

Syntactic tiers: Empirical applications and challenges

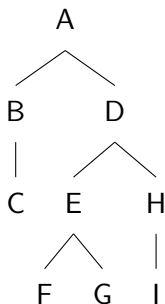
Thomas Graf

Stony Brook University
mail@thomasgraf.net

NYU Syntax Brown Bag
November 11, 2022

The talk in a nutshell

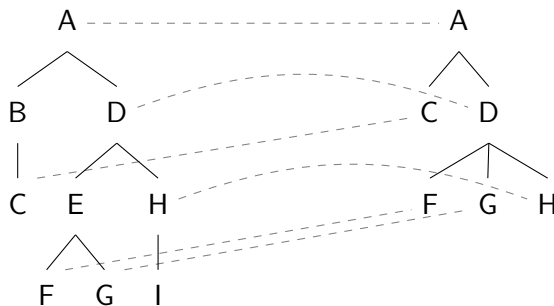
- Tiers, in syntax



- Can we use this for anything?
Way more than you'd think...!

The talk in a nutshell

- Tiers, in syntax



- Can we use this for anything?
Way more than you'd think...!

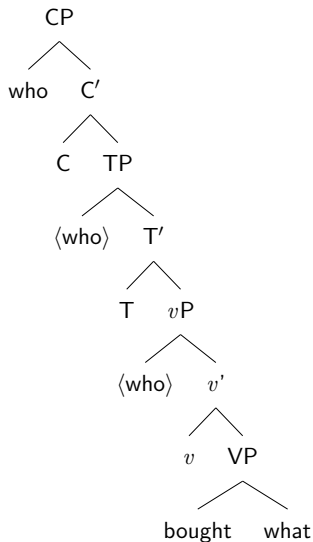
Applications

- ▶ Movement
- ▶ Multiple wh-movement
- ▶ Grafts/amalgams/reprojection
- ▶ Island effects (with exceptions and gradience)
- ▶ Extraction morphology
- ▶ complementizer alternations in Lio
- ▶ German wh-copying

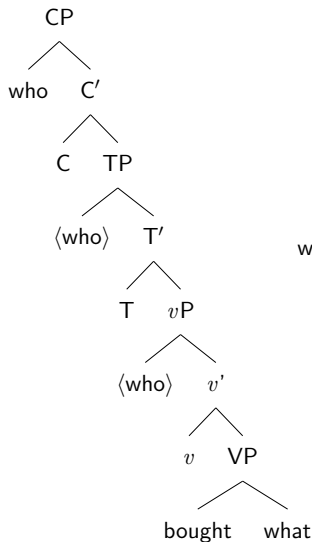
Outline

- 1 A primer on tree tiers
- 2 Mini-analyses to strengthen your intuition
 - Multiple wh-movement
 - Grafts/amalgams/reprojection
- 3 Islands
- 4 Extraction morphology
- 5 The limits of tiers

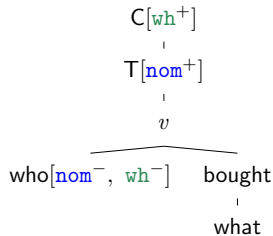
Derivations as dependency trees



Derivations as dependency trees

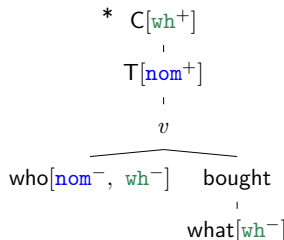


Derivation



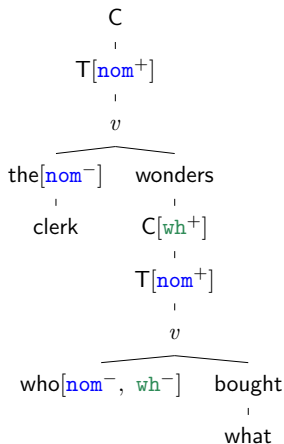
Some basic assumptions about movement

- ▶ wh^- = “you must wh -move”
- ▶ wh^+ = “something must wh -move to you”
- ▶ **Empirical fact**
Mover targets closest landing site of suitable type
- ▶ **Convenient assumption**
One-to-one match of movers and landing sites



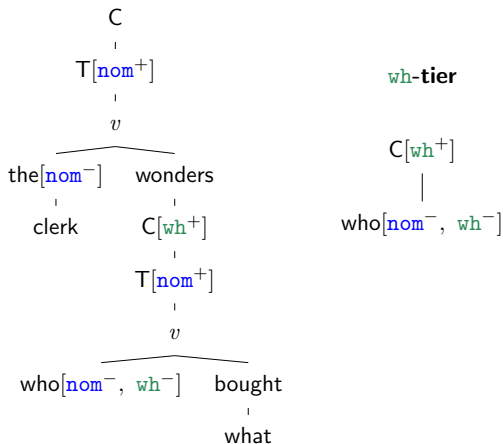
A cognitive/computational question about movement

