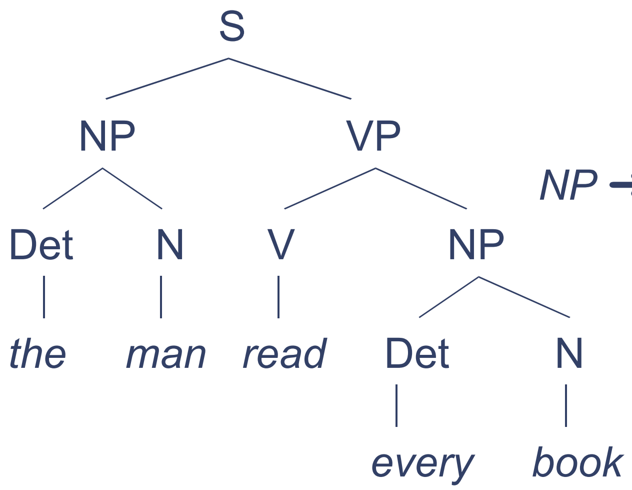
Context-free grammar rules:

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



S → NP VP

NP → Det N　　*VP → V NP*

NP → Det N

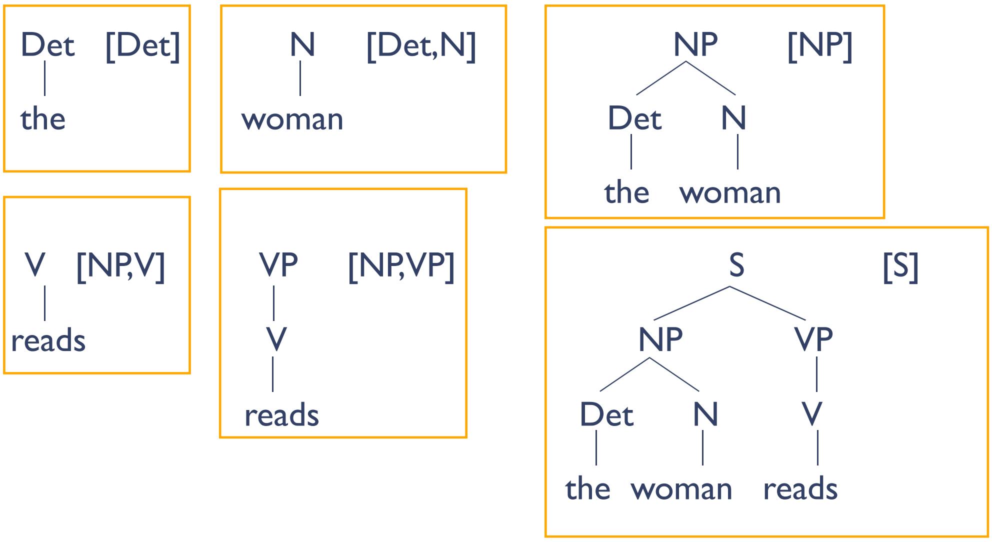
Parsing Algorithms for PSGs

An algorithm to recover the parse tree for an utterance, given that it is in the language

- Dimensions of variation:
  - 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"

