

Curbing Feature Coding: Strictly Local Feature Assignment

Thomas Graf

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

SCiL 2020
January 4, 2020

Download
the slides



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.

Outline

- 1** Subcategorization is too powerful
- 2** Local feature recoverability

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 from Minimalist grammars

Subcategorization in MGs (Stabler 1997)

- ▶ **Category features** (F^-)
- ▶ **Selector features** (F^+)
- ▶ Subcategorization: matching features of opposite polarity

A very simple MG

foo :: X^- foo :: X^+X^-

bar :: X^- bar :: X^+X^-

ε :: X^+C^-

An example from Minimalist grammars

Subcategorization in MGs (Stabler 1997)

- ▶ **Category features** (F^-)
 - ▶ **Selector features** (F^+)
 - ▶ Subcategorization: matching features of opposite polarity

A very simple MG

foo :: X⁻ foo :: X⁺X⁻

$\bar{b} \bar{a} \bar{r} \bar{b} \bar{a} \bar{r} \bar{b} \bar{a} \bar{r} \bar{b} \bar{a} \bar{r} \bar{b}$

$\varepsilon :: X^+ C^-$ foo :: X^-

An example from Minimalist grammars

Subcategorization in MGs (Stabler 1997)

- ▶ **Category features** (F^-)
- ▶ **Selector features** (F^+)
- ▶ Subcategorization: matching features of opposite polarity

A very simple MG

foo :: X^- foo :: X^+X^-

bar :: X^- bar :: X^+X^-

ε :: X^+C^-

bar :: X^+X^-

|

foo :: X^-

An example from Minimalist grammars

Subcategorization in MGs (Stabler 1997)

- ▶ **Category features** (F^-)
- ▶ **Selector features** (F^+)
- ▶ Subcategorization: matching features of opposite polarity

A very simple MG

		$\varepsilon :: X^+ C^-$
foo :: X^-	foo :: $X^+ X^-$	$\begin{array}{c} \\ \text{bar} :: X^+ X^- \end{array}$
bar :: X^-	bar :: $X^+ X^-$	$\begin{array}{c} \\ \text{foo} :: X^- \end{array}$
	$\varepsilon :: X^+ C^-$	

An example from Minimalist grammars

Subcategorization in MGs (Stabler 1997)

- ▶ **Category features** (F^-)
- ▶ **Selector features** (F^+)
- ▶ Subcategorization: matching features of opposite polarity

A very simple MG

		$\varepsilon :: X^+ C^-$
foo :: X^-	foo :: $X^+ X^-$	bar :: $X^+ X^-$
bar :: X^-	bar :: $X^+ X^-$	foo :: X^-
	$\varepsilon :: X^+ C^-$	foo :: X^-

An example from Minimalist grammars

Subcategorization in MGs (Stabler 1997)

- ▶ **Category features** (F^-)
- ▶ **Selector features** (F^+)
- ▶ Subcategorization: matching features of opposite polarity

A very simple MG

		$\varepsilon :: X^+ C^-$
foo :: X^-	foo :: $X^+ X^-$	\mid
bar :: X^-	bar :: $X^+ X^-$	\mid
	$\varepsilon :: X^+ C^-$	foo :: X^-
		bar :: $X^+ X^-$
		\mid
		foo :: X^-

An example from Minimalist grammars

Subcategorization in MGs (Stabler 1997)

- ▶ **Category features** (F^-)
- ▶ **Selector features** (F^+)
- ▶ Subcategorization: matching features of opposite polarity

A very simple MG

		$\varepsilon :: X^+ C^-$	
foo :: X^-	foo :: $X^+ X^-$	bar :: $X^+ X^-$	bar :: $X^+ X^-$
bar :: X^-	bar :: $X^+ X^-$	foo :: X^-	bar :: $X^+ X^-$
	$\varepsilon :: X^+ C^-$		
			foo :: X^-

An example from Minimalist grammars

Subcategorization in MGs (Stabler 1997)

- ▶ **Category features** (F^-)
- ▶ **Selector features** (F^+)
- ▶ Subcategorization: matching features of opposite polarity

