Syntactic tiers for movement and agreement

Day 3: Tree Rewriting & Externalization

Thomas Graf

Stony Brook University
mail@thomasgraf.net
https://thomasgraf.net

KU Leuven Lecture Series
December 5–7, 2023

Outline

Take-home message

- **Overgeneration problem in syntax**
  Subcategorization can express very unnatural constraints, due to **category refinement**.
- **A linguistically fertile solution**
  Category features don't come for free.
  They must be inferable from the local context.

Hidden power of subcategorization

Every formalism with subcategorization can express **undesirable constraints**. (Graf 2017)

Counting	every DP contains at least five LIs
Symmetry closure	every reflexive c-commands its antecedent
Complement	sentence well-formed iff ill-formed in English
Boolean closure	sentence must obey either V2 or Principle A, unless there are less than 7 pronounced LIs
Domain blindness	a sentence is well-formed iff there are at least two words that display word-final devoicing
Is(n't)lands	an adjunct is an island iff it is inside an embedded clause or it contains no animate nouns

Why?

- Complex constraints can be lexicalized
  by decomposing them into refined categories.
- They are then enforced via subcategorization.
- It's a generalized version of slash feature percolation.
  (Gazdar et al. 1985; Graf 2011; Kobele 2011)

An example

Subcategorization (Stabler 1997)

- **Category features** ($\texttt{F}^-$)
- **Selector features** ($\texttt{F}^+$)
- Subcategorization: matching features of opposite polarity

A very simple grammar

foo :: $\texttt{X}^-$	foo :: $\texttt{X}^+\texttt{X}^-$
bar :: $\texttt{X}^-$	bar :: $\texttt{X}^+\texttt{X}^-$
	ε :: $\texttt{X}^+\texttt{C}^-$

An example

Subcategorization (Stabler 1997)

- **Category features** ($\mathtt{F}^-$)
- **Selector features** ($\mathtt{F}^+$)
- Subcategorization: matching features of opposite polarity

A very simple grammar

$$\text{foo} :: \mathtt{X}^- \qquad \text{foo} :: \mathtt{X}^+\mathtt{X}^-$$

$$\text{bar} :: \mathtt{X}^- \qquad \text{bar} :: \mathtt{X}^+\mathtt{X}^-$$

$$\varepsilon :: \mathtt{X}^+\mathtt{C}^-$$

$$\text{foo} :: \mathtt{X}^-$$

An example

Subcategorization (Stabler 1997)

- **Category features** ($\mathtt{F}^-$)
- **Selector features** ($\mathtt{F}^+$)
- Subcategorization: matching features of opposite polarity

A very simple grammar

$\text{foo} :: \mathtt{X}^-$ $\quad$ $\text{foo} :: \mathtt{X}^+\mathtt{X}^-$

$\text{bar} :: \mathtt{X}^-$ $\quad$ $\text{bar} :: \mathtt{X}^+\mathtt{X}^-$

$\varepsilon :: \mathtt{X}^+\mathtt{C}^-$

$\text{bar} :: \mathtt{X}^+\mathtt{X}^-$

|

$\text{foo} :: \mathtt{X}^-$

An example

Subcategorization (Stabler 1997)

- **Category features** ($\mathtt{F}^-$)
- **Selector features** ($\mathtt{F}^+$)
- Subcategorization: matching features of opposite polarity

A very simple grammar

$\text{foo} :: \mathtt{X}^-$ $\quad$ $\text{foo} :: \mathtt{X}^+\mathtt{X}^-$

$\text{bar} :: \mathtt{X}^-$ $\quad$ $\text{bar} :: \mathtt{X}^+\mathtt{X}^-$

$\varepsilon :: \mathtt{X}^+\mathtt{C}^-$

$\varepsilon :: \mathtt{X}^+\mathtt{C}^-$

|

$\text{bar} :: \mathtt{X}^+\mathtt{X}^-$

|

$\text{foo} :: \mathtt{X}^-$

An example

Subcategorization (Stabler 1997)

- **Category features** ($\mathtt{F}^-$)
- **Selector features** ($\mathtt{F}^+$)
- Subcategorization: matching features of opposite polarity

A very simple grammar

foo :: $\mathtt{X}^-$

bar :: $\mathtt{X}^-$

foo :: $\mathtt{X}^+\mathtt{X}^-$

bar :: $\mathtt{X}^+\mathtt{X}^-$

ε :: $\mathtt{X}^+\mathtt{C}^-$

ε :: $\mathtt{X}^+\mathtt{C}^-$
|
bar :: $\mathtt{X}^+\mathtt{X}^-$
|
foo :: $\mathtt{X}^-$

foo :: $\mathtt{X}^-$

An example

Subcategorization (Stabler 1997)

- **Category features** ($\mathtt{F}^-$)
- **Selector features** ($\mathtt{F}^+$)
- Subcategorization: matching features of opposite polarity

A very simple grammar

foo :: $\mathtt{X}^-$

bar :: $\mathtt{X}^-$

foo :: $\mathtt{X}^+\mathtt{X}^-$

bar :: $\mathtt{X}^+\mathtt{X}^-$

ε :: $\mathtt{X}^+\mathtt{C}^-$

ε :: $\mathtt{X}^+\mathtt{C}^-$
|
bar :: $\mathtt{X}^+\mathtt{X}^-$
|
foo :: $\mathtt{X}^-$

bar :: $\mathtt{X}^+\mathtt{X}^-$
|
foo :: $\mathtt{X}^-$

An example

Subcategorization (Stabler 1997)

- **Category features** ($\mathtt{F}^-$)
- **Selector features** ($\mathtt{F}^+$)
- Subcategorization: matching features of opposite polarity

A very simple grammar

foo :: $\mathtt{X}^-$

bar :: $\mathtt{X}^-$

foo :: $\mathtt{X}^+\mathtt{X}^-$

bar :: $\mathtt{X}^+\mathtt{X}^-$

ε :: $\mathtt{X}^+\mathtt{C}^-$

ε :: $\mathtt{X}^+\mathtt{C}^-$
|
bar :: $\mathtt{X}^+\mathtt{X}^-$
|
foo :: $\mathtt{X}^-$

bar :: $\mathtt{X}^+\mathtt{X}^-$
|
bar :: $\mathtt{X}^+\mathtt{X}^-$
|
foo :: $\mathtt{X}^-$

An example

Subcategorization (Stabler 1997)

- **Category features** ($\texttt{F}^-$)
- **Selector features** ($\texttt{F}^+$)
- Subcategorization: matching features of opposite polarity

A very simple grammar

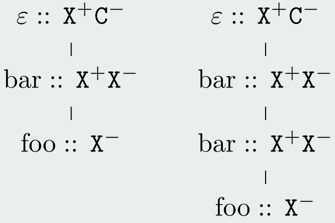
foo :: $\texttt{X}^-$

bar :: $\texttt{X}^-$

foo :: $\texttt{X}^+\texttt{X}^-$

bar :: $\texttt{X}^+\texttt{X}^-$

ε :: $\texttt{X}^+\texttt{C}^-$



Adding modulo counting

- Suppose every tree must have an **even number of nodes**
- **Refinement:** $\mathtt{X}^- \Rightarrow \mathtt{O}^-$ and $\mathtt{E}^-$ for Odd and Even

Refined grammar with even/odd distinction

$\text{foo} :: \mathtt{O}^-$	$\text{foo} :: \mathtt{E}^+\mathtt{O}^-$
	$\text{foo} :: \mathtt{O}^+\mathtt{E}^-$
$\text{bar} :: \mathtt{O}^-$	$\text{bar} :: \mathtt{E}^+\mathtt{O}^-$
	$\text{bar} :: \mathtt{O}^+\mathtt{E}^-$
	$\varepsilon :: \mathtt{O}^+\mathtt{C}^-$