- How hard is it to verify that all movement steps in a derivation meet these conditions?
- One particularly simple solution: tree tiers



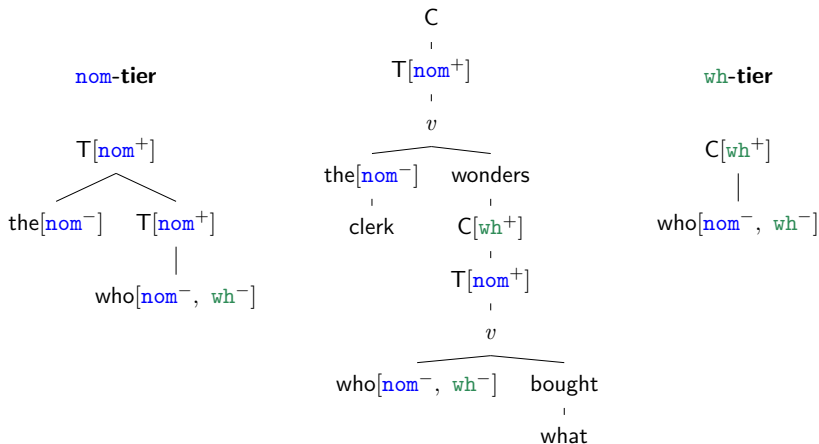
A cognitive/computational question about movement

- How hard is it to verify that all movement steps in a derivation meet these conditions?
- One particularly simple solution: tree tiers



A cognitive/computational question about movement

- ▶ How hard is it to verify that all movement steps in a derivation meet these conditions?
- ▶ One particularly simple solution: tree tiers



How tree tiers work

- ▶ finite number of tiers
- ▶ for each tier **T**:
 - ▶ whether node **N** projects to **T** depends only on **N** itself
features, phonetic exponent, ...
 - ▶ finite number of constraints on mother-daughter configurations

Example: Tiers for movement

- ▶ for every movement type f , construct an **f -tier**
- ▶ on each f -tier:
 - ▶ f^- must have f^+ mother
 - ▶ f^+ must have exactly one daughter with f^-

Um, why would we do this?

- ▶ **Philosophical answer**

We're not doing anything, this is just an inherent property of the computational nature of movement.

- ▶ **Computational answer**

This is a generalization of the formal class TSL from strings to trees, and TSL is awesome:

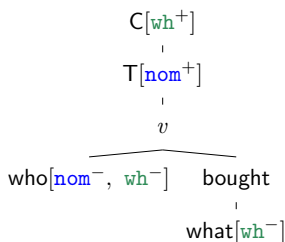
- ▶ very restricted typology
- ▶ easy to study
- ▶ efficient learning algorithms

- ▶ **Linguistic answer**

Tree tiers provide a **unified perspective** on many phenomena.

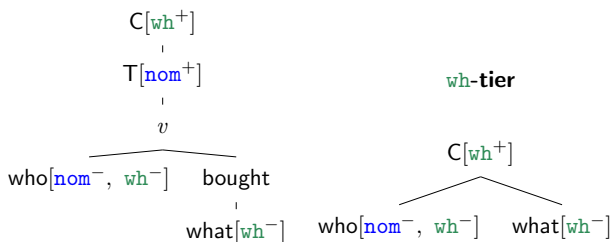
Multiple wh-movement

- ▶ Existence of multiple wh-movement unsurprising
- ▶ **Reason:** a computational device that can enforce “exactly 1” can also enforce “at least 1”



Multiple wh-movement

- Existence of multiple wh-movement unsurprising
- **Reason:** a computational device that can enforce “exactly 1” can also enforce “at least 1”



But that's barely an analysis...

That's a nice starting point, but we need to account for...

- ▶ why there is no multiple **nom**-movement
- ▶ linearization,
- ▶ superiority effects/c-command,
- ▶ multiple movers with distinct targets,
- ▶ the 2-by-2 typology of wh-movement

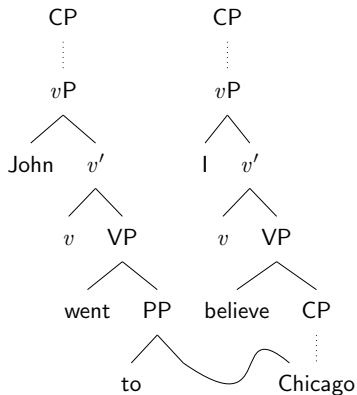
Multiple wh	Superiority	Example language
-	-	Spanish
-	+	English
+	-	Russian
+	+	Bulgarian

Grafts as long-distance selection (Van Riemsdijk)

- (1) John went to I believe it was Chicago.
- (2) I like [what you bought].

