

Phrasal Weight Effect on Word Order

ver. quick and dirty

Lei Liu

November 30, 2022

The presentation in bullet points:

The presentation in bullet points:

- Language users choose between alternative word orders based on **structural complexity**.

The presentation in bullet points:

- Language users choose between alternative word orders based on structural complexity.
 - Heavy NP shift order is preferred because it is structurally simple.

The presentation in bullet points:

- Language users choose between alternative word orders based on **structural complexity**.
 - **Heavy NP shift** order is preferred because it is structurally simple.
- **Minimalist parsing** offers a quantitative way to understand structural complexity.

1. Introduction: English heavy NP shift
2. Minimalist parsing
 - Minimalist Grammar and its parser
 - Modeling word order preferences - English HNPS
3. Conclusion
 - Where we are
 - Where to next

- English heavy NP shift

- English heavy NP shift

(1) a. Max put [*DP* boxes] [*PP* in his car].

- English heavy NP shift

- (1) a. Max put [*DP* boxes] [*PP* in his car].
b. ??Max put [*PP* in his car] [*DP* boxes].

- English heavy NP shift

- (1) a. Max put [*DP* boxes] [*PP* in his car].
- b. ??Max put [*PP* in his car] [*DP* boxes].
- c. **HNPS** Max put [*PP* in his car] [*DP* all the boxes of home furnishings].

- English heavy NP shift

- (1) a. Max put [*DP* boxes] [*PP* in his car].
- b. ??Max put [*PP* in his car] [*DP* boxes].
- c. **HNPS** Max put [*PP* in his car] [*DP* all the boxes of home furnishings].
- d. ?Max put [*DP* all the boxes of home furnishings] [*PP* in his car].

- English heavy NP shift

- (1) a. Max put [*DP* boxes] [*PP* in his car].
- b. ??Max put [*PP* in his car] [*DP* boxes].
- c. **HNPS** Max put [*PP* in his car] [*DP* all the boxes of home furnishings].
- d. ?Max put [*DP* all the boxes of home furnishings] [*PP* in his car].

(1c) is preferred because it is structurally simpler than (1d).

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

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

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

(Chomsky 2014).

- lexical items
 - feature bundles

- **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 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 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 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 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

(2) Max packed boxes.

- 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

(2) Max packed boxes.

Max :: D^- *nom*⁻
cat. mvmt

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

boxes :: D^-
cat.

- 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

(2) Max packed boxes.

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

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

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

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

boxes :: D^-
cat.

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

Minimalist parsing

Max :: $D^- \text{ nom}^-$
cat. mvmt

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

boxes :: D^-
cat.

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

T :: $v^+ \text{ nom}^+ T^-$
sel. mvmt cat.

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

Minimalist parsing

Max :: $D^- \text{ nom}^-$
cat. mvmt

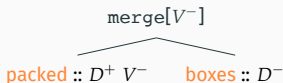
packed :: $D^+ V^-$
sel. cat.

boxes :: D^-
cat.

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

T :: $v^+ \text{ nom}^+ T^-$
sel. mvmt cat.

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



Minimalist parsing

Max :: $D^- \text{ nom}^-$
cat. mvmt

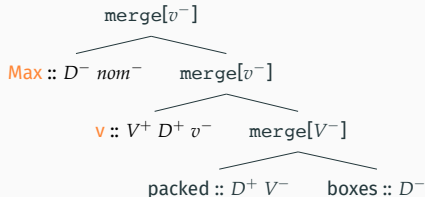
packed :: $D^+ V^-$
sel. cat.

boxes :: D^-
cat.

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

T :: $v^+ \text{ nom}^+ T^-$
sel. mvmt cat.

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



Minimalist parsing

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

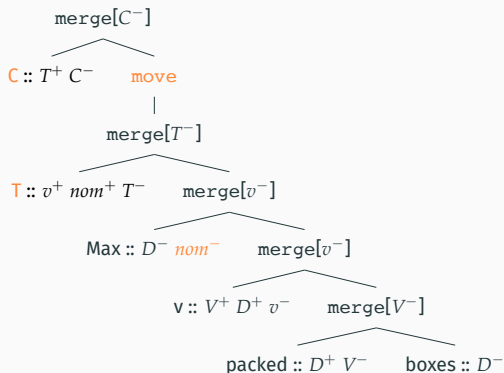
packed :: D^+ V^-
sel. cat.

boxes :: D^-
cat.

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

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

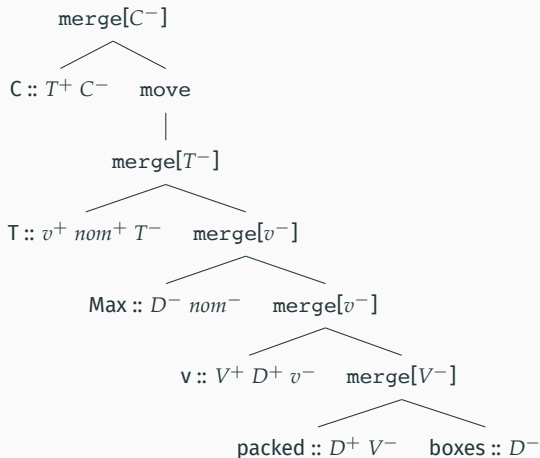
v :: V^+ D^+ v^-
sel. sel. cat.



(9) Max packed boxes.

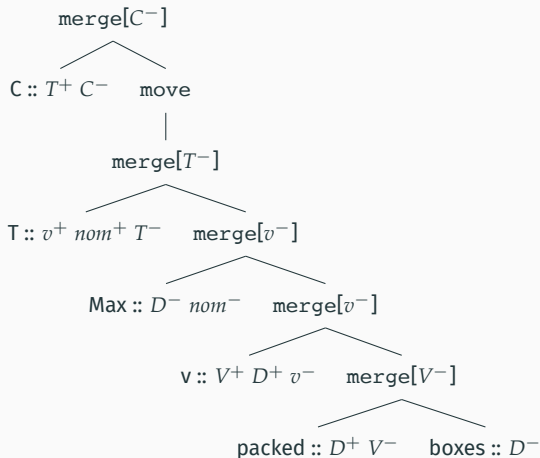
Minimalist parsing

(9) Max packed boxes.



Minimalist parsing

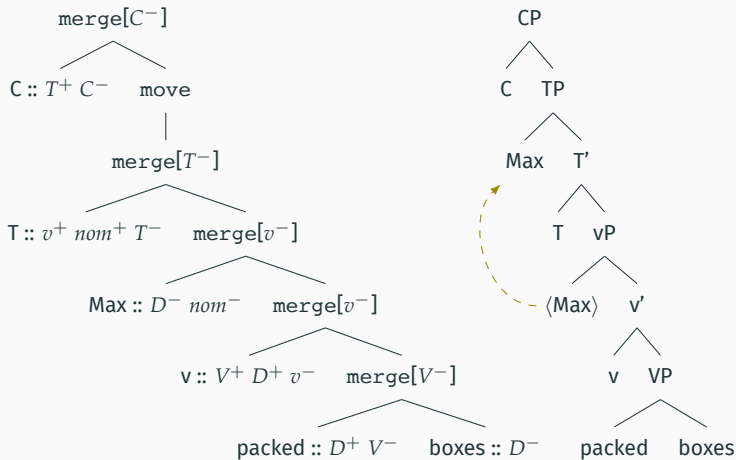
(9) Max packed boxes.