Adding modulo counting

- Suppose every tree must have an **even number of nodes**
- **Refinement:** $\texttt{X}^-$ ⇒ $\texttt{O}^-$ and $\texttt{E}^-$ for Odd and Even

Refined grammar with even/odd distinction

$$\text{foo} :: \texttt{O}^- \qquad \text{foo} :: \texttt{E}^+\texttt{O}^-$$
$$\text{foo} :: \texttt{O}^+\texttt{E}^-$$
$$\text{bar} :: \texttt{O}^- \qquad \text{bar} :: \texttt{E}^+\texttt{O}^-$$
$$\text{bar} :: \texttt{O}^+\texttt{E}^- \qquad \text{foo} :: \texttt{O}^-$$
$$\varepsilon :: \texttt{O}^+\texttt{C}^-$$

Adding modulo counting

- Suppose every tree must have an **even number of nodes**
- **Refinement:** $\texttt{X}^- \Rightarrow \texttt{O}^-$ and $\texttt{E}^-$ for Odd and Even

Refined grammar with even/odd distinction

$\text{foo} :: \texttt{O}^-$

$\text{foo} :: \texttt{E}^+\texttt{O}^-$
$\text{foo} :: \texttt{O}^+\texttt{E}^-$

$\text{bar} :: \texttt{O}^-$

$\text{bar} :: \texttt{E}^+\texttt{O}^-$
$\text{bar} :: \texttt{O}^+\texttt{E}^-$

$\varepsilon :: \texttt{O}^+\texttt{C}^-$

$\text{bar} :: \texttt{O}^+\texttt{E}^-$
|
$\text{foo} :: \texttt{O}^-$

Adding modulo counting

- Suppose every tree must have an **even number of nodes**
- **Refinement:** $\mathtt{X}^- \Rightarrow \mathtt{O}^-$ and $\mathtt{E}^-$ for Odd and Even

Refined grammar with even/odd distinction

$\text{foo} :: \mathtt{O}^-$

$\text{foo} :: \mathtt{E}^+\mathtt{O}^-$

$\text{foo} :: \mathtt{O}^+\mathtt{E}^-$

$\text{bar} :: \mathtt{O}^-$

$\text{bar} :: \mathtt{E}^+\mathtt{O}^-$

$\text{bar} :: \mathtt{O}^+\mathtt{E}^-$

$\varepsilon :: \mathtt{O}^+\mathtt{C}^-$

$\varepsilon :: \mathtt{O}^+\mathtt{C}^-$
|
$\text{bar} :: \mathtt{O}^+\mathtt{E}^-$
|
$\text{foo} :: \mathtt{O}^-$

Adding modulo counting

- Suppose every tree must have an **even number of nodes**
- **Refinement:** $\texttt{X}^- \Rightarrow \texttt{O}^-$ and $\texttt{E}^-$ for Odd and Even

Refined grammar with even/odd distinction

$\text{foo} :: \texttt{O}^-$

$\text{foo} :: \texttt{E}^+\texttt{O}^-$

$\text{foo} :: \texttt{O}^+\texttt{E}^-$

$\text{bar} :: \texttt{O}^-$

$\text{bar} :: \texttt{E}^+\texttt{O}^-$

$\text{bar} :: \texttt{O}^+\texttt{E}^-$

$\varepsilon :: \texttt{O}^+\texttt{C}^-$

$\varepsilon :: \texttt{O}^+\texttt{C}^-$
|
$\text{bar} :: \texttt{O}^+\texttt{E}^-$
|
$\text{foo} :: \texttt{O}^-$

$\text{foo} :: \texttt{O}^-$

Adding modulo counting

- Suppose every tree must have an **even number of nodes**
- **Refinement:** $\mathtt{X}^-$ $\Rightarrow$ $\mathtt{O}^-$ and $\mathtt{E}^-$ for Odd and Even

Refined grammar with even/odd distinction

$\text{foo} :: \mathtt{O}^-$

$\text{foo} :: \mathtt{E}^+\mathtt{O}^-$

$\text{foo} :: \mathtt{O}^+\mathtt{E}^-$

$\text{bar} :: \mathtt{O}^-$

$\text{bar} :: \mathtt{E}^+\mathtt{O}^-$

$\text{bar} :: \mathtt{O}^+\mathtt{E}^-$

$\varepsilon :: \mathtt{O}^+\mathtt{C}^-$

$\varepsilon :: \mathtt{O}^+\mathtt{C}^-$
|
$\text{bar} :: \mathtt{O}^+\mathtt{E}^-$
|
$\text{foo} :: \mathtt{O}^-$

$\text{bar} :: \mathtt{O}^+\mathtt{E}^-$
|
$\text{foo} :: \mathtt{O}^-$

Adding modulo counting

- Suppose every tree must have an **even number of nodes**
- **Refinement:** $\texttt{X}^-$ ⇒ $\texttt{O}^-$ and $\texttt{E}^-$ for Odd and Even

Refined grammar with even/odd distinction

foo :: $\texttt{O}^-$

foo :: $\texttt{E}^+\texttt{O}^-$

foo :: $\texttt{O}^+\texttt{E}^-$

bar :: $\texttt{O}^-$

bar :: $\texttt{E}^+\texttt{O}^-$

bar :: $\texttt{O}^+\texttt{E}^-$

ε :: $\texttt{O}^+\texttt{C}^-$

ε :: $\texttt{O}^+\texttt{C}^-$
|
bar :: $\texttt{O}^+\texttt{E}^-$
|
foo :: $\texttt{O}^-$

bar :: $\texttt{E}^+\texttt{O}^-$
|
bar :: $\texttt{O}^+\texttt{E}^-$
|
foo :: $\texttt{O}^-$

Adding modulo counting

- Suppose every tree must have an **even number of nodes**
- **Refinement:** $\mathtt{X}^-$ $\Rightarrow$ $\mathtt{O}^-$ and $\mathtt{E}^-$ for Odd and Even

Refined grammar with even/odd distinction

foo :: $\mathtt{O}^-$

foo :: $\mathtt{E}^+\mathtt{O}^-$

foo :: $\mathtt{O}^+\mathtt{E}^-$

bar :: $\mathtt{O}^-$

bar :: $\mathtt{E}^+\mathtt{O}^-$

bar :: $\mathtt{O}^+\mathtt{E}^-$

ε :: $\mathtt{O}^+\mathtt{C}^-$

ε :: $\mathtt{O}^+\mathtt{C}^-$
|
bar :: $\mathtt{O}^+\mathtt{E}^-$
|
foo :: $\mathtt{O}^-$

ε :: $\mathtt{O}^+\mathtt{C}^-$
|
bar :: $\mathtt{E}^+\mathtt{O}^-$
|
bar :: $\mathtt{O}^+\mathtt{E}^-$
|
foo :: $\mathtt{O}^-$

The problem with subcategorization

- Even very complex constraints can be
  1. compiled into the category system and
  2. enforced via subcategorization.
- works for all MSO constraints ⇒ massive overgeneration (Graf 2011; Kobele 2011)
- Linguistic criteria for determining categories are too weak to prevent this.
  - morphology
  - syntactic distribution
  - semantics

The central issue

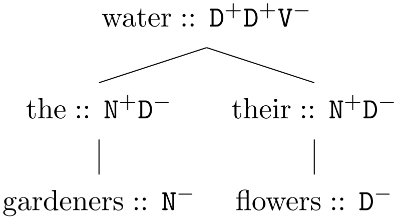
We need a more restrictive notion of category!

