

# Computational Psycholinguistics

## Lecture 4: Gaps & Reanalysis

Matthew W. Crocker

Marty Mayberry



## Long Distance Dependencies

- Active Filler Strategy: (“Gap as a first resort”)

- When a filler has been identified, rank the possibility of assigning it to a gap above all other options.

- Wh-Fillers:

- Who<sub>i</sub> did Fred tell Mary e<sub>i</sub> left the country?* **dispreferred**

- Who<sub>i</sub> did Fred tell e<sub>i</sub> Mary left the country?* **preferred**

- Subject-Relative preference:

- I met the man<sub>i</sub> that John likes e<sub>i</sub>.* **dispreferred**

- I met the man<sub>i</sub> that e<sub>i</sub> likes John.* **preferred**



# Further observations ...

🍷 Filled-Gap effect:

- 🍷 *My brother wanted to know who<sub>i</sub> e<sub>i</sub> will bring us home at Christmas*
- 🍷 *My brother wanted to know who<sub>i</sub> Ruth will bring (\*e<sub>i</sub>) us home to e<sub>i</sub> at Christmas*

🍷 Intuitively easy:

- 🍷 *Who (e<sub>i</sub>) did you want (e<sub>i</sub>) Mother to bake (e<sub>i</sub>) a cake for e<sub>i</sub>?*
- 🍷 ... despite 3 possible earlier gap locations



# Gaps versus Dependencies

🍷 Consider:

- 🍷 *[In which tin]<sub>i</sub> did you put the cake e<sub>i</sub> ?*
- 🍷 *[In which tin]<sub>i</sub> did you put<sub>i</sub> the cake ?*

🍷 If keeping the filler in memory causes difficulty, we can compare:

- 🍷 *[In which tin]<sub>i</sub> did you put<sub>i</sub> the cake that your little sister baked for you e<sub>i</sub> ?* **Easy**

\_\_\_\_\_

\_\_\_\_\_

- 🍷 *[Which tin]<sub>i</sub> did you put the cake that your little sister baked for you in<sub>i</sub> e<sub>i</sub> ?* **Hard**

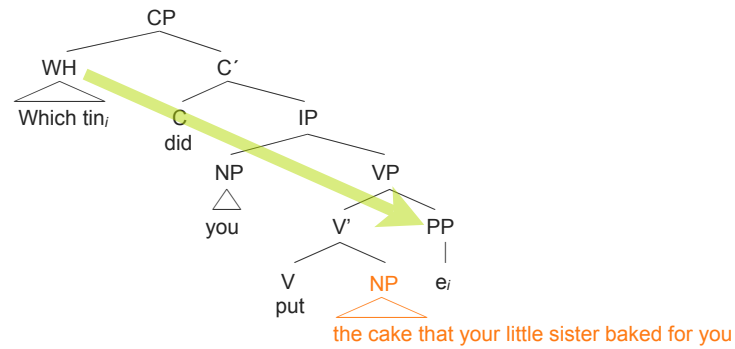
\_\_\_\_\_

\_\_\_\_\_



# Parsing in 2 dimensions

- ☘ Gaps don't exist in the input, so we needn't wait until they are found
- ☘ We can associate a filler & gap as soon as the structure licenses it:



Consider: *Den Hund<sub>i</sub> sah<sub>j</sub> Maria e<sub>j</sub> e<sub>i</sub>.*



# Reanalysis Mechanisms Revisited

- ☘ Incrementality leads to local ambiguity
  - ☘ Local ambiguity leads to reanalysis: How does the HSPM recover?
    - ☘ Serial: Backtracking, Deterministic, Parse-repair, Underspecification
    - ☘ Parallel: Re-rank or change activation
- ☘ Modeling different 'degrees' of reanalysis:
  - ☘ "John knows the truth **hurts**" *easy*
  - ☘ "While John was walking the dog **barked**" *hard*
  - ☘ "The boat sailed down the river **sank**" *v. hard*



# Deterministic Parsing

- Alternative: ensure only one possible parsing action at any point in parsing:
  - Avoid/delay rule selection until it is fully determined
    - How: bottom up (e.g. S/R or LR) plus lookahead [Marcus, Berwick & Weinberg, and Abney]
- Advantage: very fast, clear predictions
- Disadvantages:
  - not fully incremental
  - unsuccessful for head-final languages
  - wrong predictions!