derivation tree

Minimalist parsing

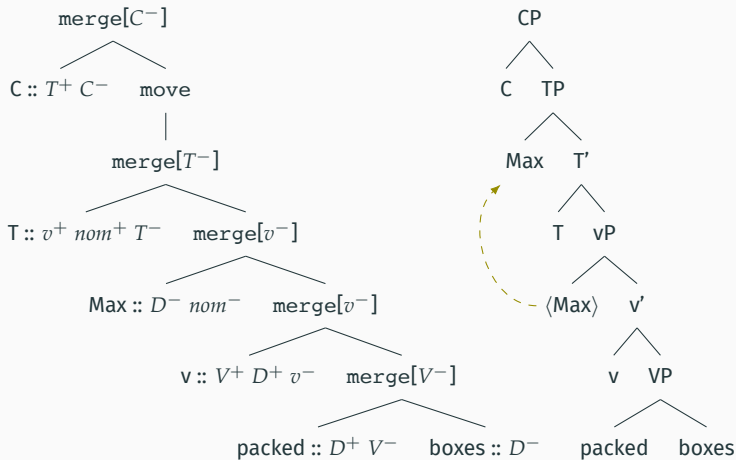
(9) Max packed boxes.



derivation tree

Minimalist parsing

(9) Max packed boxes.

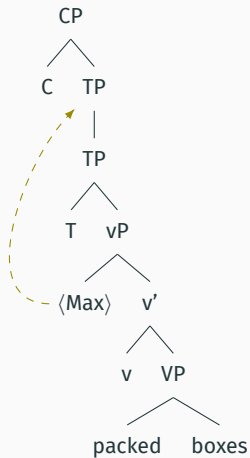


derivation tree

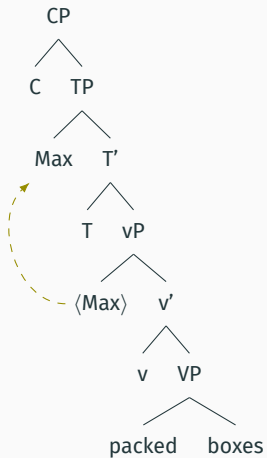
derived tree

Minimalist parsing

(9) Max packed boxes.



derivation tree



derived tree

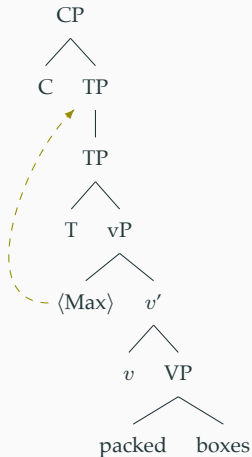
An MG parser (Stabler 2013, Graf et al. 2015a) is a **top-down** parsers that:

- takes as **input** a string with pronounced and unpronounced nodes and,
- **outputs** derivation trees

Minimalist parsing

(9) • C Max T v packed boxes.

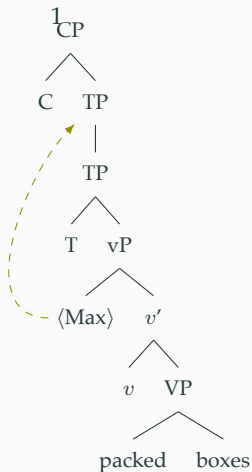
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) • C Max T v packed boxes.

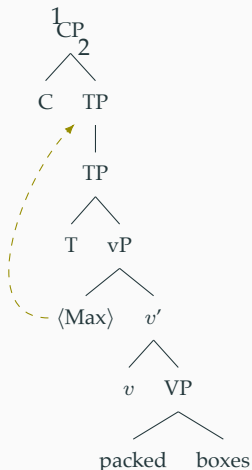
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) • C Max T v packed boxes.

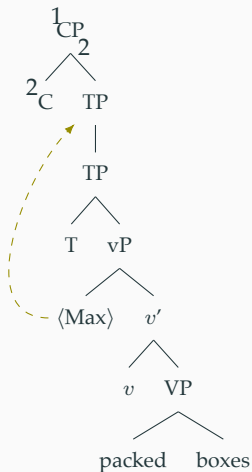
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) • C Max T v packed boxes.

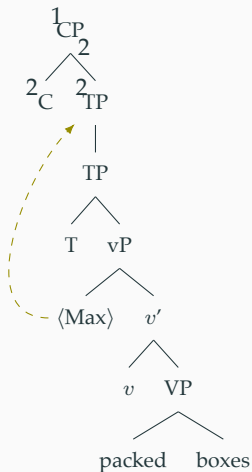
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) • C Max T v packed boxes.

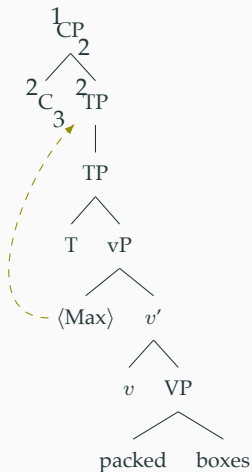
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) C • Max T v packed boxes.

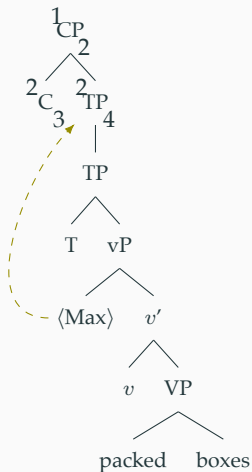
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) C • Max T v packed boxes.

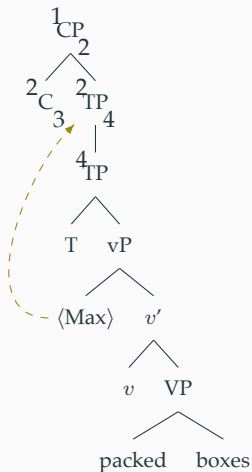
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) C • Max T v packed boxes.

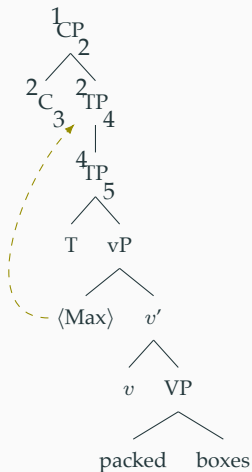
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) C • Max T v packed boxes.

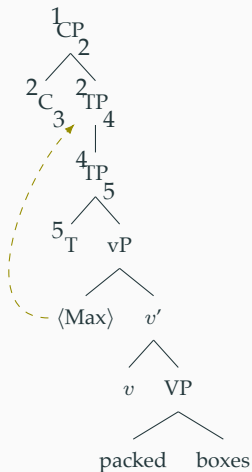
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) C • Max T v packed boxes.

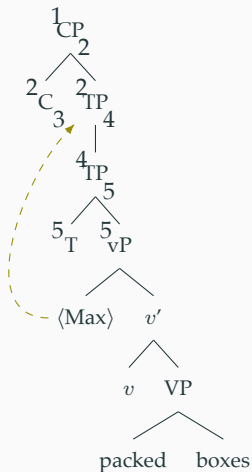
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) C • Max T v packed boxes.