A formal notion of complexity

- We need to restrict the power of subcategorization, but how?
- Features currently come for free.
- We must **measure the cost of features**.

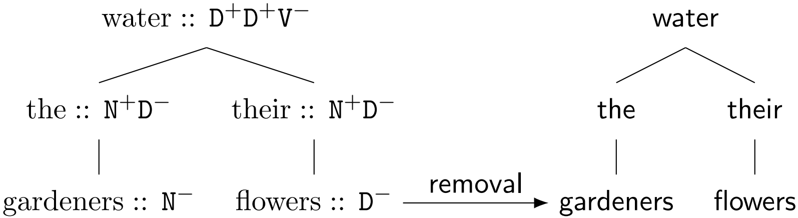
A formal notion of complexity

- We need to restrict the power of subcategorization, but how?
- Features currently come for free.
- We must **measure the cost of features**.



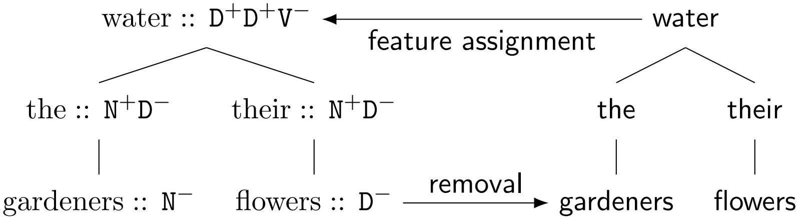
A formal notion of complexity

- We need to restrict the power of subcategorization, but how?
- Features currently come for free.
- We must **measure the cost of features**.



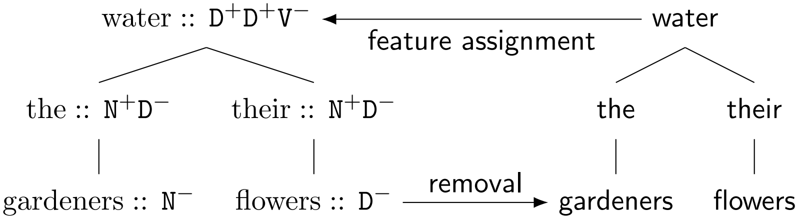
A formal notion of complexity

- We need to restrict the power of subcategorization, but how?
- Features currently come for free.
- We must **measure the cost of features**.



A formal notion of complexity

- We need to restrict the power of subcategorization, but how?
- Features currently come for free.
- We must **measure the cost of features**.



Local feature recoverability

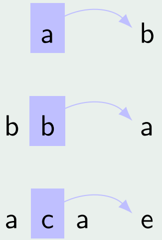
Features must be recoverable in an **ISL** fashion.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandlee 2014)

Rewrite each symbol in a string based on its local input context.

An ISL-3 relabeling



- ISL is a very weak class,
  yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandlee 2014)

Rewrite each symbol in a string based on its local input context.

An ISL-3 relabeling

a → b

b b → a

a c a → e

a b b a c a

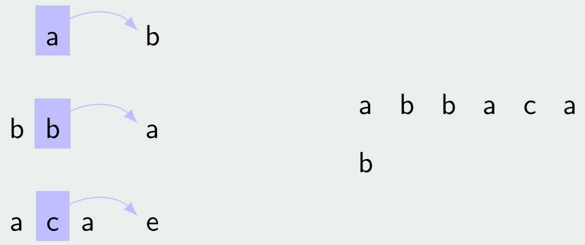
- ISL is a very weak class,
  yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandlee 2014)

Rewrite each symbol in a string based on its local input context.

An ISL-3 relabeling



- ISL is a very weak class,
  yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandlee 2014)

Rewrite each symbol in a string based on its local input context.

An ISL-3 relabeling

a → b

b b → a

a c a → e

a b b a c a

b b

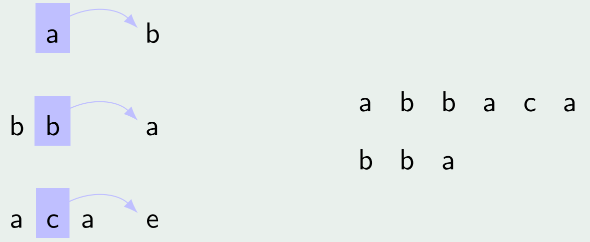
- ISL is a very weak class,
  yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandlee 2014)

Rewrite each symbol in a string based on its local input context.

An ISL-3 relabeling



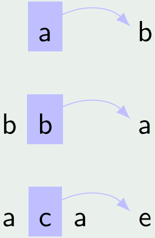
- ISL is a very weak class,
  yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandlee 2014)

Rewrite each symbol in a string based on its local input context.

An ISL-3 relabeling



a b b a c a

b b a b

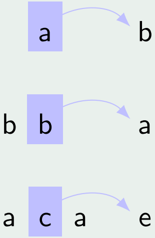
- ISL is a very weak class,
  yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandlee 2014)

Rewrite each symbol in a string based on its local input context.

An ISL-3 relabeling



a b b a c a

b b a b e

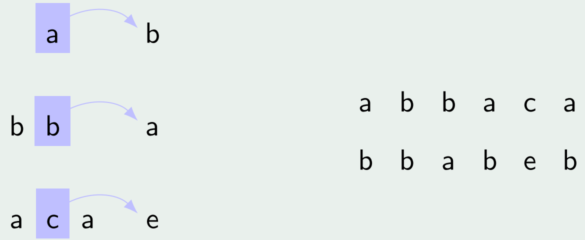
- ISL is a very weak class,
  yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandlee 2014)

Rewrite each symbol in a string based on its local input context.

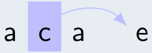
An ISL-3 relabeling



- ISL is a very weak class,
  yet widely found in phonology and morphology.

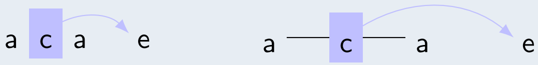
Lifting ISL relabelings to trees

String contexts as tree contexts



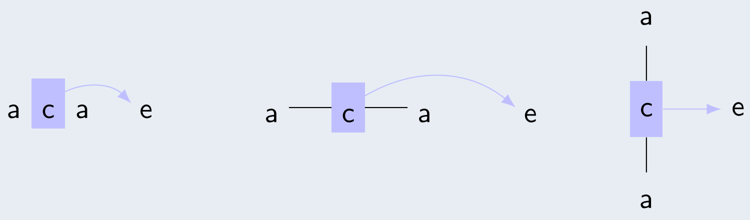
Lifting ISL relabelings to trees

String contexts as tree contexts



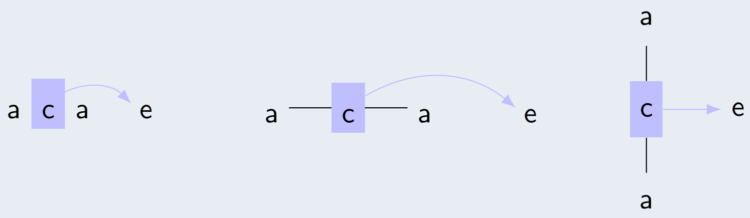
Lifting ISL relabelings to trees

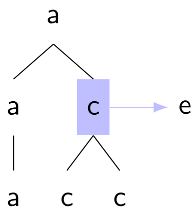
String contexts as tree contexts



Lifting ISL relabelings to trees

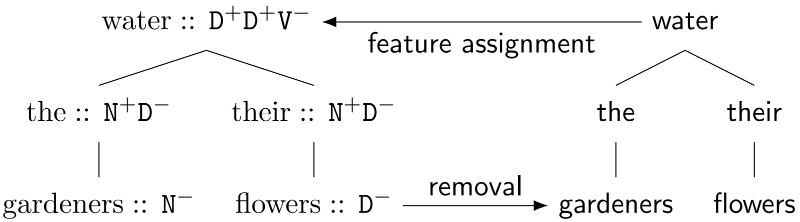
String contexts as tree contexts





Reminder: ISL for feature inference

- Feature cost ≈ how hard to assign by transduction?