# Monotonic Parsing

- Inspired by determinism and reanalysis:
  - many local ambiguities seem to cause little difficulty
  - contra predictions of naive backtracking
  - Gorrell, Weinberg, Sturt & Crocker
- Provides a richer set of 'tree-building' operations which means destructive backtracking is not always required
- Predicts 'reanalysis' *outwith* these operations to be difficult



# Talking about Talking about Trees

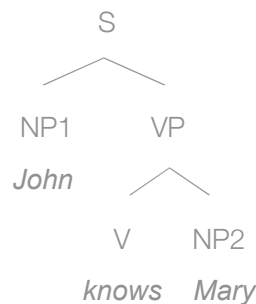
- ❁ Traditional theories locate reanalysis in the parser: “re-parsing”
- ❁ Can local ambiguity be handled using **underspecified** representations?
  - ❁ Representations which allow some ambiguity to remain, and be later removed without (destructive) re-parsing.
- ❁ Description-Theory: (Marcus, Hindle & Fleck, 1983)
  - ❁ Uses *tree descriptions*, not trees: e.g. dominance and precedence
  - ❁ Allows subsequent insertion of nodes & branches
    - ❁ Gorrell (1995), Weinberg (1993), Sturt & Crocker (1996)



# Monotonic Parsing [Gorrell; Sturt & Crocker]

- ❁ Trees are described as a set of nodes, and a set of precedence and dominance relations:

❁ *John knows Mary*



{dom(S,NP<sub>1</sub>), dom(S,VP), dom(S,V), dom(S,NP<sub>2</sub>), prec(NP<sub>1</sub>,VP),  
dom(VP,V), dom(VP,NP<sub>2</sub>), prec(V,NP<sub>2</sub>) ...}



# Properties of Trees

- Single root condition: a single root node dominates all nodes

$$\exists x \forall y \cdot \text{dom}(x, y)$$

- Exclusivity condition: no two nodes can stand in dom & prec relations

$$\forall x, y \cdot \text{prec}(x, y) \vee \text{prec}(y, x) \leftrightarrow \neg \text{dom}(x, y) \wedge \text{dom}(y, x)$$

- Inheritance: nodes inherit precedence properties of their ancestors

$$\forall w, x, y, z \cdot \text{prec}(x, y) \wedge \text{dom}(x, w) \wedge \text{dom}(y, z) \rightarrow \text{prec}(w, z)$$

- dom* and *prec* are transitive relations

- dom* is reflexive, *prec* is irreflexive



# Constraints on the Model

- Strict incrementality:** words are connected to the tree description as they are encountered

- Coherence:** tree properties must always be satisfied, the tree must be grammatically licensed

- Full specification of nodes:** no features on nodes (e.g. bar-level) can be left unspecified

- Informational monotonicity:** the tree description at state  $n$  is a subset of the description at state  $n+1$