Grafts as long-distance selection (Van Riemsdijk)

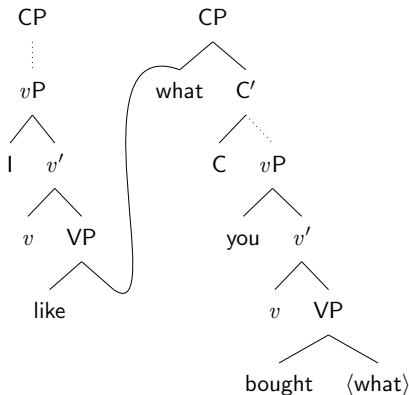
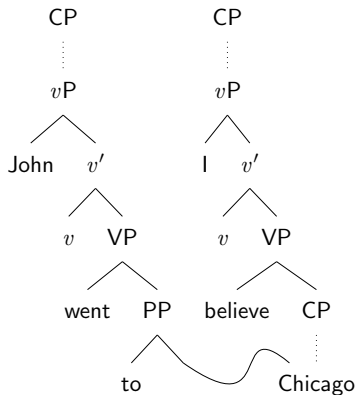
- (1) John went to I believe it was Chicago.
- (2) I like [what you bought].



Grafts as long-distance selection (Van Riemsdijk)

(1) John went to I believe it was Chicago.

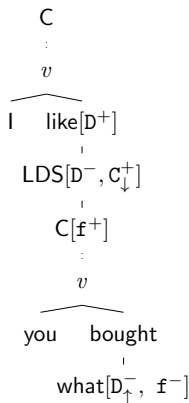
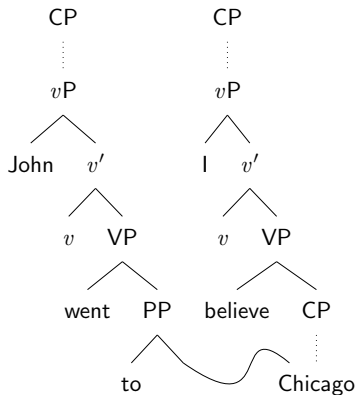
(2) I like [what you bought].



Grafts as long-distance selection (Van Riemsdijk)

(1) John went to I believe it was Chicago.

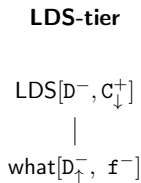
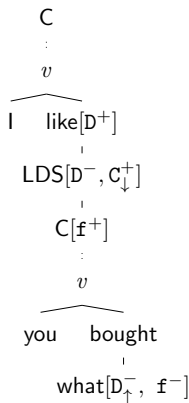
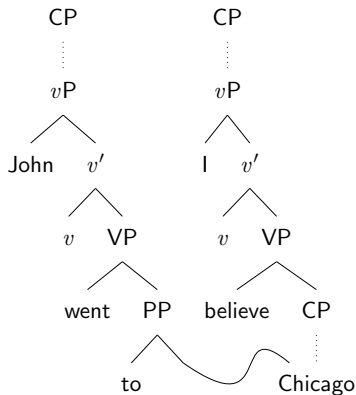
(2) I like [what you bought].



Grafts as long-distance selection (Van Riemsdijk)

(1) John went to I believe it was Chicago.

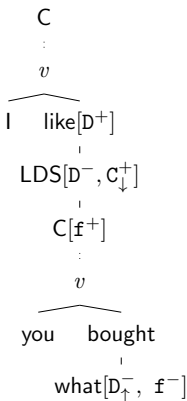
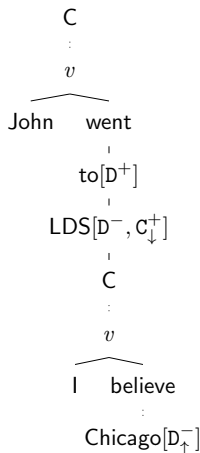
(2) I like [what you bought].



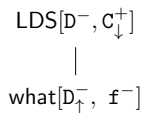
Grafts as long-distance selection (Van Riemsdijk)

(1) John went to I believe it was Chicago.

(2) I like [what you bought].



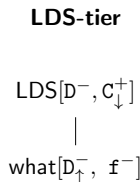
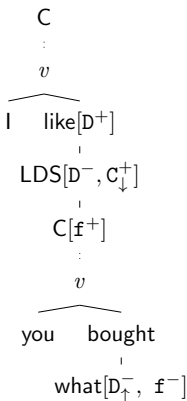
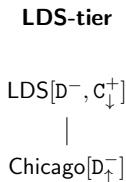
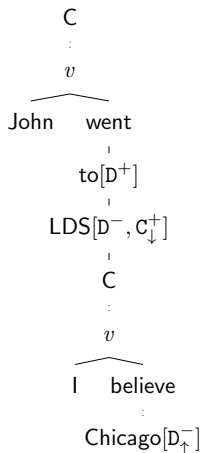
LDS-tier



Grafts as long-distance selection (Van Riemsdijk)

(1) John went to I believe it was Chicago.