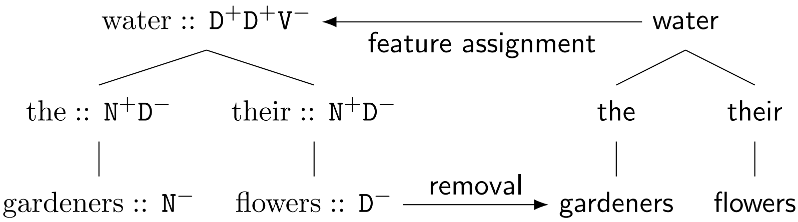


Local feature recoverability

Features must be recoverable in an **ISL** fashion.

Reminder: ISL for feature inference

- Feature cost $\approx$ how hard to assign by transduction?



Local feature recoverability

Features must be recoverable in an **ISL** fashion.

Intuition
Categorial ambiguity can be resolved within local context

Modulo counting is not ISL recoverable

$\varepsilon :: \texttt{O}^{+}\texttt{C}^{-}$

|

$\text{bar} :: \texttt{E}^{+}\texttt{O}^{-}$

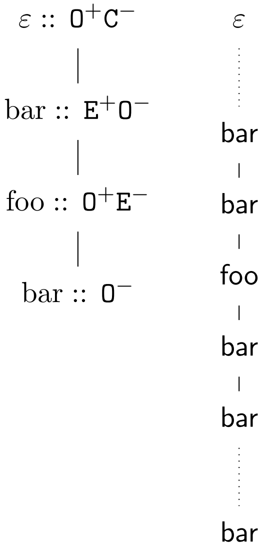
|

$\text{foo} :: \texttt{O}^{+}\texttt{E}^{-}$

|

$\text{bar} :: \texttt{O}^{-}$

Modulo counting is not ISL recoverable



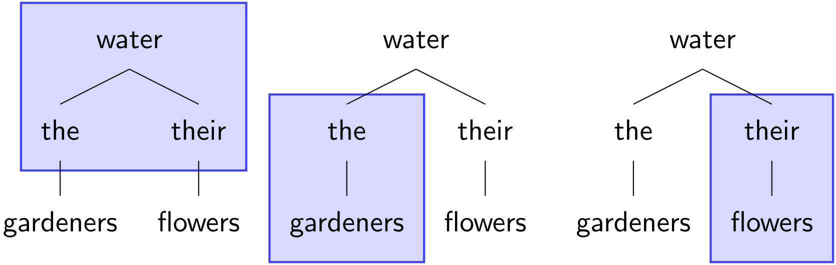
- Can you determine the features of **foo**?
  1. $\mathtt{O}^{+}\ \mathtt{E}^{-}$
  2. $\mathtt{E}^{+}\ \mathtt{O}^{-}$
- No, that's impossible.
- You need more than local information.
- Modulo counting is not ISL recoverable.

An empirical conjecture

SL-2 recoverability conjecture

The category and selector features of lexical items are

- recoverable from feature-less dependency trees
- using only a window of size 2.



Implications and open issues

Implications

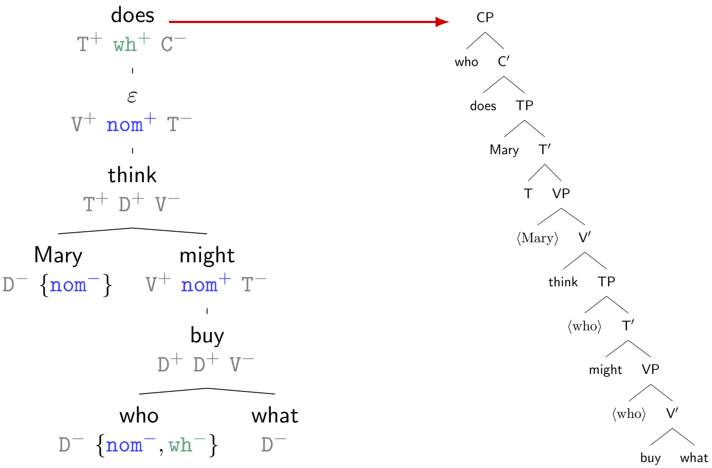
- We avoid tons of overgeneration.
- Heads only select for arguments, not arguments of arguments.

Open issues

- Needs to be tested across many languages
- Depends on theoretical assumptions
  - distribution of empty heads
  - lexical items fully inflected or bare roots? (Hale and Keyser 1993; Marantz 1997)
- SL-2 may be too tight, but SL-k recoverability seems safe
- Move features are not ISL recoverable!

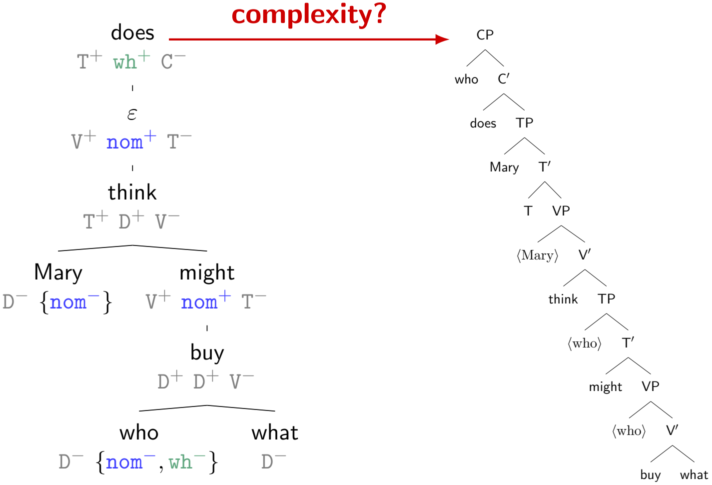
An incomplete picture

Movement isn't just a syntactic dependency, it **affects the output**.



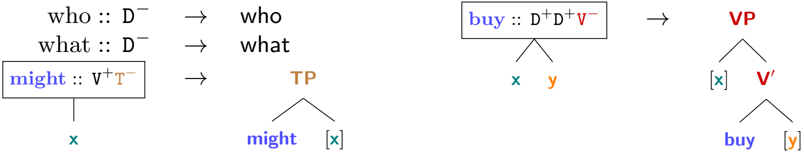
An incomplete picture

Movement isn't just a syntactic dependency, it **affects the output**.



