Memory load of left-corner MG parsers

Lei Liu

June 1, 2023

Cyclop Retreat SS23

The presentation in bullet points:

The presentation in bullet points:

- When building tree structures for sentences, how long nodes are held in memory (tenure) is a reliable metric for processing difficulties.

The presentation in bullet points:

- When building tree structures for sentences, how long nodes are held in memory (tenure) is a reliable metric for processing difficulties.
- Reported tenure-related difficulty metrics are based on top-down parsing schemata, which are argued to be less a precise capture of how human parse sentences than, say, a left-corner parsing scheme.

The presentation in bullet points:

- When building tree structures for sentences, how long nodes are held in memory (tenure) is a reliable metric for processing difficulties.
- Reported tenure-related difficulty metrics are based on top-down parsing schemata, which are argued to be less a precise capture of how human parse sentences than, say, a left-corner parsing scheme.
- This presentation asks how tenure works based on left-corner parsers for Minimalist Grammars

The presentation in bullet points:

- When building tree structures for sentences, how long nodes are held in memory (tenure) is a reliable metric for processing difficulties.
- Reported tenure-related difficulty metrics are based on top-down parsing schemata, which are argued to be less a precise capture of how human parse sentences than, say, a left-corner parsing scheme.
- This presentation asks how tenure works based on left-corner parsers for Minimalist Grammars
- (...and provides no concrete answers.)

Outline

1. Introduction
   - Tenure
   - Parser directions
2. Tenure and the usual suspects
   - Right- vs. center embedding
   - Heavy NP shift
3. Questions and next steps

Introduction: tenure

- Tenure: how long a parse item is held in memory

(1) a. The **reporter** who [1]*the senator* [2]*attacked* **disliked** the editor.

b. The **reporter** who [1]*the senator* who [2]*John* [3]*met* [4]*attacked* **disliked** the editor.

(Gibson 2000)

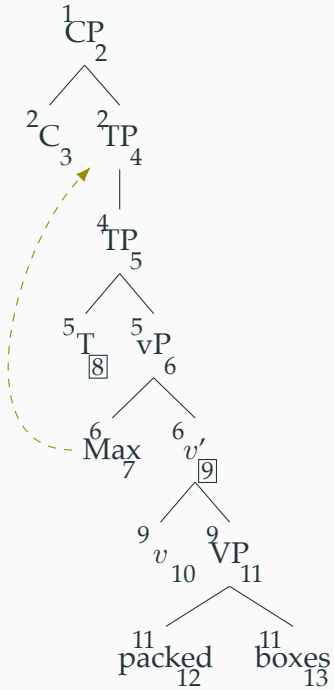
Introduction: tenure

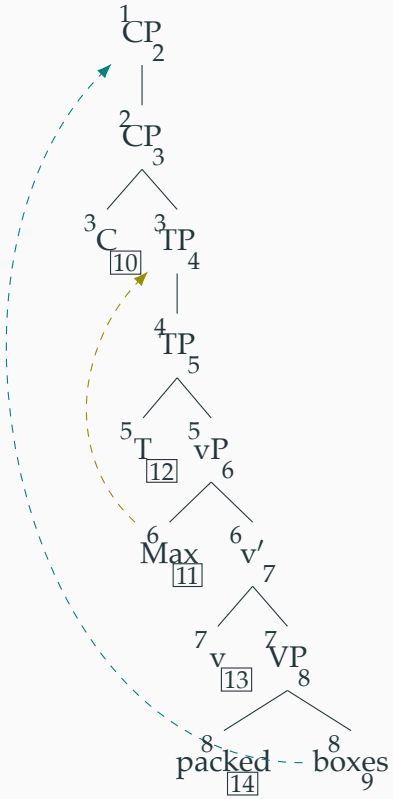
- Tenure: how long a parse item is held in memory

(2) a. Max packed boxes.

b. Boxes, Max packed.

(Liu 2022)





Introduction: tenure

- Tenure correctly predict processing preferences cross-linguistically!
    - Across constructions

Introduction: tenure

- Tenure correctly predict processing preferences cross-linguistically!
    - Across constructions
        - right- vs. center-embedding (Kobele et al. 2013)

Introduction: tenure

- Tenure correctly predict processing preferences cross-linguistically!
    - Across constructions
        - right- vs. center-embedding (Kobele et al. 2013)
        - crossing vs. nested dependencies (Kobele et al. 2013)

Introduction: tenure

- Tenure correctly predict processing preferences cross-linguistically!
  - Across constructions
    - right- vs. center-embedding (Kobele et al. 2013)
    - crossing vs. nested dependencies (Kobele et al. 2013)
    - subject vs. object relative clauses (Graf et al. 2017, Zhang 2017, De Santo 2020)