(2) I like [what you bought].



But that's barely an analysis... again

- ▶ only select few CPs can be grafted

- (3) a. * John went to Al Capone ruled Chicago.
b. ?? John went to I established it was Chicago.

- ▶ only select few DPs can be LDS-ed within a graft

- (4) a. * I like you bought what.
(meaning *what you bought*)
b. * I like you slept.
(meaning *I like you & you slept*)

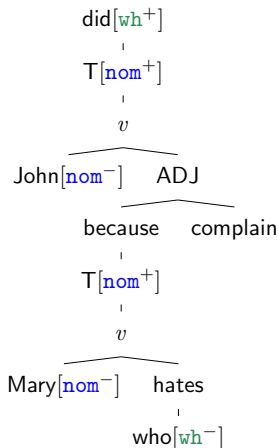
- ▶ grafts bridge locality domains

- (5) John nominated [_{CP} I believe [_{CP} it was himself]].

- ▶ grafts have other applications (Meinunger on numerals)

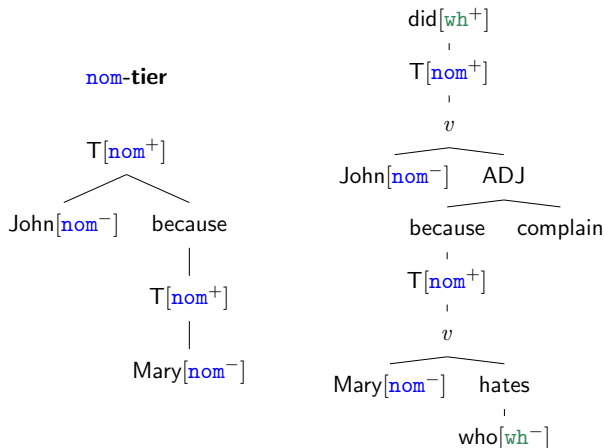
Islands

- ▶ (Strong) islands are easy to capture with tiers
- ▶ island = node that disrupts licensing on movement tier



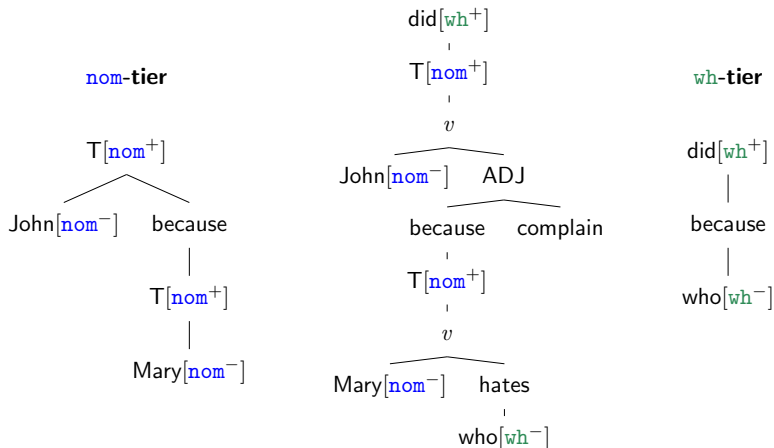
Islands

- ▶ (Strong) islands are easy to capture with tiers
- ▶ island = node that disrupts licensing on movement tier

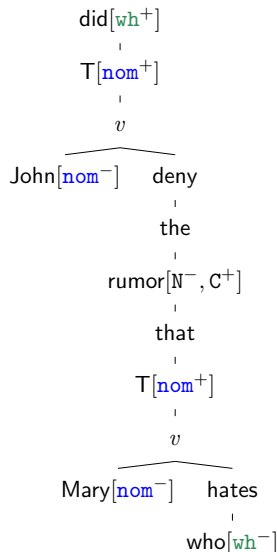


Islands

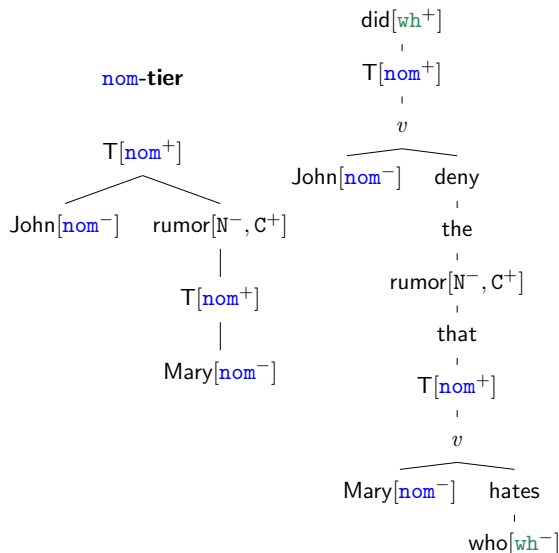
- (Strong) islands are easy to capture with tiers
- island = node that disrupts licensing on movement tier