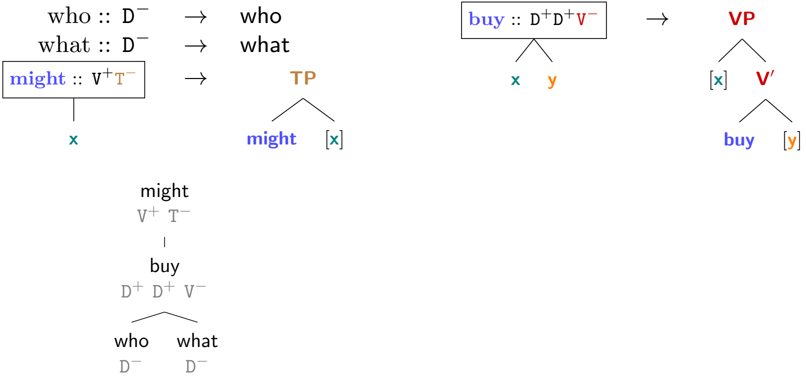
Phrase structure trees without movement

no movement $\Rightarrow$ easy translation to phrase structure trees



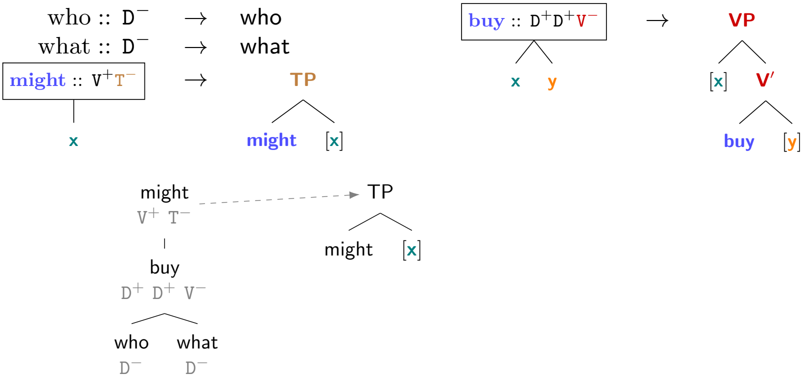
Phrase structure trees without movement

no movement ⇒ easy translation to phrase structure trees



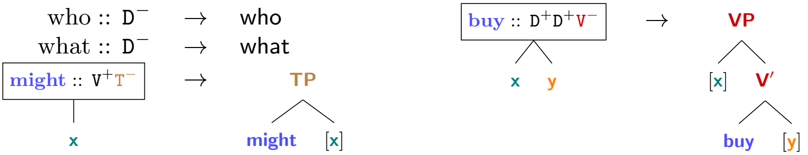
Phrase structure trees without movement

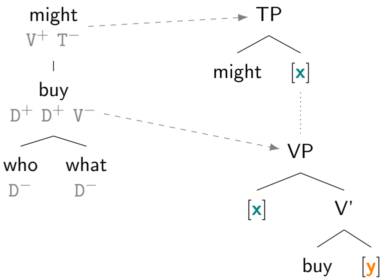
no movement ⇒ easy translation to phrase structure trees



Phrase structure trees without movement

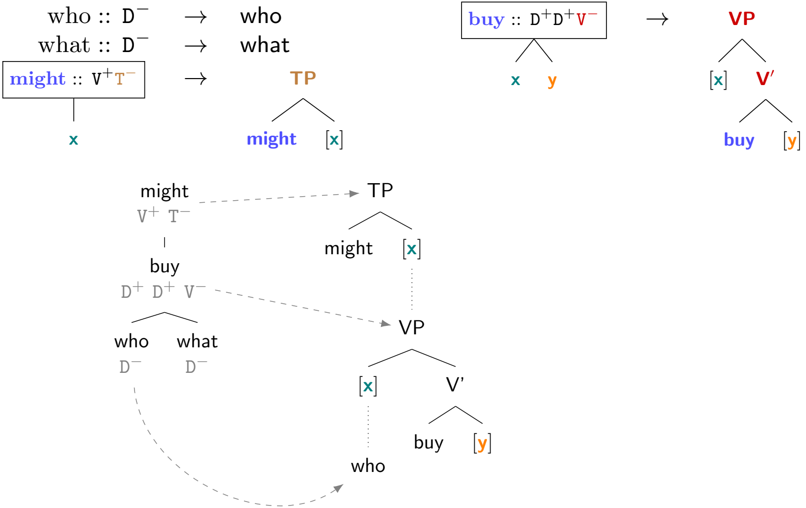
no movement ⇒ easy translation to phrase structure trees





Phrase structure trees without movement

no movement ⇒ easy translation to phrase structure trees

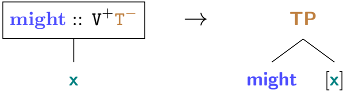


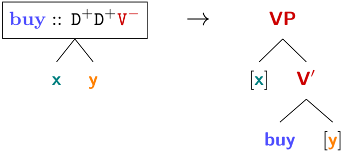
Phrase structure trees without movement

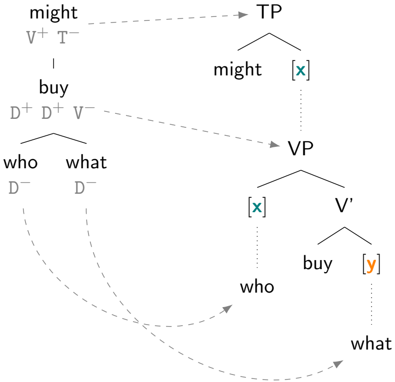
no movement ⇒ easy translation to phrase structure trees

who :: D^- → who

what :: D^- → what

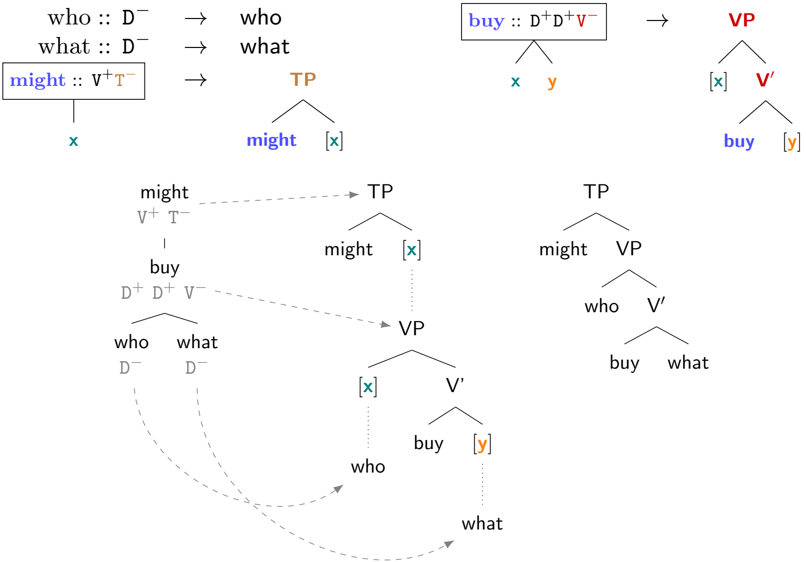






Phrase structure trees without movement

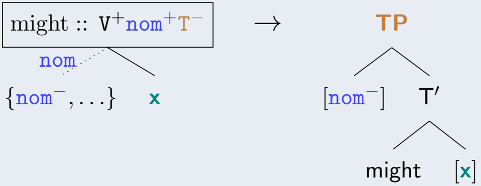
no movement ⇒ easy translation to phrase structure trees



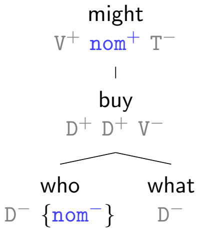
Adding tier relations

- ISL limited to local contexts ⇔ unbounded movement
- **But:** movement is **local over tiers**
- suffices to enrich ISL rewrite rules with tier-daughter relation

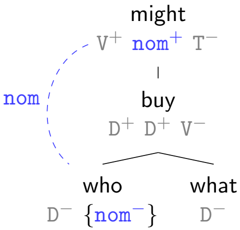
ISL rewrite rule with nom-daughter relation