Introduction: tenure

- Tenure correctly predict processing preferences cross-linguistically!
  - Across constructions
    - right- vs. center-embedding (Kobele et al. 2013)
    - crossing vs. nested dependencies (Kobele et al. 2013)
    - subject vs. object relative clauses (Graf et al. 2017, Zhang 2017, De Santo 2020)
    - attachment ambiguity (Lee 2018)

Introduction: tenure

- Tenure correctly predict processing preferences cross-linguistically!
  - Across constructions
    - right- vs. center-embedding (Kobele et al. 2013)
    - crossing vs. nested dependencies (Kobele et al. 2013)
    - subject vs. object relative clauses (Graf et al. 2017, Zhang 2017, De Santo 2020)
    - attachment ambiguity (Lee 2018)
    - heavy NP shift (Liu 2018)

Introduction: tenure

- Tenure correctly predict processing preferences cross-linguistically!
  - Across constructions
    - right- vs. center-embedding (Kobele et al. 2013)
    - crossing vs. nested dependencies (Kobele et al. 2013)
    - subject vs. object relative clauses (Graf et al. 2017, Zhang 2017, De Santo 2020)
    - attachment ambiguity (Lee 2018)
    - heavy NP shift (Liu 2018)
  - Across languages

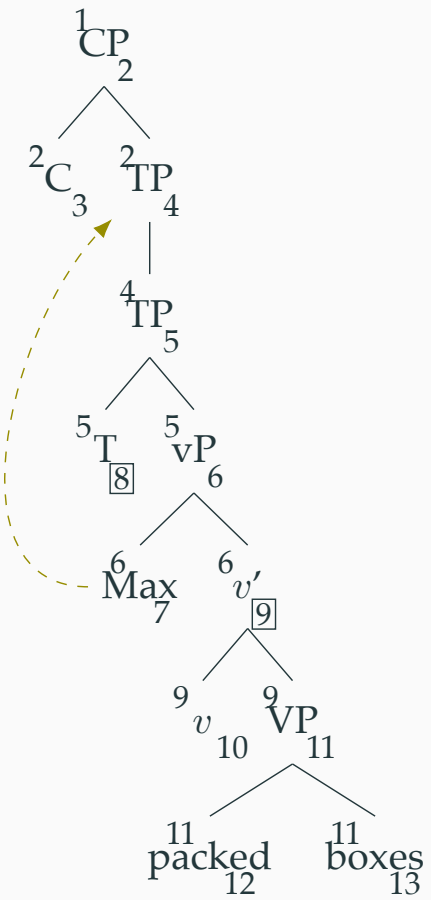
Introduction: tenure

- Tenure correctly predict processing preferences cross-linguistically!
  - Across constructions
    - right- vs. center-embedding (Kobele et al. 2013)
    - crossing vs. nested dependencies (Kobele et al. 2013)
    - subject vs. object relative clauses (Graf et al. 2017, Zhang 2017, De Santo 2020)
    - attachment ambiguity (Lee 2018)
    - heavy NP shift (Liu 2018)
  - Across languages
    - English, German, Italian, Korean, Japanese, Chinese, Persian...

Introduction: tenure

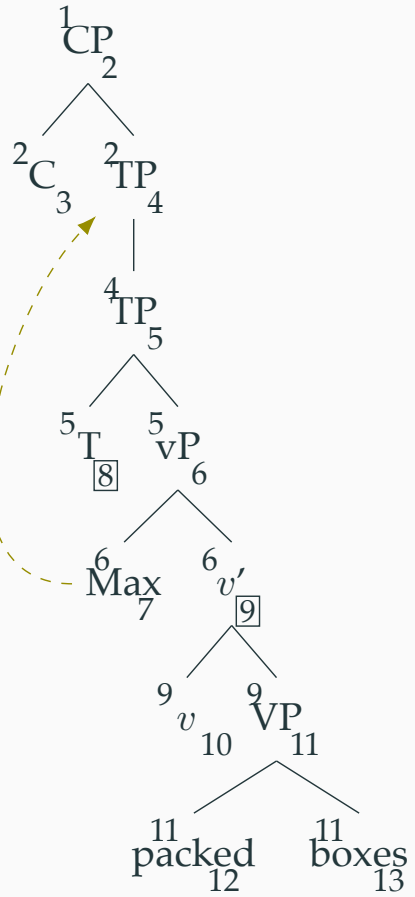
- Tenure correctly predict processing preferences cross-linguistically!
  - Across constructions
    - right- vs. center-embedding (Kobele et al. 2013)
    - crossing vs. nested dependencies (Kobele et al. 2013)
    - subject vs. object relative clauses (Graf et al. 2017, Zhang 2017, De Santo 2020)
    - attachment ambiguity (Lee 2018)
    - heavy NP shift (Liu 2018)
  - Across languages
    - English, German, Italian, Korean, Japanese, Chinese, Persian...
  - All based on a version of top-down parsers