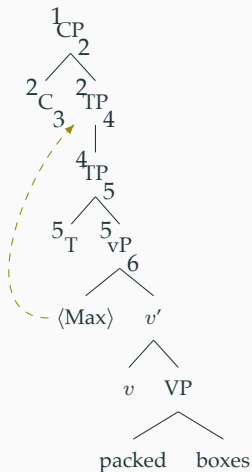
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) C • Max T v packed boxes.

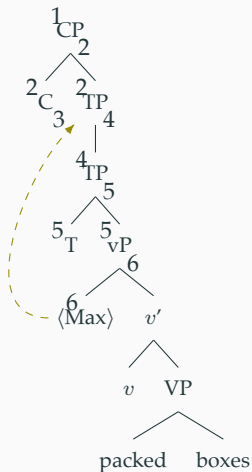
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) C • Max T v packed boxes.

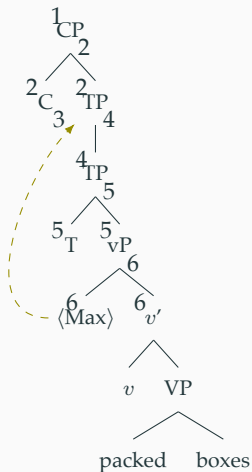
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) C • Max T v packed boxes.

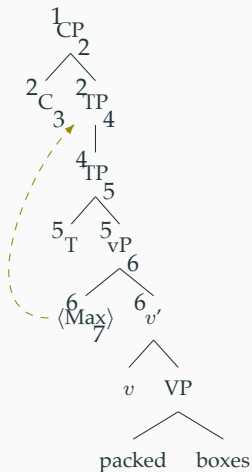
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) C Max • T *v* packed boxes.

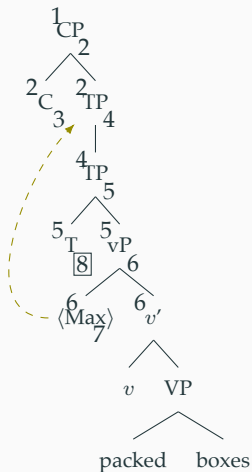
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for <i>v</i>
Step 9	<i>v</i> ' expands to <i>v</i> and VP	look for <i>v</i>
Step 10	<i>v</i> 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



Minimalist parsing

(9) C Max T • *v* packed boxes.

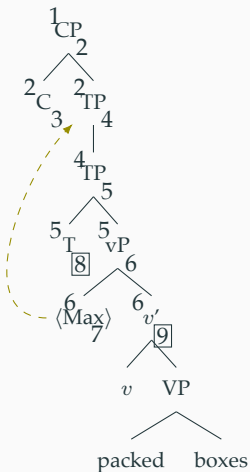
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to T	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) C Max T • v packed boxes.

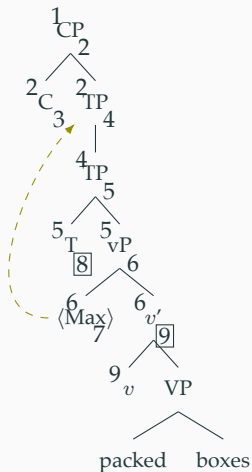
- | | | |
|---------|--------------------------------|-----------------|
| Step 1 | CP is conjectured | look for C |
| Step 2 | CP expands to C and TP | look for C |
| Step 3 | C is found | look for Max |
| Step 4 | TP expands to TP | look for Max |
| Step 5 | TP expands to T and vP | look for Max |
| 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 |



Minimalist parsing

(9) C Max T • *v* packed boxes.

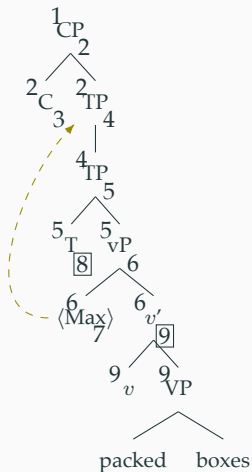
- | | | |
|---------|---------------------------------------|-------------------|
| Step 1 | CP is conjectured | look for C |
| Step 2 | CP expands to C and TP | look for C |
| Step 3 | C is found | look for Max |
| Step 4 | TP expands to TP | look for Max |
| Step 5 | TP expands to T and vP | look for 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 <i>v</i> |
| Step 9 | <i>v</i> ' expands to <i>v</i> and VP | look for <i>v</i> |
| Step 10 | <i>v</i> 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 |



Minimalist parsing

(9) C Max T • *v* packed boxes.

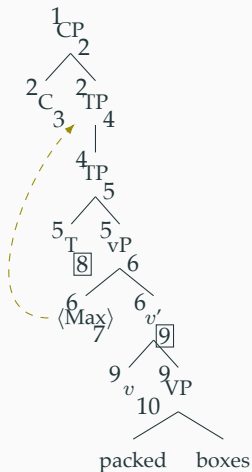
- | | | |
|---------|---------------------------------------|-------------------|
| Step 1 | CP is conjectured | look for C |
| Step 2 | CP expands to C and TP | look for C |
| Step 3 | C is found | look for Max |
| Step 4 | TP expands to TP | look for Max |
| Step 5 | TP expands to T and vP | look for 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 <i>v</i> |
| Step 9 | <i>v</i> ' expands to <i>v</i> and VP | look for <i>v</i> |
| Step 10 | <i>v</i> 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 |



Minimalist parsing

(9) C Max T v • packed boxes.

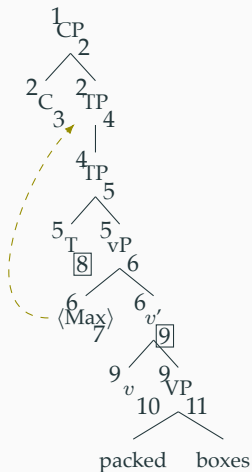
- | | | |
|---------|--------------------------------|-----------------|
| Step 1 | CP is conjectured | look for C |
| Step 2 | CP expands to C and TP | look for C |
| Step 3 | C is found | look for Max |
| Step 4 | TP expands to TP | look for Max |
| Step 5 | TP expands to T and vP | look for Max |
| 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 |



Minimalist parsing

(9) C Max T v • packed boxes.

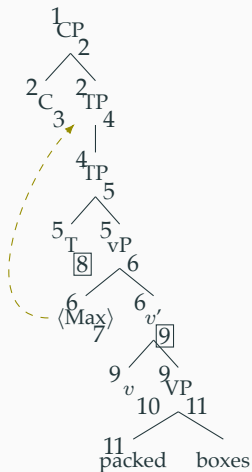
- | | | |
|---------|--------------------------------|-----------------|
| Step 1 | CP is conjectured | look for C |
| Step 2 | CP expands to C and TP | look for C |
| Step 3 | C is found | look for Max |
| Step 4 | TP expands to TP | look for Max |
| Step 5 | TP expands to T and vP | look for Max |
| 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 |



Minimalist parsing

(9) C Max T v • packed boxes.

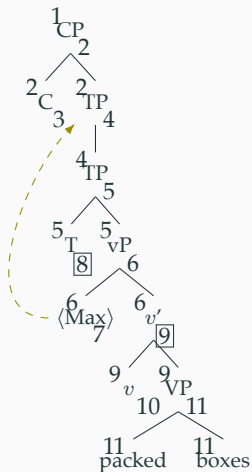
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) C Max T v • packed boxes.