A very simple MG

		$\varepsilon :: X^+C^-$		$\varepsilon :: X^+C^-$
foo :: X^-	foo :: X^+X^-	bar :: X^+X^-	bar :: X^+X^-	bar :: X^+X^-
bar :: X^-	bar :: X^+X^-	foo :: X^-	bar :: X^+X^-	bar :: X^+X^-
	$\varepsilon :: X^+C^-$			foo :: X^-

Adding modulo counting

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

Refined MG with even/odd distinction

foo :: O^- foo :: E^+O^-
 foo :: O^+E^-

bar :: O^- bar :: E^+O^-
 bar :: O^+E^-

$\varepsilon :: O^+C^-$

Adding modulo counting

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

Refined MG with even/odd distinction

foo :: O^- foo :: E^+O^-
 foo :: O^+E^-

bar :: O^- bar :: E^+O^-
 bar :: O^+E^- foo :: O^-

$\varepsilon :: O^+C^-$

Adding modulo counting

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

Refined MG with even/odd distinction

foo :: O^- foo :: E^+O^-
 foo :: O^+E^-

bar :: O^- bar :: E^+O^-
 bar :: O^+E^-

bar :: O^+E^-

|

foo :: O^-

$\varepsilon :: O^+C^-$

Adding modulo counting

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

Refined MG with even/odd distinction

foo :: O^-	foo :: E^+O^-	$\varepsilon :: O^+C^-$
	foo :: O^+E^-	
bar :: O^-	bar :: E^+O^-	bar :: O^+E^-
	bar :: O^+E^-	
		foo :: O^-
	$\varepsilon :: O^+C^-$	

Adding modulo counting

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

Refined MG with even/odd distinction

foo :: O^-	foo :: E^+O^-	$\varepsilon :: O^+C^-$
	foo :: O^+E^-	
bar :: O^-	bar :: E^+O^-	bar :: O^+E^-
	bar :: O^+E^-	
		foo :: O^-
	$\varepsilon :: O^+C^-$	
		foo :: O^-

Adding modulo counting

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

Refined MG with even/odd distinction

foo :: O^-	foo :: E^+O^-	$\varepsilon :: O^+C^-$
	foo :: O^+E^-	
bar :: O^-	bar :: E^+O^-	bar :: O^+E^-
	bar :: O^+E^-	
	$\varepsilon :: O^+C^-$	foo :: O^-
		bar :: O^+E^-
		foo :: O^-

Adding modulo counting

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

Refined MG with even/odd distinction

foo :: O^-	foo :: E^+O^-	$\varepsilon :: O^+C^-$
	foo :: O^+E^-	
bar :: O^-	bar :: E^+O^-	bar :: O^+E^-
	bar :: O^+E^-	
	$\varepsilon :: O^+C^-$	bar :: O^+E^-
		foo :: O^-

Adding modulo counting

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

Refined MG with even/odd distinction

foo :: O^-	foo :: E^+O^-	$\varepsilon :: O^+C^-$	$\varepsilon :: O^+C^-$
	foo :: O^+E^-		
bar :: O^-	bar :: E^+O^-	bar :: O^+E^-	bar :: E^+O^-
	bar :: O^+E^-		
	$\varepsilon :: O^+C^-$	foo :: O^-	bar :: O^+E^-
			foo :: 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?
- ▶ Linguistic restrictions on categories don't work.
- ▶ **Subregular complexity** provides a fix...

(Heinz 2009, 2010, 2018; Chandee 2014; Jardine 2016; McMullin 2016; Aksënova et al. 2016; Graf 2018; Shafiei and Graf 2020)

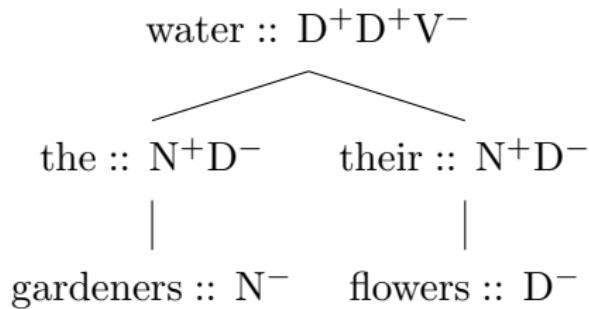


The key idea

- ▶ Features currently come for free.
- ▶ We must **measure the cost of features**.

