

Adjunction in (T)SL Syntax

Kenneth Hanson

Stony Brook University & Yale University

SCiL 2025

July 18–20, 2025

Properties of adjunction

Properties of adjunction

1. **Optionality:** an adjunct may be freely added or removed
 - ▶ a (big) house

Properties of adjunction

1. **Optionality:** an adjunct may be freely added or removed
 - ▶ a (big) house
2. **Iteration:** if one adjunct may be added, then any number may be added
 - ▶ a big house
 - ▶ a big blue house
 - ▶ a big big big blue house

Properties of adjunction

1. **Optionality:** an adjunct may be freely added or removed
 - ▶ a (big) house
2. **Iteration:** if one adjunct may be added, then any number may be added
 - ▶ a big house
 - ▶ a big blue house
 - ▶ a big big big blue house
3. **Ordering restrictions:** non-canonical orders are often degraded
 - ▶ ✓ a big blue wooden house
 - ▶ ?? a blue big house
 - ▶ ?? a wooden big house
 - ▶ ?? a wooden blue house

Properties of adjunction

1. **Optionality:** an adjunct may be freely added or removed
 - ▶ a (big) house
2. **Iteration:** if one adjunct may be added, then any number may be added
 - ▶ a big house
 - ▶ a big blue house
 - ▶ a big big big blue house
3. **Ordering restrictions:** non-canonical orders are often degraded
 - ▶ ✓ a big blue wooden house
 - ▶ ?? a blue big house
 - ▶ ?? a wooden big house
 - ▶ ?? a wooden blue house
4. **Invisibility to selection:** the phrase inherits all properties from the head
 - ▶ **this** [big blue wooden **house**]
 - ▶ **these** [big blue wooden **houses**]

Overview

- **Adjunction** is intuitively a **local** operation, but its **computational complexity** depends on:
 - ▶ the geometry of the syntactic representation
 - ▶ the model of adjunction assumed

Overview

- **Adjunction** is intuitively a **local** operation, but its **computational complexity** depends on:
 - ▶ the geometry of the syntactic representation
 - ▶ the model of adjunction assumed
- **This work:** I propose a model of adjunction which is **strictly local (SL)** over Minimalist Grammar (MG) dependency trees

Overview

- **Adjunction** is intuitively a **local** operation, but its **computational complexity** depends on:
 - ▶ the geometry of the syntactic representation
 - ▶ the model of adjunction assumed
- **This work:** I propose a model of adjunction which is **strictly local (SL)** over Minimalist Grammar (MG) dependency trees
- **This covers** our **four core properties:**
 - 1) optionality, 2) iteration, 3) ordering restrictions, 4) invisibility to selection

Overview

- **Adjunction** is intuitively a **local** operation, but its **computational complexity** depends on:
 - ▶ the geometry of the syntactic representation
 - ▶ the model of adjunction assumed
- **This work:** I propose a model of adjunction which is **strictly local (SL)** over Minimalist Grammar (MG) dependency trees
- **This covers** our **four core properties**:
1) optionality, 2) iteration, 3) ordering restrictions, 4) invisibility to selection
- **Strict locality is preserved** in cases of **recursive adjunction**

Overview

- **Adjunction** is intuitively a **local** operation, but its **computational complexity** depends on:
 - ▶ the geometry of the syntactic representation
 - ▶ the model of adjunction assumed
- **This work:** I propose a model of adjunction which is **strictly local (SL)** over Minimalist Grammar (MG) dependency trees
- **This covers** our **four core properties**:
1) optionality, 2) iteration, 3) ordering restrictions, 4) invisibility to selection
- **Strict locality is preserved** in cases of **recursive adjunction**
- **Bonus:** An interesting treatment of some other properties of adjunction

Roadmap

1. Background

- ▶ Computational complexity
- ▶ MG dependency trees

2. Strictly local adjunction

- ▶ Capturing the core properties
- ▶ Handling recursive adjunction
- ▶ Adjunction + selection in one SL grammar

3. Predictions, extensions

Background

Computational complexity

Modeled using MG dependency trees, syntactic dependencies are **subregular**, and **close to the bottom** of the complexity hierarchy.

$$\underbrace{\text{FIN} \subset \text{SL} \subset \text{TSL} \subset \dots}_{\text{"subregular"}} \subset \text{REG} \subset \dots \subset \text{CFL} \subset \dots$$

Computational complexity

Modeled using MG dependency trees, syntactic dependencies are **subregular**, and **close to the bottom** of the complexity hierarchy.