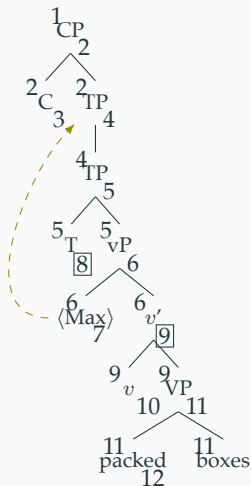
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

(9) C Max T v packed • boxes.

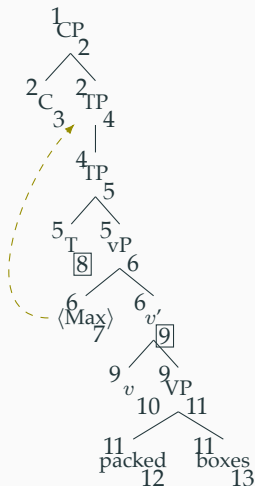
Step 1	CP is conjectured	look for C
Step 2	CP expands to C and TP	look for C
Step 3	C is found	look for Max
Step 4	TP expands to TP	look for Max
Step 5	TP expands to T and vP	look for Max
Step 6	vP expands to Max and v'	look for Max
Step 7	Max is found	look for T
Step 8	T is found	look for v
Step 9	v' 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



Minimalist parsing

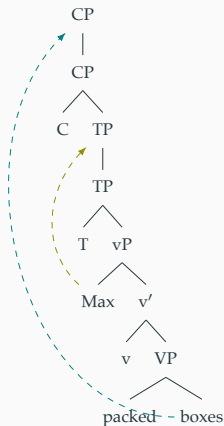
(9) C Max T *v* packed boxes. ✕

- | | | |
|---------|---------------------------------------|-------------------|
| 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 T | 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 <i>v</i> |
| Step 9 | <i>v</i> ' expands to <i>v</i> and VP | look for <i>v</i> |
| Step 10 | <i>v</i> 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 |



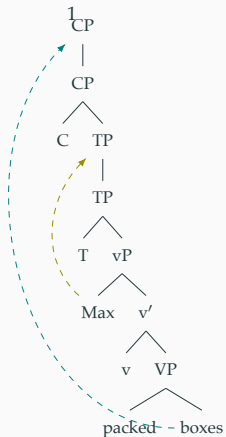
Minimalist parsing

- (10) • Boxes, C Max T *v* packed.



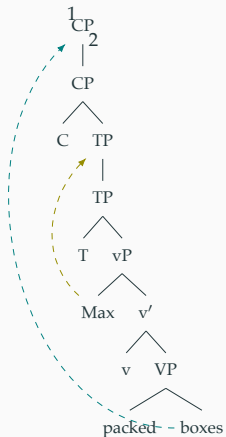
Minimalist parsing

- (10) • Boxes, C Max T *v* packed.



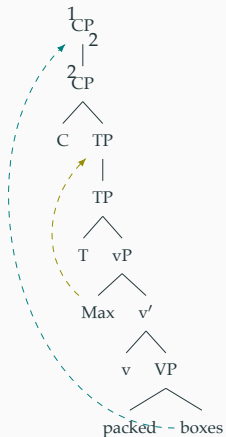
Minimalist parsing

- (10) • Boxes, C Max T *v* packed.



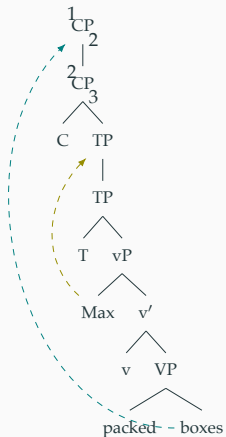
Minimalist parsing

- (10) • Boxes, C Max T *v* packed.



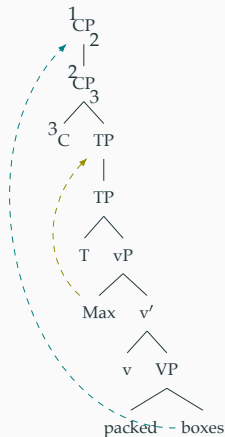
Minimalist parsing

- (10) • Boxes, C Max T *v* packed.



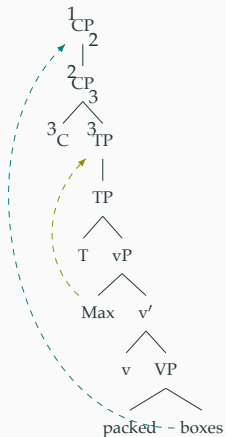
Minimalist parsing

- (10) • Boxes, C Max T v packed.



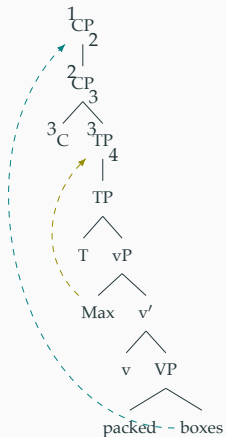
Minimalist parsing

- (10) • Boxes, C Max T v packed.



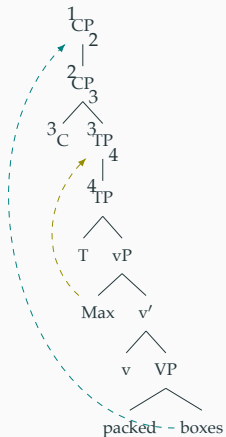
Minimalist parsing

- (10) • Boxes, C Max T v packed.



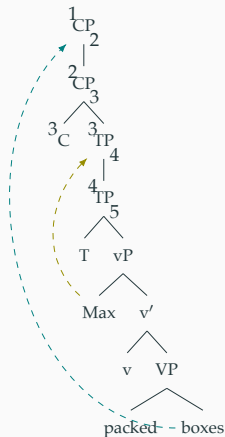
Minimalist parsing

- (10) • Boxes, C Max T v packed.



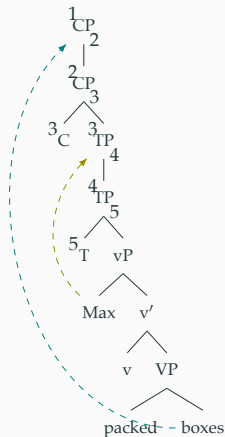
Minimalist parsing

(10) • Boxes, C Max T *v* packed.



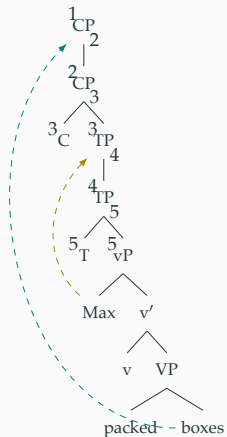
Minimalist parsing

- (10) • Boxes, C Max T *v* packed.



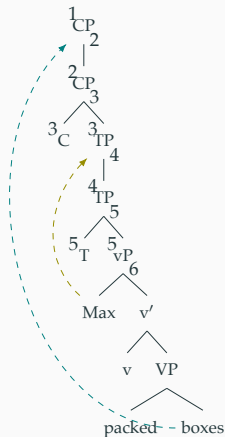
Minimalist parsing

- (10) • Boxes, C Max T *v* packed.



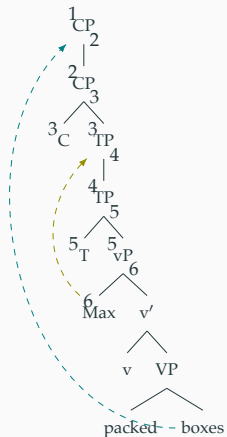
Minimalist parsing

(10) • Boxes, C Max T *v* packed.