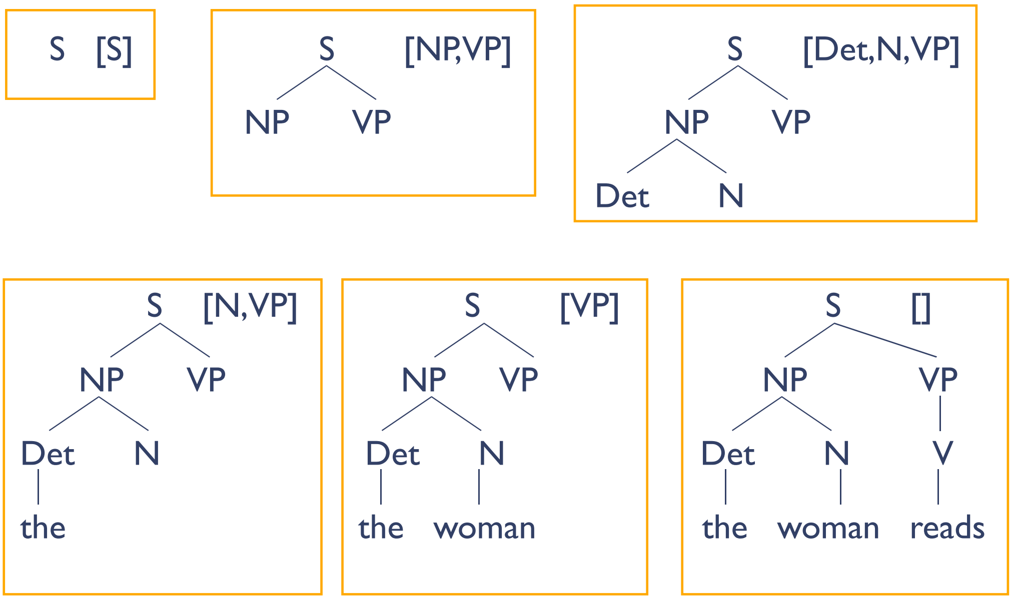


Shift-reduce Algorithm

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

❶ Initialise *Stack* = [S]

❷ If top(*Stack*) is a non-terminal, *N:*

- Select rule *N* ➝ *RHS*;
- pop(*N)* off the stack and push(*RHS*) on the stack;

❸ 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;

❹ No more words to process?

- If *Stack* = [], then done;

❺ Goto ②

Evaluating top-down & bottom-up

Are these parsers psychologically plausible?

Incrementality:

- Bottom-up: no
- Top-down: yes

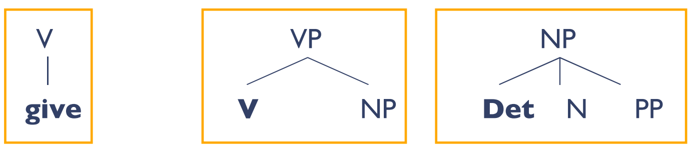
Input-driven:

- Bottom-up: yes
- Top-down: no
  - Problems with left-recursion

A Psychologically Plausible Parser

Left-Corner Parsing

Rules are 'activated' by their 'left-corner'

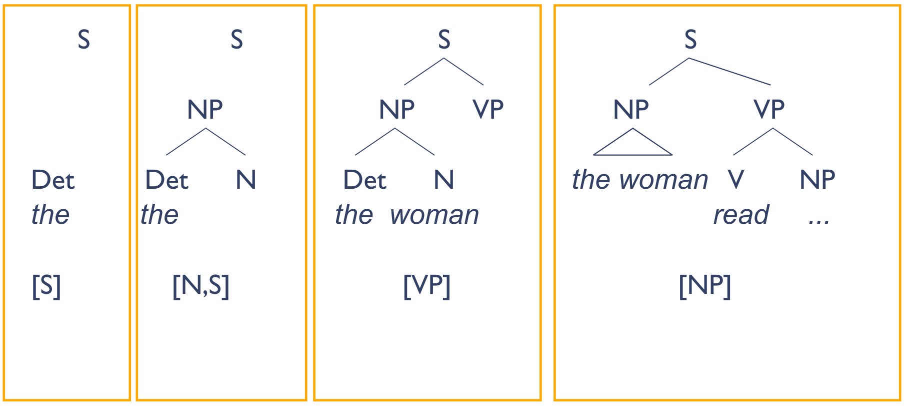


Combines input-driven with top-down

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 (LE) is easy:

- [[[John's brother]'s car door]'s handle] broke off.