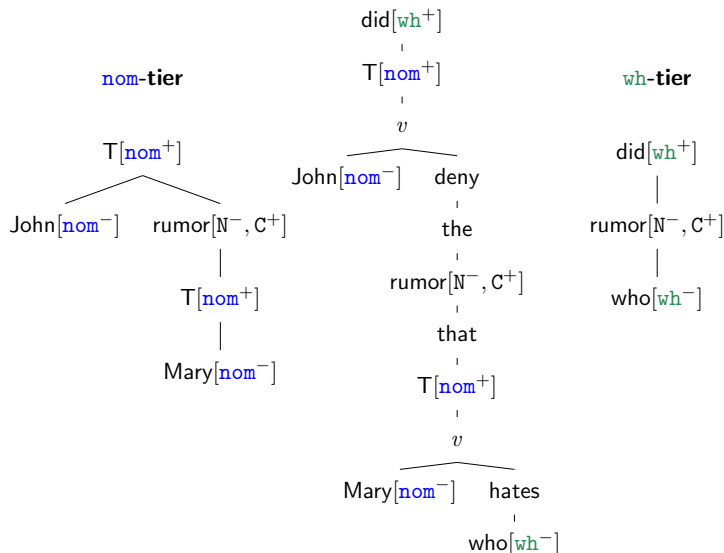
Another example: Complex NP constraint



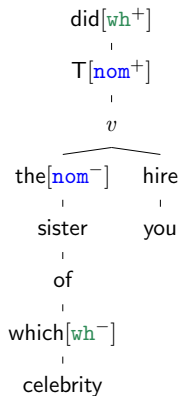
Another example: Complex NP constraint



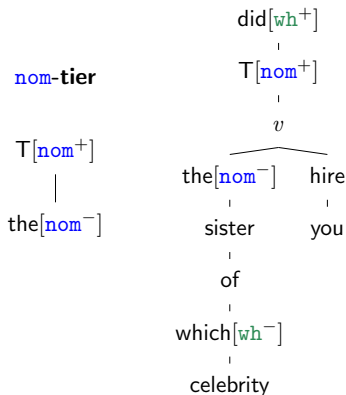
Another example: Complex NP constraint



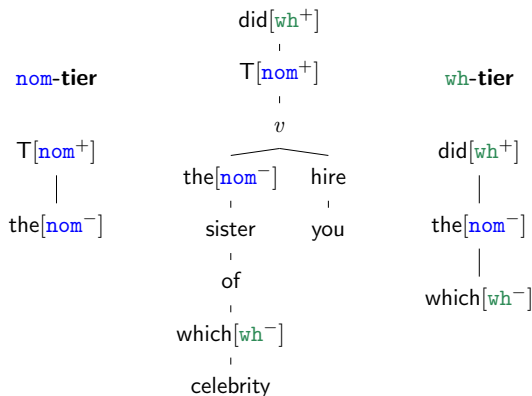
Another example: Subject island constraint



Another example: Subject island constraint



Another example: Subject island constraint



Disclaimer

- ▶ Yes, there's tons of **exceptions!**
(Chaves and Dery 2019) and references therein
- (6) Which president would [the impeachment of _] cause outrage?
- (7) What did [the attempt to find _] end in failure?
(sounds awful to me)
- (8) This is the man who [friends of _] think that [enemies of _] are everywhere.
- ▶ Tiers have some ways of dealing with exceptions, not clear yet if it works with these.

Constraints I've looked at so far

- 1 Adjunct islands: project adjuncts
- 2 Complex NP constraint: project items with N^- and C^+
- 3 Relative clause islands: project items with rel^+
- 4 Subject island constraint: project items with nom^-
- 5 *that*-trace and anti-*that*-trace: works, but it's complicated

Gradience

- Tier projection can be made probabilistic to capture gradience (Mayer 2021)

Calculating gradience with probabilistic wh-tier

- 1 construct all possible versions of the wh-tier
- 2 filter out illicit tiers
- 3 sum up probabilities of remaining tiers

Example calculation

- (9) * Who did Bill complain after having a meeting with
⟨who⟩ because he's always angry.

Item	Probability
any wh^+	1
any wh^-	1
because	.9
after	.8

► Possible tiers

- 1 Project wh^+ , wh^- , because, after: .72, illicit
 - 2 Project wh^+ , wh^- , because: .18, licit
 - 3 Project wh^+ , wh^- , after: .08, illicit
 - 4 Project wh^+ , wh^- : .02, licit
- Overall probability: .2

Ongoing project



Charlie Torres
UCI



Connor Mayer
UCI

Kenneth Hanson
SBU

(Too) Many degrees of freedom

► Three predictions

- 1 Island constraints can vary across languages.
- 2 Islands may appear on some but not all movement tiers.
- 3 There may be lexical exceptions to what projects on what tier.