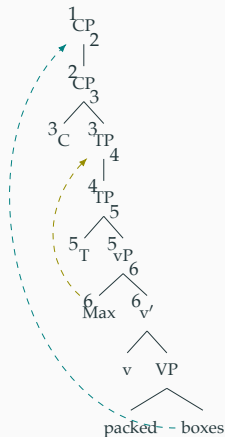
Minimalist parsing

- (10) • Boxes, C Max T *v* packed.



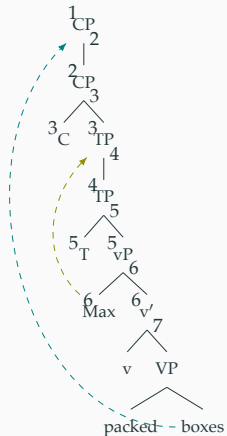
Minimalist parsing

(10) • Boxes, C Max T *v* packed.



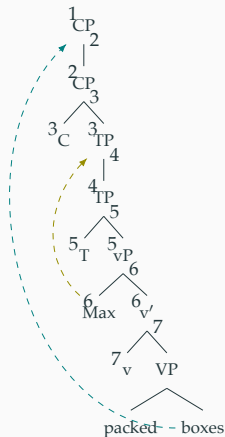
Minimalist parsing

(10) • Boxes, C Max T *v* packed.



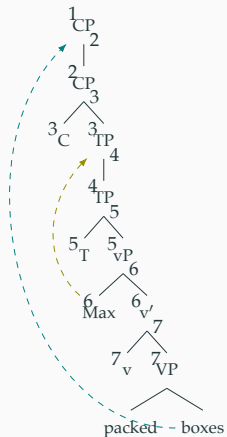
Minimalist parsing

(10) • Boxes, C Max T *v* packed.



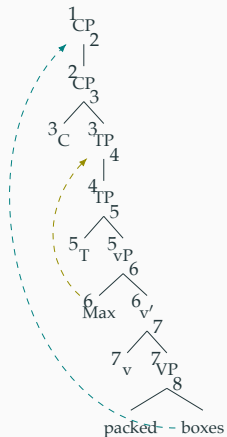
Minimalist parsing

(10) • Boxes, C Max T *v* packed.



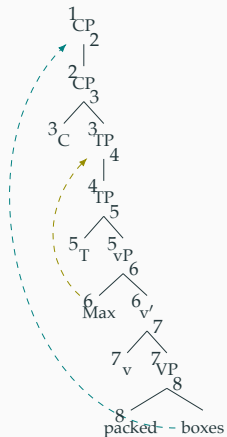
Minimalist parsing

(10) • Boxes, C Max T *v* packed.



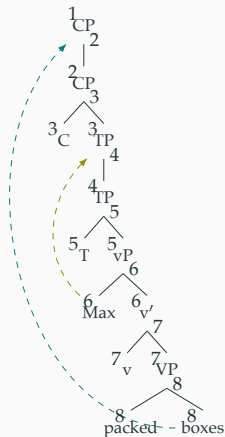
Minimalist parsing

(10) • Boxes, C Max T *v* packed.



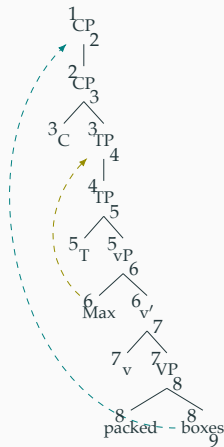
Minimalist parsing

(10) • Boxes, C Max T *v* packed.



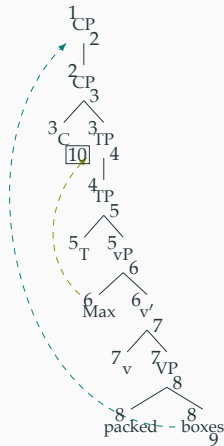
Minimalist parsing

(10) Boxes, • C Max T v packed.



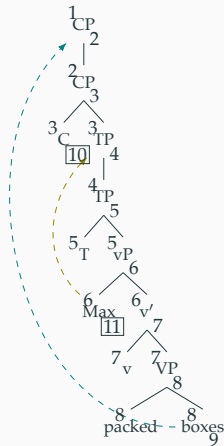
Minimalist parsing

(10) Boxes, C • Max T *v* packed.



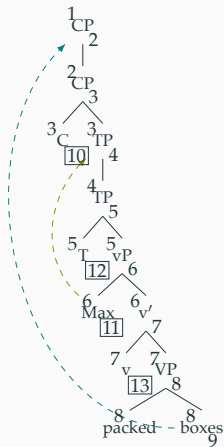
Minimalist parsing

(10) Boxes, C Max • T *v* packed.



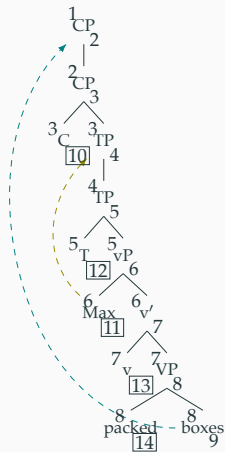
Minimalist parsing

(10) Boxes, C Max T *v* • packed.

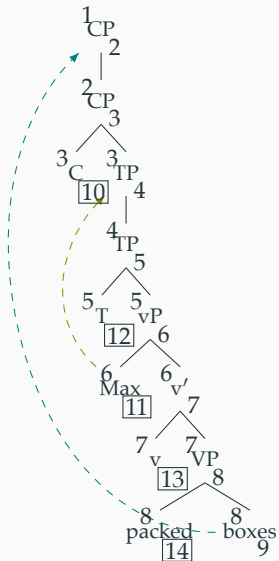
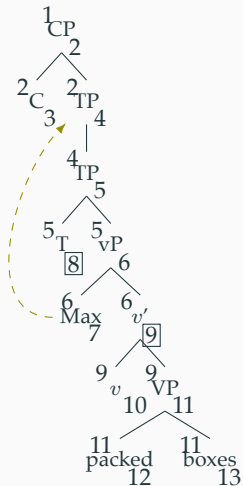


Minimalist parsing

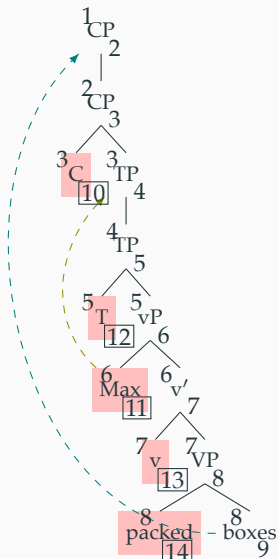
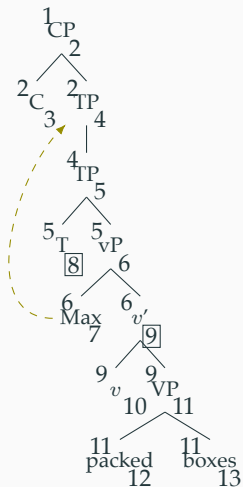
(10) Boxes, C Max T *v* packed.×