$$\underbrace{\text{FIN} \subset \text{SL} \subset \text{TSL} \subset \dots}_{\text{"subregular"}} \subset \text{REG} \subset \dots \subset \text{CFL} \subset \dots$$

SL	{	Selection	(Graf 2018)
		Functional hierarchies	(Hanson 2024)
		Adjunction	(Hanson 2024, this work)
TSL	{	Movement	(Graf 2022)
		Case	(Hanson 2025a)
		Agreement	(Hanson 2025b)

Minimalist Grammars

Minimalist Grammars (MGs, Stabler 1997, 2011):

- Operations include category selection, movement, and possibly others
- Lexical item is a pair (phonological exponent, string of features/diacritics)
- F^+ = selects category F, F^- = has category F

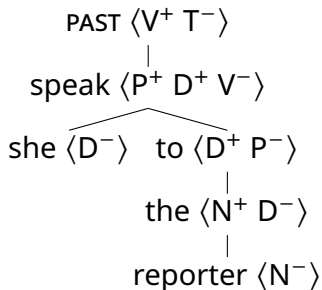
Example MG lexical items

she $\langle D^- \rangle$	- no arguments, makes a DP
reporter $\langle N^- \rangle$	- no arguments, makes an NP
the $\langle N^+ D^- \rangle$	- selects an NP, makes a DP
to $\langle D^+ P^- \rangle$	- selects an NP, makes a PP
speak $\langle P^+ D^+ V^- \rangle$	- selects a PP, then a DP, makes a VP

MG dependency trees

Every node is a **lexical item**. The daughters of a node are its **arguments** in reverse order of selection (Graf and Kostyszyn 2021).

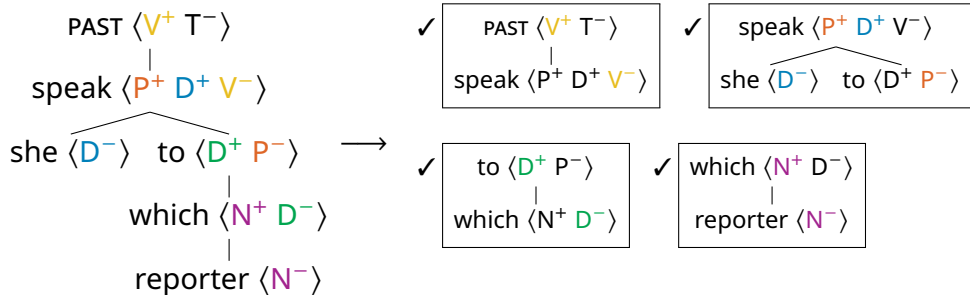
She spoke to the reporter



Also see Kobele (2012).

Selection

Selection is **strictly local (SL)** – just need to look at a node and its daughters.
(Graf 2018)



F^+ = selects category F F^- = has category F

Strictly local adjunction

Adding in adjunction

We introduce a new feature type F^{\approx} for adjuncts (Frey and Gärtner 2002).

- Adjectives are not A^- but N^{\approx}

Adding in adjunction

We introduce a new feature type F^{\approx} for adjuncts (Frey and Gärtner 2002).

- Adjectives are not A^- but N^{\approx}

Adjuncts are daughters of their head in the dependency tree.
(Not the other way around!)

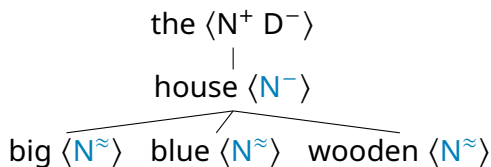
Adding in adjunction

We introduce a new feature type F^{\approx} for adjuncts (Frey and Gärtner 2002).

- Adjectives are not A^- but N^{\approx}

Adjuncts are daughters of their head in the dependency tree.
(Not the other way around!)

the big blue wooden house



The daughter string language

The set of all possible daughter strings involving adjunction is *not finite*, but it is **SL with a window of 2 (SL-2)**.

The daughter string language

The set of all possible daughter strings involving adjunction is *not finite*, but it is **SL with a window of 2 (SL-2)**.

Ex. Licit daughter strings for category N

- As regular expression: $N^{\approx*}$ where N^{\approx} stands for any node bearing N $^{\approx}$
- The string language:
 $\{\epsilon, N^{\approx}, N^{\approx} N^{\approx}, N^{\approx} N^{\approx} N^{\approx}, \dots\}$
- As SL-2 grammar:
 $G^+ = \{\times\times, \times N^{\approx}, N^{\approx} N^{\approx}, N^{\approx} \times\}$ where \times/\times = beginning/end of string

The daughter string language

The set of all possible daughter strings involving adjunction is *not finite*, but it is **SL with a window of 2 (SL-2)**.

