Processing Advantages of Eng-weight

Lei Liu

Leipzig University
lei.liu@uni-leipzig.de
leiliu.net

SCiL2023
June 15, 2023
updated on June 25, 2023

The presentation in bullet points:

The presentation in bullet points:

- End-weight configuration facilitates processing.

The presentation in bullet points:

- End-weight configuration facilitates processing.
- The facilitation effect is accounted for by the memory resources needed to parse the end-weight syntactic tree.

The presentation in bullet points:

- End-weight configuration facilitates processing.
- The facilitation effect is accounted for by the memory resources needed to parse the end-weight syntactic tree.
- This memory-based account has implications on the opposite, initial-weight preferences.

Outline

1. Introduction
   - End Weight
   - MG Parsing
2. Parsing End-weight Configurations
   - End-weight in Heavy NP Shift
   - End-weight in Particle Verb
3. Discussion

Intro: Heavy NP Shift

- English heavy NP shift (HNPS)

Intro: Heavy NP Shift

- English heavy NP shift (HNPS)

(1) a. Emma explained [$_{DO}$ the regulations] to [$_{IO}$ Jim].

Intro: Heavy NP Shift

- English heavy NP shift (HNPS)

(1) a. Emma explained [$_{DO}$ the regulations] to [$_{IO}$ Jim].

b. Emma explained to [$_{IO}$ Jim] [$_{DO}$ all the regulations regarding import and export taxes for pottery].

Intro: Heavy NP Shift

- English heavy NP shift (HNPS)

(1) a. Emma explained [$_{DO}$ the regulations] to [$_{IO}$ Jim].

b. Emma explained to [$_{IO}$ Jim] [$_{DO}$ all the regulations regarding import and export taxes for pottery].

c. ? Emma explained to [$_{IO}$ Jim] [$_{DO}$ the regulations].

(Stallings and MacDonald 2011)

Intro: Heavy NP Shift

- English heavy NP shift (HNPS)

(1) a. Emma explained [$_{DO}$ the regulations] to [$_{IO}$ Jim].

b. Emma explained to [$_{IO}$ Jim] [$_{DO}$ all the regulations regarding import and export taxes for pottery].

c. ? Emma explained to [$_{IO}$ Jim] [$_{DO}$ the regulations].

(Stallings and MacDonald 2011)

→ HNPS order (Verb-IO-DO) preferred when DO is heavy

Intro: Particle Verb

- English Particle Verb

Intro: Particle Verb

- English Particle Verb

(2) a. I **looked up** the word (in the dictionary).

b. I **looked** the word **up** (in the dictionary).

Intro: Particle Verb

- English Particle Verb

(2) a. I **looked up** the word (in the dictionary).

b. I **looked** the word **up** (in the dictionary).

(3) a. … I **looked up** [a person who answered a query I posted on the internet]…

Intro: Particle Verb

- English Particle Verb

(2) a. I **looked up** the word (in the dictionary).

b. I **looked** the word **up** (in the dictionary).

(3) a. … I **looked up** [a person who answered a query I posted on the internet]…

b. *I **looked** [a person who answered a query I posted on the internet] **up**…

(Cappelle 2005, 19)

Intro: Particle Verb

- English Particle Verb

(2) a. I **looked up** the word (in the dictionary).

b. I **looked** the word **up** (in the dictionary).

(3) a. … I **looked up** [a person who answered a query I posted on the internet]…

b. *I **looked** [a person who answered a query I posted on the internet] **up**…

(Cappelle 2005, 19)

→ Joined order (**Verb**-**particle**-[object]) preferred when object is heavy

Intro: End Weight

(4) a. Emma explained to [$_{IO}$ Jim] [$_{DO}$ all the … for pottery].
light heavy

b. I looked up [a person … on the internet].
light heavy

Intro: End Weight

(4) a. Emma explained to [$_{IO}$ Jim] [$_{DO}$ all the … for pottery].
light heavy

b. I looked up [a person … on the internet].
light heavy

- Heavy how?

Intro: End Weight

(4) a. Emma explained to [$_{IO}$ Jim] [$_{DO}$ all the … for pottery].
light heavy

b. I looked up [a person … on the internet].
light heavy

- Heavy how?
  - Syntactic structure matters

Intro: End Weight

(4) a. Emma explained to [$_{IO}$ Jim] [$_{DO}$ all the … for pottery].
light heavy

b. I looked up [a person … on the internet].
light heavy

- Heavy how?
  - Syntactic structure matters
  - Heavy relative to other constituents

Intro: End Weight

(4) a. Emma explained to [$_{IO}$ Jim] [$_{DO}$ all the … for pottery].
light heavy

b. I looked up [a person … on the internet].
light heavy

- Heavy how?
  - Syntactic structure matters
  - Heavy relative to other constituents
- Proposal:

Intro: End Weight

(4) a. Emma explained to [$_{IO}$ Jim] [$_{DO}$ all the ... for pottery].
light heavy

b. I looked up [a person ... on the internet].
light heavy

- Heavy how?
  - Syntactic structure matters
  - Heavy relative to other constituents
- Proposal:
  - End-weight preference follows from the processing difficulties associated with the syntactic structure of competing word orders.

Intro: End Weight

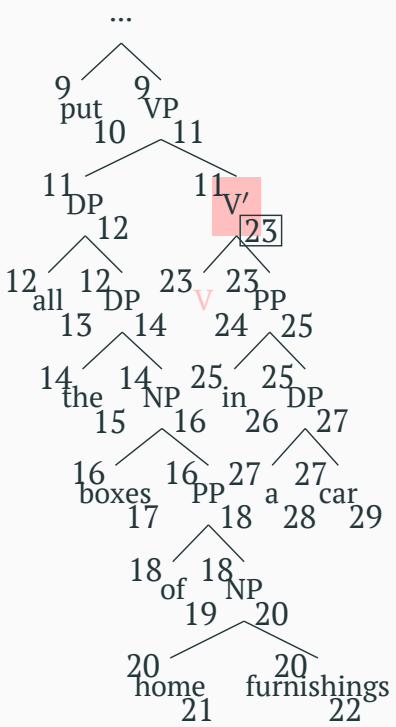
(4) a. Emma explained to [$_{IO}$ Jim] [$_{DO}$ all the ... for pottery].
light heavy

b. I looked up [a person ... on the internet].
light heavy

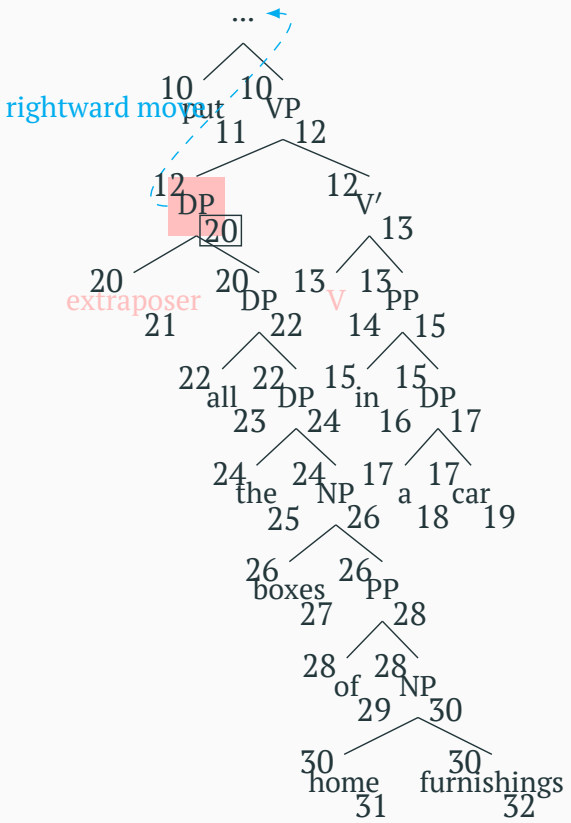
- Heavy how?
  - Syntactic structure matters
  - Heavy relative to other constituents
- Proposal:
  - End-weight preference follows from the processing difficulties associated with the syntactic structure of competing word orders.

Intro: End Weight

(5) put [DP …boxes…] [PP in…]
canonical order

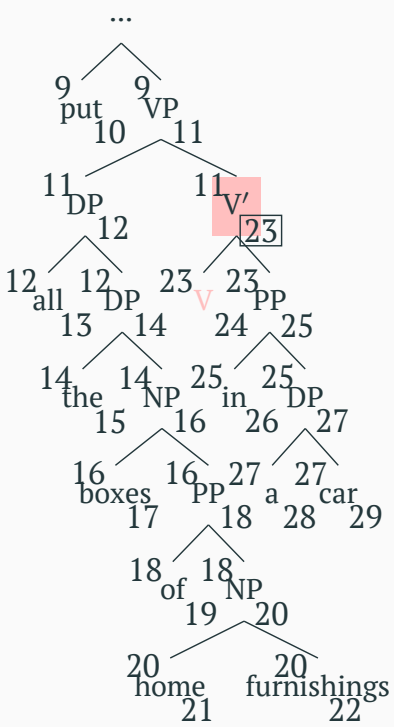


(6) put [PP in…] [DP …boxes…]
HNPS order

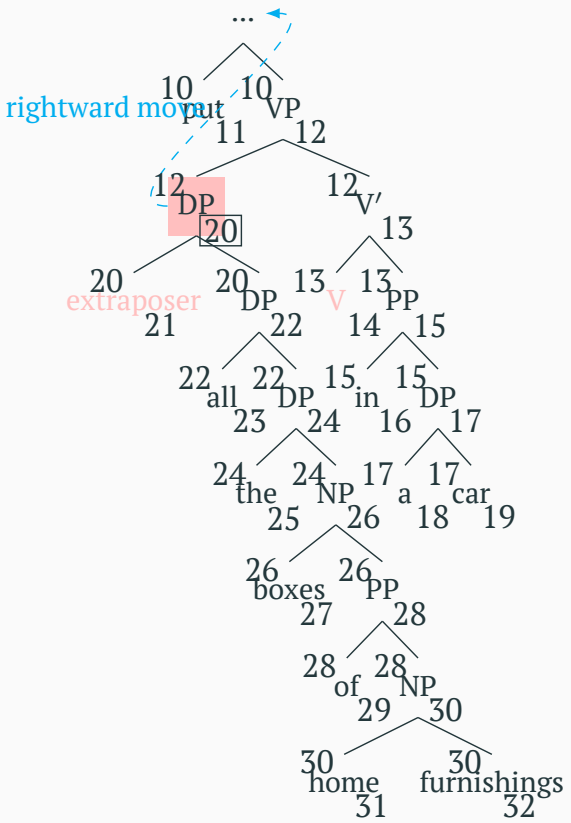


Intro: End Weight

(5) put [DP …boxes…] [PP in…]
canonical order



(6) put [PP in…] [DP …boxes…]
HNPS order



whoa!

Minimalist Parsing

- Minimalist Grammars (Stabler 1996; 2011, Graf 2013): lexicalized grammar formalisms based on the Minimalist Program (Chomsky 2014).

Minimalist Parsing

- Minimalist Grammars (Stabler 1996; 2011, Graf 2013): lexicalized grammar formalisms based on the Minimalist Program (Chomsky 2014).
    - lexical items

Minimalist Parsing

- Minimalist Grammars (Stabler 1996; 2011, Graf 2013): lexicalized grammar formalisms based on the Minimalist Program (Chomsky 2014).
    - lexical items
        - feature bundles

Minimalist Parsing

- Minimalist Grammars (Stabler 1996; 2011, Graf 2013): lexicalized grammar formalisms based on the Minimalist Program (Chomsky 2014).
    - lexical items
        - feature bundles
        - {phonetics, category, selection, movement}

Minimalist Parsing

- Minimalist Grammars (Stabler 1996; 2011, Graf 2013): lexicalized grammar formalisms based on the Minimalist Program (Chomsky 2014).
    - lexical items
        - feature bundles
        - {phonetics, category, selection, movement}
    - operations

Minimalist Parsing

- Minimalist Grammars (Stabler 1996; 2011, Graf 2013): lexicalized grammar formalisms based on the Minimalist Program (Chomsky 2014).
  - lexical items
    - feature bundles
    - {phonetics, category, selection, movement}
  - operations
    - `merge, move`

Minimalist parsing

Max :: D^- nom^-
cat. mvmt

packed :: D^+ V^-
sel. cat.

boxes :: D^-
cat.

C :: T^+ C^-
sel. cat.

T :: v^+ nom^+ T^-
sel. mvmt cat.

v :: V^+ D^+ v^-
sel. sel. cat.

Minimalist parsing

Max :: D^- nom^-
cat. mvmt

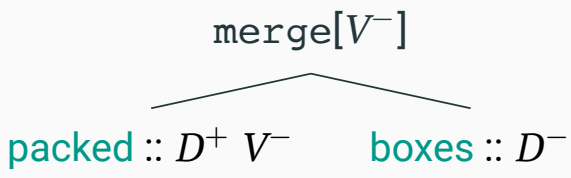
packed :: D^+ V^-
sel. cat.

boxes :: D^-
cat.

C :: T^+ C^-
sel. cat.

T :: v^+ nom^+ T^-
sel. mvmt cat.

v :: V^+ D^+ v^-
sel. sel. cat.



Minimalist parsing

Max :: D^- nom^-
cat. mvmt

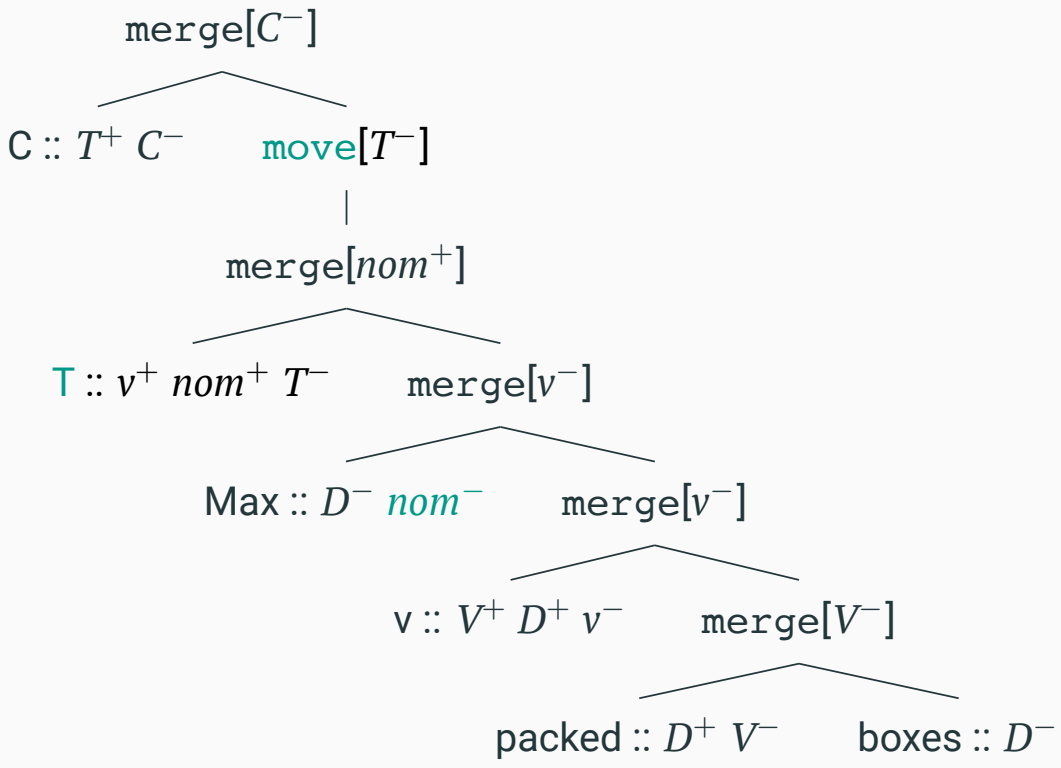
packed :: D^+ V^-
sel. cat.

boxes :: D^-
cat.

C :: T^+ C^-
sel. cat.

T :: v^+ nom^+ T^-
sel. mvmt cat.

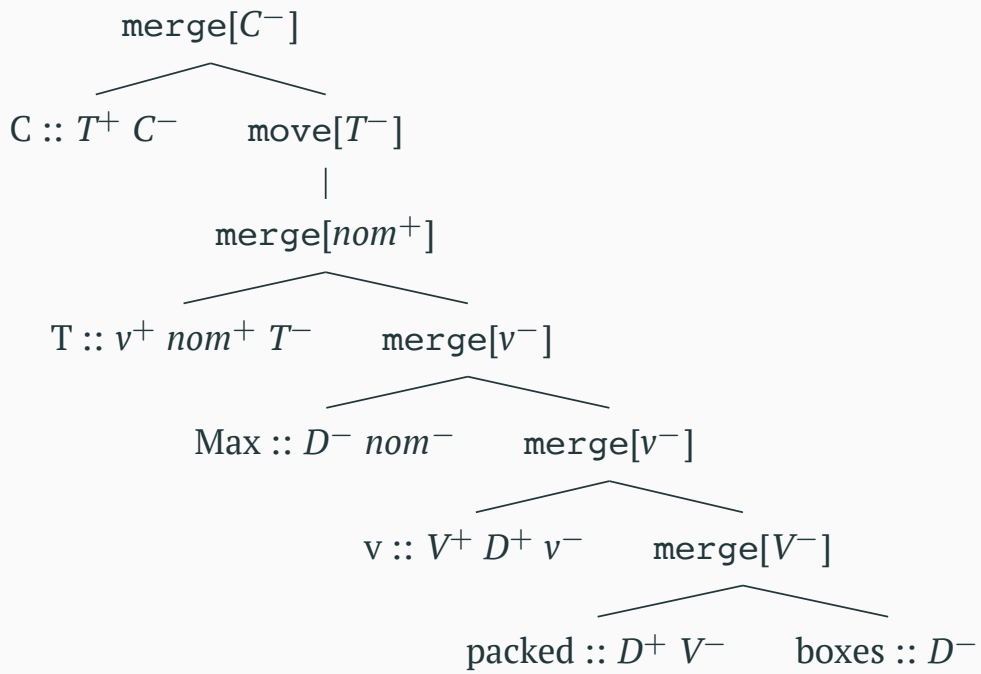
v :: V^+ D^+ v^-
sel. sel. cat.



Minimalist Parsing

- Minimalist Grammars (Stabler 1996; 2011, Graf 2013): lexicalized grammar formalisms based on the Minimalist Program (Chomsky 2014).

