Graph Transductions and Typological Gaps in Morphological Paradigms

Thomas Graf

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

MOL 2017
July 13–14, 2017
Prelude: So Many Boring Problems

- Theoretical linguists obsess about many problems that are boring to mathematical linguists.

Example: Person Case Constraint (PCC; Bonet 1994)

The well-formedness of clitic combinations is contingent on their person specification.

(1) Roger le/*me leur a présenté.
Roger 3SG.ACC/1SG.ACC 3PL.DAT has shown
‘Roger has shown me/him to them.’

- The existence of the PCC is unremarkable.
  - captured by bigram model (very simple)
  - small problem space ⇒ no learnability issues
Take-Home Message: Boring = Interesting At Close-Up

- Boring problems are interesting once we take a closer look.

Why the PCC is Interesting

- Out of 64 conceivable PCC variants, only 4 are attested.
- The attested PCCs form a mathematically natural class.
- And the mathematical account extends to seemingly unrelated phenomena in morphosyntax.

- Moral: We should study all linguistic phenomena, not just the usual suspects.
Morphosyntactic phenomena can be given a natural explanation via **three components:**

1. an independently motivated base hierarchy
   - person, number, adjectival gradation, . . . 
2. maximally simple graph transductions to modify this hierarchy
3. a simple interpretation of the output graphs
Outline

1. The *ABA Generalization & Monotonicity

2. *ABA Revisited: Graph-Theoretic Approach
   - Application to Pronoun Syncretism
   - Beyond 3-Cell Systems

3. Person Case Constraint

4. Subregularity of Weakly Non-Inverting Graph Mappings
A Case Study: *ABA in Morphological Paradigms

Syntretism multiple forms built from same base

*ABA Generalization (Bobaljik 2012)

Two paradigmatic cells cannot be syncretic to the exclusion of any intervening cell.

Example: Adjectival Gradation

(2) a. smart, smarter, smartest (AAA)
    b. good, better, best (ABB)
    c. *good, better, goodest (ABA)
*ABA Across Morphological Paradigms

Example: Pronoun Syncretism (Harbour 2015, 2016)

(3) a. mi, ni, ehi (ABC)  
    b. n!aa, n!uu, n!uu (ABB)  
    c. ne, ne, e (AAB)  
    d. *I, you, I (ABA)  

Example: Case Syncretism in Russian (Caha 2009)

<table>
<thead>
<tr>
<th>Case</th>
<th>window.Sg</th>
<th>teacher.Pl</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nom</td>
<td>okn-o</td>
<td>ucitel-ja</td>
<td>st-o</td>
</tr>
<tr>
<td>Acc</td>
<td>okn-o</td>
<td>ucitel-ej</td>
<td>st-o</td>
</tr>
<tr>
<td>Gen</td>
<td>okn-a</td>
<td>ucitel-ej</td>
<td>st-a</td>
</tr>
<tr>
<td>Dat</td>
<td>okn-u</td>
<td>ucitel-jam</td>
<td>st-a</td>
</tr>
<tr>
<td>Inst</td>
<td>okn-om</td>
<td>ucitel-ami</td>
<td>st-a</td>
</tr>
</tbody>
</table>
*ABA Across Morphological Paradigms

Example: Pronoun Syncretism (Harbour 2015, 2016)

(3) a. mi, ni, ehi (ABC)  
    b. n!aa, n!uu, n!uu (ABB)  
    c. ne, ne, e (AAB)  
    d. *I, you, I (ABA)

Example: Case Syncretism in Russian (Caha 2009)

<table>
<thead>
<tr>
<th>Case</th>
<th>window.Sg</th>
<th>teacher.Pl</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nom</td>
<td>okn-o</td>
<td>ucitel-ja</td>
<td>st-o</td>
</tr>
<tr>
<td>Acc</td>
<td>okn-o</td>
<td>ucitel-ej</td>
<td>st-o</td>
</tr>
<tr>
<td>Gen</td>
<td>okn-a</td>
<td>ucitel-ej</td>
<td>st-a</td>
</tr>
<tr>
<td>Dat</td>
<td>okn-u</td>
<td>ucitel-jam</td>
<td>st-a</td>
</tr>
<tr>
<td>Inst</td>
<td>okn-om</td>
<td>ucitel-ami</td>
<td>st-a</td>
</tr>
</tbody>
</table>
ABA: A First Account

- A mapping that produces ABA violates monotonicity.