► This is a **boon**!

Consider recent data from Norwegian (Bondevik et al. 2021), where extraction is

- allowed from finite adjunct clauses with *om* 'if'
- forbidden from finite adjunct clauses with *når* 'when' or *fordi* 'because'

► This is a **bane**!

- Typological overgeneration
- Why are island effects very systematic overall?

Summary: Islands

- ▶ Computational mechanisms behind movement also allow for islands
- ▶ **Flexible**
 - ▶ covers many common island phenomena
 - ▶ gradience easy to add if desired
 - ▶ allows for lexical exceptions
- ▶ **Too flexible**
needs to be combined with some other theory of islandhood

Extraction morphology: An example from Wolof u-chains

Extraction morphology morphology conditioned by the presence of movement

- ▶ Consider **Wolof u-chains** (Niger-Congo)
 - ▶ (covert) wh-phrase moves to matrix Spec,CP
 - ▶ highest C-head *u* along the movement path must agree with wh-phrase in Class
 - ▶ intermediate C-heads may agree with wh-phrase in Class

- (10) [ε **k-u** Kumba wax [ne **k/l-u** Isaa defe [
 Q CL-u Kumba say FRC CL/EXPL-u Isaa think
 ne **k/l-u** Maryam dóór *t*]]]
 FRC CL/EXPL-u Maryam hit
 ‘Who did Kumba say that Isaa thought that Maryam hit?’
 (Torrence 2012:1171)

A typology of extraction morphology

based on Georgi (2017), a terrific paper

1 What is agreeing?

- L landing site (Wolof u-chains)
- X some other head X in the clause of the landing site (Duala *no*-marking on T)

2 What clauses display agreement?

- F the clause of the **F**inal landing site (Chamorro complementizer agreement)
- N clauses with **N**on-final landing sites (Kiitharaka focus marking on verb)

3 agreement is. . .

- M **M**andatory
- O **O**ptional

Example

Wolof u-chains are L-F_M-N_O

Extraction morphology and tiers

- ▶ Tiers work well for extraction morphology.
- ▶ the various options fall out nicely for L-patterns
- ▶ X-patterns are harder

Central idea

- ▶ treat it as a distribution problem (not mapping/spell-out)
- ▶ use tiers to ensure correct distribution of agreeing forms

Two simplifications due to data gaps

- ▶ exactly one A'-mover per sentence
- ▶ only one type of A'-movement per sentence
(no *which car that John bought did we trash*)

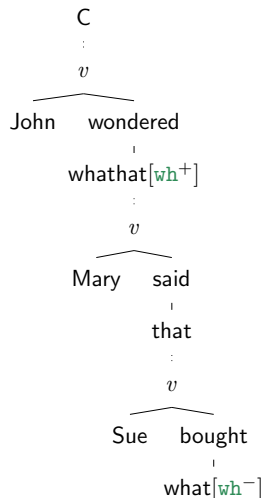
Toy example 1: F_M English *whathat*

- ▶ Suppose English had a special C-head *whathat*, which has to be the final landing site of some *wh*-mover.

Analysis: Lexical accident

The lexicon happens to be such that

- ▶ *whathat* must carry *wh*⁺,
- ▶ *that* must not carry *wh*⁺,
- ▶ empty C must not carry *wh*⁺.

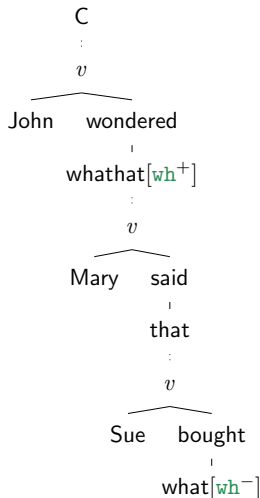


Toy example 2: F_M English *whathat/whothat*

- Suppose English also had *whothat*, and *Xthat* must agree in animacy with the *wh*⁺-mover.

Lexical accident + tier match

- wh*⁺ only on *whathat* and *whothat*
- on *wh*-tier, *wh*⁺-head must match animacy of *wh*⁻-daughter



Toy example 2: F_M English *whathat/whothat*

- ▶ Suppose English also had *whothat*, and *Xthat* must agree in animacy with the *wh*-mover.