Introduction: parser directions

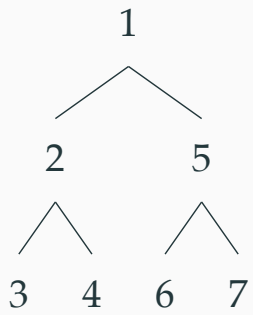


Max packed boxes

Introduction: parser directions

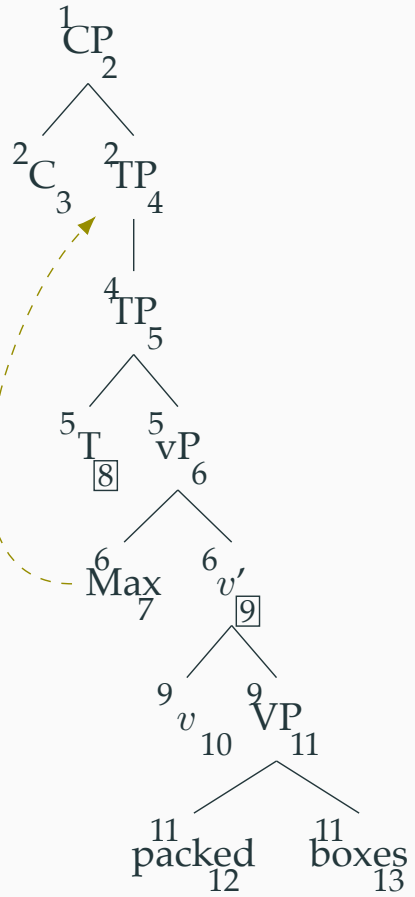


Max packed boxes

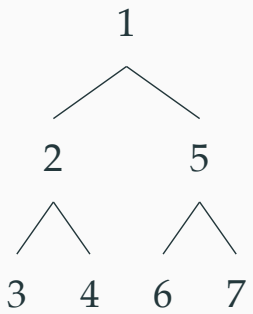


Top-down

Introduction: parser directions



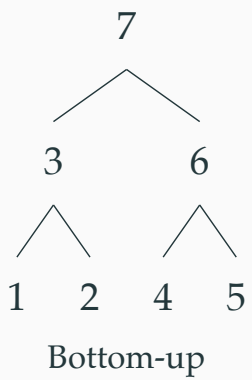
Max packed boxes



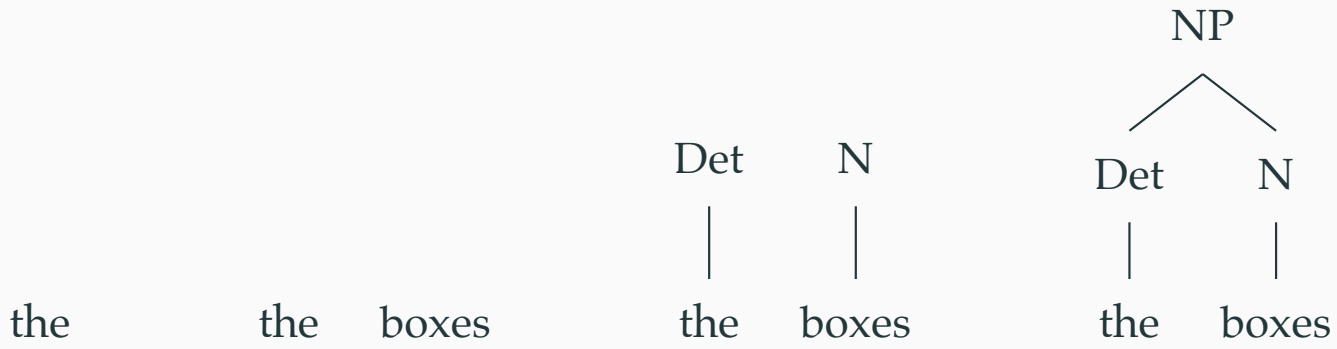
Top-down

- Pros
  - predictive
  - incremental
- Cons
  - terminal symbols do not guide prediction