Ex. Licit daughter strings for category N

- As regular expression: $N^{\approx*}$ where N^{\approx} stands for any node bearing N $^{\approx}$
- The string language:
 $\{\epsilon, N^{\approx}, N^{\approx} N^{\approx}, N^{\approx} N^{\approx} N^{\approx}, \dots\}$
- As SL-2 grammar:
 $G^+ = \{\times\times, \times N^{\approx}, N^{\approx} N^{\approx}, N^{\approx} \times\}$ where \times/\times = beginning/end of string

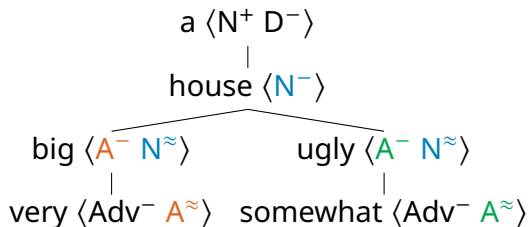
This covers 1) optionality, 2) iteration, and 3) invisibility to selection.

Recursive adjunction

Adjuncts can themselves host adjuncts.

To allow this, they should have a category feature (cf. Graf 2018).

a very big somewhat ugly house

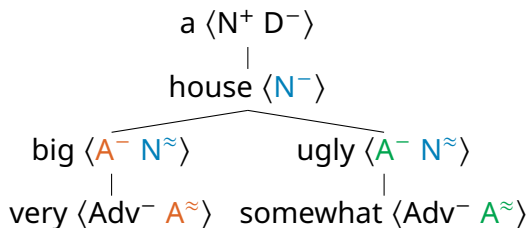


Recursive adjunction

Adjuncts can themselves host adjuncts.

To allow this, they should have a category feature (cf. Graf 2018).

a very big somewhat ugly house

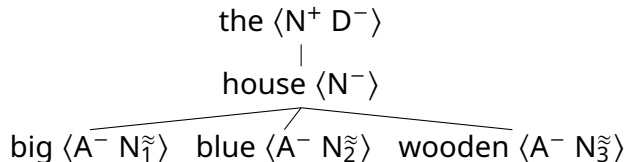


Strict locality is maintained!

See appendix for why locality is lost with traditional derivation trees.

Ordering restrictions

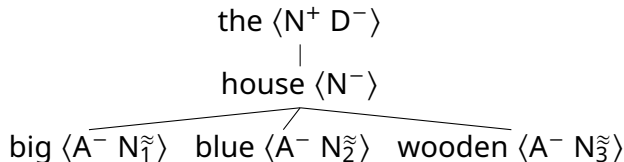
To get ordering restrictions without hacks, we need a [pair of features](#).
One encodes [the position in the ordering hierarchy](#) (Fowlie 2013; Hanson 2024).



Alternative: divide A into S(ize), C(olor), M(aterial), etc., and leave N^{\approx} alone.

Ordering restrictions

To get ordering restrictions without hacks, we need a **pair of features**.
One encodes **the position in the ordering hierarchy** (Fowlie 2013; Hanson 2024).



Language/grammar for adjective order

Language:

$$N_1^{\approx*} \cdot N_2^{\approx*} \cdot N_3^{\approx*}$$

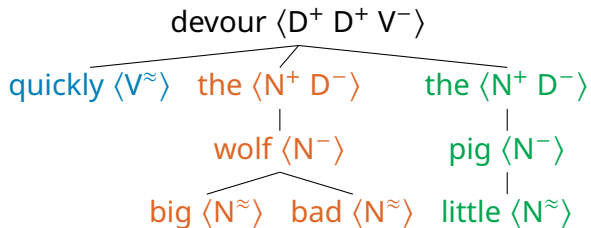
$$G^+ = \left\{ \begin{array}{l} \times \times, \times N_1^{\approx}, \times N_2^{\approx}, \times N_3^{\approx}, \\ N_1^{\approx} N_2^{\approx}, N_2^{\approx} N_3^{\approx}, N_1^{\approx} N_3^{\approx}, \\ N_1^{\approx} N_1^{\approx}, N_2^{\approx} N_2^{\approx}, N_3^{\approx} N_3^{\approx}, \\ N_1^{\approx} \times, N_2^{\approx} \times, N_3^{\approx} \times \end{array} \right\}$$

Alternative: divide A into S(ize), C(olor), M(aterial), etc., and leave N^{\approx} alone.