- Obligatory assertion of precedence:** precedence must be specified for sisters



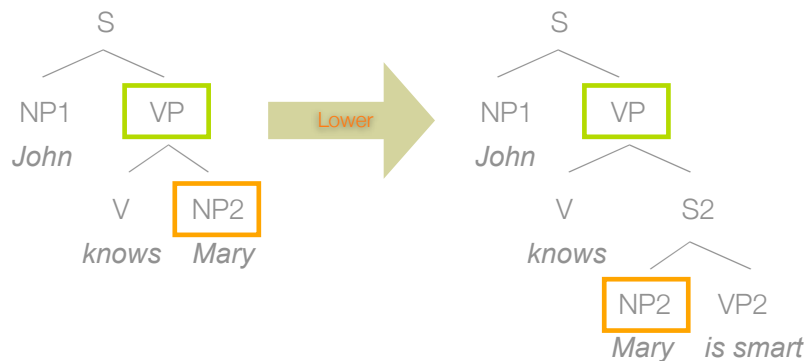
# Monotonic Parsing and Reanalysis

- ❏ **Easy:** *Monotonic* reanalysis      *“John knows Mary is smart”*
- ❏ **Hard:** *Non-monotonic* reanalysis      *“While John walked the dog barked”*
- ❏ How does the parser work:
  - ❏ The monotonic parser uses “macro” operations, which meet the constraints.
  - ❏ This simplifies the parsers search process
  - ❏ Non-monotonic operations are not permitted
- ❏ Easy reanalysis is simply “monotonic” parsing
- ❏ Hard reanalysis is either not possible, or requires re-parsing



# Monotonic Parsing Operation

- ❏ Tree-lowering: “John knows Mary ... “      “John knows Mary is smart”

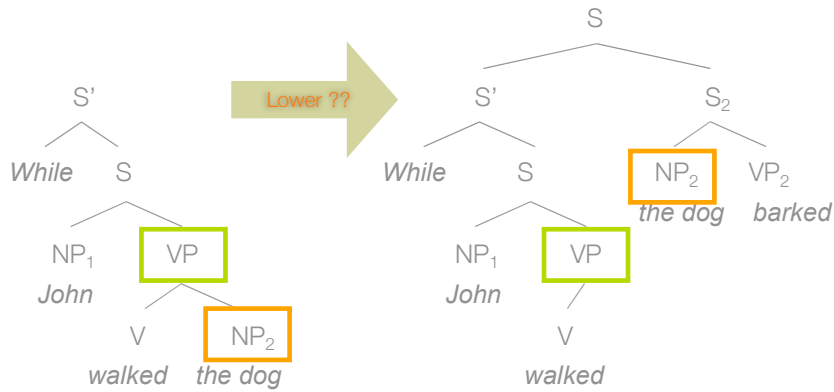


{dom(S,NP1), dom(S,VP), dom(S,V), dom(S,NP2), prec(NP1,VP), dom(VP,V), dom(VP,NP2), prec(V,NP2), **dom(VP,S2), dom(S2,NP2), prec(NP2,VP2) ...**}



# Non-Monotonic Parsing

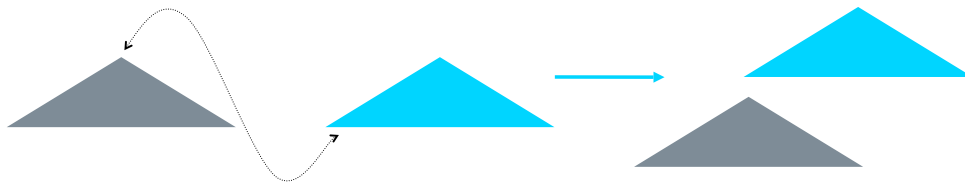
☘ Predicting difficult reanalysis: While John walked the dog ... barked.



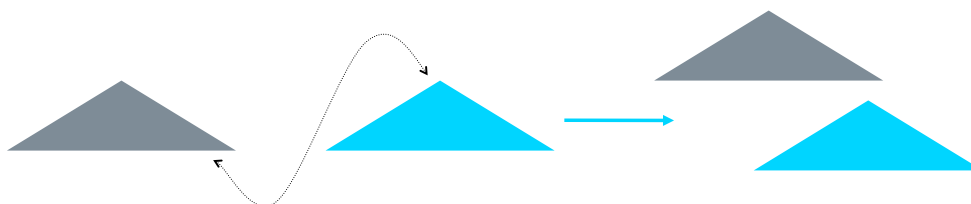
... dom(VP, NP<sub>2</sub>) ... **but** ... dom(VP, NP<sub>2</sub>) & prec(VP, NP<sub>2</sub>) ...