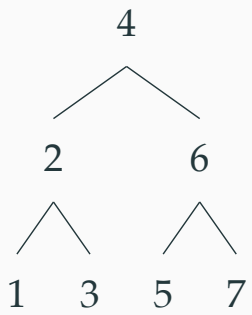
Introduction: parser directions



- Pros
  - input driven
  - (can be) incremental
- Cons
  - not predicative



Introduction: parser directions

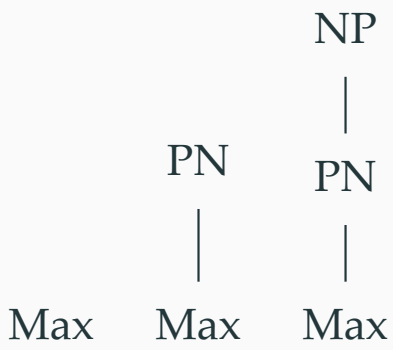


Left-corner

- Integrates
  - top-down prediction
  - bottom-up reduction

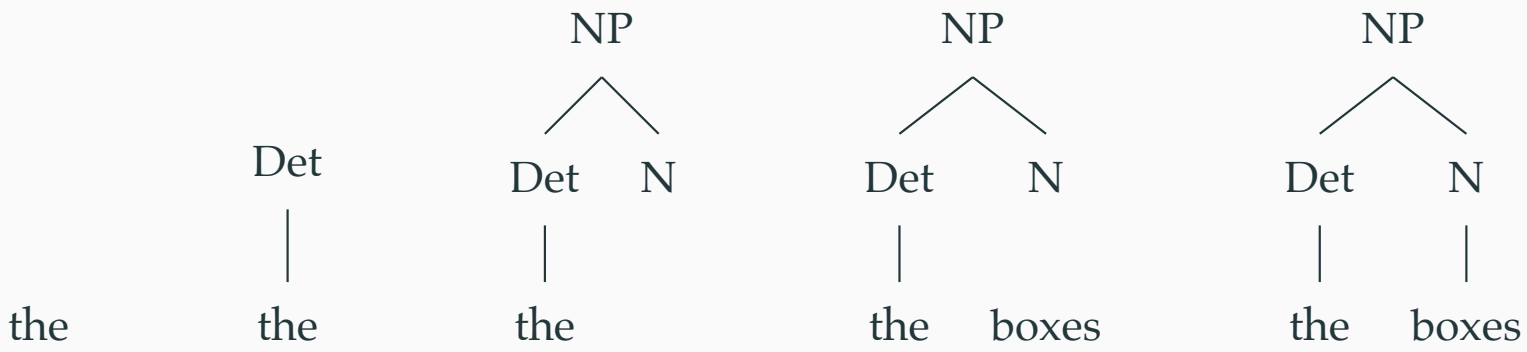
Left-corner parser: intuitions

1. `S -> NP VP`
2. `NP -> PN`
3. `NP -> Det N`
4. `VP -> Vt NP`
5. `PN -> Max`
6. `Vt -> packed`
7. `Det -> the`
8. `N -> boxes`



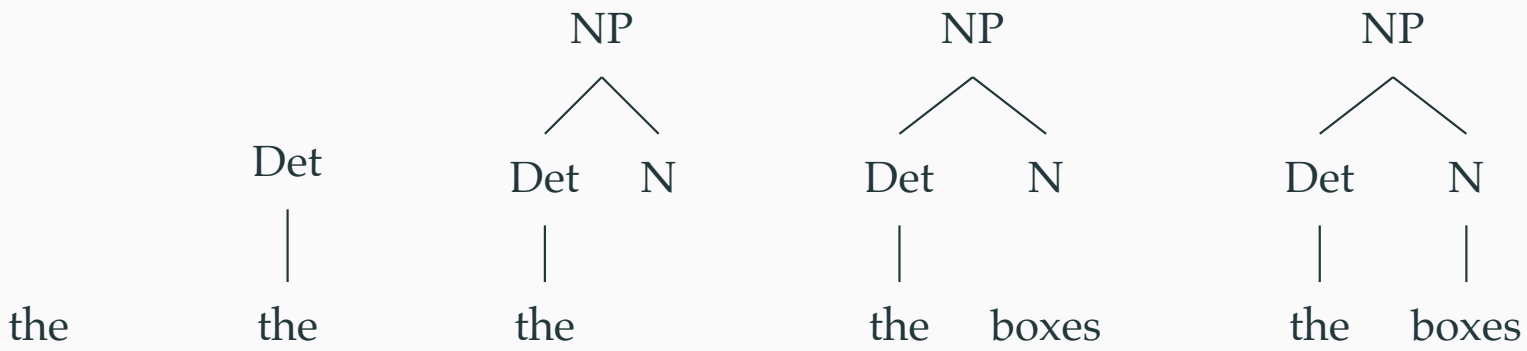
Left-corner parser: intuitions

1. `S -> NP VP`
2. `NP -> PN`
3. `NP -> Det N`
4. `VP -> Vt NP`
5. `PN -> Max`
6. `Vt -> packed`
7. `Det -> the`
8. `N -> boxes`