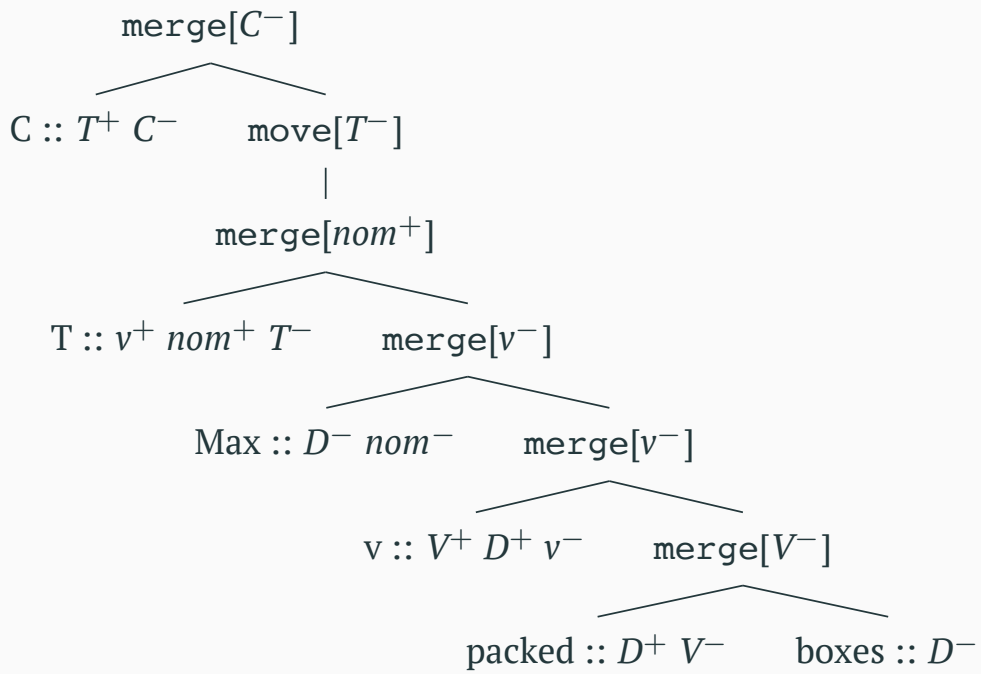
(7) Max packed boxes.



Minimalist Parsing

- Minimalist Grammars (Stabler 1996; 2011, Graf 2013): lexicalized grammar formalisms based on the Minimalist Program (Chomsky 2014).

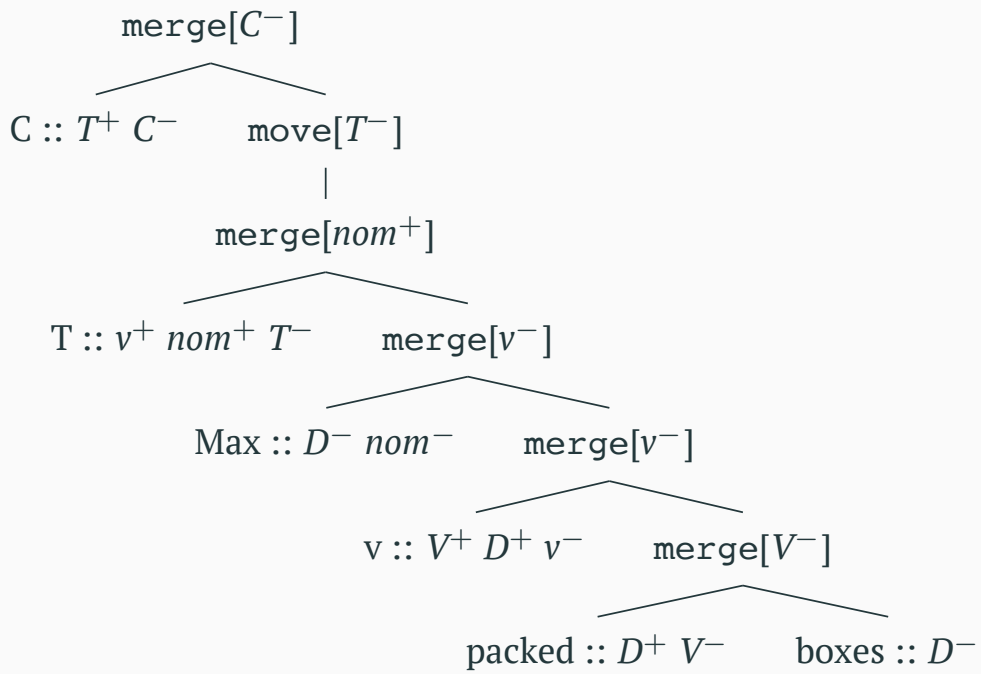
(7) Max packed boxes.



Minimalist Parsing

- Minimalist Grammars (Stabler 1996; 2011, Graf 2013): lexicalized grammar formalisms based on the Minimalist Program (Chomsky 2014).

(7) Max packed boxes.

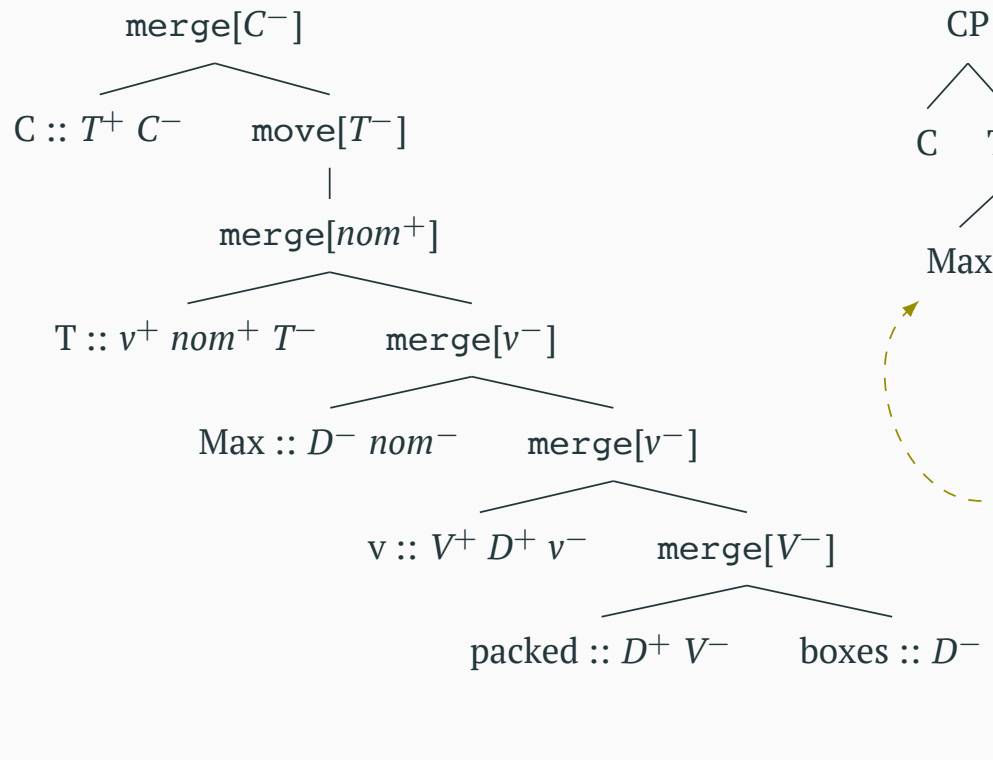


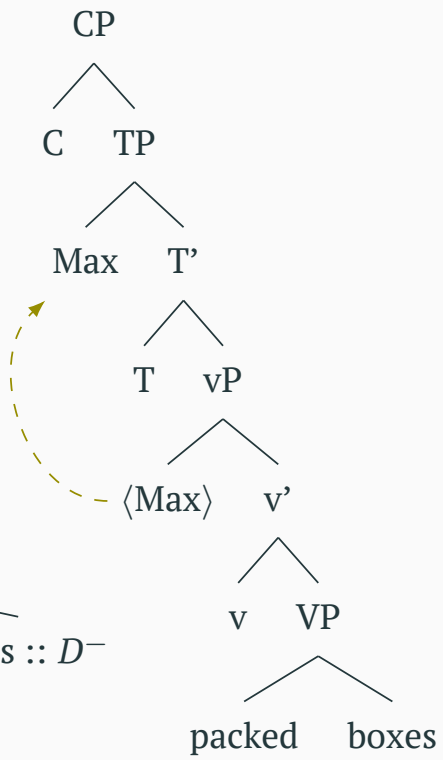
derivation tree

Minimalist Parsing

- Minimalist Grammars (Stabler 1996; 2011, Graf 2013): lexicalized grammar formalisms based on the Minimalist Program (Chomsky 2014).

(7) Max packed boxes.



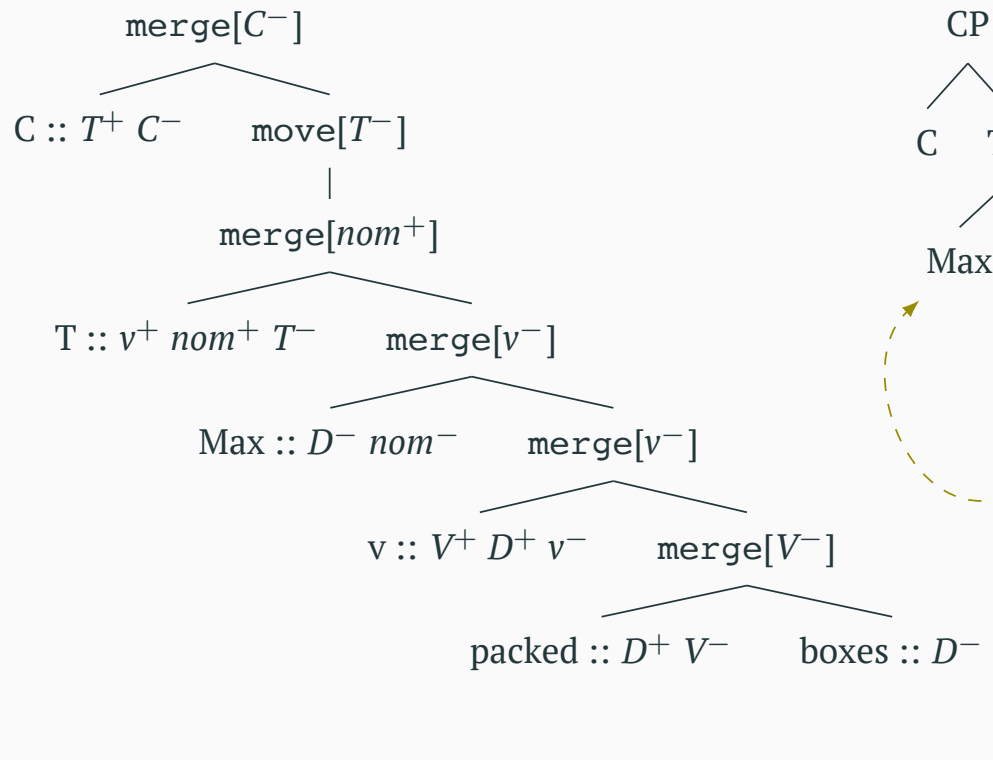


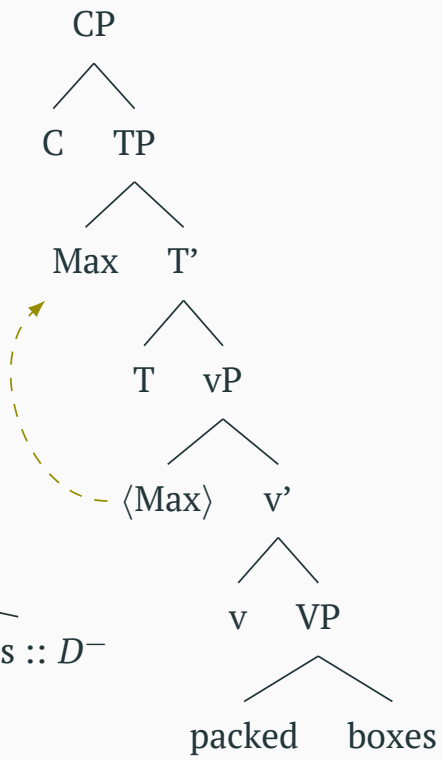
derivation tree

Minimalist Parsing

- Minimalist Grammars (Stabler 1996; 2011, Graf 2013): lexicalized grammar formalisms based on the Minimalist Program (Chomsky 2014).

(7) Max packed boxes.





derivation tree

derived tree

Minimalist Parsing

- Minimalist Grammars (Stabler 1996; 2011, Graf 2013): lexicalized grammar formalisms based on the Minimalist Program (Chomsky 2014).
- Rightward movement possible (Torr and Stabler 2016)
  - Extraposer :: $D^{-}\ D^{+}\ v^{\sim}$
  - no complexity increase

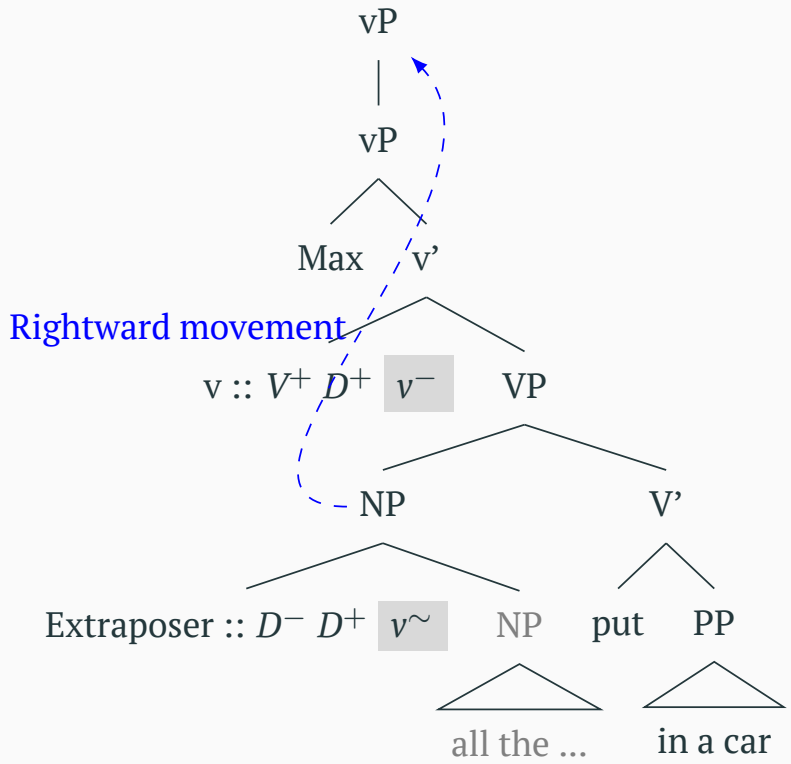
Minimalist Parsing

- Minimalist Grammars (Stabler 1996; 2011, Graf 2013): lexicalized grammar formalisms based on the Minimalist Program (Chomsky 2014).
- Rightward movement possible (Torr and Stabler 2016)
  - Extraposer :: $D^{-}\ D^{+}\ v^{\sim}$
  - no complexity increase

Minimalist Parsing

- Minimalist Grammars (Stabler 1996; 2011, Graf 2013): lexicalized grammar formalisms based on the Minimalist Program (Chomsky 2014).
- Rightward movement possible (Torr and Stabler 2016)
  - Extraposer :: $D^{-}\ D^{+}\ v\sim$
  - no complexity increase

(8) put [PP in a car] [DP all the boxes...]



Minimalist Parsing

- Minimalist Grammars (Stabler 1996; 2011, Graf 2013): lexicalized grammar formalisms based on the Minimalist Program (Chomsky 2014).

- Rightward movement possible

  (Torr and Stabler 2016)

  - Extraposer :: $D^{-}\ D^{+}\ v^{\sim}$
  - no complexity increase

- Head movement possible

  (Kobele et al. 2013)

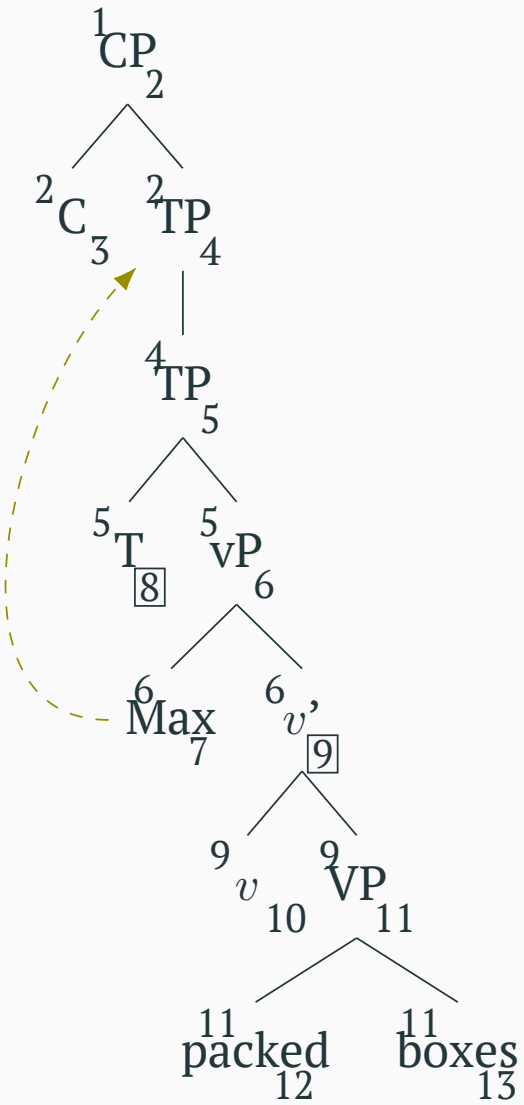
  - also discussed later on

Minimalist Parsing

A top-down parser for MGs (Stabler 2013, Graf et al. 2015a):