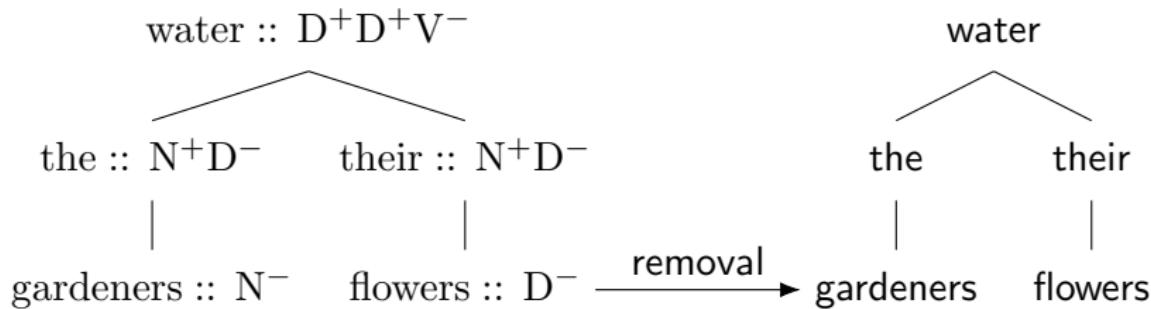
The key idea

- ▶ Features currently come for free.
- ▶ We must **measure the cost of features**.



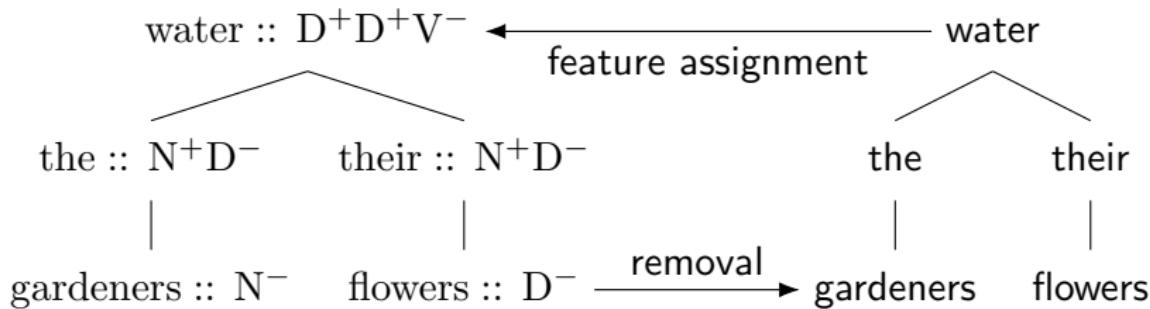
The key idea

- ▶ Features currently come for free.
- ▶ We must **measure the cost of features**.



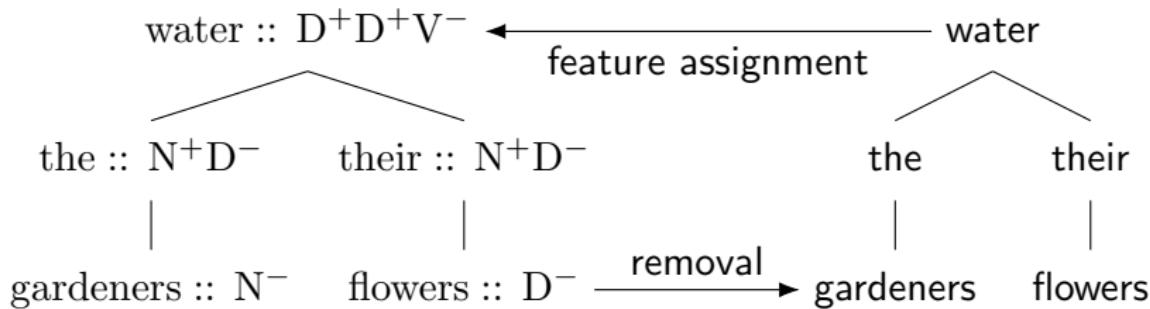
The key idea

- ▶ Features currently come for free.
- ▶ We must **measure the cost of features**.



The key idea

- ▶ Features currently come for free.
- ▶ We must **measure the cost of features**.