Left-corner parser: intuitions

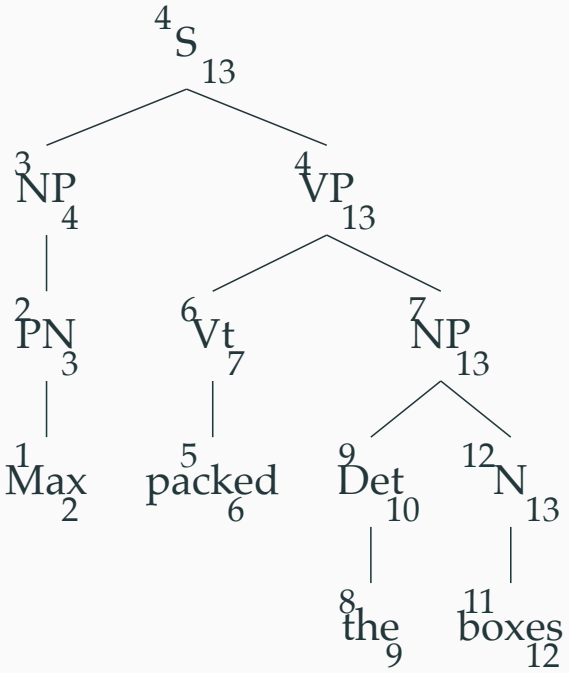
1. `S -> NP VP`
2. `NP -> PN`
3. `NP -> Det N`
4. `VP -> Vt NP`
5. `PN -> Max`
6. `Vt -> packed`
7. `Det -> the`
8. `N -> boxes`



- Left-corner: the leftmost symbol on the righthand side of the rewrite arrow.

Left-corner parser: intuitions

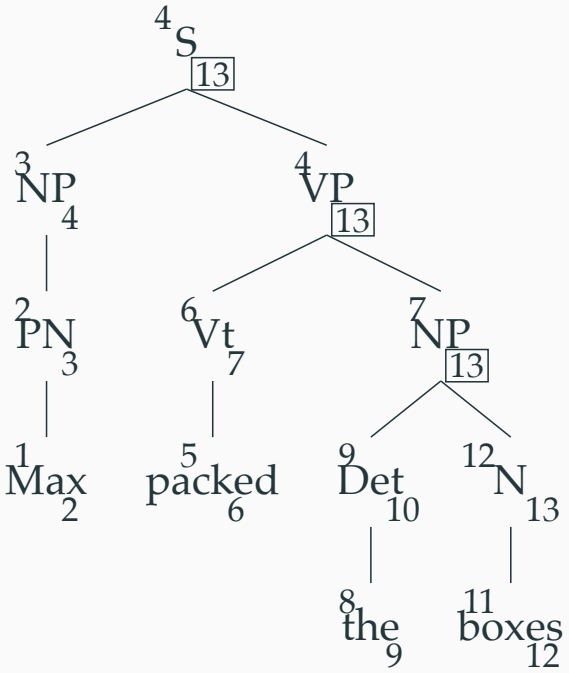
1. `S -> NP VP`
2. `NP -> PN`
3. `NP -> Det N`
4. `VP -> Vt NP`
5. `PN -> Max`
6. `Vt -> packed`
7. `Det -> the`
8. `N -> boxes`



Max packed the boxes.