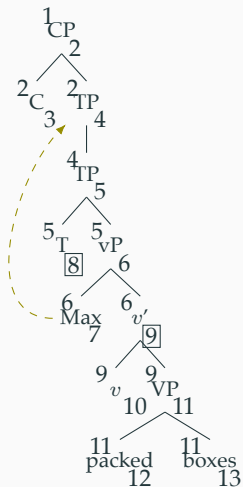
Minimalist parsing



Minimalist parsing

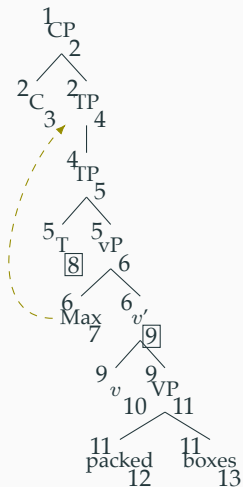


Minimalist parsing



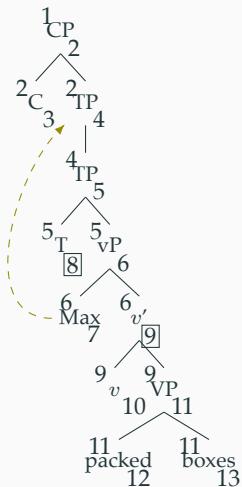
- An annotated tree is a record of the parser's behavior

Minimalist parsing



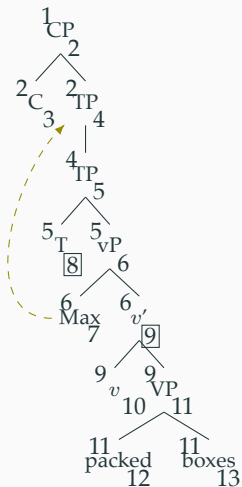
- An annotated tree is a record of the parser's behavior
- 1CP_2

Minimalist parsing



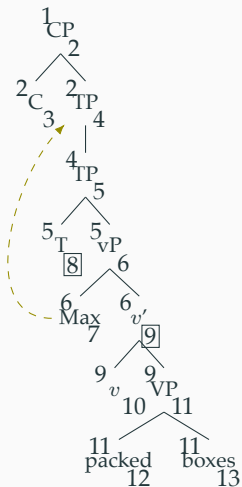
- An annotated tree is a record of the parser's behavior
- ${}^1\text{CP}_2$
 - **index**: when the parser puts the node **in** the memory

Minimalist parsing



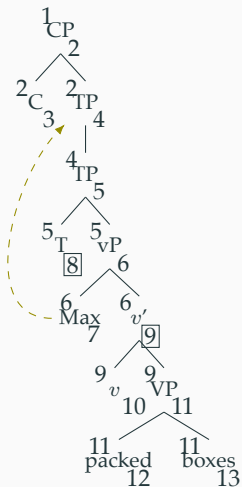
- An annotated tree is a record of the parser's behavior
- 1CP_2
 - **index**: when the parser puts the node **in** the memory
 - **outdex**: when the parser throws the node **out** of the memory

Minimalist parsing



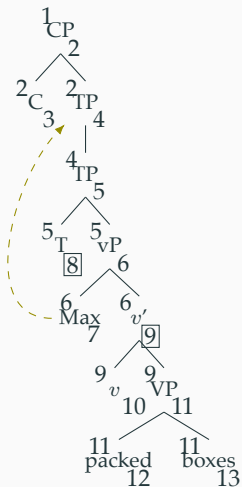
- An annotated tree is a record of the parser's behavior
- $^1_{CP}_2$
 - **index**: when the parser puts the node **in** the memory
 - **outdex**: when the parser throws the node **out** of the memory
- **Memory usage** (Kobele et al. 2013, Graf et al. 2015b)

Minimalist parsing



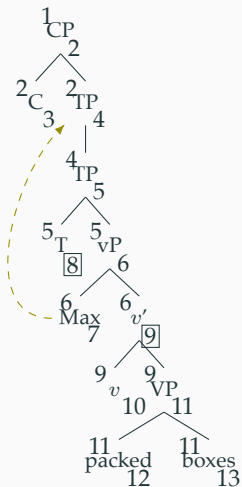
- An annotated tree is a record of the parser's behavior
- $^1_{CP}_2$
 - **index**: when the parser puts the node **in** the memory
 - **outdex**: when the parser throws the node **out** of the memory
- **Memory usage** (Kobele et al. 2013, Graf et al. 2015b)
 - **Tenure**: how long a parse item is held in memory

Minimalist parsing



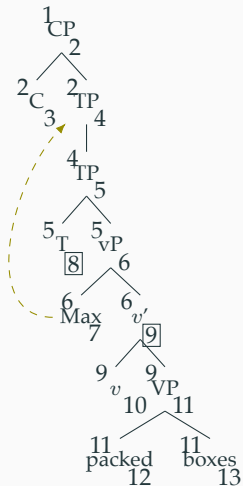
- An annotated tree is a record of the parser's behavior
- ${}^1\text{CP}_2$
 - **index**: when the parser puts the node **in** the memory
 - **outdex**: when the parser throws the node **out** of the memory
- **Memory usage** (Kobele et al. 2013, Graf et al. 2015b)
 - **Tenure**: how long a parse item is held in memory
 - **Payload**: how many items are held in memory throughout a parse

Minimalist parsing



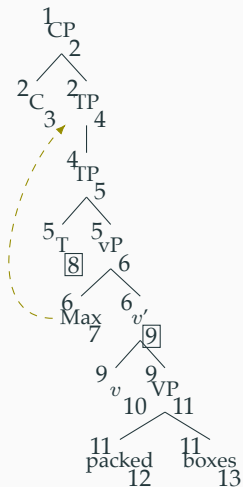
- An annotated tree is a record of the parser's behavior
- ${}^1\text{CP}_2$
 - **index**: when the parser puts the node **in** the memory
 - **outdex**: when the parser throws the node **out** of the memory
- **Memory usage** (Kobele et al. 2013, Graf et al. 2015b)
 - **Tenure**: how long a parse item is held in memory
 - **Payload**: how many items are held in memory throughout a parse
 - **Size**: the length of movement dependencies

Minimalist parsing



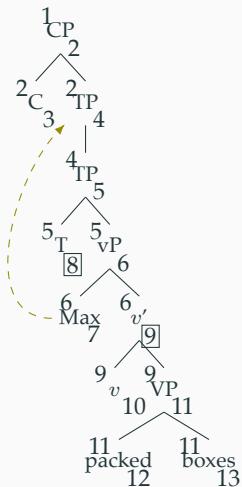
- An annotated tree is a record of the parser's behavior

Minimalist parsing



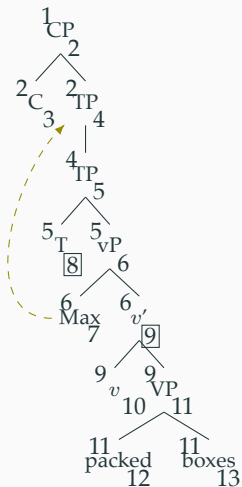
- An annotated tree is a record of the parser's behavior
- Memory usage formalized: complexity metrics (Kobele et al. 2013, Graf et al. 2015b)

Minimalist parsing



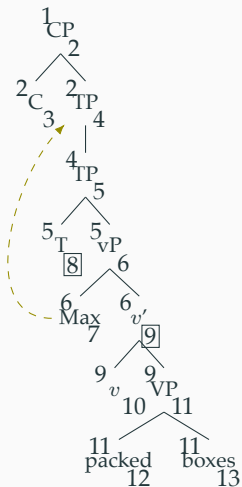
- An annotated tree is a record of the parser's behavior
- Memory usage formalized: complexity metrics (Kobele et al. 2013, Graf et al. 2015b)
- **MaxT** := $\max(\text{tenure-of}(n) | n \in T)$