Local feature recoverability

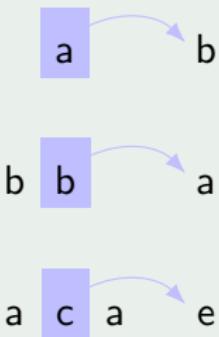
Features must be recoverable in an **ISL** fashion.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandee 2014)

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

An ISL-3 relabeling



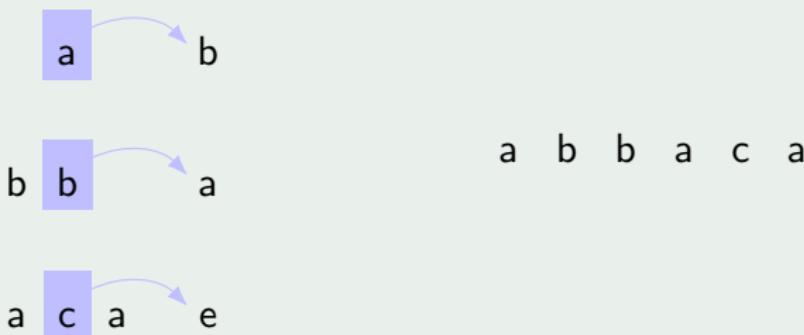
- ▶ ISL is a very weak class,
yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandee 2014)

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

An ISL-3 relabeling



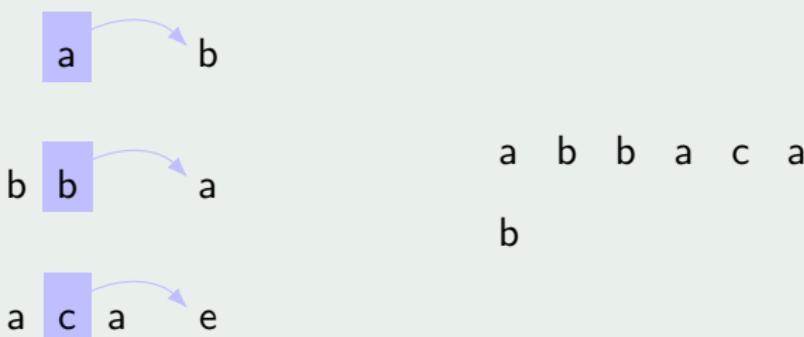
- ▶ ISL is a very weak class,
yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandee 2014)

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

An ISL-3 relabeling



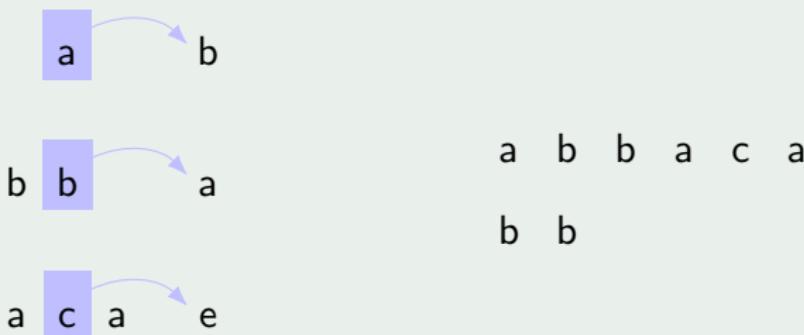
- ▶ ISL is a very weak class,
yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandee 2014)

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

An ISL-3 relabeling



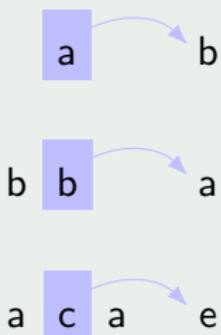
- ▶ ISL is a very weak class,
yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandee 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