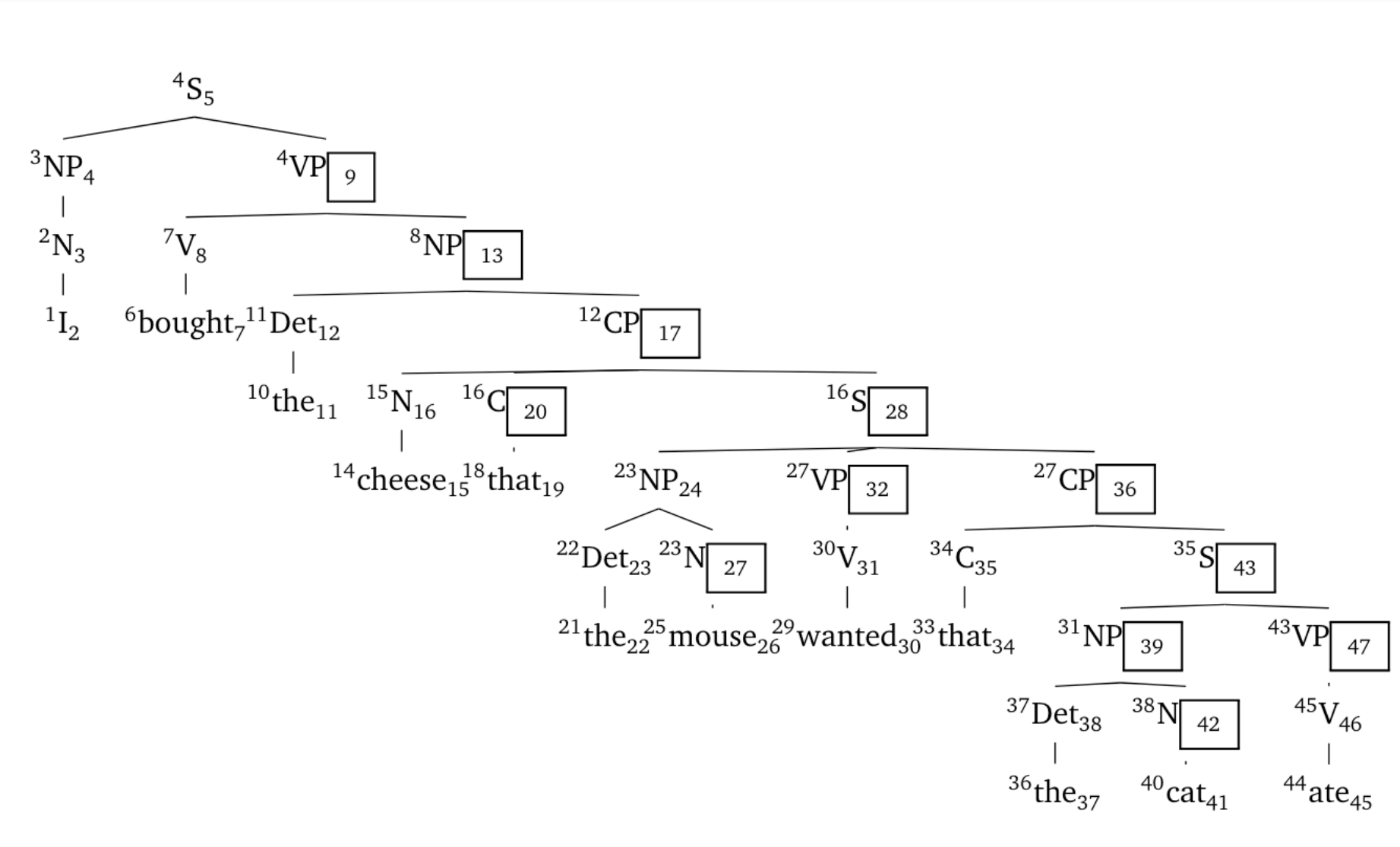
Left-corner parser: intuitions

1. `S -> NP VP`
2. `NP -> PN`
3. `NP -> Det N`
4. `VP -> Vt NP`
5. `PN -> Max`
6. `Vt -> packed`
7. `Det -> the`
8. `N -> boxes`



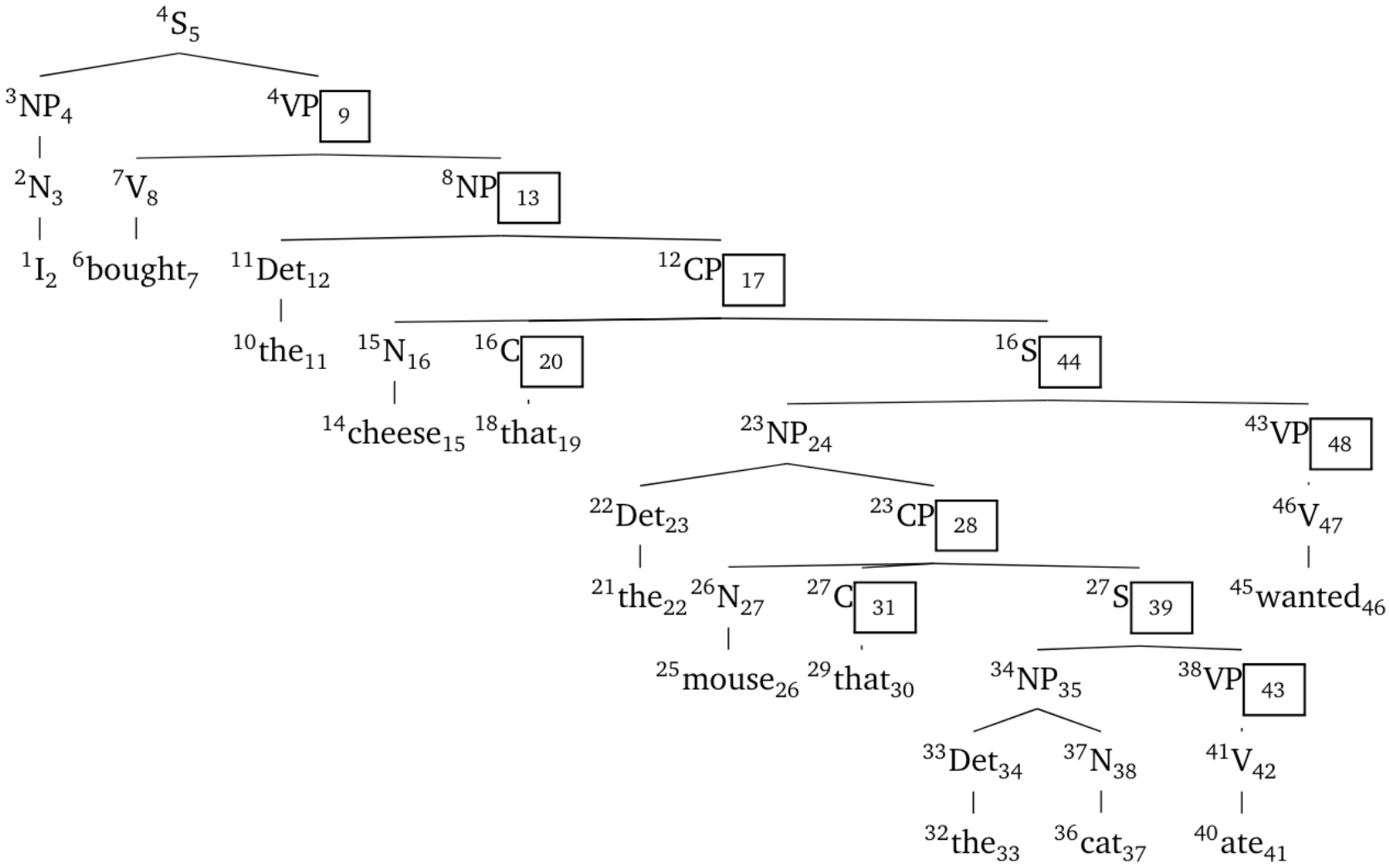
Max packed the boxes.