So is right-embedding (RE):

- John believes [Bill knows [Mary said [she likes cats]]]

But centre-embedding (CE) is hard:

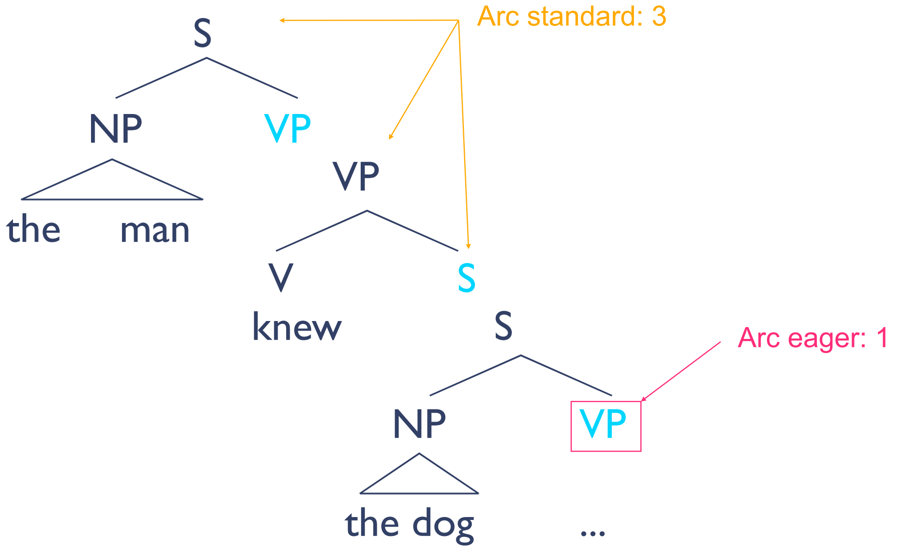
- [The mouse [the cat [the dog bit] chased] died]

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

Oracle increases the top-down component, reduce ambiguity

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

Ambiguity in Parsing

Rule selection: *what if more than one rule can be selected?*

- Local ambiguity: a parse derivation may fail later
- Global ambiguity: multiple parses can succeed

How can we handle local and global ambiguities during parsing:

- Backtracking
- Parallelism
- Determinism
- Underspecification

Backtracking Parsers

Parsing is a sequence of rule selections

If at one point, more than one rule can be applied, this is called a choice point

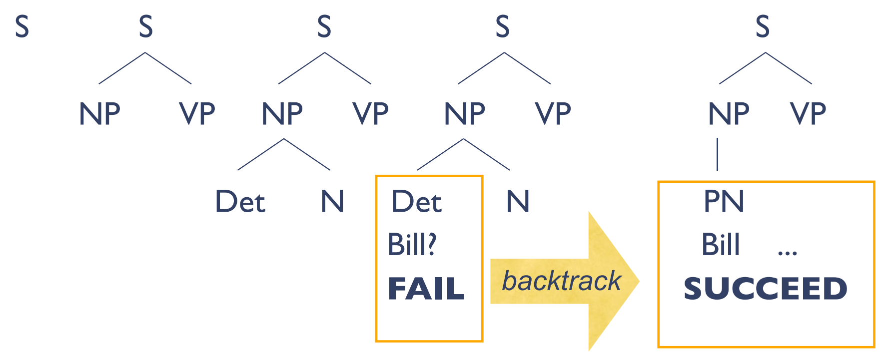
- Make a decision, based on some selection rule
- If subsequently parsing 'blocks', return to a choice point and re-parse from there

Which choice point to return to?

- usually the last, why?
- what other choice point selection rules could be used

Backtracking: an example

"Bill reads"