Minimalist parsing



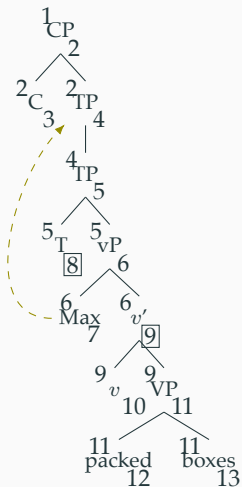
- An annotated tree is a record of the parser's behavior
- Memory usage formalized: complexity metrics (Kobele et al. 2013, Graf et al. 2015b)
- **MaxT** := $\max(\text{tenure-of}(n) | n \in T)$

Minimalist parsing



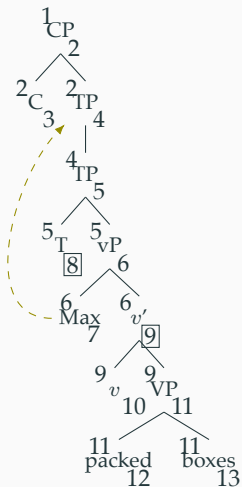
- An annotated tree is a record of the parser's behavior
- Memory usage formalized: complexity metrics (Kobele et al. 2013, Graf et al. 2015b)
- **MaxT** := $\max(\text{tenure-of}(n) | n \in T) = 3$

Minimalist parsing



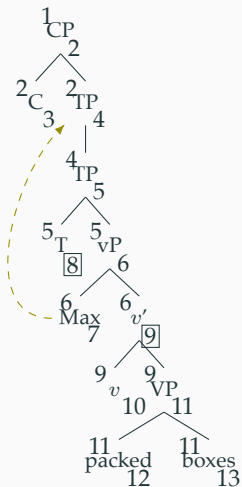
- An annotated tree is a record of the parser's behavior
- Memory usage formalized: complexity metrics (Kobelev et al. 2013, Graf et al. 2015b)
- **MaxT** := $\max(\text{tenure-of}(n) | n \in T) = 3$
- **SumT** := $\sum_{n \in T} \text{tenure-of}(n)$

Minimalist parsing



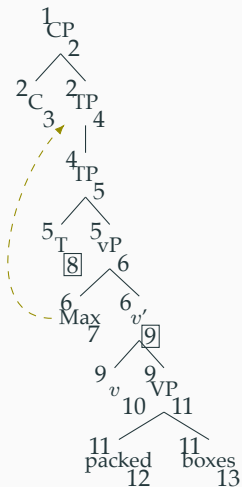
- An annotated tree is a record of the parser's behavior
- Memory usage formalized: complexity metrics (Kobelev et al. 2013, Graf et al. 2015b)
- **MaxT** := $\max(\text{tenure-of}(n) | n \in T) = 3$
- **SumT** := $\sum_{n \in T} \text{tenure-of}(n)$

Minimalist parsing



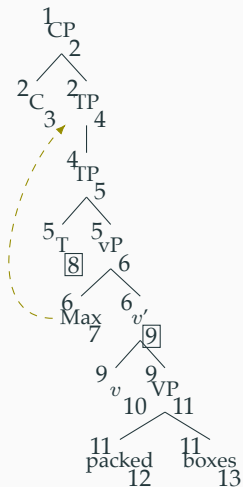
- An annotated tree is a record of the parser's behavior
- Memory usage formalized: complexity metrics (Kobele et al. 2013, Graf et al. 2015b)
- **MaxT** := $\max(\text{tenure-of}(n) | n \in T) = 3$
- **SumT** := $\sum_{n \in T} \text{tenure-of}(n) = 6$

Minimalist parsing

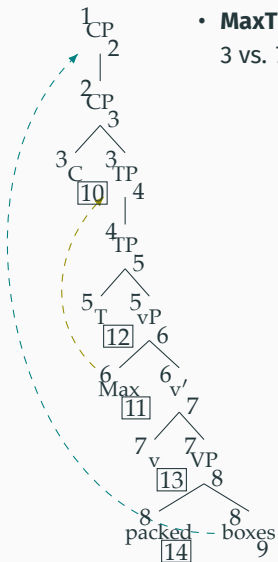


- An annotated tree is a record of the parser's behavior
- Memory usage formalized: complexity metrics (Kobelev et al. 2013, Graf et al. 2015b)
- **MaxT** := $\max(\text{tenure-of}(n) | n \in T) = 3$
- **SumT** := $\sum_{n \in T} \text{tenure-of}(n) = 6$

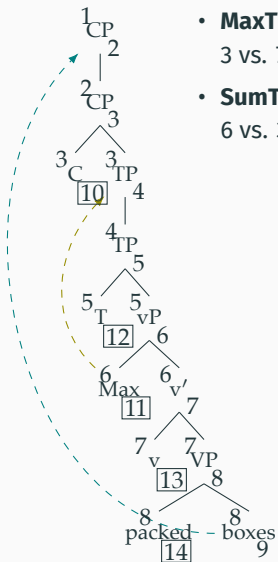
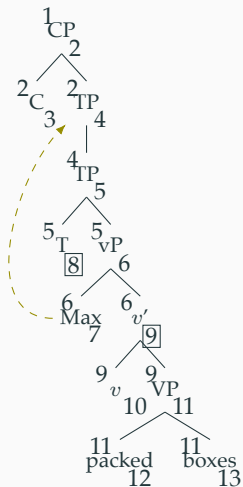
Minimalist parsing



- **MaxT:**
3 vs. 7



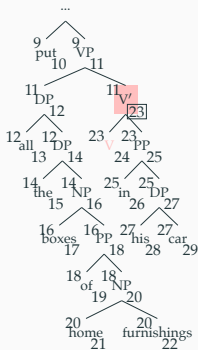
Minimalist parsing



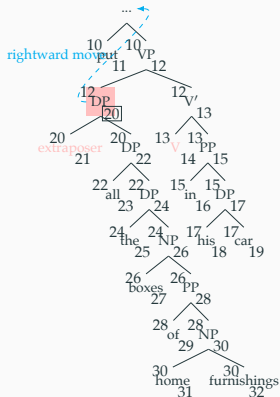
- **MaxT:**
3 vs. 7
- **SumT:**
6 vs. 31

Modeling word order preferences - English HNPS

- (16) put [DP ...boxes...] [PP in...]
canonical order



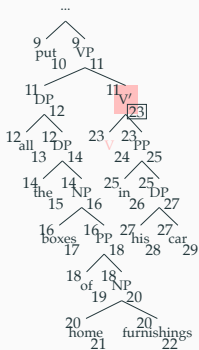
- (17) put [PP in...] [DP ...boxes...]
HNPS order



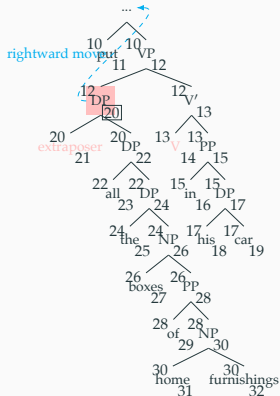
MaxT: 12/v' (canonical)