### Monotonicity for Pronoun Syncretism

- Suppose $3 < 2 < 1$ (Zwicky 1977)
- A function $f$ is monotonic iff $x \leq y$ implies $f(x) \leq f(y)$.
- No monotonic function from $\{1, 2, 3\}$ to $\{A, B, C\}$ can produce ABA!
- This holds irrespective of the ordering of $\{A, B, C\}$. 
Monotonicity is similar to No Crossing Branches constraint in autosegmental phonology. (Goldsmith 1976)

Patterns:

1 2 3

A B C
Monotonicity is similar to No Crossing Branches constraint in autosegmental phonology. (Goldsmith 1976)

Patterns:
Illustrating Monotonicity

Monotonicity is similar to No Crossing Branches constraint in autosegmental phonology. (Goldsmith 1976)

Patterns: ABC,
Illustrating Monotonicity

Monotonicity is similar to No Crossing Branches constraint in autosegmental phonology. (Goldsmith 1976)

Patterns: ABC,
Illustrating Monotonicity

Monotonicity is similar to No Crossing Branches constraint in autosegmental phonology. (Goldsmith 1976)

Patterns: ABC, AAB = AAC,
Illustrating Monotonicity

Monotonicity is similar to No Crossing Branches constraint in autosegmental phonology. (Goldsmith 1976)

Patterns: ABC, AAB = AAC,
Monotonicity is similar to No Crossing Branches constraint in autosegmental phonology. (Goldsmith 1976)

Patterns: ABC, AAB = AAC, ABB = ACC,
Illustrating Monotonicity

Monotonicity is similar to No Crossing Branches constraint in autosegmental phonology. (Goldsmith 1976)

Patterns: ABC, AAB = AAC, ABB = ACC,
Illustrating Monotonicity

Monotonicity is similar to No Crossing Branches constraint in autosegmental phonology. (Goldsmith 1976)

Patterns: ABC, AAB = AAC, ABB = ACC, AAA

1 — 2 — 3
A — B — C
Illustrating Monotonicity

Monotonicity is similar to No Crossing Branches constraint in autosegmental phonology. (Goldsmith 1976)

Patterns: ABC, AAB = AAC, ABB = ACC, AAA
Why Monotonicity?

- Why should spell-out functions be monotonic?
- **Idea:** Monotonicity matters in other areas.
  - NPI licensing in downward entailing contexts
  - Direction-preserving nature of movement in MGs
- **But:** Those are just-so stories.
  - Downward entailingness is neither necessary nor sufficient.
  - Various MG movement types are not direction-presevering.
- Maybe monotonicity is not the best characterization...
## A More General View: Graph Structure Preservation

### The General Idea

- *ABA is about structure preservation.*
- Syncretism is modification of a base graph.
- Modification must not contradict orderings of base graph.

### Definition (Weakly Non-Inverting Graph Mappings)

- Given input graph $G$ and output graph $G'$
  - $x \triangleleft y$ iff $y$ is reachable from $x$ in $G$, 
  - $x \blacktriangleleft y$ iff $y$ is reachable from $x$ in $G'$.
- A mapping from $G$ to $G'$ is **weakly non-inverting** iff $x \triangleleft y \land y \blacktriangleleft x \rightarrow x \blacktriangleleft y$
Since we want graphs to encode hierarchies, they must be \textit{weakly connected}: ignoring the direction of arrows, all nodes are mutually reachable.

And the mapping must be weakly non-inverting:
\[
x \triangleleft y \land y \triangleright x \rightarrow x \triangleright y
\]
Weakly Non-Inverting Graph Mappings

- Since we want graphs to encode hierarchies, they must be *weakly connected*: ignoring the direction of arrows, all nodes are mutually reachable.
- And the mapping must be weakly non-inverting:
  \[ x \prec y \land y \preceq x \rightarrow x \preceq y \]

![Diagram of weakly non-inverting graph mappings]
Weakly Non-Inverting Graph Mappings

- Since we want graphs to encode hierarchies, they must be \textit{weakly connected}: ignoring the direction of arrows, all nodes are mutually reachable.
- And the mapping must be weakly non-inverting:
  \[ x \triangleleft y \land y \triangleright x \rightarrow x \triangleright y \]
Since we want graphs to encode hierarchies, they must be weakly connected: ignoring the direction of arrows, all nodes are mutually reachable.