Parallel Parsers

Build parse trees through successive rule selections

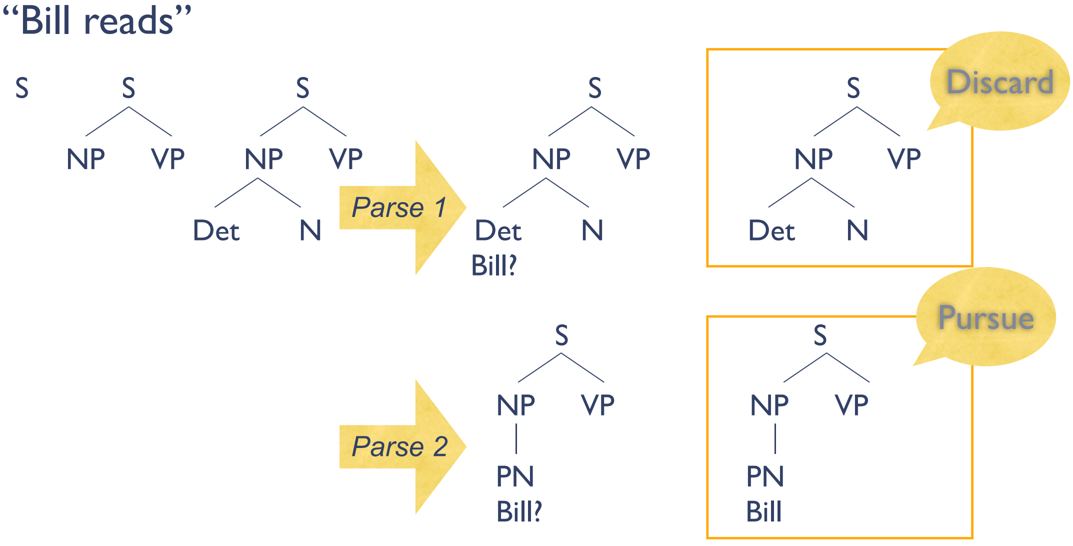
- If more than one rule may be applied, create a new parse derivation for each possibility
- Pursue all parses in parallel
- If any of the parses ‘blocks’, discard it

Because of multiple local ambiguities, the number of parallel derivation grows exponentially

- Bounded parallelism: pursue a fixed number
- How do we choose which ones to keep?

Parallel: an example

“Bill reads”



Parsing and ambiguity resolution

LC-eager parser is motivated by memory and incrementality

What predictions does it make for ambiguity resolution?

- Consider the following high-low attachment ambiguity:
  - "Two sisters reunited after eighteen years in a checkout counter
  - "John said that he will go to Edinburgh last week

Perceived as odd:

- people prefer to attach the modifier low but it must be attached high (to the main verb)

Theories of Sentence Processing

Relate the theory/model to some observed measure

- Typically impossible to predict measures completely

Theories of parsing typically determine …

- what **mechanism** is used to construct interpretations?
- which **information** sources are used by the mechanism?
- which **representation** is preferred/constructed when ambiguity arises?

Linking Hypothesis:

- Preferred sentence structures should have faster reading times in the disambiguating region than dispreferred

Garden-Path Theory: Frazier (1978)

What **architecture** is assumed?

- Modular syntactic processor, with restricted lexical (category) and semantic knowledge

What **mechanisms** is used to construct interpretations?

- Incremental, serial parsing, with reanalysis

What **information** is used to determine preferred structure?

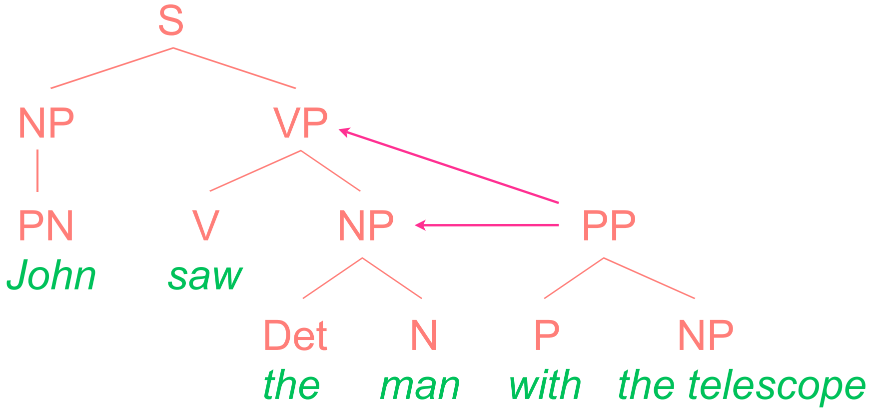
- General syntactic principles based on the current phrase stucture