Adjunction plus selection

What if we have both arguments and adjuncts?

ex. the big bad wolf quickly devoured the little pig



Adjunction plus selection (2)

Just one SL daughter string language is enough.

- The adjuncts go first: Adj1, Adj2, Adj3, ..., Spec, Comp.
- Caveat: we might need a bigger window to accommodate selection.

A length 2 substring should be padded to produce every length 3 substring which contains it.

Adjunction plus selection (2)

Just one SL daughter string language is enough.

- The adjuncts go first: Adj1, Adj2, Adj3, ..., Spec, Comp.
- Caveat: we might need a bigger window to accommodate selection.

Ex. Grammar for *devour* (SL-3)

Language:

$$Adv_1^* \cdot Adv_2^* \cdot Adv_2^* \cdot D \cdot D$$

$$G^+ = \left\{ \begin{array}{l} \times \times, \times Adv_1, \times Adv_2, \times Adv_3, \\ Adv_1 Adv_2, Adv_2 Adv_3, Adv_1 Adv_3, \\ Adv_1 Adv_1, Adv_2 Adv_2, Adv_3 Adv_3, \\ Adv_1 D, Adv_2 D, Adv_3 D, \\ \times \times D, \times DD, DD \times, D \times \times, \end{array} \right\}$$

A length 2 substring should be padded to produce every length 3 substring which contains it.

Predictions, extensions

Typological predictions

Previous work in subregular syntax shows how differences in the SL/TSL grammar derive linguistic variation while ruling out unattested patterns.

Typological predictions

Previous work in subregular syntax shows how differences in the SL/TSL grammar derive linguistic variation while ruling out unattested patterns.

Some examples:

- Single/multiple *wh*-movement (Graf and Kostyszyn 2021)
- Islands, extraction morphology (Graf 2022)
- Case sensitivity, upward agreement (Hanson 2025b)
- Case alignment, differential argument marking (Hanson 2025a)

Typological predictions

Previous work in subregular syntax shows how differences in the SL/TSL grammar derive linguistic variation while ruling out unattested patterns.

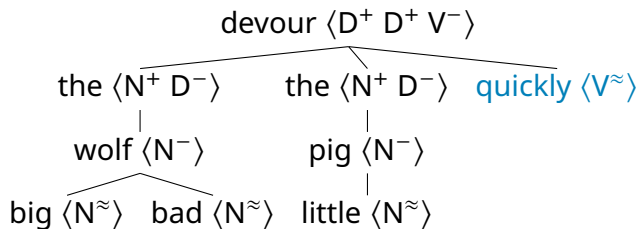
Some examples:

- Single/multiple *wh*-movement (Graf and Kostyszyn 2021)
- Islands, extraction morphology (Graf 2022)
- Case sensitivity, upward agreement (Hanson 2025b)
- Case alignment, differential argument marking (Hanson 2025a)

We might predict other kinds of adjunction configurations to be possible as long as the combined selection + adjunction grammar is SL.

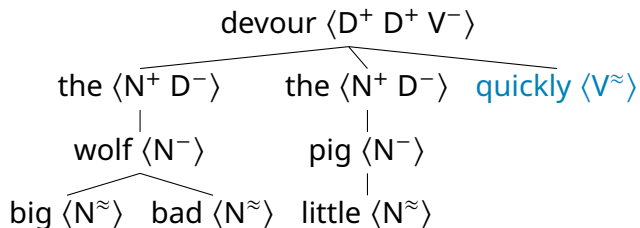
Low adjunction

Frey and Gärtner (2002) assume low adjunction of manner adverbs in German: they attach to the verb below the object.



Low adjunction

Frey and Gärtner (2002) assume low adjunction of manner adverbs in German: they attach to the verb below the object.



No problem! Put the manner adjuncts at the end.

- $D \cdot D \cdot Adv_M^*$

Left and right adjuncts

Right adjuncts seem to be c-commanded by the object (cf. Ernst 1994).

- ✓ John saw [no one] [anywhere].
- ✗ John saw [anyone] [nowhere].