Right- vs. center-embedding



I bought the cheese that the mouse wanted that the cat ate.

Right- vs. center-embedding



I bought the cheese that the mouse that the cat ate wanted.

Right- vs. center-embedding

(3) I bought the cheese that the mouse wanted that the cat ate.
right-embedding
`MaxT = 12`

(4) I bought the cheese that the mouse that the cat ate wanted.
center-embedding
`MaxT = 28`

Right- vs. center-embedding

(3) I bought the cheese that the mouse wanted that the cat ate.
right-embedding
`MaxT = 12`

(4) I bought the cheese that the mouse that the cat ate wanted.
center-embedding
`MaxT = 28`

success

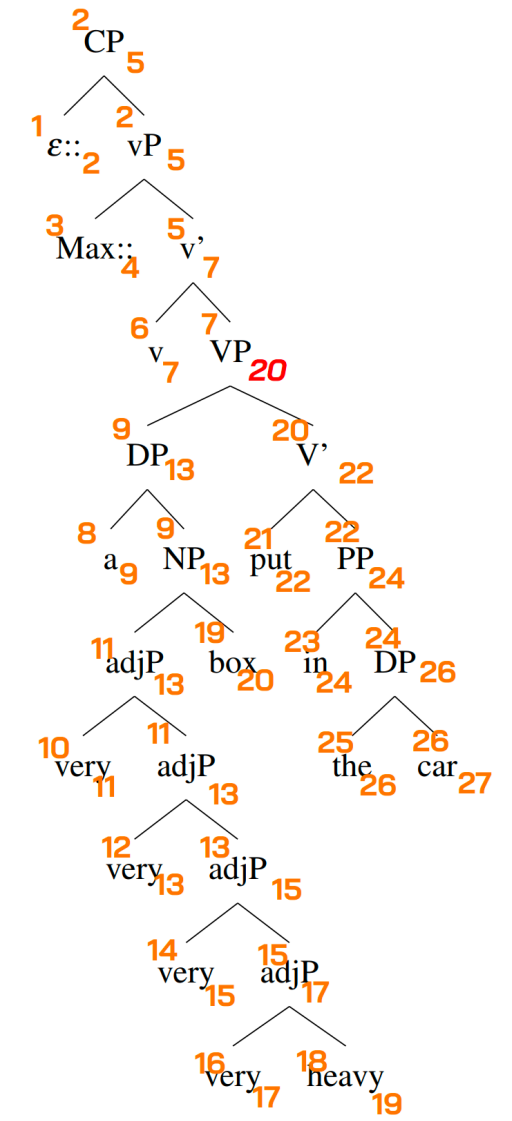
Right- vs. center-embedding

(3) I bought the cheese that the mouse wanted that the cat ate.
right-embedding
`MaxT = 12`

(4) I bought the cheese that the mouse that the cat ate wanted.
center-embedding
`MaxT = 28`