Linking Hypothesis:

- Parse complexity and reanalysis cause increased RTs

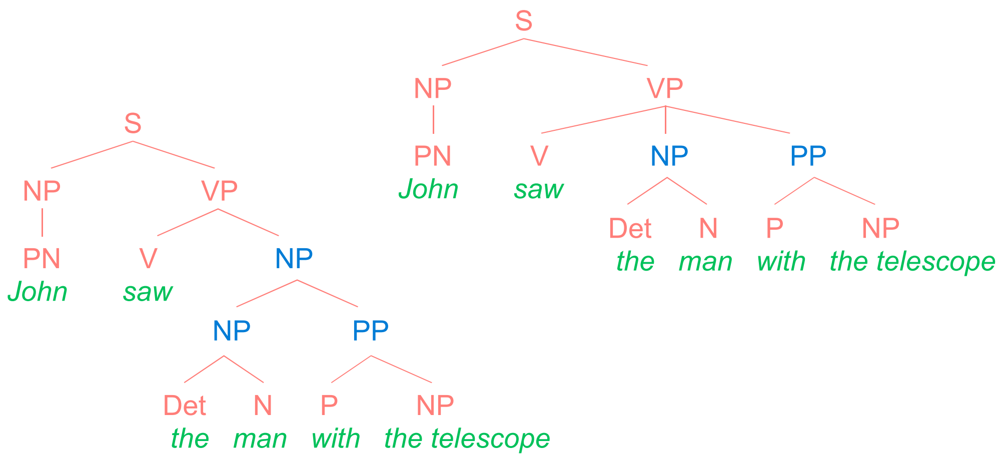
The Garden Path Theory (Frazier)

Prepositional Phase Attachment:



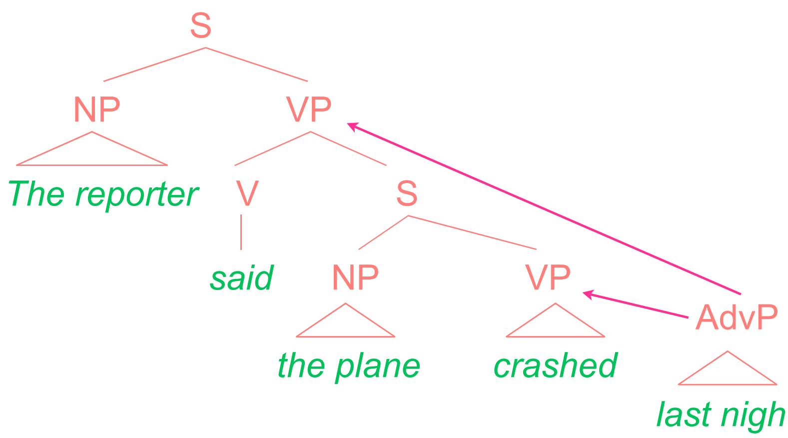
First Strategy: Minimal Attachment

Minimal Attachment: Adopt the analysis which requires postulating the fewest nodes



Second Strategy: Late Closure

Late Closure: Attach material into the most recently constructed phrase marker



Summary of Frazier

Parsing preferences are guided by general principles:

- Serial structure building
- Reanalyze based on syntactic conflict
- Reanalyze based on low plausibility (“thematic fit”)

Psychological assumptions:

- Modularity: only syntactic (not lexical, not semantic) information used for initial structure building
- Resources: emphasizes importance of memory limitations
- Processing strategies are universal, innate

Reading & Next Week

Read: Altmann, G. Ambiguity in Sentence Processing. *Trends in Cognitive Sciences*, 2:4, 1998.

- How do the accounts Altmann discusses relate to the notion of linguistic modularity?
- What kinds of information is used during processing?
- We will return later in the course to:
  - theories of ambiguity resolution later
  - connectionist and constraint-based processing models

Next two lectures: Experimental method (PK)