# Parsing Operations: Attachment

☘ Left attachment:



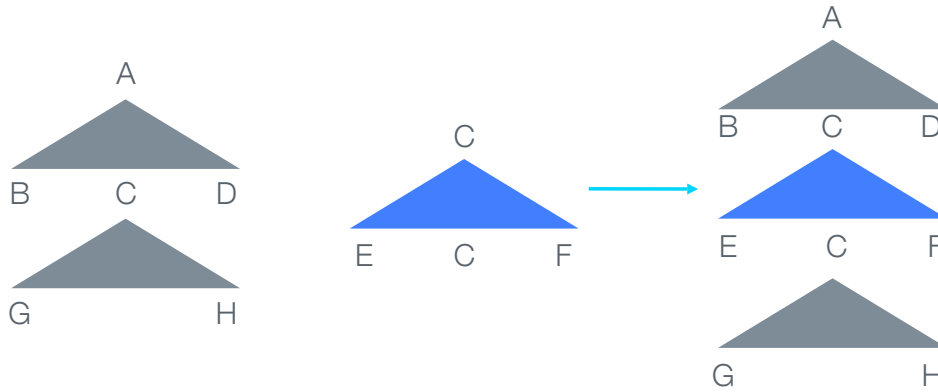
☘ Right attachment:



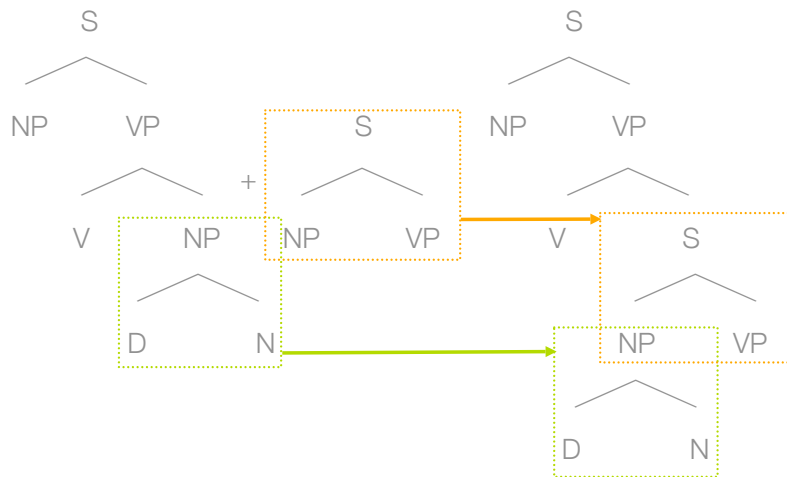


# TAG Adjunction

☘ The operations of the monotonic parser resemble those of Tree Adjoining Grammar.



# Tree Lowering



# Psycholinguistic Evidence

☘ Are there really two types of reanalysis?

☘ **NP/S (A):** *"The woman saw the famous doctor had been drinking"*

☘ **NP/Z (A):** *"Before the woman visited the famous doctor had been drinking"*

☘ **NP/S (U):** *"The woman saw that the famous doctor had been drinking"*

☘ **NP/Z (U):** *"Before the woman visited, the famous doctor had been drinking"*

☘ All verbs are biased (BNC) towards NP complement

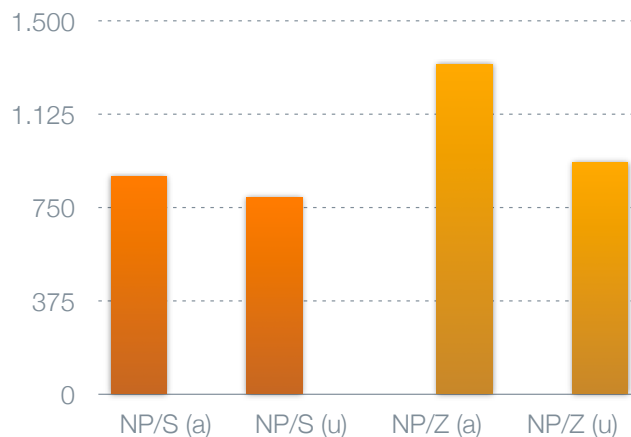
☘ To make sure the object attachment is initially adopted, forcing reanalysis

☘ Plausibility of the the direct object analysis is similar (pre-test).