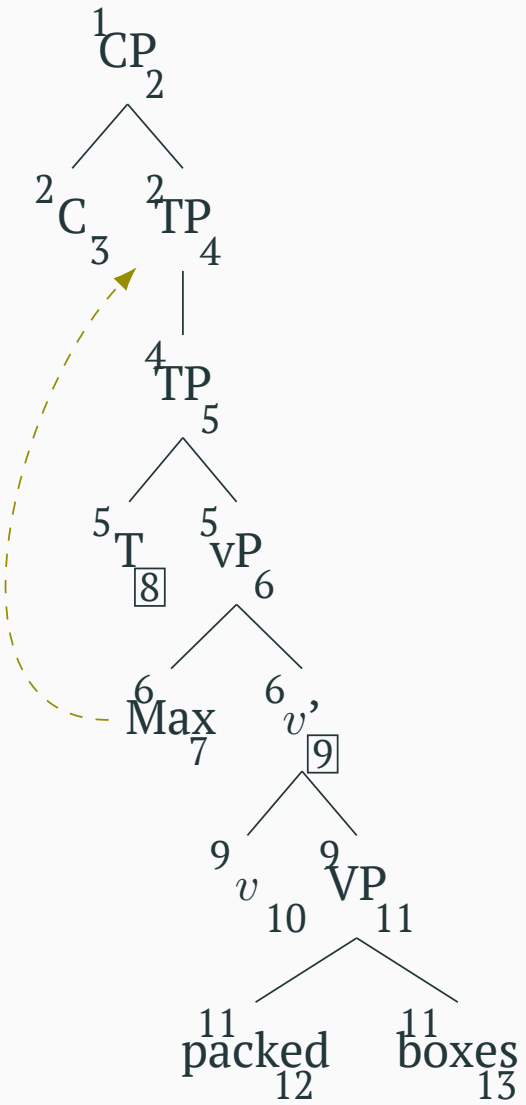
- takes as input a string with pronounced and unpronounced nodes,
- based on MG rules,
- outputs (derivation) trees

Minimalist Parsing



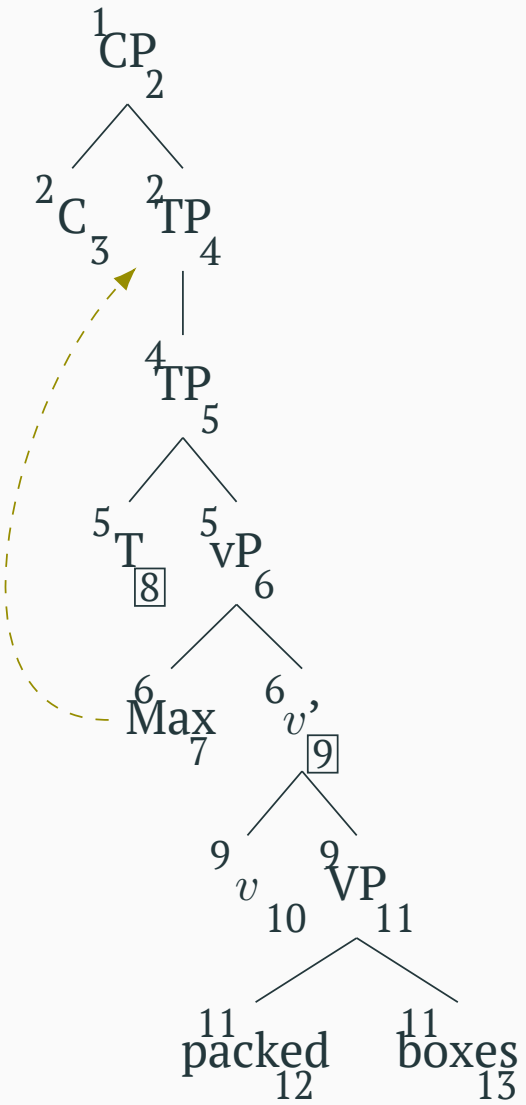
- An annotated tree is a record of the parser's behavior

Minimalist Parsing



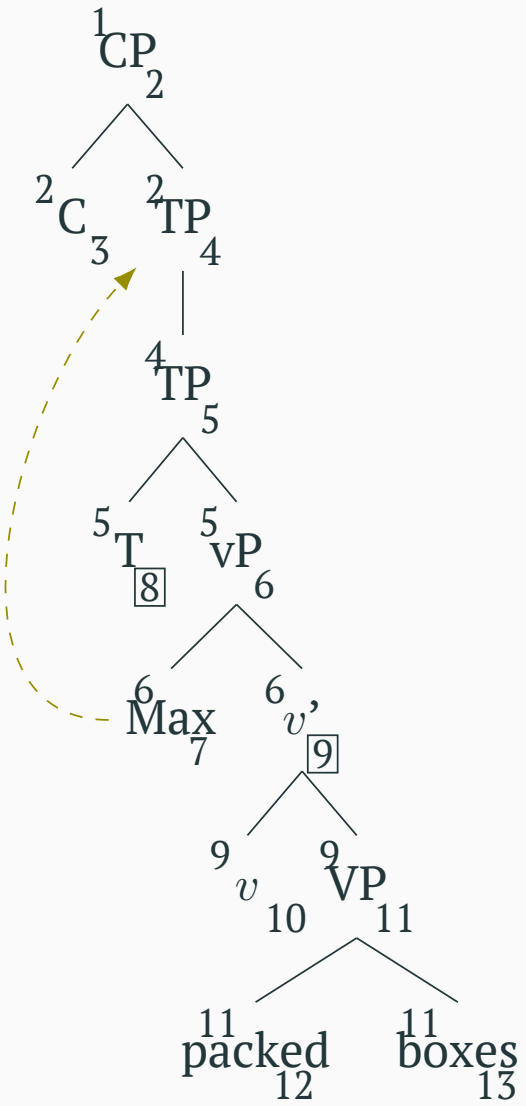
- An annotated tree is a record of the parser's behavior
- ${}^{1}\mathrm{CP}_{2}$

Minimalist Parsing



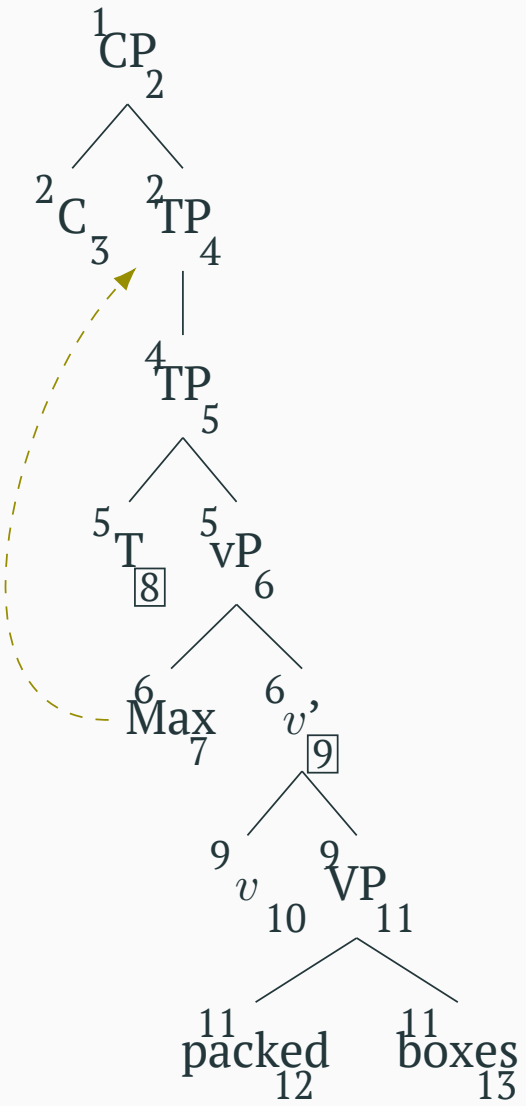
- An annotated tree is a record of the parser's behavior
- ${}^{1}\mathrm{CP}_{2}$
  - index: when the parser puts the node in the memory

Minimalist Parsing



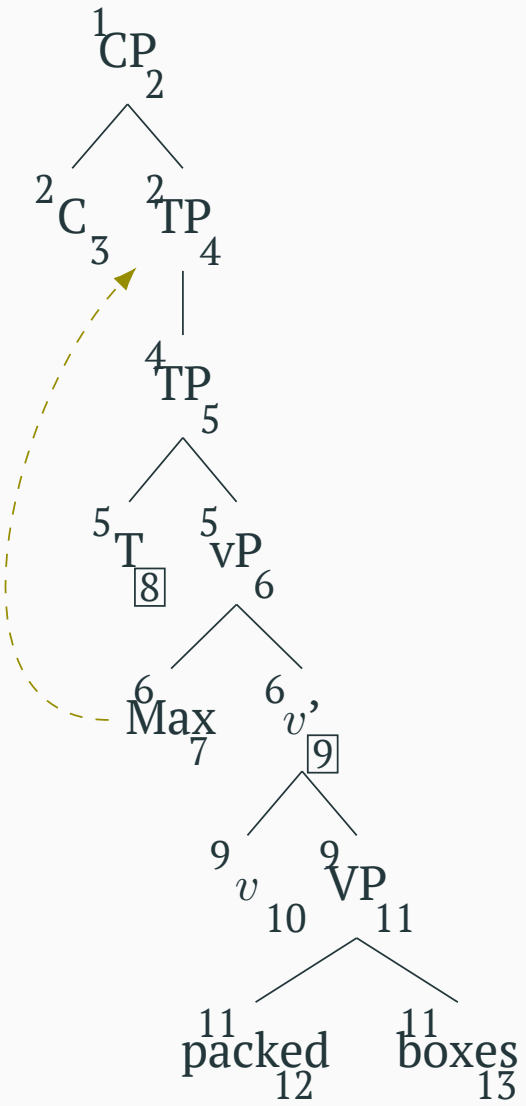
- An annotated tree is a record of the parser's behavior
- ${}^{1}\mathrm{CP}_{2}$
  - index: when the parser puts the node in the memory
  - outdex: when the parser throws the node out of the memory

Minimalist Parsing



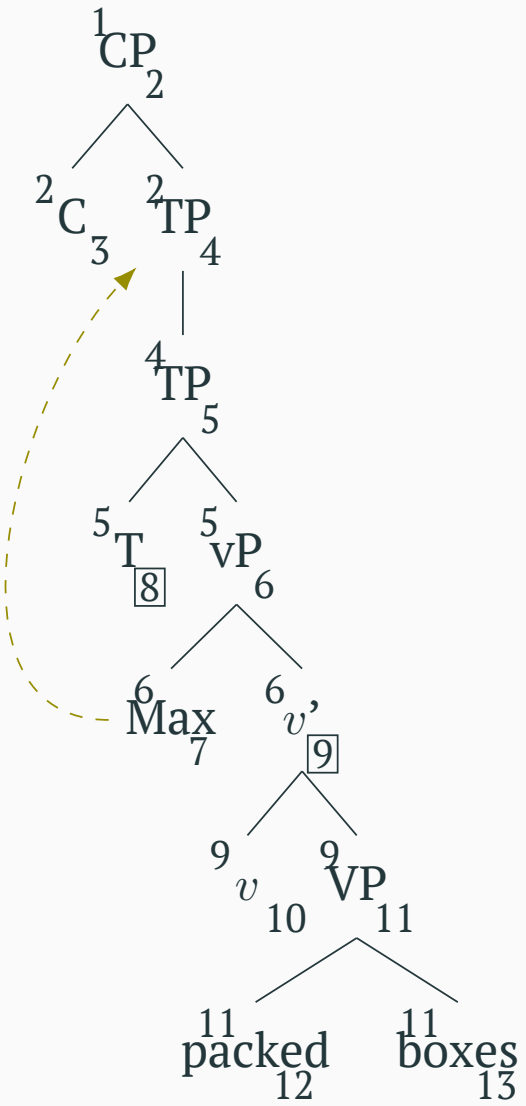
- An annotated tree is a record of the parser's behavior
- ${}^{1}\mathrm{CP}_{2}$
  - index: when the parser puts the node in the memory
  - outdex: when the parser throws the node out of the memory
- Memory usage (Kobele et al. 2013, Graf et al. 2015b)

Minimalist Parsing



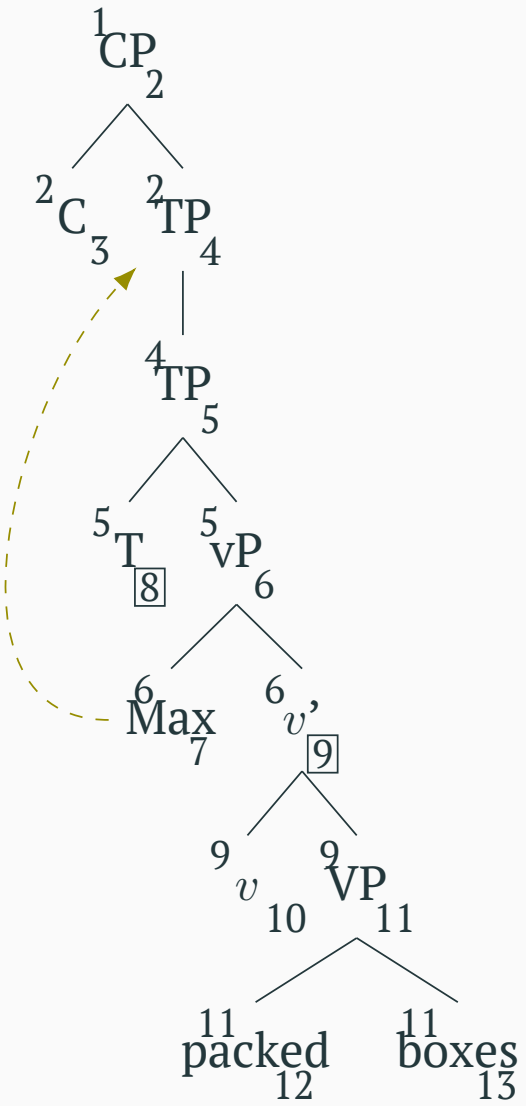
- An annotated tree is a record of the parser's behavior
- ${}^{1}CP_{2}$
  - index: when the parser puts the node in the memory
  - outdex: when the parser throws the node out of the memory
- Memory usage (Kobele et al. 2013, Graf et al. 2015b)
  - Tenure: how long a parse item is held in memory

Minimalist Parsing



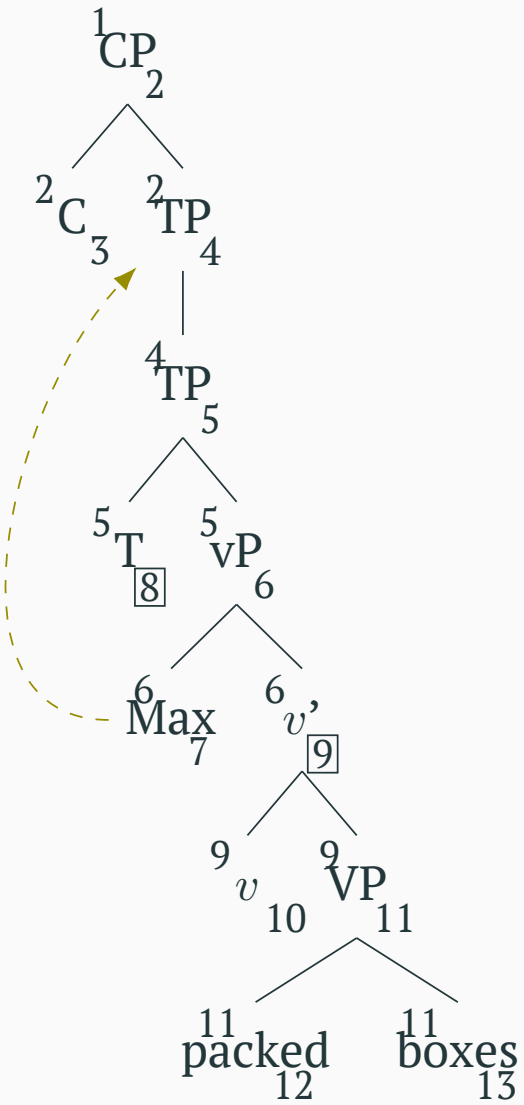
- An annotated tree is a record of the parser's behavior
- ${}^{1}\mathrm{CP}_{2}$
  - index: when the parser puts the node in the memory
  - outdex: when the parser throws the node out of the memory
- Memory usage (Kobele et al. 2013, Graf et al. 2015b)
  - Tenure: how long a parse item is held in memory
  - Payload: how many items are held in memory throughout a parse

Minimalist Parsing



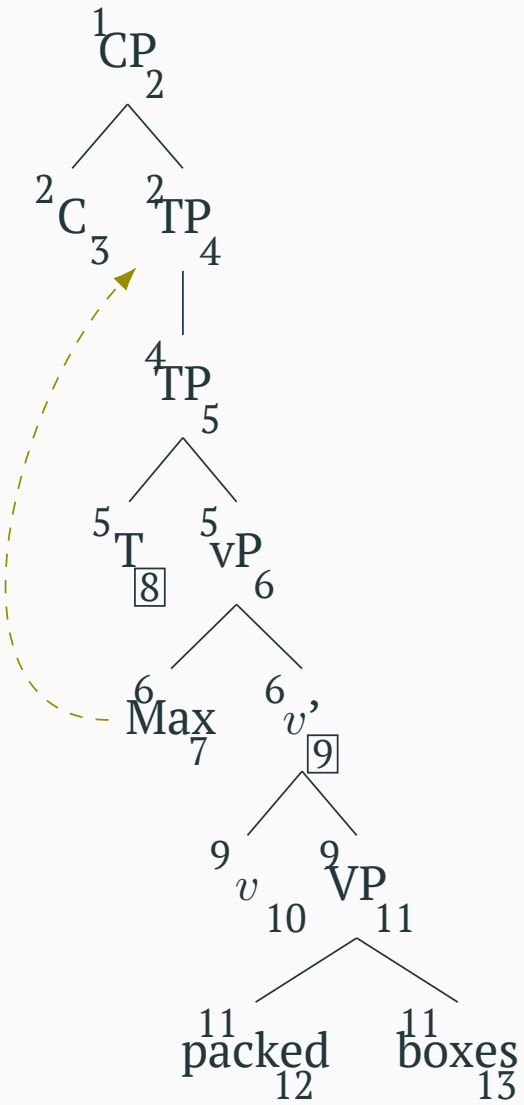
- An annotated tree is a record of the parser's behavior
- ${}^{1}\mathrm{CP}_{2}$
  - index: when the parser puts the node in the memory
  - outdex: when the parser throws the node out of the memory
- Memory usage (Kobele et al. 2013, Graf et al. 2015b)
  - Tenure: how long a parse item is held in memory
  - Payload: how many items are held in memory throughout a parse
  - Size: the length of movement dependencies