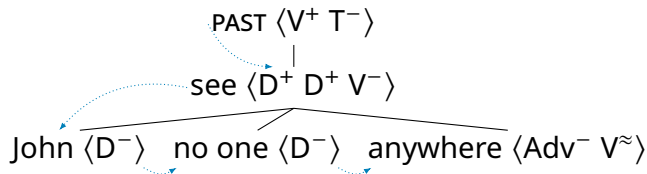
Left and right adjuncts

Right adjuncts seem to be c-commanded by the object (cf. Ernst 1994).

- ✓ John saw [no one] [anywhere].
- ✗ John saw [anyone] [nowhere].

Solution: right adjuncts go at the end (left adjuncts are unchanged).

- $Adv_1^* \cdot Adv_2^* \cdot Adv_3^* \cdot D \cdot D \cdot Adv_R^*$



Impossible adjunction patterns

If adjunction + selection is SL, these patterns should be impossible:

- Ordered APs and unordered PPs, interleaved

$$P^* \cdot A_1^* \cdot P^* \cdot A_2^* \cdot P^* \cdot A_3^* \cdot P^*$$

- Arguments and unordered adjuncts, interleaved

$$P^* \cdot D \cdot P^* \cdot D \cdot P^*$$

- Left/right adjuncts duplicate the same order

$$Adv_1^* \cdot Adv_2^* \cdot Adv_3^* \cdot D \cdot D \cdot Adv_1^* \cdot Adv_2^* \cdot Adv_3^*$$

If they do exist, the complexity of the daughter string language increases to M[ulti]TSL (De Santo and Graf 2019).

Conclusion

Conclusion

- We have achieved an SL model of adjunction which:
 - ▶ Encapsulates our four core properties
 - ▶ Handles recursive adjunction

Conclusion

- We have achieved an SL model of adjunction which:
 - ▶ Encapsulates our four core properties
 - ▶ Handles recursive adjunction
- Adjunction is unified with selection (and functional hierarchies)

Conclusion

- We have achieved an SL model of adjunction which:
 - ▶ Encapsulates our four core properties
 - ▶ Handles recursive adjunction
- Adjunction is unified with selection (and functional hierarchies)
- The complexity distinction between local (SL) and non-local (TSL) dependencies is upheld

Conclusion

- We have achieved an SL model of adjunction which:
 - ▶ Encapsulates our four core properties
 - ▶ Handles recursive adjunction
- Adjunction is unified with selection (and functional hierarchies)
- The complexity distinction between local (SL) and non-local (TSL) dependencies is upheld
- Puzzles like the left/right adjunct distinction can potentially be explained

Thank you!

References

- De Santo, Aniello and Thomas Graf (2019). Structure Sensitive Tier Projection: Applications and Formal Properties. *Formal Grammar*. Ed. by Raffaella Bernardi, Gregory Koble, and Sylvain Pogodalla. Berlin, Heidelberg: Springer, pp. 35–50. DOI: 10.1007/978-3-662-59648-7_3.
- Ernst, Thomas (1994). M-Command and Precedence. *Linguistic Inquiry* 25.2, pp. 327–335.
- Fowlie, Meaghan (2013). Order and Optionality: Minimalist Grammars with Adjunction. *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*. Ed. by András Kornai and Marco Kuhlmann. Association for Computational Linguistics, pp. 12–20. URL: <https://aclanthology.org/W13-3002>.
- Frey, Werner and Hans-Martin Gärtner (2002). On the treatment of scrambling and adjunction in minimalist grammars. *Proceedings of Formal Grammar 2002*. Ed. by G. Jäger et al., pp. 41–52.
- Graf, Thomas (2018). Why movement comes for free once you have adjunction. *Proceedings of CLS 53*. Chicago, IL: The Chicago Linguistic Society, pp. 117–136. URL: <https://thomasgraf.net/output/graf18cls.html>.

References (2)

- Graf, Thomas (2022). Typological implications of tier-based strictly local movement. *Proceedings of the Society for Computation in Linguistics (SCiL) 2022*. Amherst, MA: University of Massachusetts Amherst, pp. 184–193. DOI: 10.7275/gb65-ht31.
- Graf, Thomas and Kalina Kostyszyn (2021). Multiple Wh-Movement is not Special: The Subregular Complexity of Persistent Features in Minimalist Grammars. *Proceedings of the Society for Computation in Linguistics (SCiL) 2021*. Ed. by Allyson Ettinger, Ellie Pavlick, and Brandon Prickett. Amherst, MA: University of Massachusetts Amherst, pp. 275–285. DOI: 10.7275/2dfv-v219.
- Graf, Thomas and Nazila Shafiei (2019). C-command dependencies as TSL string constraints. *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*. Ed. by Gaja Jarosz et al. Amherst, MA: University of Massachusetts Amherst, pp. 205–215. DOI: 10.7275/4rrx-x488.
- Hanson, Kenneth (2024). Strict Locality in Syntax. *Proceedings of CLS 59*. Ed. by Kutay Serova and M. K. Snigaroff. Chicago, IL: The Chicago Linguistic Society, pp. 131–145.