# Results

☘ Reading times: 1.500



☘ Main effects of construction type, ambiguity, and a significant interaction

☘ GP effect: NP/Z (400ms) vs. NP/S (87ms)

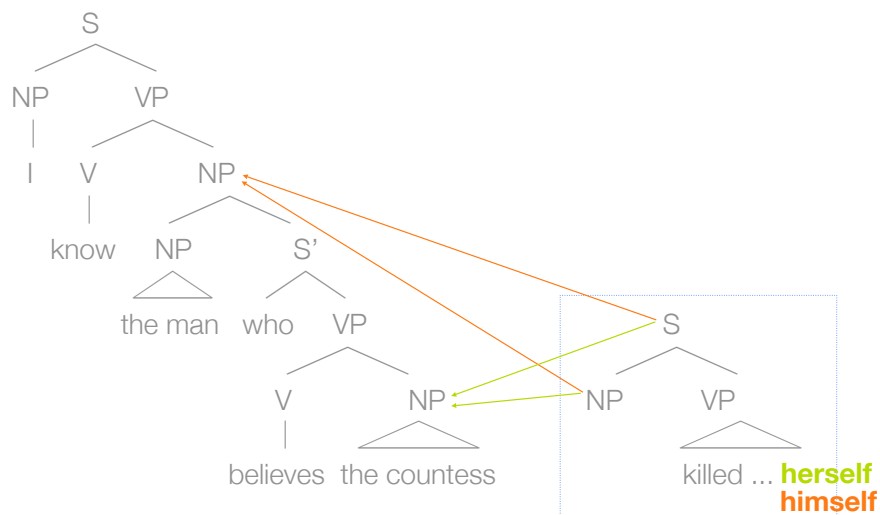


# Search in Parsing/Reanalysis

- ❁ (Some) reanalysis is simply monotonic attachment: e.g. tree-lowering.
- ❁ What if there are multiple such “lowering” attachments
- ❁ Consider a double NP/S ambiguity:
  - ❁ “I know the man who believes the countess killed herself”
  - ❁ “I know the man who believes the countess killed himself”
  - ❁ Which is easier?
- ❁ How does the parser search for an attachment?



# The Trees



# Search

- English appears to use a bottom-up search strategy for attachment
  - late closure, recency
  - *Someone shot the servant of the actress who was on the balcony*
- Possibly other influences:
  - Verb bias? Predicate proximity (Gibson)?
- Japanese seems to be top-down
  - Head final, left-branching language (Sturt & Crocker, 1996)



# Summary of Reanalysis

- Frazier: no clear account
- Pritchett: cost determined by syntactic nature of reanalysis (TRC)
- Monotonic Parsing:
  - Representations allow for some kinds of local ambiguity
    - Some reanalysis is monotonic (easy), some is destructive (difficult)
  - Similar in some respects to Pritchett's theory (of reanalysis)
  - Search mechanism still required when multiple reanalyses is possible



# Summary of Syntactic Models

## ☘ Syntactic Parsing Theories:

- ☘ Frazier: emphasis on syntactic structure/form
- ☘ Pritchett: emphasis on syntactic dependencies/content
- ☘ Assume serial, incremental parsing. Reanalysis causes difficulty
- ☘ Preference to associate fillers with role-assigners immediately
- ☘ Monotonic models enable some local ambiguities to be revised without destructive reanalysis
  - ☘ distinguish easy and difficult “garden paths”



# Computational Modeling

- ☘ Consider these theories in light of the parsing algorithms we’ve seen
  - ☘ How could you implement Frazier within one of the parsing algorithms?
  - ☘ How might Pritchett’s proposals be implemented?
  - ☘ What kinds of parser/representations do active gap filling assume?
- ☘ Monotonic parsing & reanalysis:
  - ☘ can this be done using a standard parser?
  - ☘ can you imagine an algorithm that could parse in this way?
- ☘ What does trying to build “working” models do for us?