Minimalist Parsing



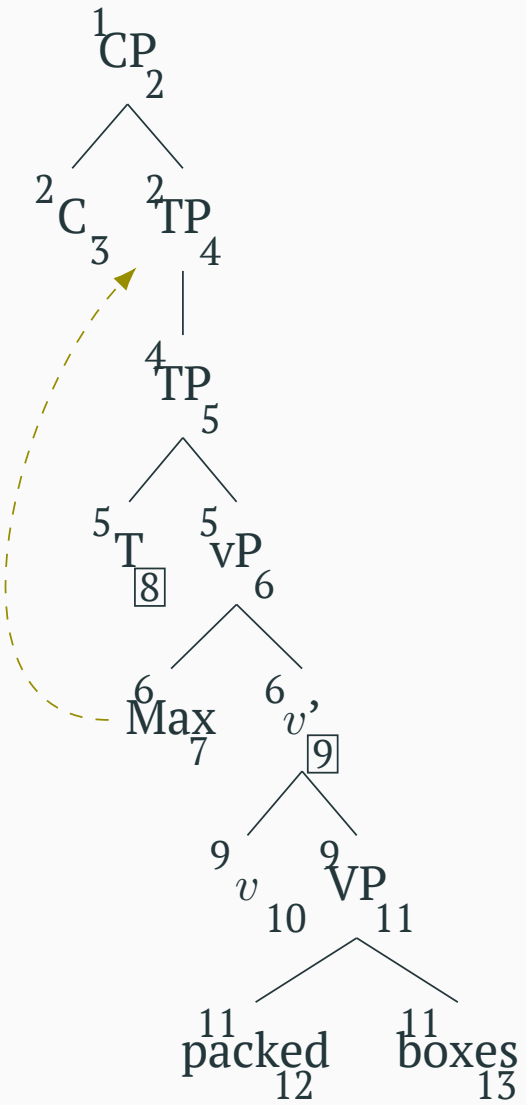
- An annotated tree is a record of the parser's behavior

Minimalist Parsing



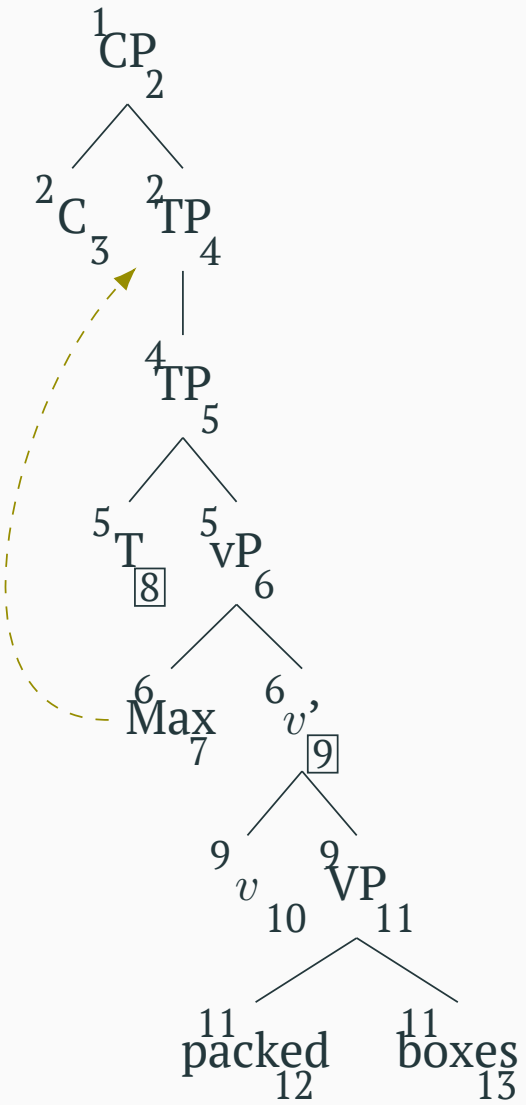
- An annotated tree is a record of the parser's behavior
- Memory usage formalized: complexity metrics (Kobele et al. 2013, Graf et al. 2015b)

Minimalist Parsing



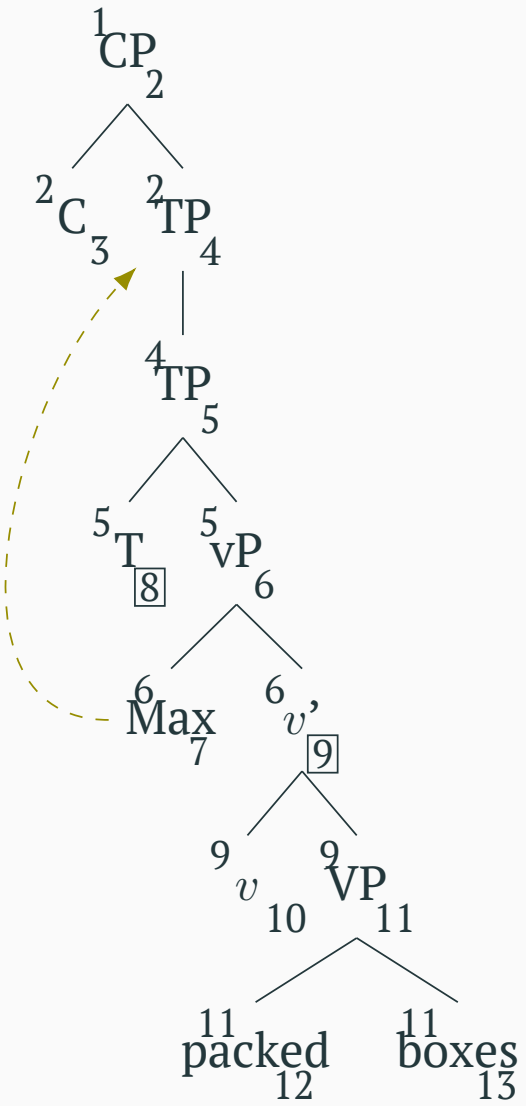
- An annotated tree is a record of the parser's behavior
- Memory usage formalized: complexity metrics (Kobele et al. 2013, Graf et al. 2015b)
- $\textbf{MaxT} := max(\text{tenure-of}(n) | n \in T)$

Minimalist Parsing



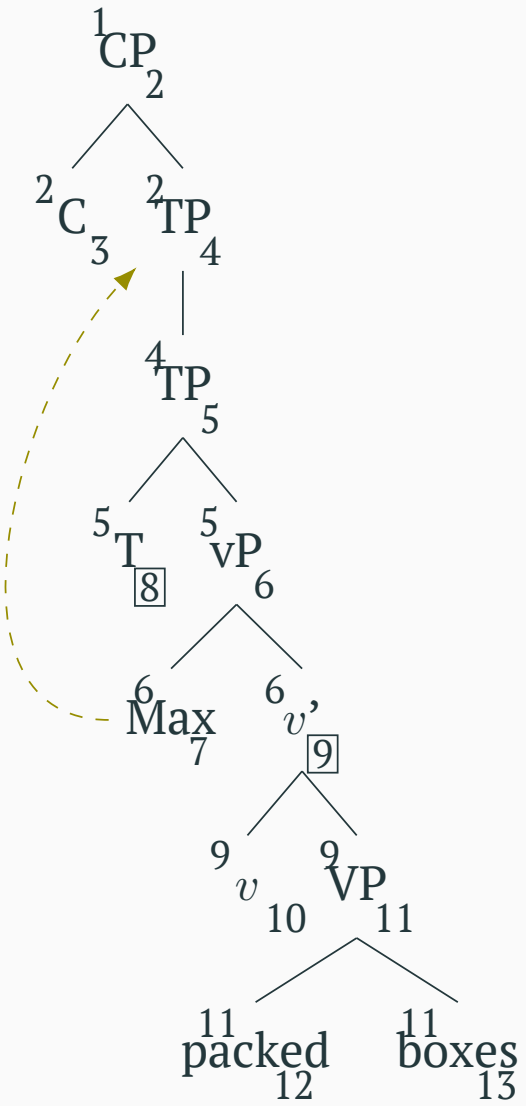
- An annotated tree is a record of the parser's behavior
- Memory usage formalized: complexity metrics (Kobele et al. 2013, Graf et al. 2015b)
- $\textbf{MaxT} := max(\text{tenure-of}(n) | n \in T)$

Minimalist Parsing



- An annotated tree is a record of the parser's behavior
- Memory usage formalized: complexity metrics (Kobele et al. 2013, Graf et al. 2015b)
- **MaxT** $:= max(\text{tenure-of}(n)|n \in T) = 3$

Minimalist Parsing



- An annotated tree is a record of the parser's behavior
- Memory usage formalized: complexity metrics (Kobele et al. 2013, Graf et al. 2015b)
- **MaxT** $:= max(\text{tenure-of}(n) | n \in T) = 3$

  SumT $:= \sum_{n \in T} \text{tenure-of}(n) = 6$

Parsing End-weight - HNPS

(9) a. Max put [DP boxes] [PP in a car]. (short-DP short-PP)

b. Max put [PP in a car] [DP boxes]. (short-PP short-DP)

(10) a. Max put [DP boxes] [PP in a car made in Stuttgart]. (short-DP long-PP)

b. Max put [PP in a car made in Stuttgart] [DP boxes]. (long-PP short-DP)

(11) a. Max put [DP all the boxes of home furnishings] [PP in a car].
(heavy NP)

b. Max put [PP in a car] [DP all the boxes of home furnishings].
(heavy NP shift)

(12) a. Max put [DP all the boxes of home furnishings] [PP in a car made in Stuttgart].
(long-DP long-PP)

b. Max put [PP in a car made in Stuttgart] [DP all the boxes of home furnishings].
(long-PP long-DP)

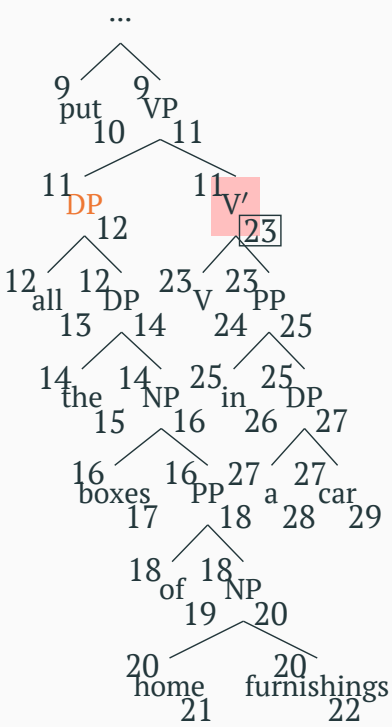
Parsing End-weight - HNPS

Weight config.	Shift advantage?	Parser prediction
Both light	No	No
Heavy PP	No	No
Heavy NP	Yes	Yes (`MaxT`: 8 vs. 12)
Both Heavy	No	No (`MaxT`: 14 vs. 12)

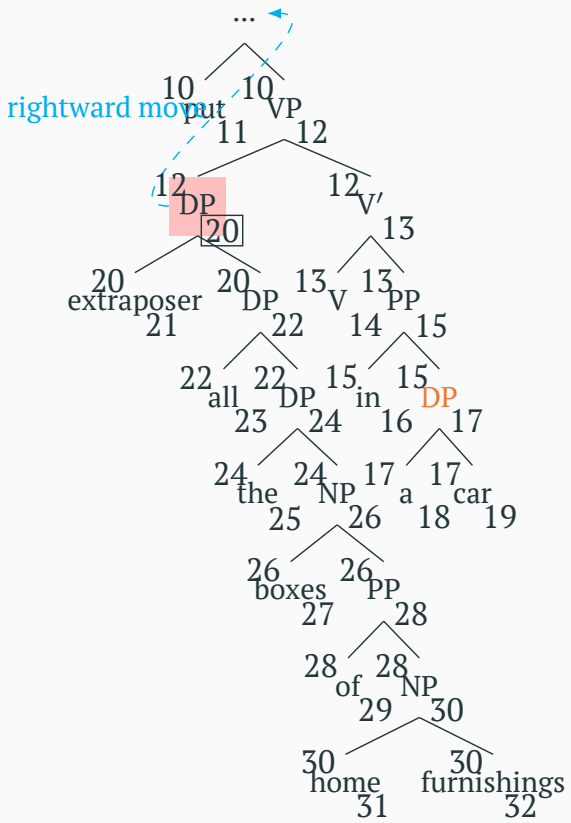
Table 1: Summary of the predictions for each weight configuration in object shift constructions

Parsing End-weight - HNPS

(13) put [DP …boxes…] [PP in…]
canonical order

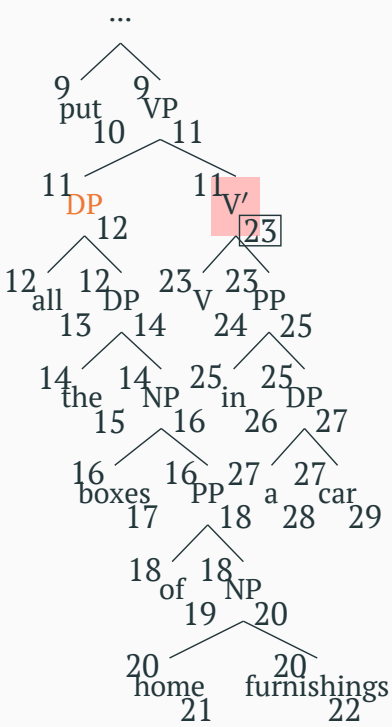


(14) put [PP in…] [DP …boxes…]
HNPS order

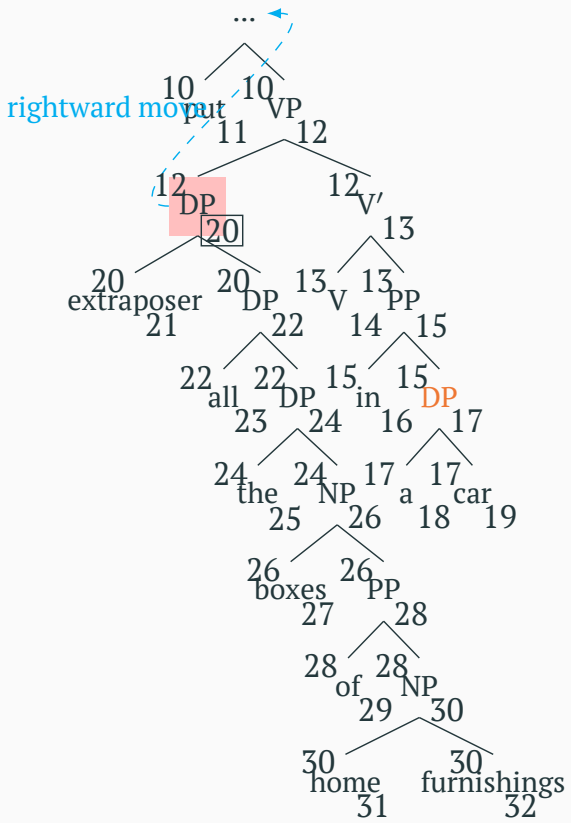


Parsing End-weight - HNPS

(13) put [DP …boxes…] [PP in…]
canonical order



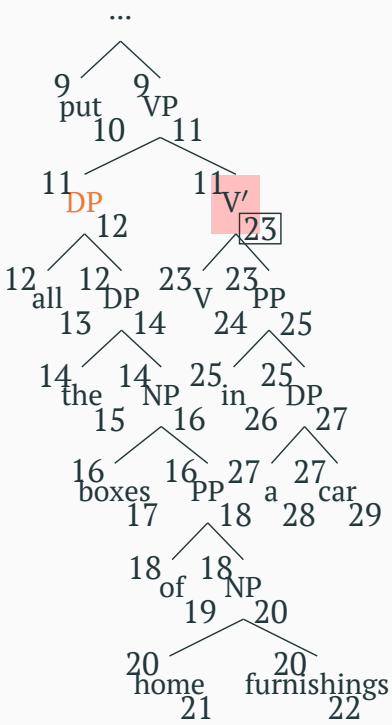
(14) put [PP in…] [DP …boxes…]
HNPS order



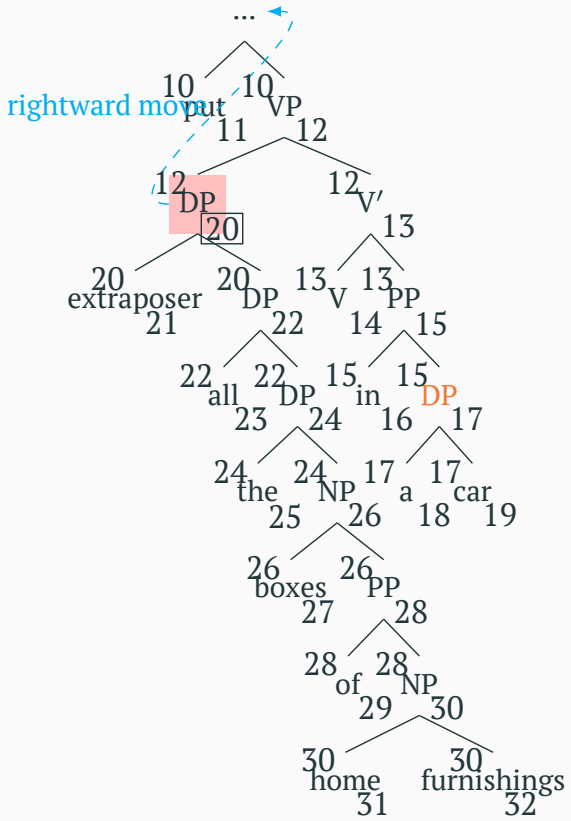
- Weight → Steps

Parsing End-weight - HNPS

(13) put [DP …boxes…] [PP in…]
canonical order



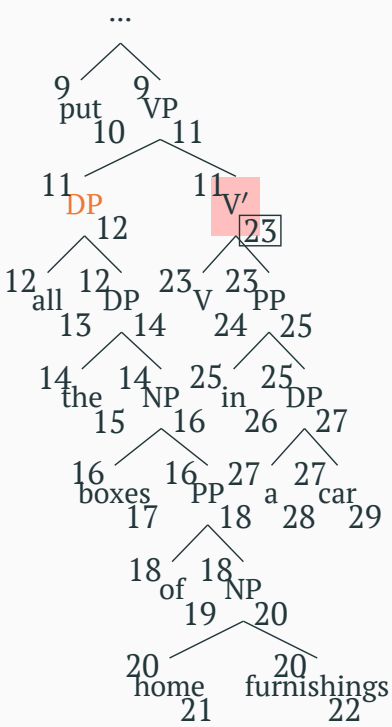
(14) put [PP in…] [DP …boxes…]
HNPS order