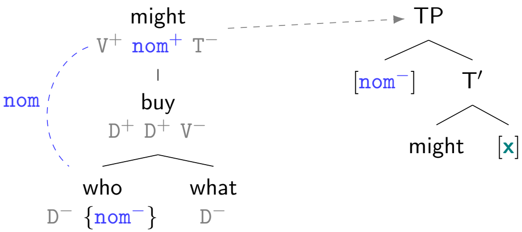
Example with subject movement



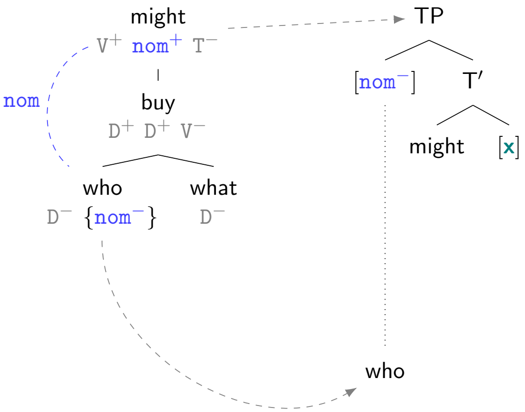
Example with subject movement



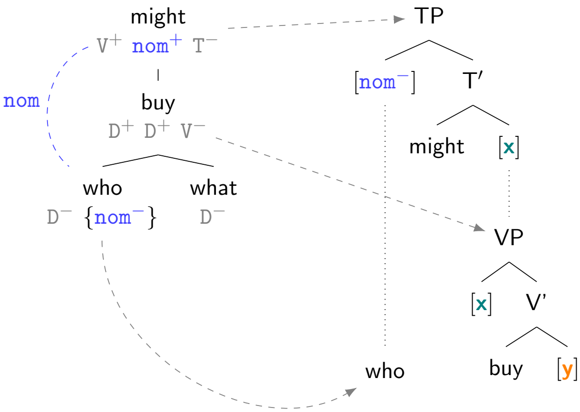
Example with subject movement



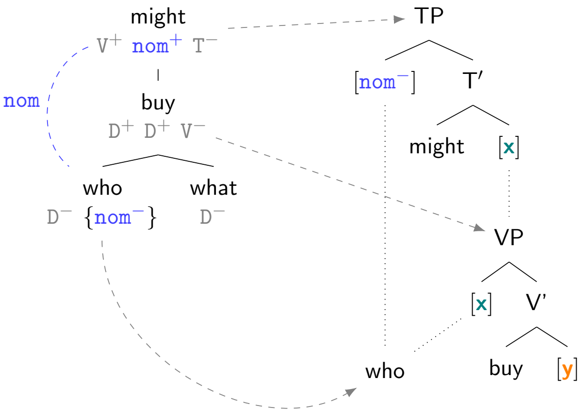
Example with subject movement



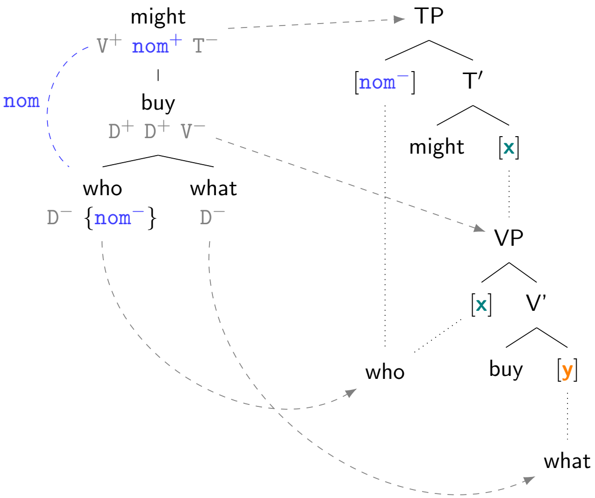
Example with subject movement



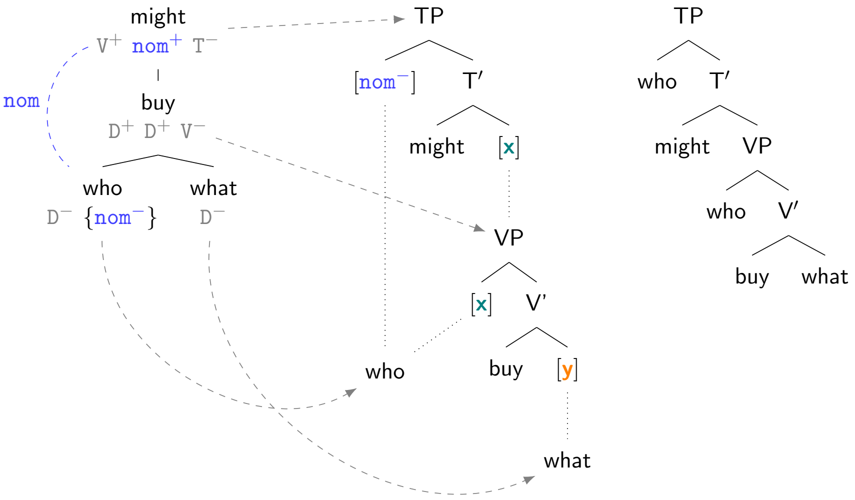
Example with subject movement



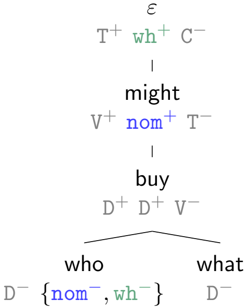
Example with subject movement



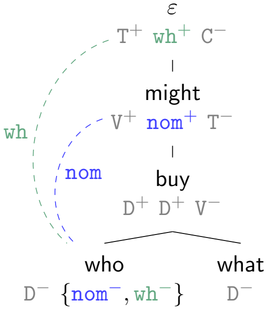
Example with subject movement



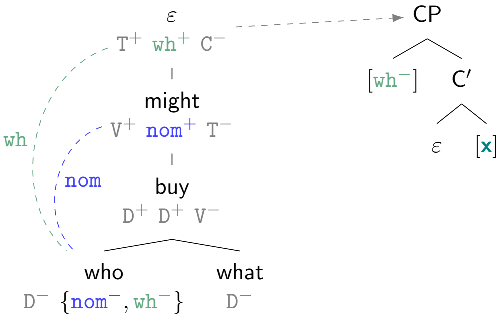
Multiple copies are straightforward



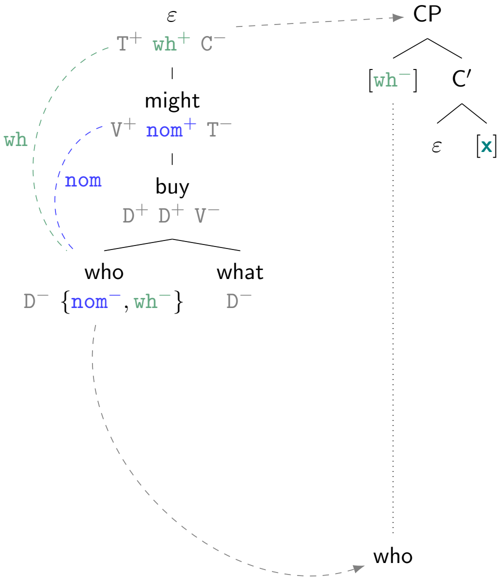
Multiple copies are straightforward



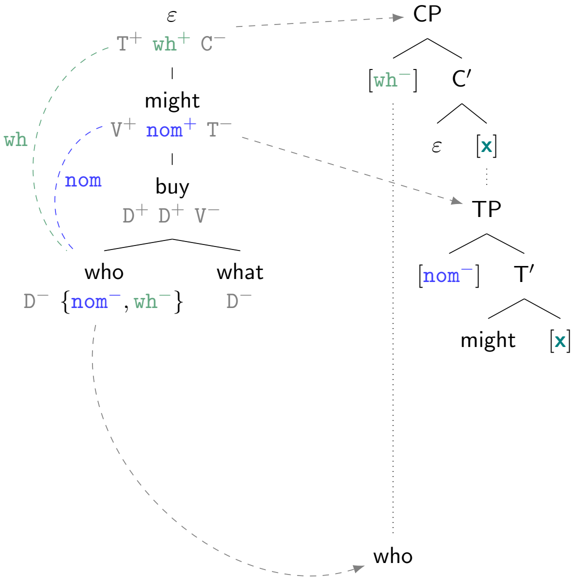
Multiple copies are straightforward



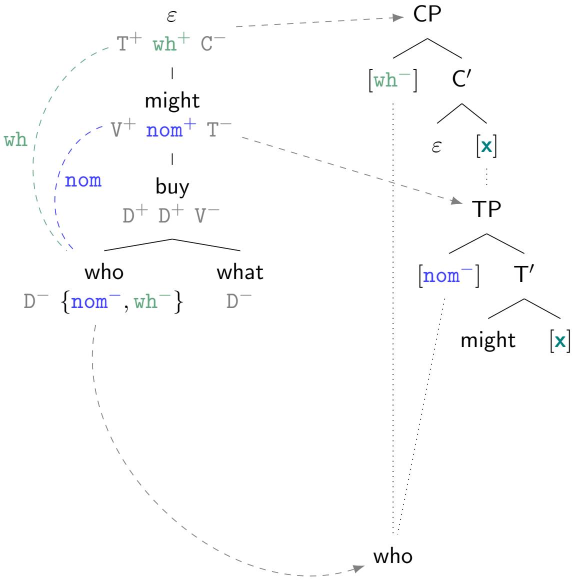
Multiple copies are straightforward



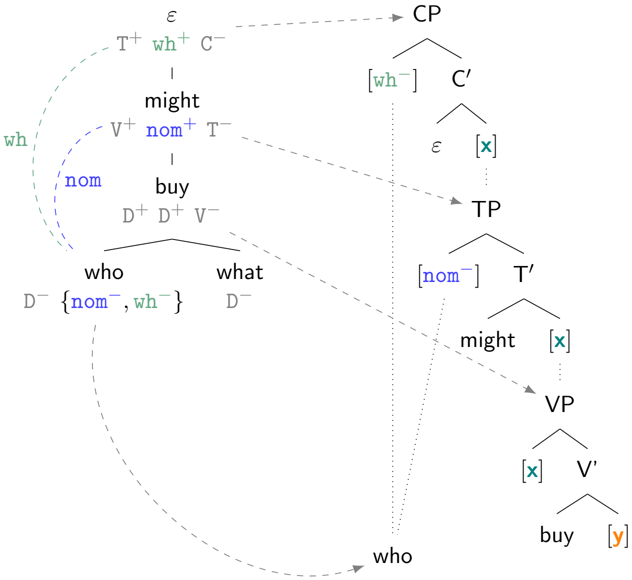
Multiple copies are straightforward



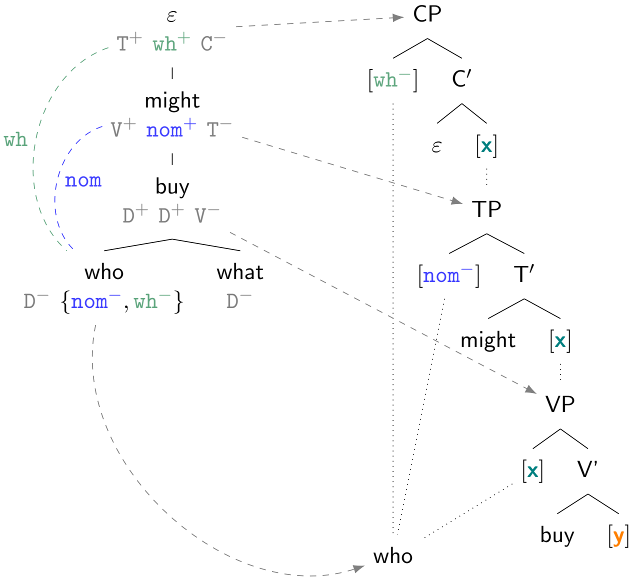
Multiple copies are straightforward



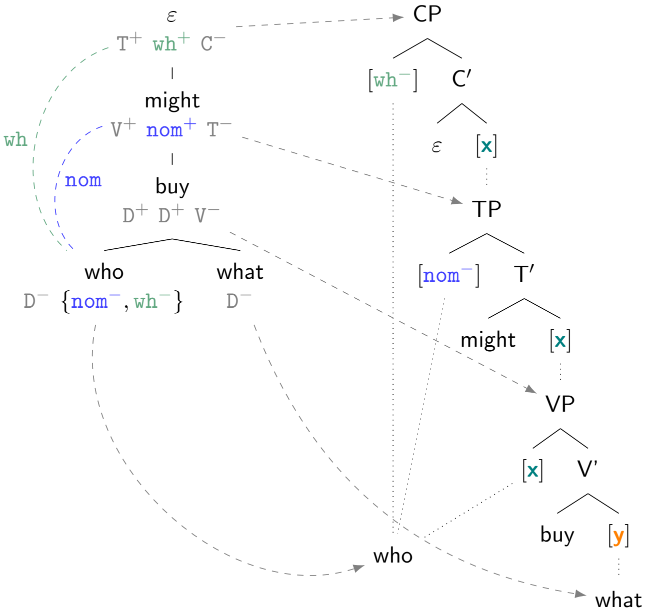
Multiple copies are straightforward



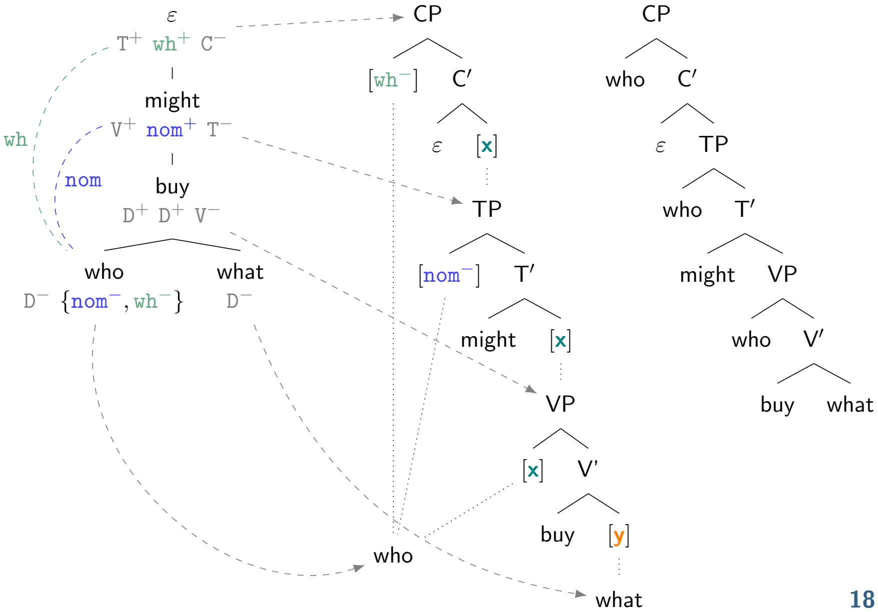
Multiple copies are straightforward



Multiple copies are straightforward



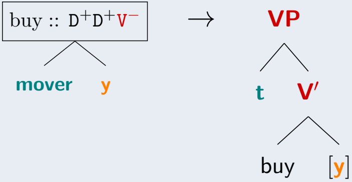
Multiple copies are straightforward



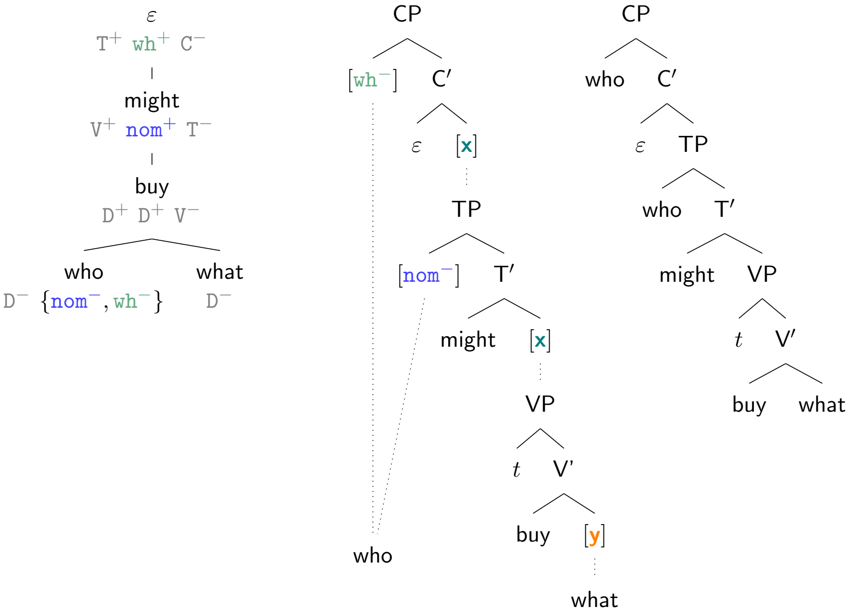
Linearization/traces: tricky

- Copies don't tell us how to pronounce the tree.
- **Traces**: unpronounced landing sites of movers

A delinking rule for base positions

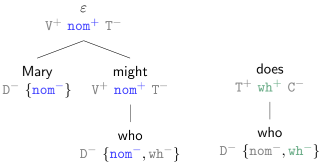


Base delinking example



Lexical predictability

- Given two landing sites **x** and **y** on different tiers, one cannot tell from the tiers whether **x** or **y** is higher.
- We **cannot distinguish final from non-final landing sites**.



Lexical predictability requirement of delinking