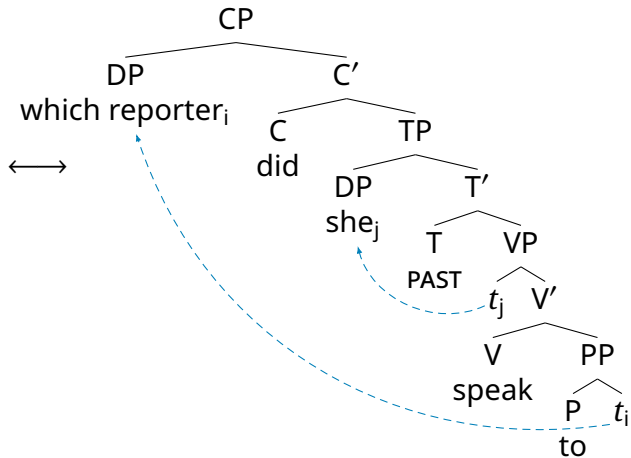
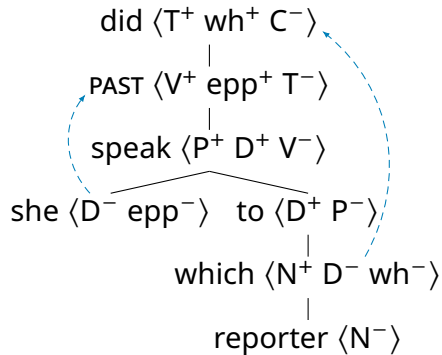
References (3)

- Hanson, Kenneth (2025a). The Typology of TSL Case Assignment. *Proceedings of the 18th Meeting on the Mathematics of Language*. To appear.
- Hanson, Kenneth (2025b). Tier-based strict locality and the typology of agreement. *Journal of Language Modelling* 13.1, pp. 43–97. DOI: 10.15398/jlm.v13i1.411.
- Kobele, Gregory M. (2012). Eliding the derivation: A minimalist formalization of ellipsis. *Proceedings of the International Conference on Head-Driven Phrase Structure Grammar*. CSLI Publications, Oct. DOI: 10.21248/hpsg.2012.23.
- Stabler, Edward P. (1997). Derivational minimalism. *Logical Aspects of Computational Linguistics. First International Conference, LACL '96, Nancy, France, September 23-25, 1996. Selected Papers*. Ed. by Christian Retoré. Lecture Notes in Computer Science 1328. Berlin, Heidelberg: Springer. DOI: 10.1007/BFb0052152.
- Stabler, Edward P. (2011). Computational perspectives on Minimalism. *Oxford handbook of linguistic Minimalism*. Ed. by Cedric Boeckx. Oxford: Oxford University Press, pp. 617–643. DOI: 10.1093/oxfordhb/9780199549368.013.0027.

Appendix

MG dependency trees with movement

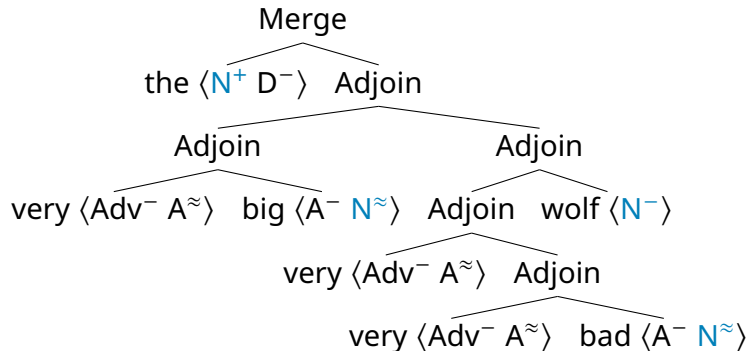
Which reporter did she speak to?



See Stabler (1997, 2011) and Graf and Kostyszyn (2021).

Derivation trees and recursive adjunction

Internal nodes represent the Merge/Move/Adjoin operations, and all leaves are lexical items.



Notice: as the number of adjuncts grows, the length of the both selection and adjunction dependencies grows without bound.