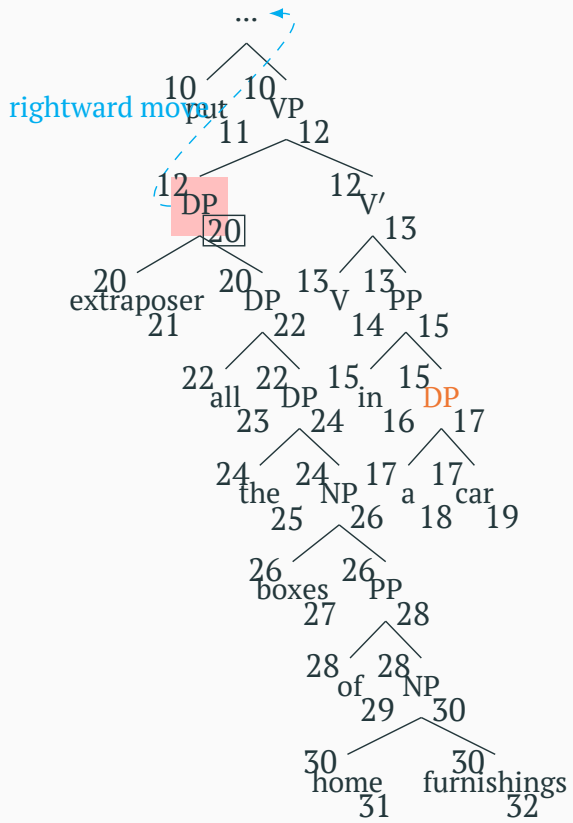
- Weight → Steps
- **MaxT**: 12/V’ (canonical)

Parsing End-weight - HNPS

(13) put [DP …boxes…] [PP in…]
canonical order



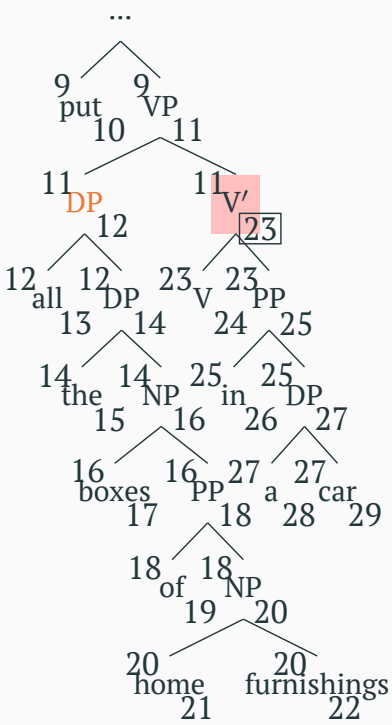
(14) put [PP in…] [DP …boxes…]
HNPS order



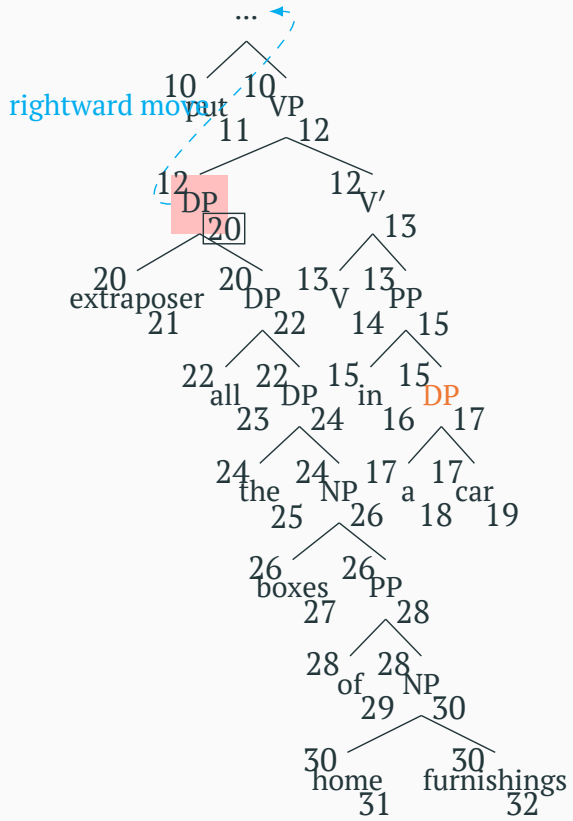
- Weight → Steps
- **MaxT**: 12/V’ (canonical) > 8/DP (HNPS)

Parsing End-weight - HNPS

(13) put [DP …boxes…] [PP in…]
canonical order



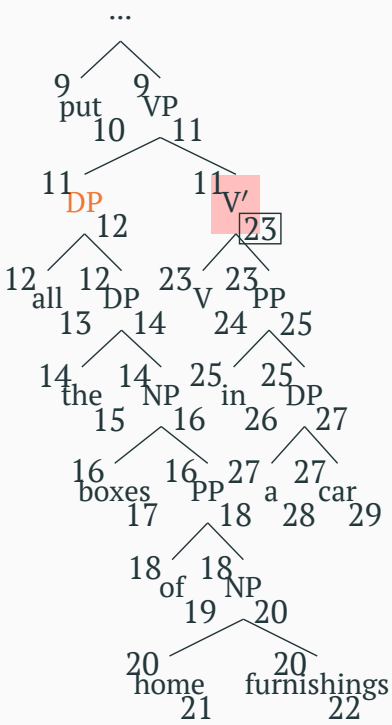
(14) put [PP in…] [DP …boxes…]
HNPS order



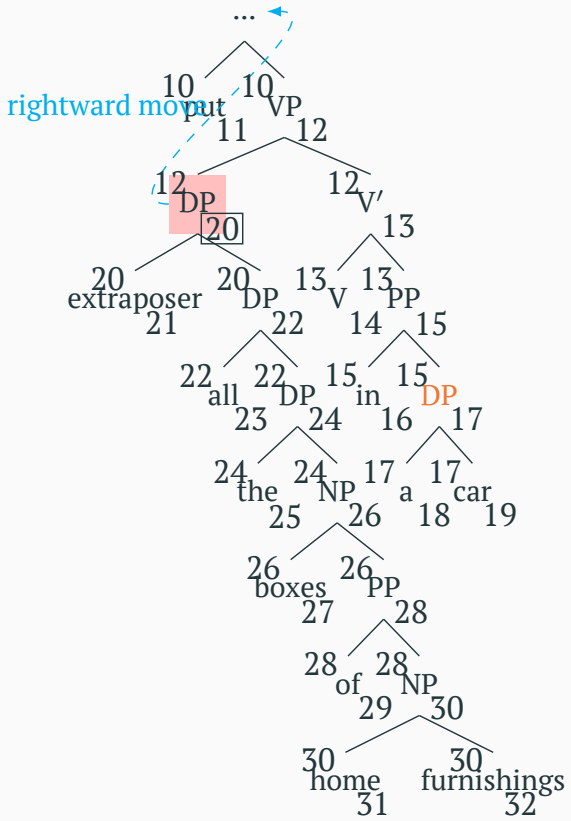
- Weight → Steps
- **MaxT**: 12/V’ (canonical) > 8/DP (HNPS) → end-weight preferred!

Parsing End-weight - HNPS

(13) put [DP …boxes…] [PP in…]
canonical order



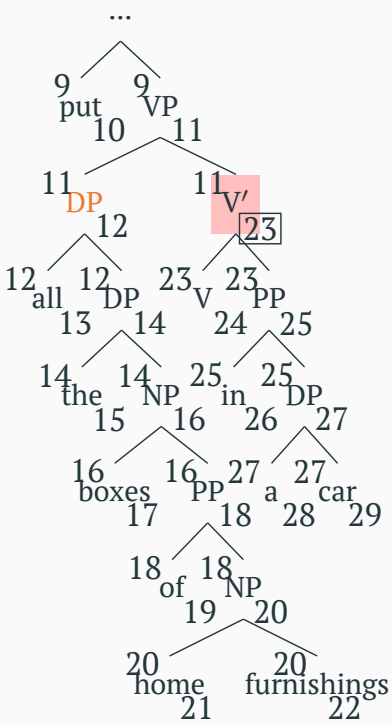
(14) put [PP in…] [DP …boxes…]
HNPS order



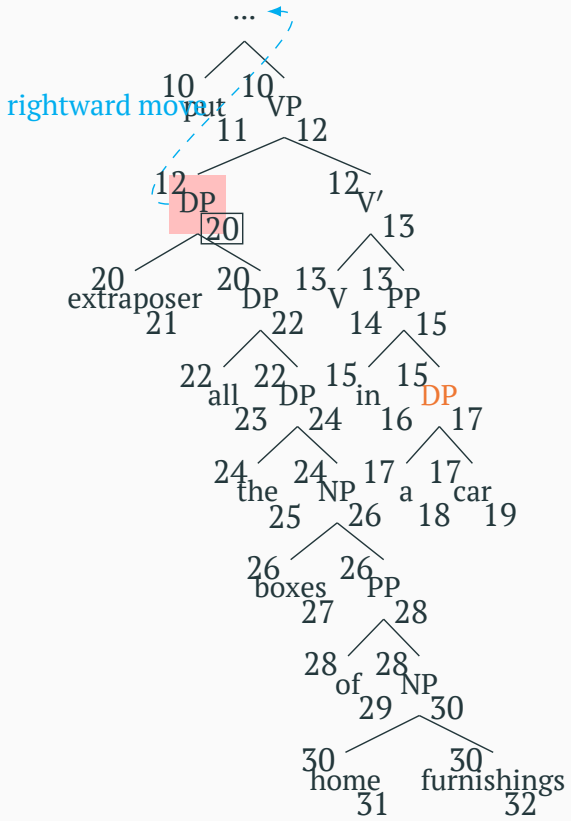
- Weight → Steps
- **MaxT**: 12/V’ (canonical) > 8/DP (HNPS) → end-weight preferred!
- **MaxT**(DP) grows with V’

Parsing End-weight - HNPS

(13) put [DP …boxes…] [PP in…]
canonical order



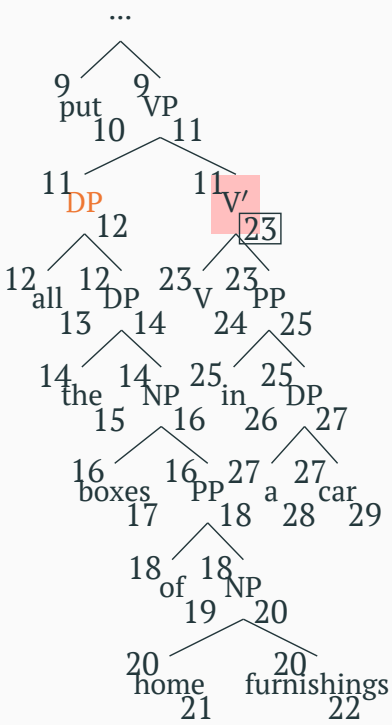
(14) put [PP in…] [DP …boxes…]
HNPS order



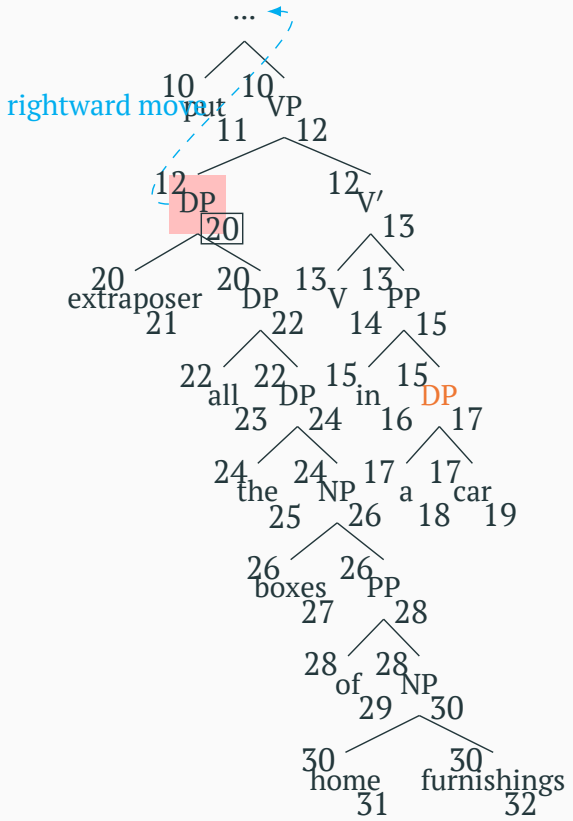
- Weight → Steps
- **MaxT**: 12/V’ (canonical) > 8/DP (HNPS) → end-weight preferred!
- **MaxT**(DP) grows with V’ → relative weight!

Parsing End-weight - HNPS

(13) put [DP …boxes…] [PP in…]
canonical order



(14) put [PP in…] [DP …boxes…]
HNPS order



- Weight → Steps
- **MaxT**: 12/V’ (canonical) > 8/DP (HNPS) → end-weight preferred!
- **MaxT**(DP) grows with V’ → relative weight!

  SumT: 18 (canonical) > 15 (HNPS) ✓

Parsing End-weight - PV

(15) short DP

a. Chris **put on** a hat.

b. Chirs **put** a hat **on**.

(16) [mod-DP]

a. Chris **put on** a very very very very expensive hat.

b. Chirs **put** a very very very very expensive hat **on**.

(17) [DP-mod]

a. Chris **put on** a hat which Alex made with love.

b. Chris **put** a hat which Alex made with love **on**.

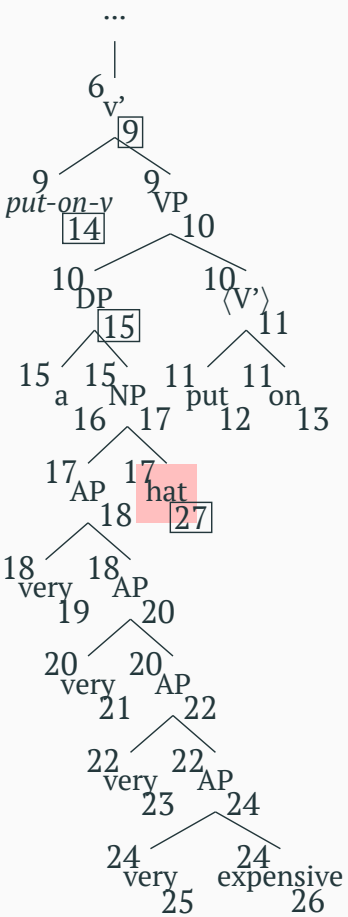
Parsing End-weight - PV

Weight config.	Joined advt?	MG parser
Short DP	No/Unclear	Yes (`MaxT` 5 vs. 6)
[mod-DP]	Yes	Yes (`MaxT` 10 vs. 16)
[DP-mod]	Yes	Yes (`MaxT` 8 vs. 24)

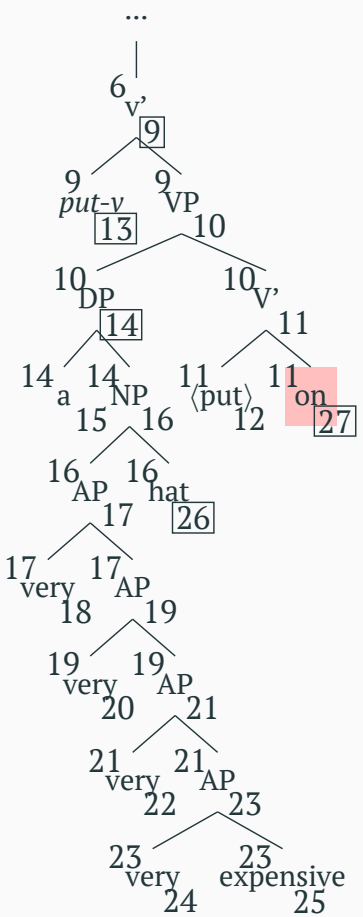
Table 2: Summary of the predictions for each weight configuration in particle verb constructions

Parsing End-weight - PV

(18) **put on** a very very…hat.
Joined order

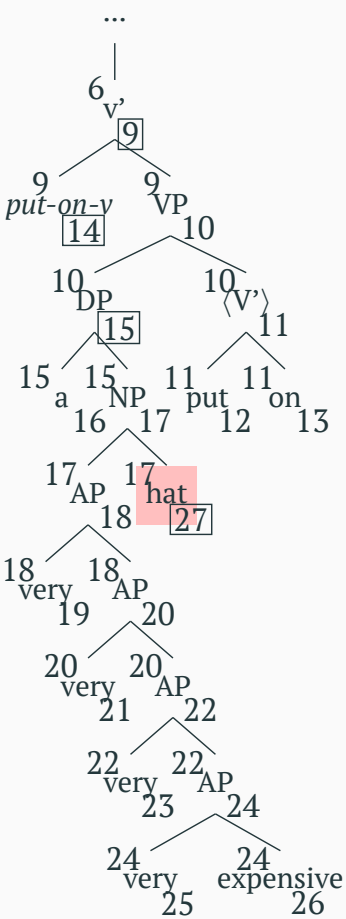


(19) **put** a very very…hat **on**.
Separated order

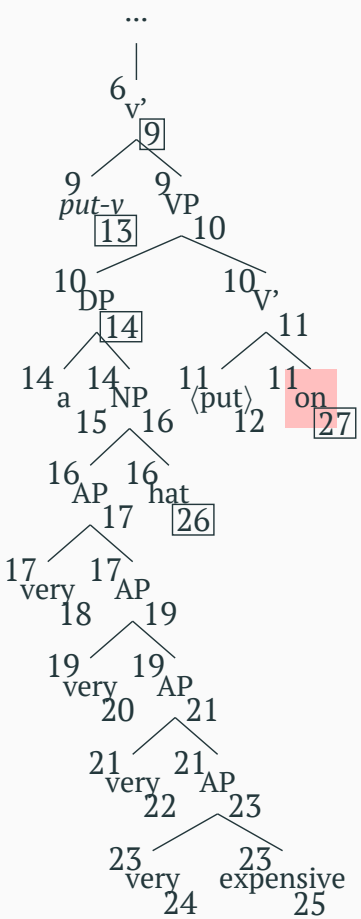


Parsing End-weight - PV

(18) **put on** a very very…hat.
Joined order



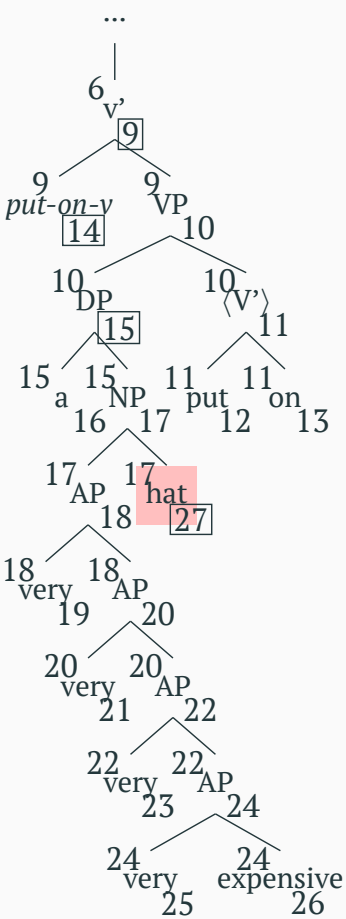
(19) **put** a very very…hat **on**.
Separated order