Delinking works only if one knows whether to

1. insert a copy, or
2. insert a trace.

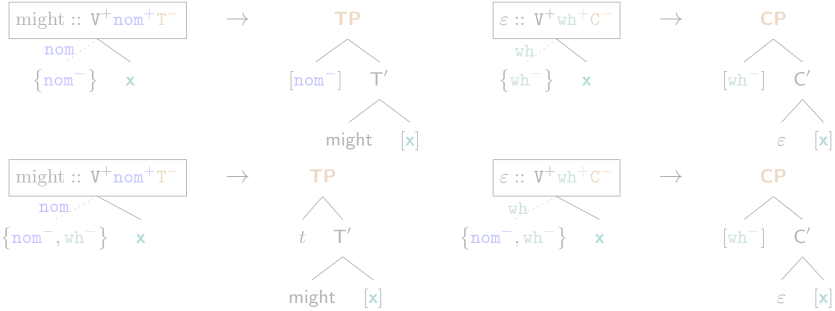
Due to the limitations of tiers,
this must be **inferable directly from the mover**.

Empirical support

- Lexical predictability holds for nom and wh movement.

Ban on Improper Movement (BoIM)

If a mover undergoes both nom and wh movement,
nom movement derivationally precedes wh movement.

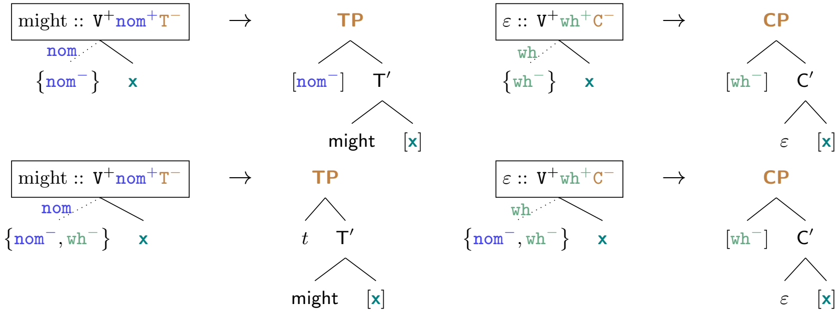


Empirical support

- Lexical predictability holds for nom and wh movement.

Ban on Improper Movement (BoIM)

If a mover undergoes both nom and wh movement,
nom movement derivationally precedes wh movement.



Output-oriented Ban on Improper Movement

- BoIM is a particular instance of a more general requirement.

Output-oriented Ban on Improper Movement (OOBoIM)

- Let l be an arbitrary lexical item with $\{\mathtt{f}^-, \mathrm{g}_0^-, \ldots, \mathrm{g}_n^-\}$.
- If l's final movement step is f-movement in **some** derivation, then l's final movement step is f-movement in **all** derivations.

- Kenneth Hanson's analysis of MG corpus supports **even stronger version**: if $\mathtt{f} \prec \mathrm{g}$ for l in some derivation, then $\mathtt{f} \prec \mathrm{g}$ for l in all derivations.
- OOBoIM permits BoIM violations
  hyperraising

Kenneth Hanson

Conclusion

- Movement is both a syntactic dependency and an operation.
- In both cases the core of movement is **local over tiers**.
- Identifying mover with all landing sites
  (copies/multidominance)
  easier than identifying output-relevant landing sites (traces)

Outlook: Bringing it all together

- TSL perspectives of all movement types
  covert, successive-cyclic, sidewards, multiple wh
- Tiers for islands, extraction morphology and path conditions
  Wolof u-chains, floating quantifiers, Germanic wh-copying
- Learning