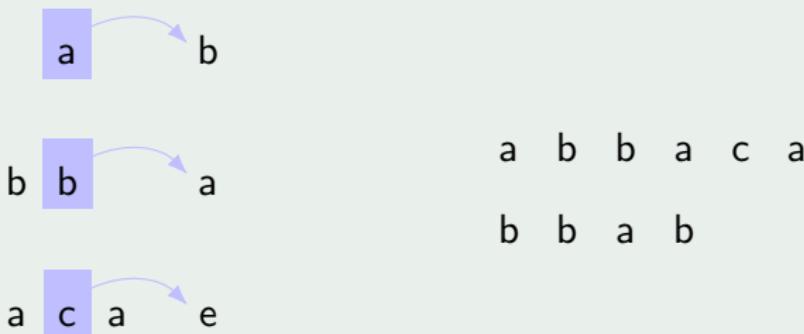
- ▶ ISL is a very weak class,
yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandee 2014)

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

An ISL-3 relabeling



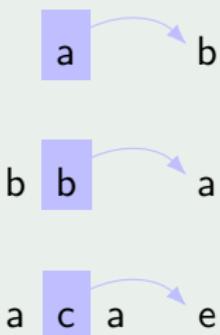
- ▶ ISL is a very weak class,
yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandee 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

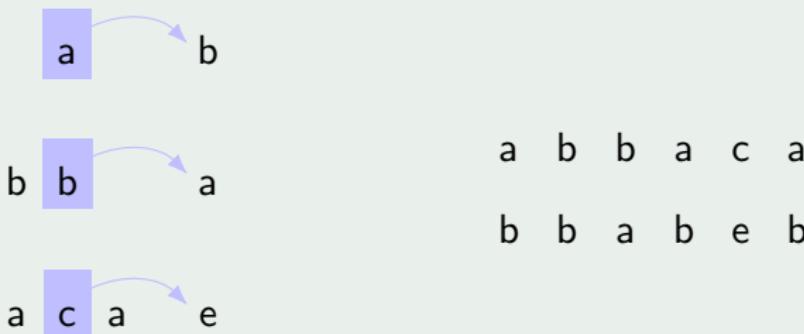
- ▶ ISL is a very weak class,
yet widely found in phonology and morphology.

Input strictly k-local relabelings

ISL string-to-string transduction (Chandee 2014)

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

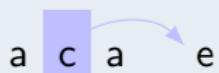
An ISL-3 relabeling



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

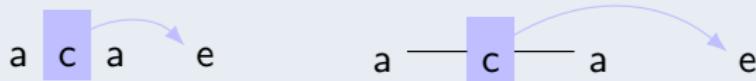
Lifting ISL relabelings to trees

String contexts as tree contexts



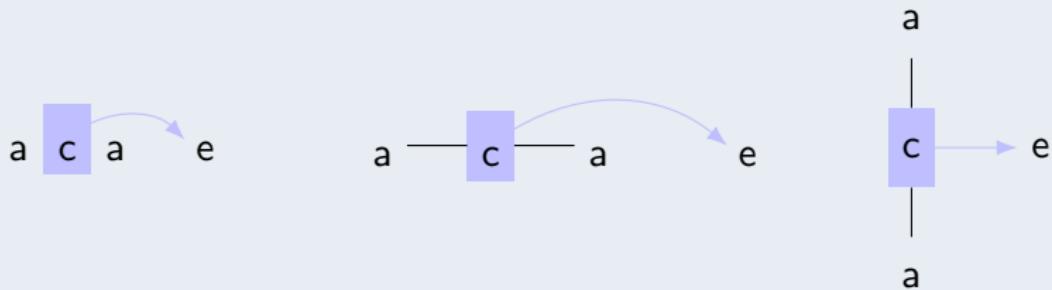
Lifting ISL relabelings to trees

String contexts as tree contexts



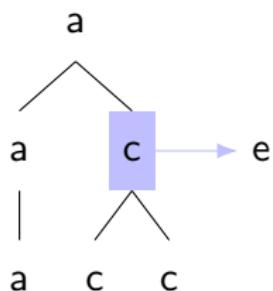
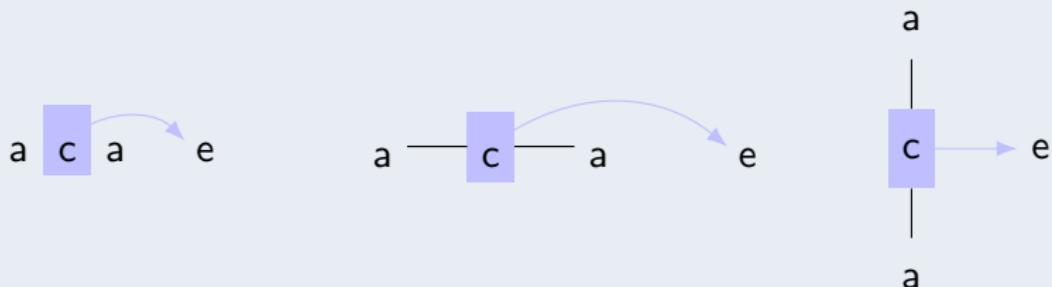
Lifting ISL relabelings to trees