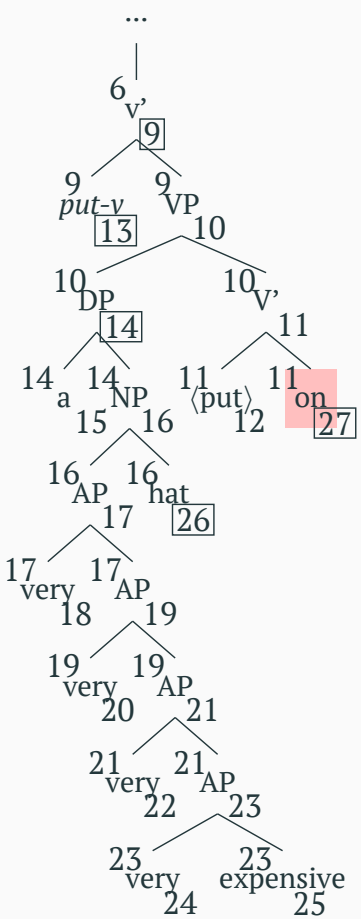
- **MaxT**: 10/hat (Joined)

Parsing End-weight - PV

(18) **put on** a very very…hat.
Joined order



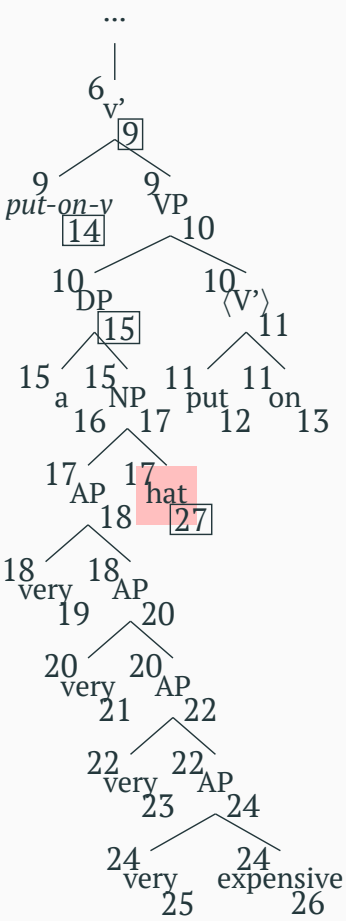
(19) **put** a very very…hat **on**.
Separated order



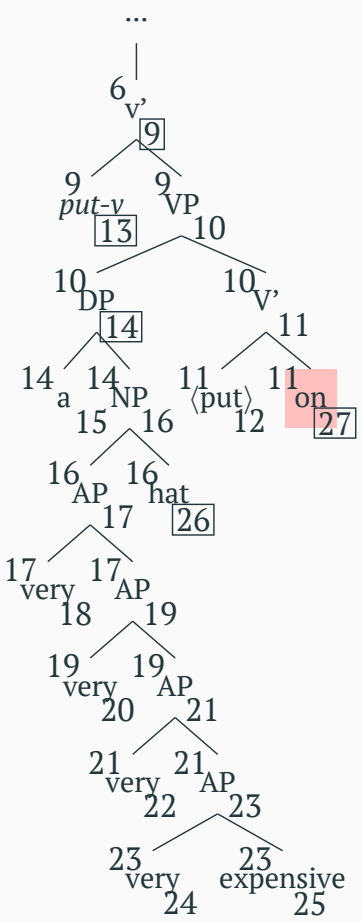
- **MaxT**: 10/hat (Joined) < 16/on (Separated.)

Parsing End-weight - PV

(18) **put on** a very very…hat.
Joined order



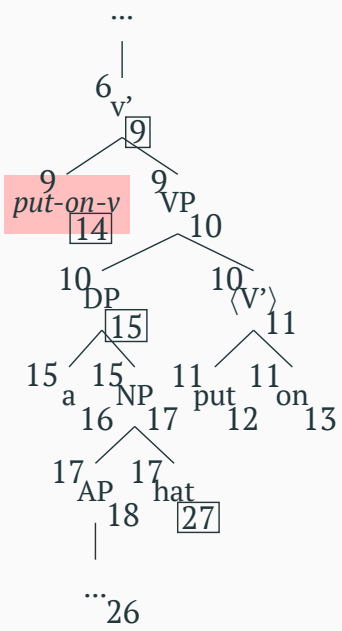
(19) **put** a very very…hat **on**.
Separated order



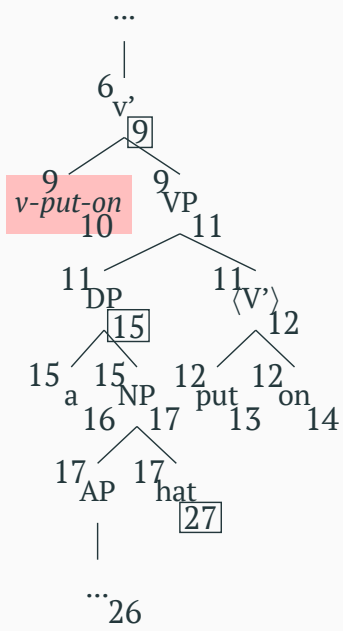
- **MaxT**: 10/hat (Joined) < 16/on (Separated.) → end-weight preferred!

Parsing End-weight - PV

(20) V-to-*v* on left.
Joined order

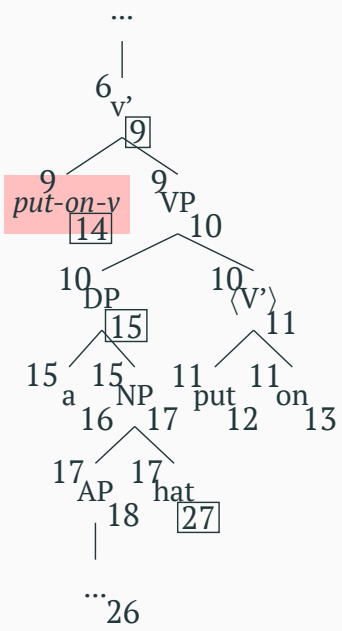


(21) V-to-*v* on right.
Joined order

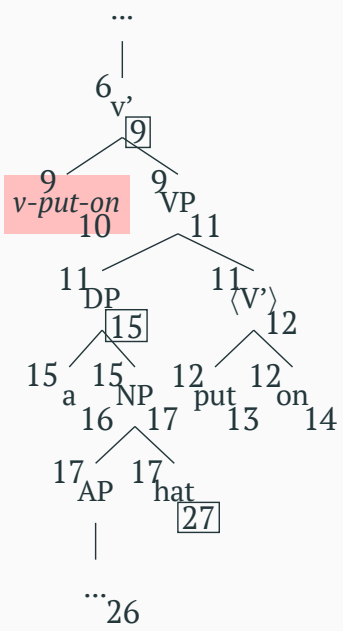


Parsing End-weight - PV

(20) V-to-*v* on left.
Joined order



(21) V-to-*v* on right.
Joined order



- V-to-*v* landing site affects tenure, but not processing prediction

Discussion

Processing phenomena ↔ Complexity metrics ↔ Syntactic structure

Discussion

Processing phenomena ↔ Complexity metrics ↔ Syntactic structure

- Processing phenomena:

Discussion

Processing phenomena ↔ Complexity metrics ↔ Syntactic structure

- Processing phenomena:
    - English end-weight preferences

Discussion

Processing phenomena ↔ Complexity metrics ↔ Syntactic structure

- Processing phenomena:
    - English end-weight preferences
- Syntactic structures:

Discussion

Processing phenomena ↔ Complexity metrics ↔ Syntactic structure

- Processing phenomena:
  - English end-weight preferences
- Syntactic structures:
  - HNPS: rightward movement ↘
  - PV: particle stranding ↗

Complexity metric ↗ ✓ ↘ ✓

Discussion

Processing phenomena ↔ Complexity metrics ↔ Syntactic structure

- Processing phenomena:
  - English end-weight preferences
- Syntactic structures:
  - HNPS: rightward movement ↘
  - PV: particle stranding ↗

Complexity metric ↗ ↘

- → End-weight preference follows from the processing difficulties associated with the syntactic structure of competing word orders.

Discussion

Processing phenomena ↔ Complexity metrics ↔ Syntactic structure

Discussion

Processing phenomena ↔ Complexity metrics ↔ Syntactic structure

- Processing phenomenon:

Discussion

Processing phenomena ↔ Complexity metrics ↔ Syntactic structure

- Processing phenomenon:
    - Japanese initial-weight preference

Discussion

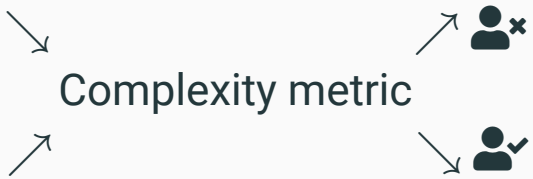
Processing phenomena ↔ Complexity metrics ↔ Syntactic structure

- Processing phenomenon:
    - Japanese initial-weight preference
- Syntactic structures:

Discussion

Processing phenomena ↔ Complexity metrics ↔ Syntactic structure

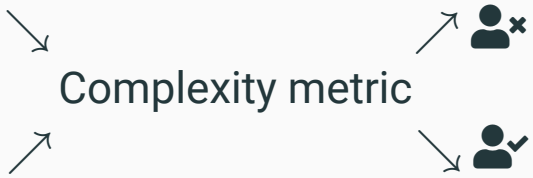
- Processing phenomenon:
  - Japanese initial-weight preference
- Syntactic structures:
  - Scrambling
  - Base-generation



Discussion

Processing phenomena ↔ Complexity metrics ↔ Syntactic structure

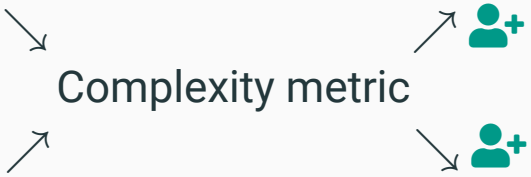
- Processing phenomenon:
  - Japanese initial-weight preference
- Syntactic structures:
  - Scrambling
  - Base-generation



Discussion

Processing phenomena ↔ Complexity metrics ↔ Syntactic structure

- Processing phenomenon:
  - Japanese initial-weight preference
- Syntactic structures:
  - Scrambling
  - Base-generation



Thank you!

References i

Cappelle, B. (2005). Particle patterns in english: A comprehensive coverage.

Chomsky, N. (2014). *The minimalist program*. MIT press.

Graf, T. (2013). *Local and transderivational constraints in syntax and semantics*. PhD thesis, UCLA.

Graf, T., Fodor, B., Monette, J., Rachiele, G., Warren, A., and Zhang, C. (2015a). A refined notion of memory usage for minimalist parsing. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, pages 1–14, Chicago, USA. Association for Computational Linguistics.

Graf, T., Fodor, B., Monette, J., Rachiele, G., Warren, A., and Zhang, C. (2015b). A refined notion of memory usage for minimalist parsing. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, pages 1–14.

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

Stabler, E. (1996). Derivational minimalism. In *International Conference on Logical Aspects of Computational Linguistics*, pages 68–95. Springer.

Stabler, E. P. (2011). Computational perspectives on minimalism. *Oxford handbook of linguistic minimalism*, pages 617–643.

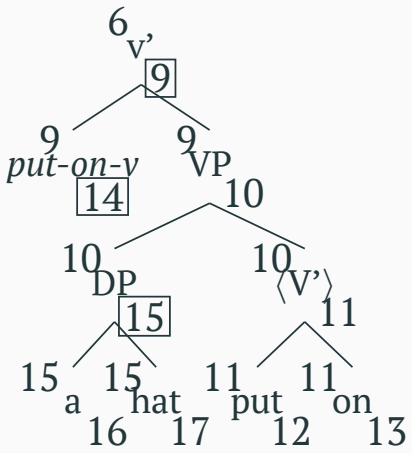
Stabler, E. P. (2013). Two models of minimalist, incremental syntactic analysis. *Topics in cognitive science*, 5(3):611–633.

Stallings, L. M. and MacDonald, M. C. (2011). It's not just the "heavy np": relative phrase length modulates the production of heavy-np shift. *Journal of psycholinguistic research*, 40(3):177–187.

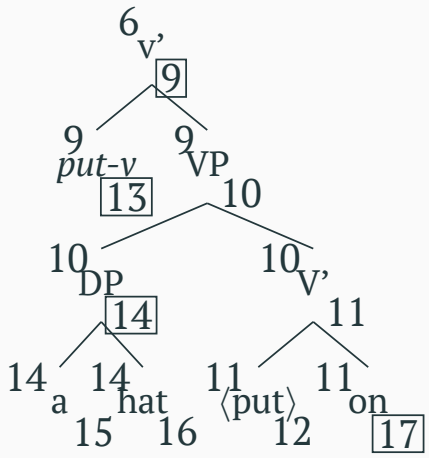
Torr, J. and Stabler, E. (2016). Coordination in minimalist grammars: Excorporation and across the board (head) movement. In *Proceedings of the 12th international workshop on tree adjoining grammars and related formalisms (TAG+ 12)*, pages 1–17.

No/Unclear Short PV?

(22) **put on** a hat



(23) **put** a hat **on**

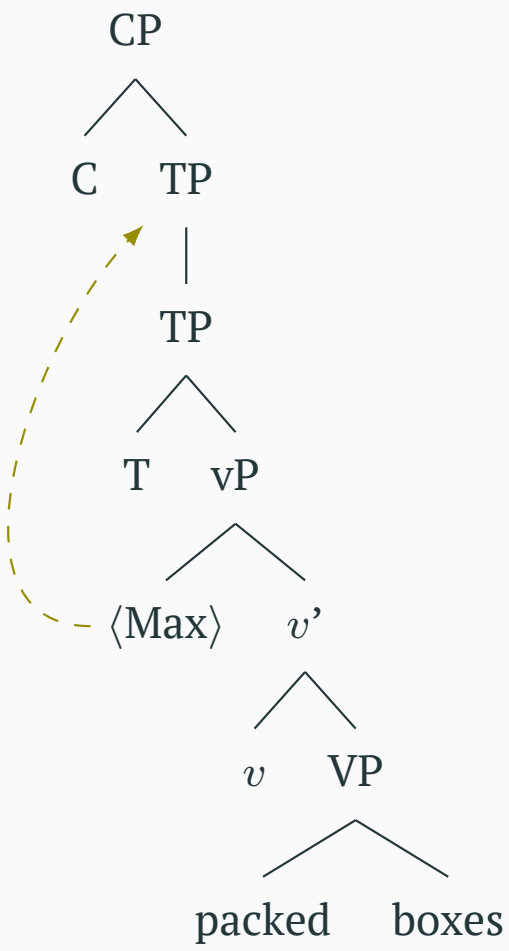


- truly default order?
- other factors?

MG parser in action

(24) • C Max T v packed boxes.

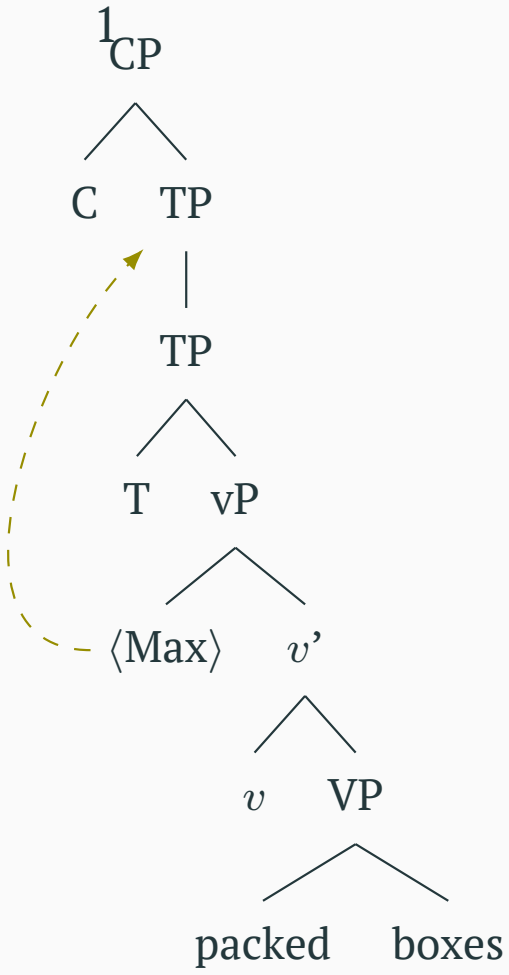
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' expands to v and VP	look for v
Step 10	v is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) • C Max T v packed boxes.

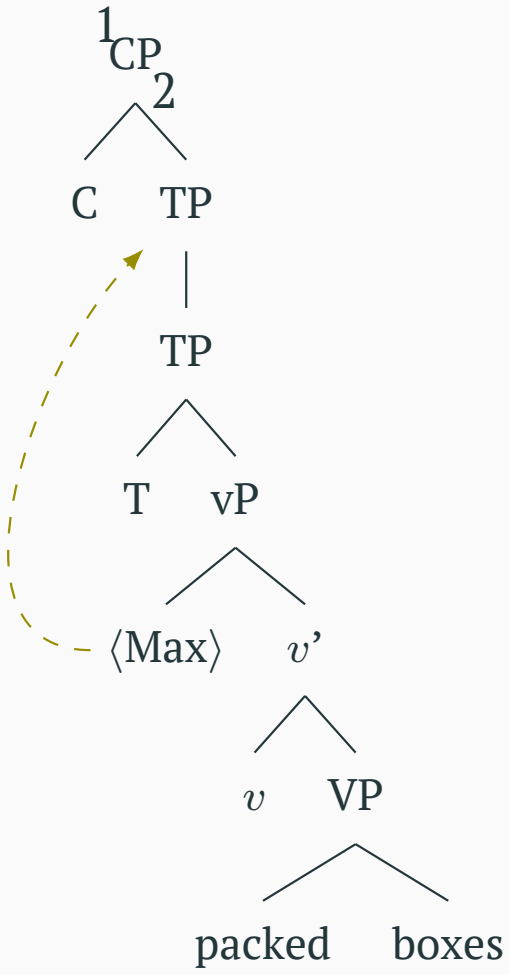
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' expands to v and VP	look for v
Step 10	v is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) • C Max T v packed boxes.