Lexical accident + tier match

- ▶ *wh*⁺ only on *whathat* and *whothat*
- ▶ on *wh*-tier, *wh*⁺-head must match animacy of *wh*⁻-daughter

wh-tier

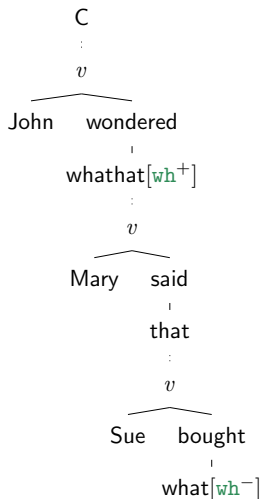
whathat[*wh*⁺]
 |
 what[*wh*⁻]

Toy example 3: F_M - N_M English *whathat/whothat*

- Suppose every C-head along a *wh*-movement path has to be *whathat/whothat*

Lexical accident + tier match + N-tier

- wh*⁺ only on *whathat* and *whothat*
- on *wh*-tier, *wh*⁺-head must match animacy of *wh*⁻-daughter
- N-tier**
 - project all *wh*⁺, all *wh*⁻, and all C-heads
 - every *Xthat* must have daughter that is *Xthat* (same X) or *wh*⁻
 - non-*Xthat* with *Xthat* daughter, daughter must carry *wh*⁺



Toy example 3: F_M - N_M English *wh**that*/*who**that*

- ▶ Suppose every C-head along a *wh*-movement path has to be *wh**that*/*who**that*

Lexical accident + tier match + N-tier

- ▶ *wh*⁺ only on *wh**that* and *who**that*
- ▶ on *wh*-tier, *wh*⁺-head must match animacy of *wh*⁻-daughter
- ▶ **N-tier**
 - ▶ project all *wh*⁺, all *wh*⁻, and all C-heads
 - ▶ every *Xthat* must have daughter that is *Xthat* (same X) or *wh*⁻
 - ▶ non-*Xthat* with *Xthat* daughter, daughter must carry *wh*⁺

wh-tier

*wh**that*[*wh*⁺]
 |
what[*wh*⁻]

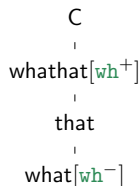
Toy example 3: F_M - N_M English *wh**that*/*wh**that*

- ▶ Suppose every C-head along a *wh*-movement path has to be *wh**that*/*wh**that*

Lexical accident + tier match + N-tier

- ▶ *wh*⁺ only on *wh**that* and *wh**that*
- ▶ on *wh*-tier, *wh*⁺-head must match animacy of *wh*⁻-daughter
- ▶ **N-tier**
 - ▶ project all *wh*⁺, all *wh*⁻, and all C-heads
 - ▶ every *Xthat* must have daughter that is *Xthat* (same X) or *wh*⁻
 - ▶ non-*Xthat* with *Xthat* daughter, daughter must carry *wh*⁺

N-tier

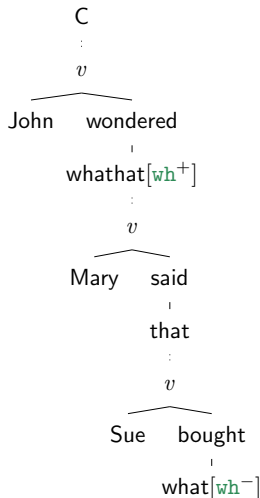


Toy example 4: F_0 - N_0 English *wh*that/*wh*that

- Suppose as before, except that *X*that is optional

No accident + fused(tier match, N-tier)

- wh^+ on *X*that **or that**
- **N-tier**
 - project all wh^+ , all wh^- , and all C-heads **that are Xthat**
 - every *X*that must have daughter that is *X*that or **matching** wh^-
 - non-*X*that with *X*that daughter, daughter must carry wh^+



Toy example 4: F_0 - N_0 English *wh**that*/*wh**that*

- ▶ Suppose as before, except that *Xthat* is optional

No accident + fused(tier match, N-tier)

- ▶ wh^+ on *Xthat* **or that**
- ▶ **N-tier**
 - ▶ project all wh^+ , all wh^- , and all C-heads **that are Xthat**
 - ▶ every *Xthat* must have daughter that is *Xthat* or **matching** wh^-
 - ▶ non-*Xthat* with *Xthat* daughter, daughter must carry wh^+

N-tier

wh*that* $[wh^+]$

|
what $[wh^-]$

Summary of analytical tricks for L-patterns

- ▶ **F-patterns arise from lexicon**
 - ▶ no F: wh^+ only on default C
 - ▶ F_M : wh^+ only on agreeing C
 - ▶ F_O : wh^+ on either one
- ▶ **N-patterns captured via tiers**
 - ▶ construct “daisy chain” of agreeing mother-daughter configurations
 - ▶ top of daisy chain: f^+
 - ▶ bottom of daisy chain: f^-
- ▶ If we need agreement with some feature of the mover, find a suitable tier to enforce the match condition on.