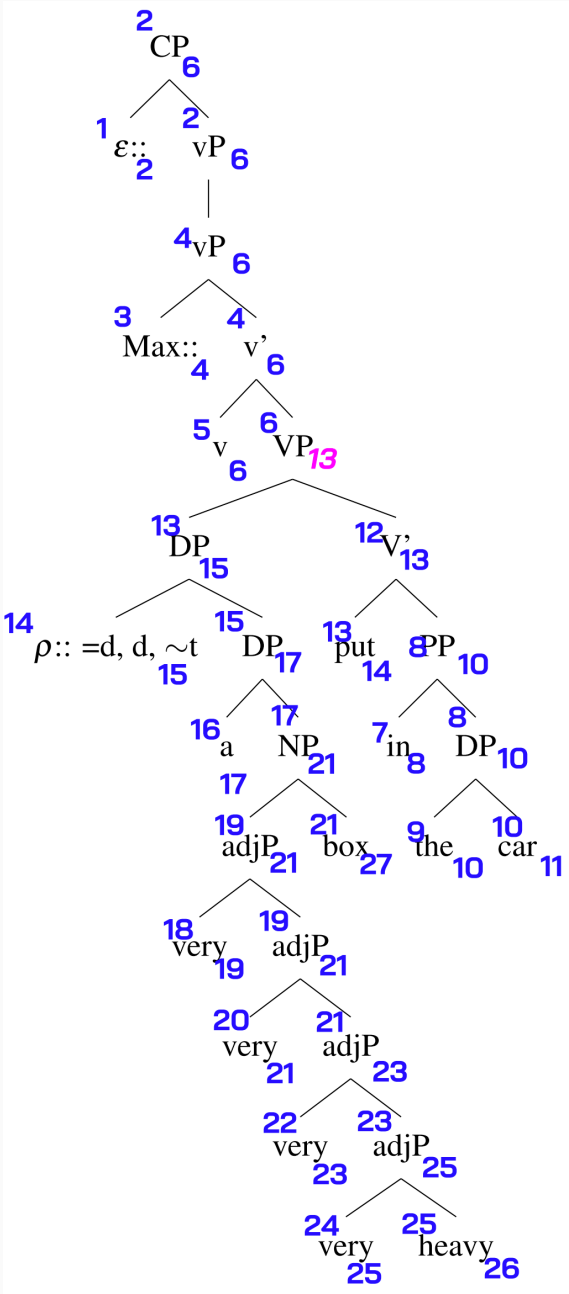
success-ish

Heavy NP shift



- ϵ Max v a very very very very heavy box put in the car.

Heavy NP shift



- ϵ Max v in the car put ρ a very very very very heavy box.

Heavy NP shift

(5) ϵ Max v a very very very very heavy box put in the car.
canonical order
`MaxT = 13`

(6) ϵ Max v in the car put ρ a very very very very heavy box.
HNPS order
`MaxT = 7`

Heavy NP shift

(5) ϵ Max v a very very very very heavy box put in the car.
canonical order
`MaxT = 13`

(6) ϵ Max v in the car put ρ a very very very very heavy box.
HNPS order
`MaxT = 7`

success

Heavy NP shift

(5) ϵ Max v a very very very very heavy box put in the car.
canonical order
`MaxT = 13`

(6) ϵ Max v in the car put ρ a very very very very heavy box.
HNPS order
`MaxT = 7`

success-ish

Questions and next steps

- Nodes vs. trees
  - memorizing a tree harder than memorizing a node?
  - metrics other than `MaxT`
    - `SumT`
    - `BoxT`
    - what about movement?
- Other benchmark constructions
  - relative clauses
  - topicalization (See 2)
- Tree annotation tools

Thank you!

References i

De Santo, A. (2020). *Structure and memory: A computational model of storage, gradience, and priming*. PhD thesis, State University of New York at Stony Brook.

Gibson, E. (2000). The dependency locality theory: A distance-based theory of linguistic complexity. *Image, language, brain*, 2000:95–126.

Graf, T., Monette, J., and Zhang, C. (2017). Relative clauses as a benchmark for minimalist parsing. *Journal of Language Modelling*, 5(1):57–106.

Kobele, G. M., Gerth, S., and Hale, J. (2013). Memory resource allocation in top-down minimalist parsing. In *Formal Grammar*, pages 32–51. Springer.

Lee, S. Y. (2018). A minimalist parsing account of attachment ambiguity in english and korean. *Journal of Cognitive Science*, 19(3):291–329.

Liu, L. (2018). Minimalist parsing of heavy np shift. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*.

Liu, L. (2022). *Phrasal Weight Effect on Word Order*. PhD thesis, State University of New York at Stony Brook.

Zhang, C. (2017). *Stacked Relatives: Their Structure, Processing and Computation*. PhD thesis, State University of New York at Stony Brook.

Merely Local Syntactic Coherence Effects

(7) The coach smiled at the player **tossed** a frisbee.

(8) The coach smiled at the player **thrown** a frisbee.

Bottom-up and coordination