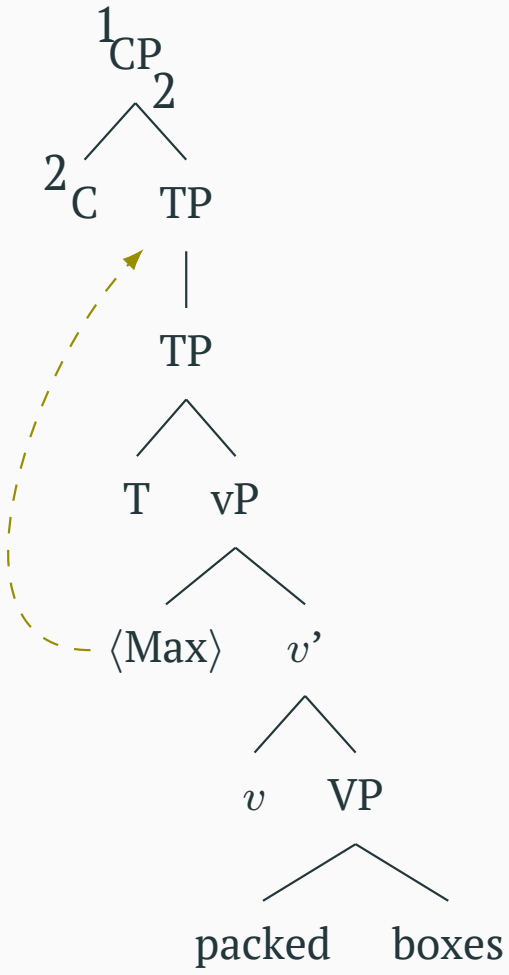
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' expands to v and VP	look for v
Step 10	v is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) • C Max T v packed boxes.

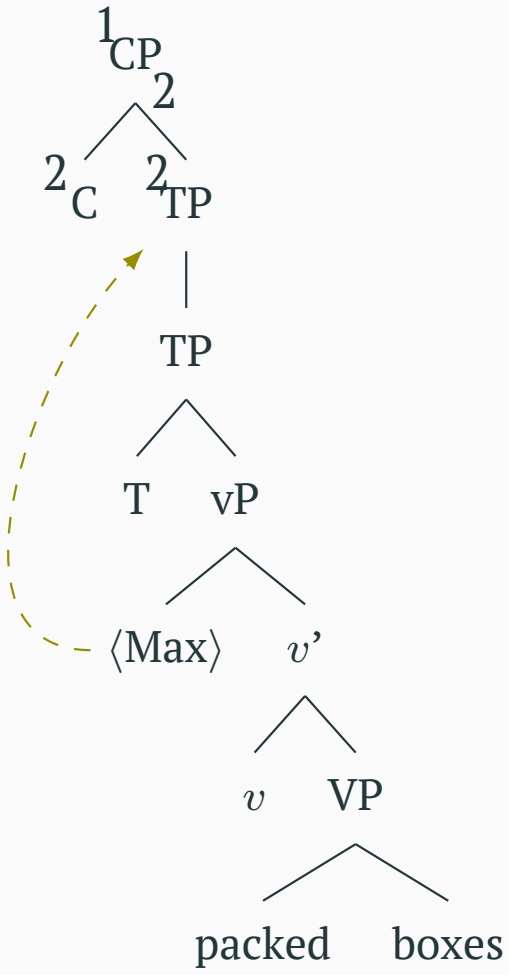
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' expands to v and VP	look for v
Step 10	v is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) • C Max T v packed boxes.

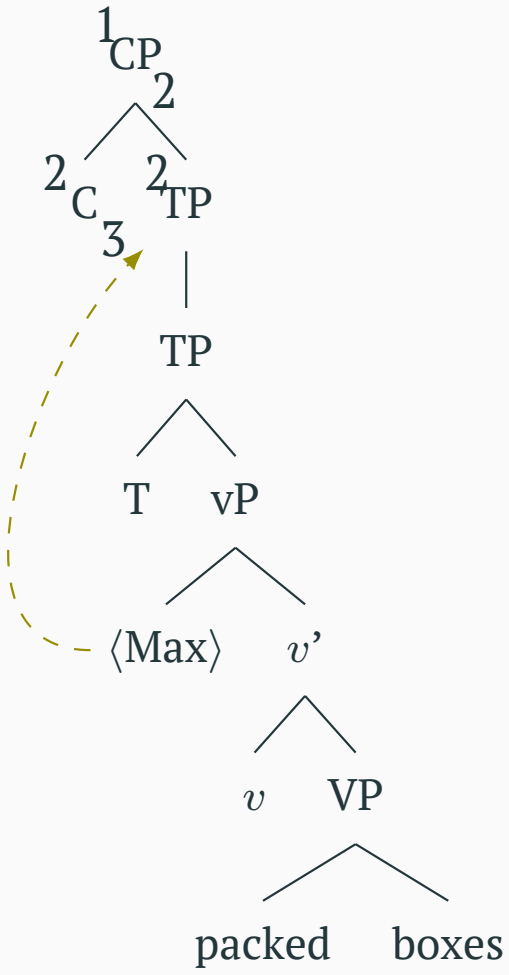
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' expands to v and VP	look for v
Step 10	v is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C • Max T *v* packed boxes.

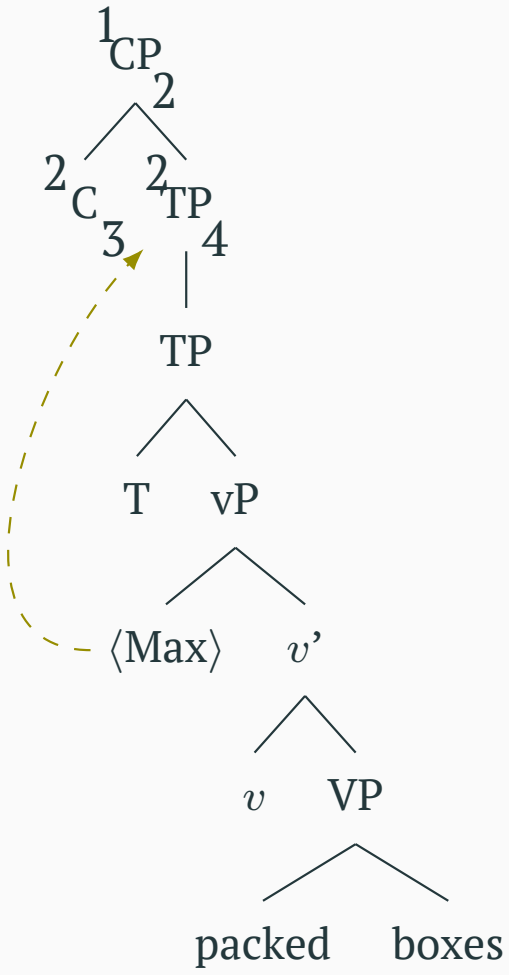
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and *v*′	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for *v*
Step 9	*v*′ expands to *v* and VP	look for *v*
Step 10	*v* is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C • Max T *v* packed boxes.

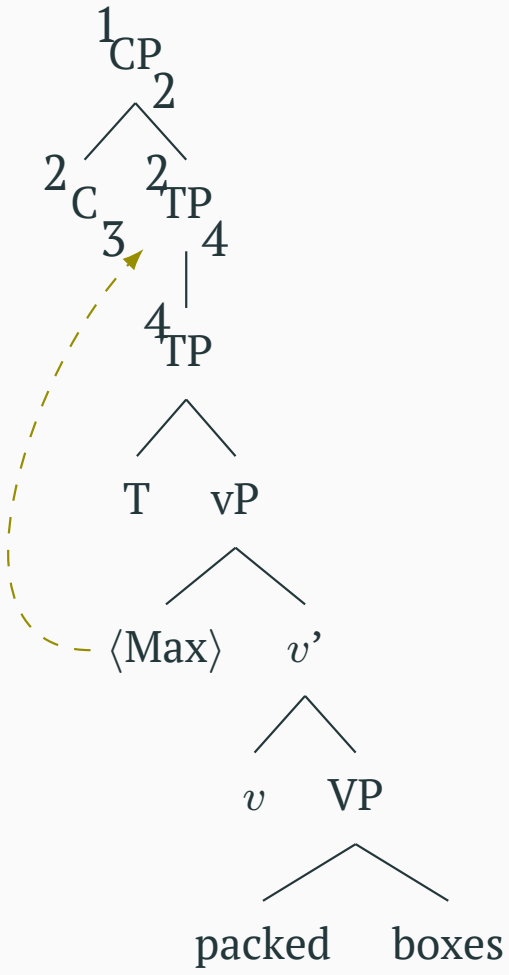
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and *v′*	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for *v*
Step 9	*v′* expands to *v* and VP	look for *v*
Step 10	*v* is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C • Max T *v* packed boxes.

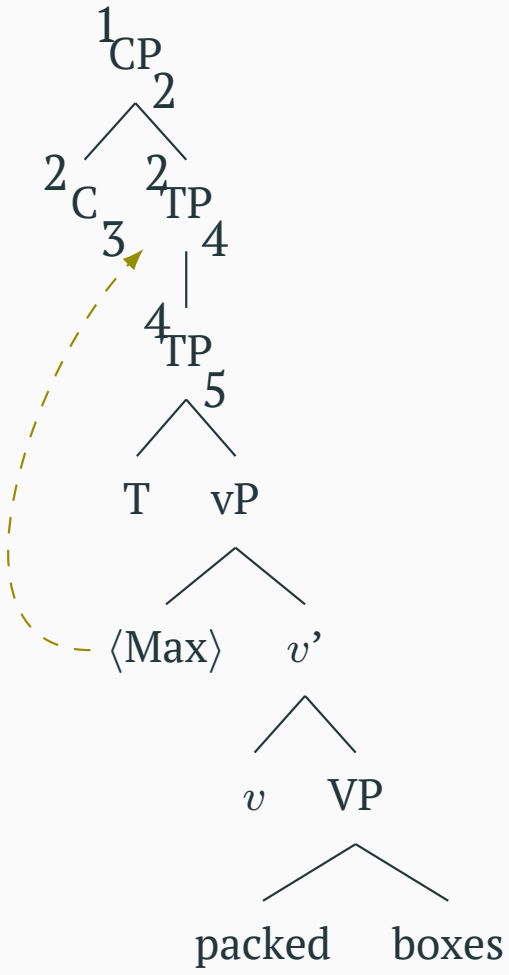
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and *v′*	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for *v*
Step 9	*v′* expands to *v* and VP	look for *v*
Step 10	*v* is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C • Max T v packed boxes.

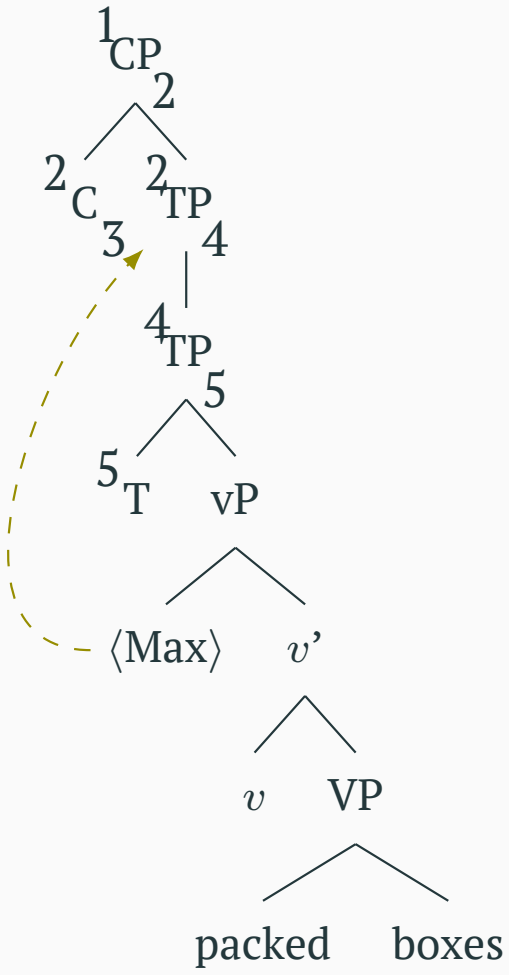
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' expands to v and VP	look for v
Step 10	v is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C • Max T v packed boxes.

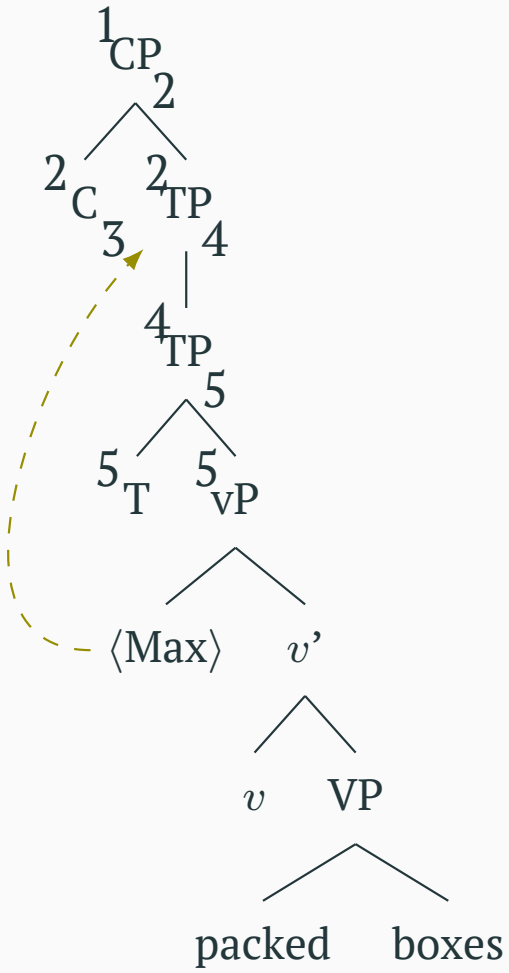
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' expands to v and VP	look for v
Step 10	v is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C • Max T *v* packed boxes.

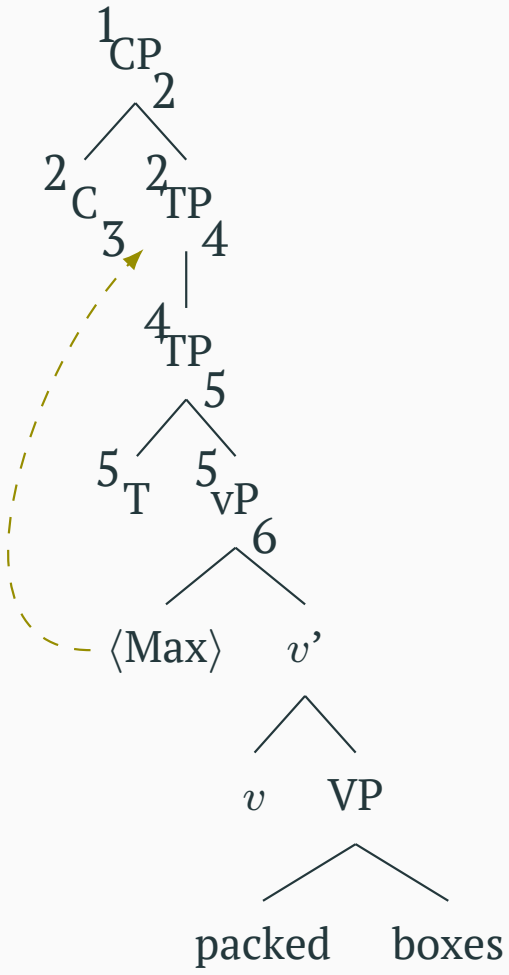
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and *v*′	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for *v*
Step 9	*v*′ expands to *v* and VP	look for *v*
Step 10	*v* is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C • Max T *v* packed boxes.

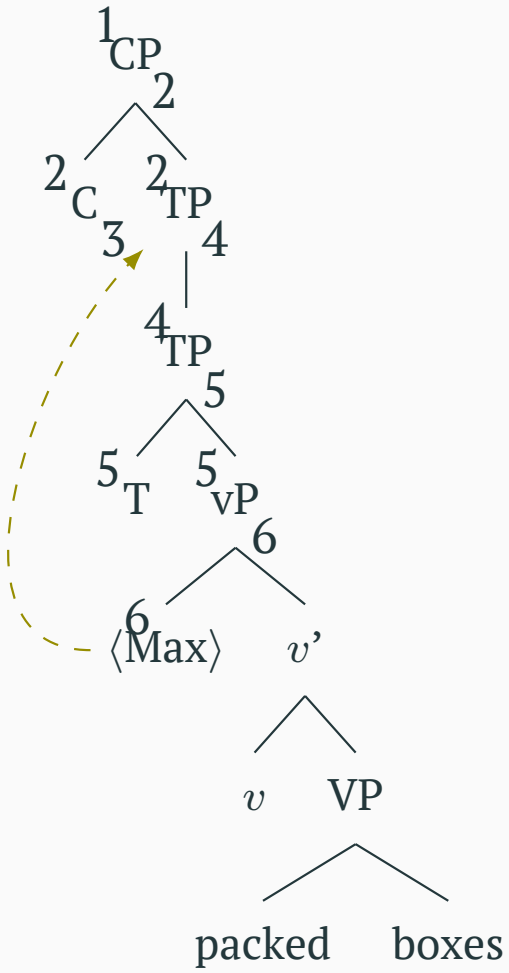
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and *v′*	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for *v*
Step 9	*v′* expands to *v* and VP	look for *v*
Step 10	*v* is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C • Max T *v* packed boxes.

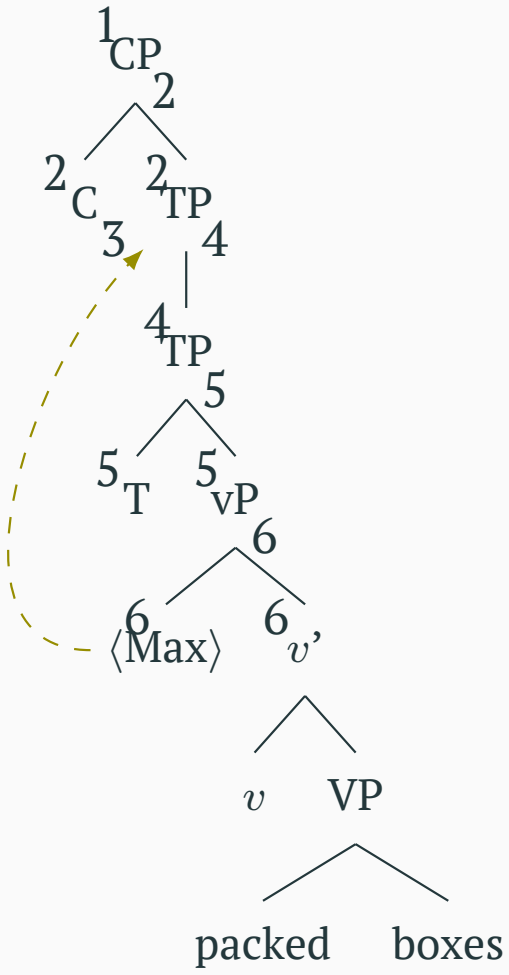
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and *v*′	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for *v*
Step 9	*v*′ expands to *v* and VP	look for *v*
Step 10	*v* is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C • Max T *v* packed boxes.