String contexts as tree contexts



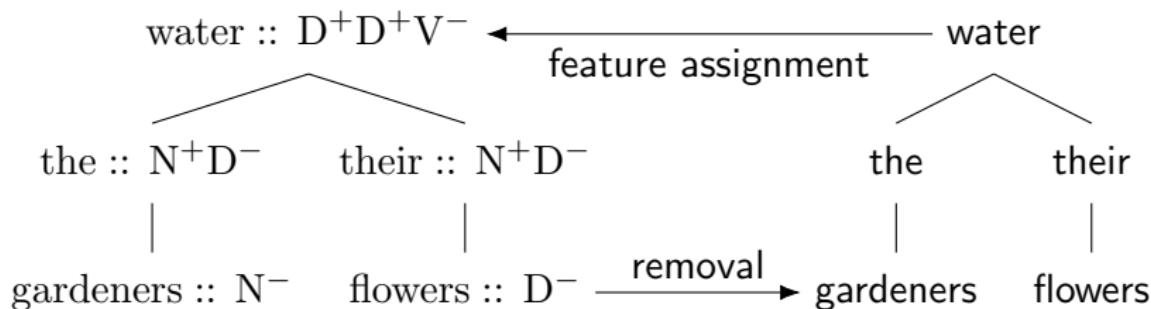
Lifting ISL relabelings to trees

String contexts as tree contexts



Reminder: ISL for feature inference

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

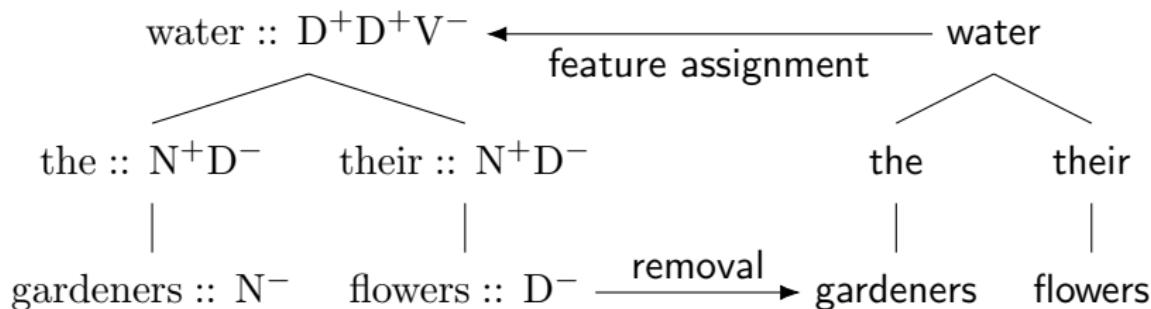


Local feature recoverability

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

$\varepsilon :: O^+ C^-$

|

bar :: $E^+ O^-$

|

foo :: $O^+ E^-$

|

bar :: O^-

Modulo counting is not ISL recoverable

$\varepsilon :: O^+ C^-$	ε
	⋮
bar :: E ⁺ O ⁻	⋮
foo :: O ⁺ E ⁻	bar
bar :: O ⁻	bar
bar	foo
bar	bar
bar	bar
⋮	⋮
	bar