  SL learning of Merge features, ??? for Move features

Acknowledgments

This work is supported by the National Science Foundation under Grant No. BCS-1845344.

References I

Chandlee, Jane. 2014. *Strictly local phonological processes*. Doctoral Dissertation, University of Delaware. URL http://udspace.udel.edu/handle/19716/13374.

Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum, and Ivan A. Sag. 1985. *Generalized phrase structure grammar*. Oxford: Blackwell.

Graf, Thomas. 2011. Closure properties of Minimalist derivation tree languages. In *LACL 2011*, ed. Sylvain Pogodalla and Jean-Philippe Prost, volume 6736 of *Lecture Notes in Artificial Intelligence*, 96–111. Heidelberg: Springer. URL https://dx.doi.org/10.1007/978-3-642-22221-4_7.

Graf, Thomas. 2017. A computational guide to the dichotomy of features and constraints. *Glossa* 2:1–36. URL https://dx.doi.org/10.5334/gjgl.212.

Hale, Kenneth, and Samuel J. Keyser. 1993. On argument structure and the lexical expression of syntactic relations. In *The view from building 20: Essays in honor of sylvain bromberger*, ed. Kenneth Hale and Samuel J. Keyser. Cambridge, MA: MIT Press.

Kobele, Gregory M. 2011. Minimalist tree languages are closed under intersection with recognizable tree languages. In *LACL 2011*, ed. Sylvain Pogodalla and Jean-Philippe Prost, volume 6736 of *Lecture Notes in Artificial Intelligence*, 129–144. URL https://doi.org/10.1007/978-3-642-22221-4_9.

Marantz, Alec. 1997. No escape from syntax: Don't try morphological analysis in the privacy of your own lexicon. *Penn Working Papers in Linguistics* 4:201–225.

References II

Stabler, Edward P. 1997. Derivational Minimalism. In *Logical aspects of computational linguistics*, ed. Christian Retoré, volume 1328 of *Lecture Notes in Computer Science*, 68–95. Berlin: Springer. URL https://doi.org/10.1007/BFb0052152.