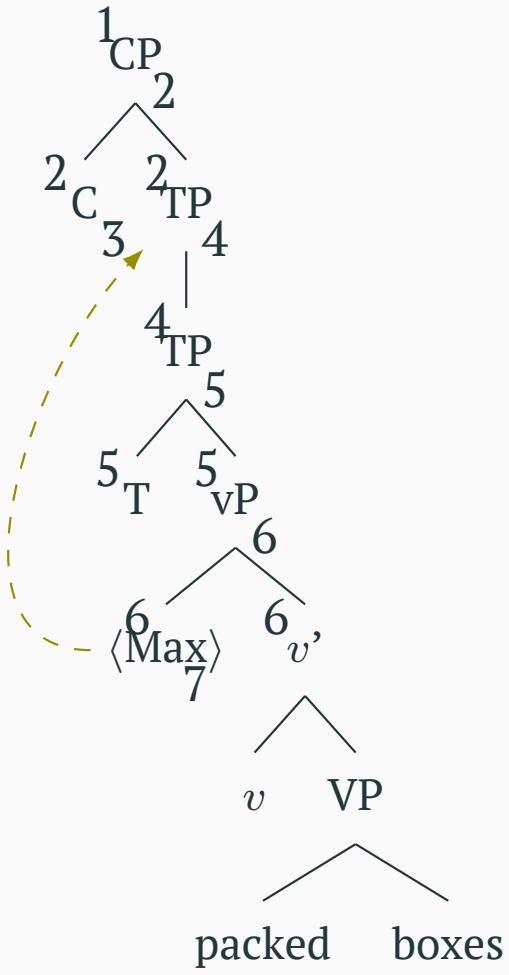
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and *v*′	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for *v*
Step 9	*v*′ expands to *v* and VP	look for *v*
Step 10	*v* is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C Max • T *v* packed boxes.

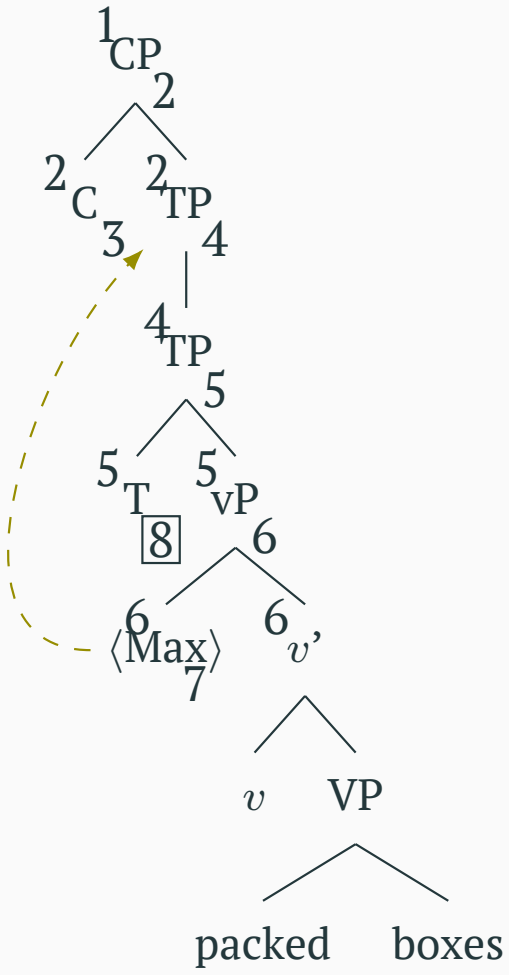
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and *v*′	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for *v*
Step 9	*v*′ expands to *v* and VP	look for *v*
Step 10	*v* is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C Max T • *v* packed boxes.

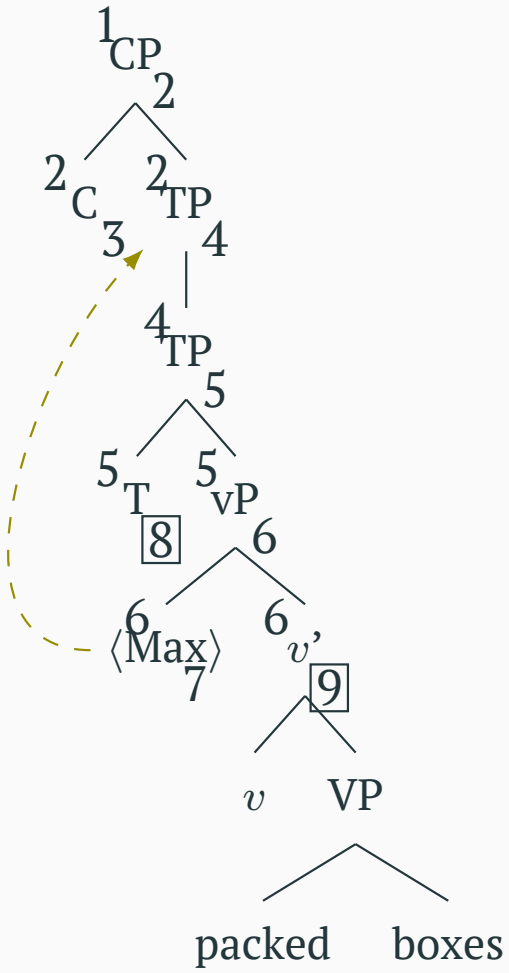
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and *v*′	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for *v*
Step 9	*v*′ expands to *v* and VP	look for *v*
Step 10	*v* is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C Max T • *v* packed boxes.

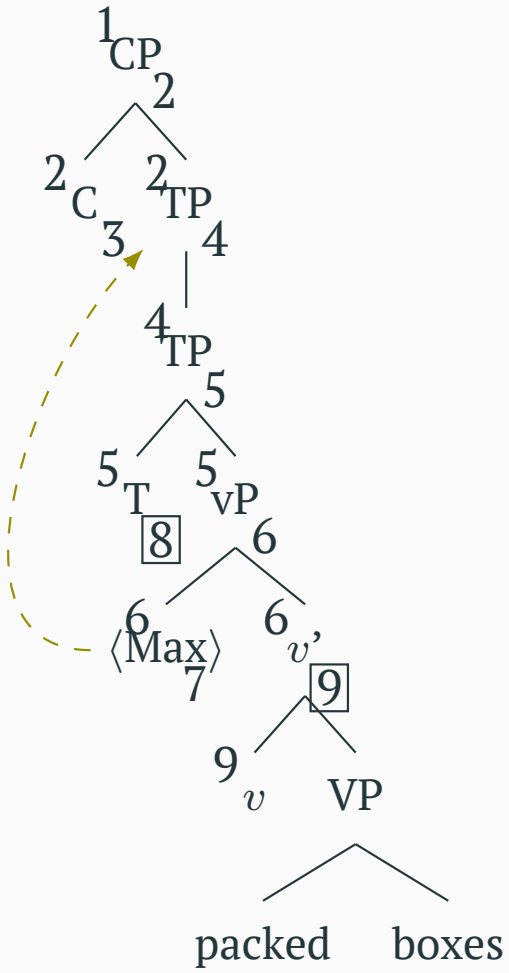
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and *v*′	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for *v*
Step 9	*v*′ expands to *v* and VP	look for *v*
Step 10	*v* is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C Max T • v packed boxes.

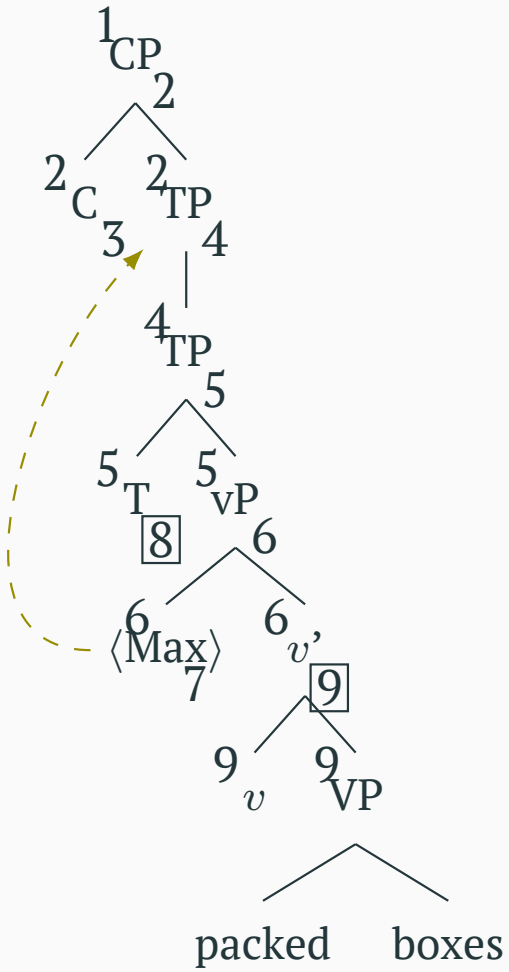
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' expands to v and VP	look for v
Step 10	v is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C Max T • v packed boxes.

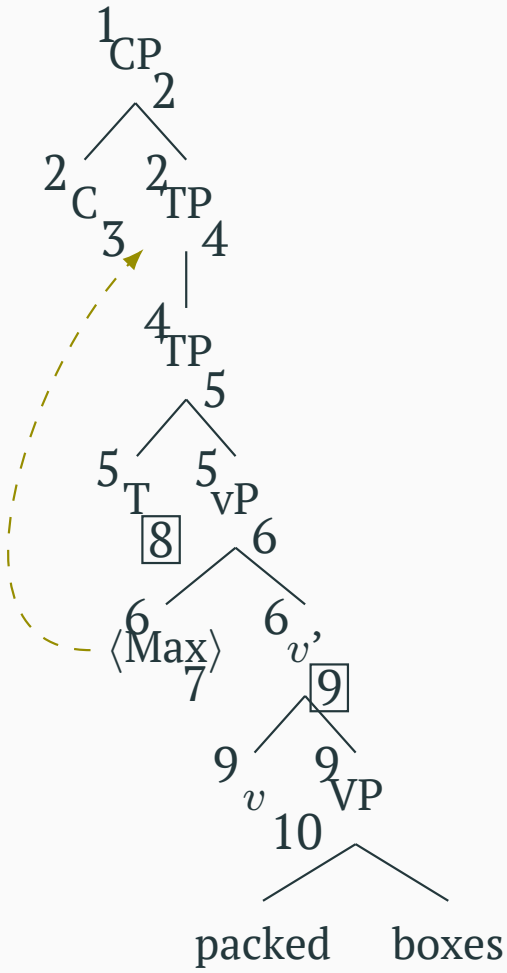
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' expands to v and VP	look for v
Step 10	v is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C Max T v • packed boxes.

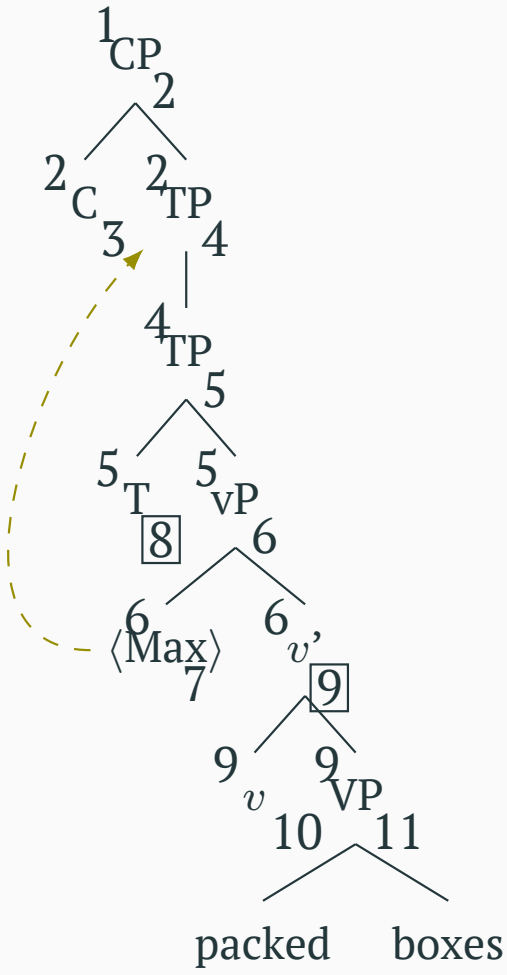
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' expands to v and VP	look for v
Step 10	v is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C Max T v • packed boxes.

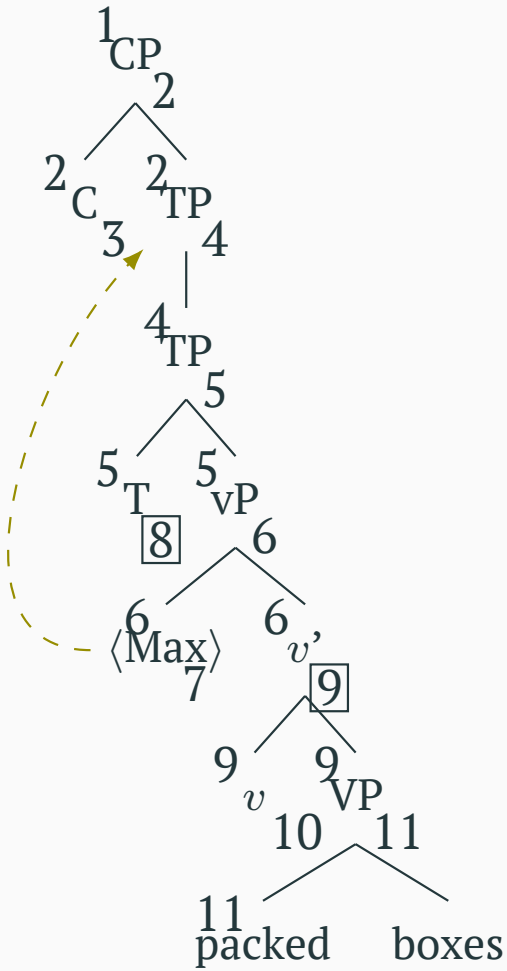
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' expands to v and VP	look for v
Step 10	v is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C Max T v • packed boxes.

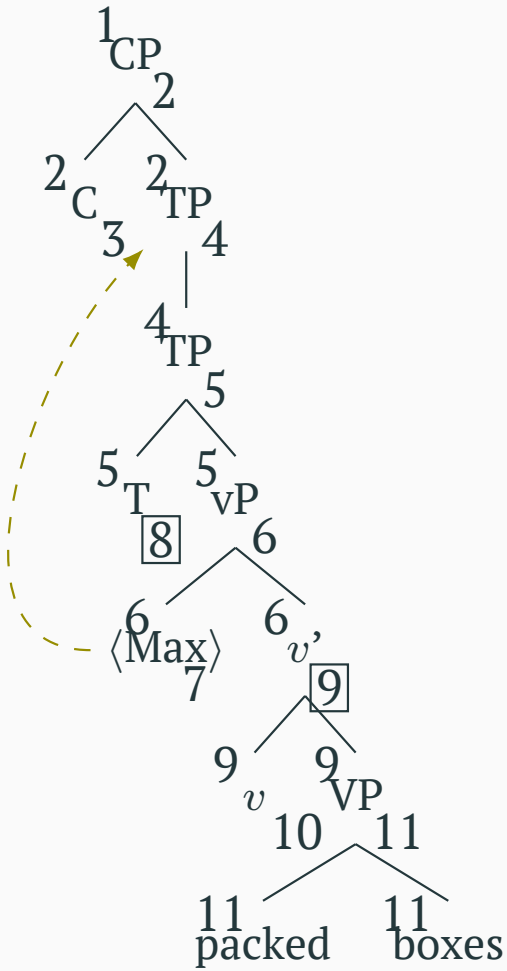
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' expands to v and VP	look for v
Step 10	v is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C Max T v • packed boxes.

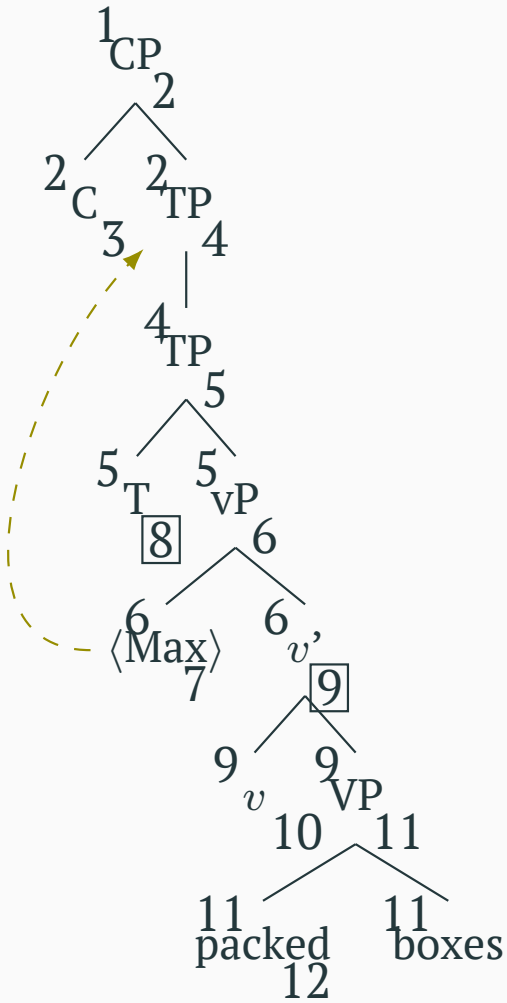
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' expands to v and VP	look for v
Step 10	v is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C Max T v packed • boxes.

Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' expands to v and VP	look for v
Step 10	v is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done



MG parser in action

(24) C Max T v packed boxes. $\ltimes$

Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' expands to v and VP	look for v
Step 10	v is found	look for packed
Step 11	VP expands to packed and boxes	look for packed
Step 12	packed is found	look for boxes
Step 13	boxes is found	done