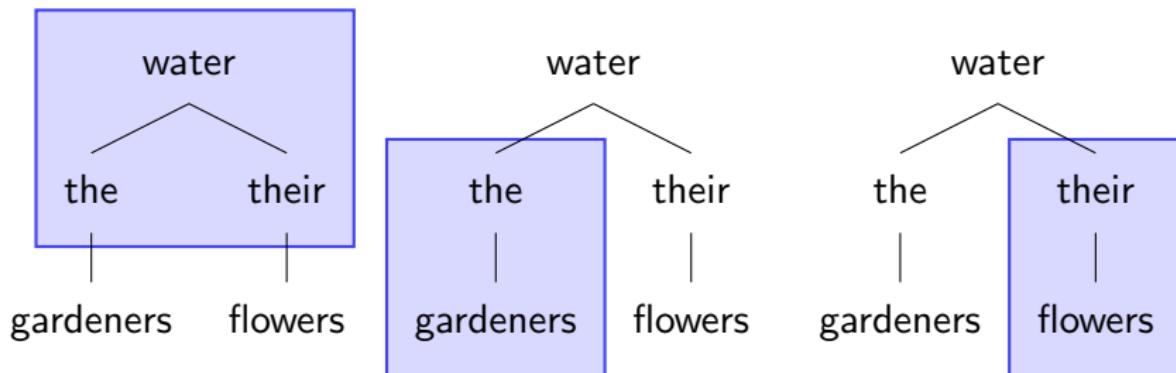
- ▶ Can you determine the features of **foo**?
 - 1 O⁺ E⁻
 - 2 E⁺ O⁻
- ▶ No, that's impossible.
- ▶ You need more than local information.
- ▶ Modulo counting is not ISL recoverable.

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!

Towards a learning algorithm for Minimalism

- ▶ Categories are a major hurdle for syntactic learning algorithms.
- ▶ Feature recoverability opens up a new strategy.

A learning paradigm for Minimalist syntax

1 Input

- ▶ string (observed)
- ▶ head-argument relations (basic semantic interpretation)
- ▶ notion of feature recoverability (UG)

2 Construct feature-free dependency tree

3 Distributional learning of categories via ISL recoverability (state merging)

4 Infer movement from string

Conclusion

- ▶ Subcategorization: major loop hole in syntactic formalisms
- ▶ Instead of substantive restrictions on categories,
we can use a formal restriction: **ISL recoverability**.
- ▶ Rules out majority of unnatural constraints/category systems
- ▶ Raises empirical questions about choice of representation

To be done

- ▶ test ISL recoverability with MGbank corpus (Torr 2017)
- ▶ how to assign Move features

Acknowledgments

Thanks to Peter Svenonius and Craig Sailor for inviting me to the Tromsø feature workshop that gave me the inspiration for this work.

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



References |

- Aksënova, Alëna, Thomas Graf, and Sedigheh Moradi. 2016. Morphotactics as tier-based strictly local dependencies. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 121–130. URL <https://www.aclweb.org/anthology/W/W16/W16-2019.pdf>.
- 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>.
- Graf, Thomas. 2018. Why movement comes for free once you have adjunction. In *Proceedings of CLS 53*, ed. Daniel Edmiston, Marina Ermolaeva, Emre Hakgüder, Jackie Lai, Kathryn Montemurro, Brandon Rhodes, Amara Sankhagowit, and Miachel Tabatowski, 117–136.

References II

- 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.
- Heinz, Jeffrey. 2009. On the role of locality in learning stress patterns. *Phonology* 26:303–351. URL <https://doi.org/10.1017/S0952675709990145>.
- Heinz, Jeffrey. 2010. Learning long-distance phonotactics. *Linguistic Inquiry* 41:623–661. URL http://dx.doi.org/10.1162/LING_a_00015.
- Heinz, Jeffrey. 2018. The computational nature of phonological generalizations. In *Phonological typology*, ed. Larry Hyman and Frank Plank, Phonetics and Phonology, chapter 5, 126–195. Mouton De Gruyter.
- Jardine, Adam. 2016. Computationally, tone is different. *Phonology* 33:247–283. URL <https://doi.org/10.1017/S0952675716000129>.
- 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 III

- McMullin, Kevin. 2016. *Tier-based locality in long-distance phonotactics: Learnability and typology*. Doctoral Dissertation, University of British Columbia.
- Shafiei, Nazila, and Thomas Graf. 2020. The subregular complexity of syntactic islands. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2020*. To appear.
- 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>.
- Torr, John. 2017. Autobank: a semi-automatic annotation tool for developing deep Minimalist grammar treebanks. In *Proceedings of the Demonstrations at the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 81–86.