X patterns

- ▶ X patterns are more convoluted, even in simple cases.
- ▶ $L-F_M$ is trivial, $X-F_M$ is challenging

Example: $X-F_M$ on verb with wh -movement

- ▶ Project all wh^+ , all wh^- , and all V
- ▶ inflected V must have wh^+ mother

Example: agreeing $X-F_M$ on verb with wh -movement

- ▶ everything as before, plus a new tier
- ▶ Project all wh^+ , all wh^- , and all inflected V
- ▶ inflected V must have matching wh^- daughter

X patterns [cont.]

Example: agreeing $X-F_M-N_O$ on verb with wh -movement

- ▶ **Tier 1:** F_M
 - ▶ Project all wh^+ , all wh^- , and all V
 - ▶ wh^+ must have inflected V daughter
- ▶ **Tier 2:** N_O
 - ▶ Project all wh^+ , all wh^- , and all inflected V
 - ▶ inflected V must have matching daughter that is wh^- or inflected V

Problems

- ▶ Some patterns not doable without extensions ($X-F_O-N_M$)
- ▶ Sometimes target of extraction morphology does not sit above the mover (e.g. V with moving subject)

Summary: Extraction morphology

- ▶ rich, varied typology
- ▶ L-patterns work very well and require no extra machinery
- ▶ X-patterns get tricky, some require more than just mother-daughter over tiers ($X-F_O-N_M$, $X-N_M$)
- ▶ But there's a clear path forward to capturing the full typology.

Speaking of problems. . .

- ▶ There is a lot of movement stuff that does not work.
smuggling, freezing, ATB, CSC, . . .

Example: Smuggling

- ▶ We always calculate locality from a mover's base position.
 - ▶ Being contained inside another mover does not alter locality.
-
- ▶ Ideally, the things that don't work are **productive failures**.

Example: Ban on Improper Movement (BoIM)

BoIM Once you A'-move, you cannot A-move anymore

(11) * John seems [_{CP} ⟨John⟩ [_{TP} ⟨John⟩ snores a lot]].

- ▶ We cannot use tiers to correctly enforce BoIM.
- ▶ **But:** tiers can handle an alternative, **probe horizons** (Keine 2016)

Modeling probe horizons with tiers

Probe horizon

[uEPP] cannot probe past C

[uF] cannot probe past T

[uwh] can probe into anything

Tier counterpart

all C-heads on **nom**-tier

all T-heads on **f**-tier

normal **wh**-tier

A cute observation

- ▶ A version of BoIM is needed for tier-based linearization.
- ▶ Perhaps a lot of tier-based stuff is the low-level implementation of more general principles.

High-level principles VS grammar implementation

- ▶ **PF interface** says: “Syntax, the PF interface congress just passed the BoIM bill to boost efficient linearization.”
- ▶ **Syntax** sighs: “Oh god, another set of super abstract regulations. But whatever, I’ll put in place a few simple procedures that ensure we’re in compliance. Hmm, probe horizons should do the trick.”

Conclusion

- ▶ Movement recently analyzed as local dependency over tiers (Graf 2018; Graf and Kostyszyn 2021)
- ▶ In fact, tiers provide a unified mechanism for
 - 1 multiple wh-movement (preliminary),
 - 2 grafts/long-distance selection (preliminary)
 - 3 islands (adjuncts, complex NPs, relative clauses, ...)
 - 4 *that*-trace effect and its exceptions
 - 5 anti-*that*-trace effect
 - 6 extraction morphology (some X-patterns are tricky)
 - 7 complementizer alternations in Lio (preliminary)
 - 8 German wh-copying (preliminary)

Take-Home Message

Tiers aren't just for phonologists anymore!

Acknowledgments

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



References I

- Bondevik, Ingrid, Dave Kush, and Terje Lohndal. 2021. Variation in adjunct islands: The case of Norwegian. *Nordic Journal of Linguistics* 44:223–254.
- Chaves, Rui P., and Jeruen E. Dery. 2019. Frequency effects in subject islands. *Journal of Linguistics* 55:475–521.
- Georgi, Doreen. 2017. Patterns of movement reflexes as the result of the order of merge and agree. *Linguistic Inquiry* 48:585–626.
- 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.
- Graf, Thomas, and Kalina Kostyszyn. 2021. Multiple wh-movement is not special: The subregular complexity of persistent features in Minimalist grammars. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2021*, 275–285.
- Keine, Stefan. 2016. *Probes and their horizons*. Doctoral Dissertation, University of Massachusetts, Amherst.
- Mayer, Connor. 2021. Capturing gradience in long-distance phonology using probabilistic tier-based strictly local grammars. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2021*, 39–50.
- Torrence, Harold. 2012. The morpho-syntax of silent *wh*-expressions in wolof. *Natural Language and Linguistic Theory* 30:1147–1184.