Modeling word order preferences - English HNPS

- (16) put [DP ...boxes...] [PP in...]
canonical order



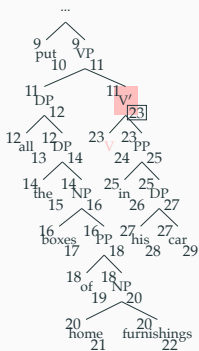
- (17) put [PP in...] [DP ...boxes...]
HNPS order



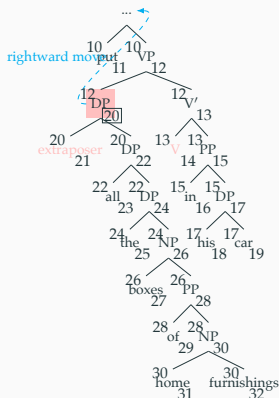
MaxT: 12/v' (canonical) > 8/DP (HNPS)

Modeling word order preferences - English HNPS

- (16) put [DP ...boxes...] [PP in...]
canonical order



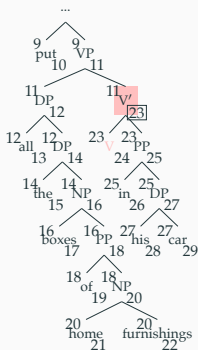
- (17) put [PP in...] [DP ...boxes...]
HNPS order



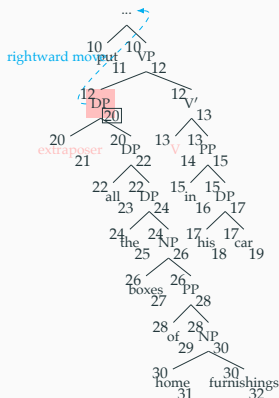
MaxT: 12/v' (canonical) > 8/DP (HNPS) ✓

Modeling word order preferences - English HNPS

- (16) put [DP ...boxes...] [PP in...]
canonical order



- (17) put [PP in...] [DP ...boxes...]
HNPS order

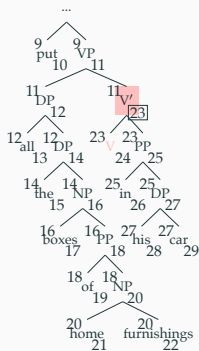


MaxT: 12/v' (canonical) > 8/DP (HNPS) ✓

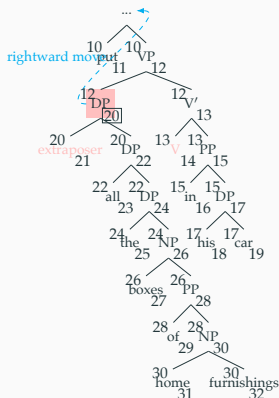
SumT: 18 (canonical)

Modeling word order preferences - English HNPS

- (16) put [DP ...boxes...] [PP in...]
canonical order



- (17) put [PP in...] [DP ...boxes...]
HNPS order

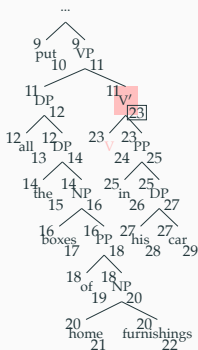


MaxT: 12/v' (canonical) > 8/DP (HNPS) ✓

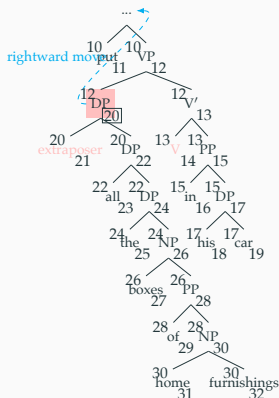
SumT: 18 (canonical) > 15 (HNPS)

Modeling word order preferences - English HNPS

- (16) put [DP ...boxes...] [PP in...]
canonical order



- (17) put [PP in...] [DP ...boxes...]
HNPS order

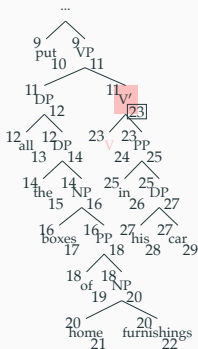


MaxT: 12/v' (canonical) > 8/DP (HNPS) ✓

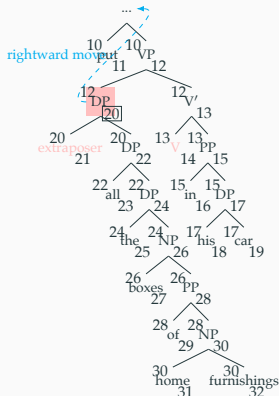
SumT: 18 (canonical) > 15 (HNPS) ✓

Modeling word order preferences - English HNPS

- (16) put [DP ...boxes...] [PP in...]
canonical order



- (17) put [PP in...] [DP ...boxes...]
HNPS order



MaxT: 12/v' (canonical) > 8/DP (HNPS) ✓

SumT: 18 (canonical) > 15 (HNPS) ✓

HNPS advantage!

- Structural simplicity accounts for word order preferences in:
 - English heavy NP shift (HNPS)
 - (18) Max put [*PP* in his car] [*DP* all the boxes of home furnishings].
 - (19) Max put [*DP* all the boxes of home furnishings] [*PP* in his car].
 - English particle verb construction (PV)
 - (20) Chris **put on** [*DP* a hat which Alex made with merino wool].
 - (21) Chris **put** [*DP* a hat which Alex made with merino wool] **on**.
 - Mandarin *ba* construction
 - (22) Zhangsan ba [*DP* daocha] fang (le) [*PP* zai yong tanmu
Z. *ba* knife.fork put (le) on use sandalwood
zhizuo de zhuozi-shang]
make de table-top
 - (23) Zhangsan [*PP* zai yong tanmu zhizuo de zhuozi-shang]
Z. on use sandalwood make de table-top
fang (le) [*DP* daocha]
put (le) knife.fork
'Z put a set of knife and fork on the table that is made of
sandalwood.'

Complexity metrics correctly predict processing preferences cross-linguistically!

- Across Many Constructions

Complexity metrics correctly predict processing preferences cross-linguistically!

- Across Many Constructions
 - right vs. center embedding (Kobele et al. 2013)

Complexity metrics correctly predict processing preferences cross-linguistically!

- Across Many Constructions
 - right vs. center embedding (Kobele et al. 2013)
 - crossing vs. nested dependencies (Kobele et al. 2013)

Complexity metrics correctly predict processing preferences cross-linguistically!

- Across Many 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)

Complexity metrics correctly predict processing preferences cross-linguistically!

- Across Many 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)

Complexity metrics correctly predict processing preferences cross-linguistically!

- Across Many 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)

Complexity metrics correctly predict processing preferences cross-linguistically!

- Across Many 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

Complexity metrics correctly predict processing preferences cross-linguistically!

- Across Many 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...

- Structure evaluation (17)

- Structure evaluation (17)
- Processing difficulties at intermediate steps

Where to next

- Structure evaluation (17)
- Processing difficulties at intermediate steps
- I'm all ears!

References i

- Chomsky, N. (2014). *The minimalist program*. MIT press.
- 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.
- 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.
- 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*.
- 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.
- Zhang, C. (2017). *Stacked Relatives: Their Structure, Processing and Computation*. PhD thesis, State University of New York at Stony Brook.