And the mapping must be weakly non-inverting:

\[ x \triangleleft y \land y \triangleright x \rightarrow x \triangleright y \]
Since we want graphs to encode hierarchies, they must be \textit{weakly connected}: ignoring the direction of arrows, all nodes are mutually reachable.

And the mapping must be weakly non-inverting:

\[ x \prec y \land y \succ x \rightarrow x \succ y \]
Weakly Non-Inverting Graph Mappings

- Since we want graphs to encode hierarchies, they must be *weakly connected*: ignoring the direction of arrows, all nodes are mutually reachable.
- And the mapping must be weakly non-inverting:
  \[ x \bowtie y \land y \leq x \rightarrow x \bowtie y \]
Since we want graphs to encode hierarchies, they must be \textit{weakly connected}: ignoring the direction of arrows, all nodes are mutually reachable.

And the mapping must be weakly non-inverting:

\[ x \triangleleft y \land y \triangleright x \rightarrow x \triangleright y \]
Weakly Non-Inverting Graph Mappings

- Since we want graphs to encode hierarchies, they must be weakly connected: ignoring the direction of arrows, all nodes are mutually reachable.
- And the mapping must be weakly non-inverting: \( x \triangleleft y \land y \triangleright x \rightarrow x \triangleright y \)

\[ \begin{array}{c}
1 \quad 2 \quad 3 \\
1 \quad 2 \\
3 \\
\end{array} \]
Weakly Non-Inverting Graph Mappings

- Since we want graphs to encode hierarchies, they must be weakly connected: ignoring the direction of arrows, all nodes are mutually reachable.

- And the mapping must be weakly non-inverting:
  \[ x \prec y \land y \succ x \rightarrow x \equiv y \]
Weakly Non-Inverting Graph Mappings

- Since we want graphs to encode hierarchies, they must be weakly connected: ignoring the direction of arrows, all nodes are mutually reachable.
- And the mapping must be weakly non-inverting:
  \[ x \triangleleft y \land y \triangleright x \rightarrow x \triangleright y \]
Weakly Non-Inverting Graph Mappings

- Since we want graphs to encode hierarchies, they must be *weakly connected*: ignoring the direction of arrows, all nodes are mutually reachable.
- And the mapping must be weakly non-inverting:
  \[ x \triangleleft y \land y \triangleright x \rightarrow x \triangleright y \]
Weakly Non-Inverting Graph Mappings

- Since we want graphs to encode hierarchies, they must be *weakly connected*: ignoring the direction of arrows, all nodes are mutually reachable.
- And the mapping must be weakly non-inverting: $x \prec y \land y \triangleleft x \rightarrow x \triangleright y$
Weakly Non-Inverting Graph Mappings

- Since we want graphs to encode hierarchies, they must be *weakly connected*: ignoring the direction of arrows, all nodes are mutually reachable.

- And the mapping must be weakly non-inverting:
  \( x \triangleleft y \land y \triangleright x \rightarrow x \triangleright y \)
Since we want graphs to encode hierarchies, they must be \textit{weakly connected}: ignoring the direction of arrows, all nodes are mutually reachable.

And the mapping must be weakly non-inverting:

\[ x \triangleleft y \land y \triangleright x \rightarrow x \triangleright y \]
Since we want graphs to encode hierarchies, they must be weakly connected: ignoring the direction of arrows, all nodes are mutually reachable.

And the mapping must be weakly non-inverting:

\[ x \triangleleft y \land y \triangleright x \rightarrow x \triangleright y \]
Weakly Non-Inverting Graph Mappings

- Since we want graphs to encode hierarchies, they must be *weakly connected*: ignoring the direction of arrows, all nodes are mutually reachable.
- And the mapping must be weakly non-inverting:

\[ x \triangleleft y \land y \triangleright x \rightarrow x \triangleright y \]
Graphs and Syncretism

- Suppose two cells may be syncretic iff they are mutually reachable in a graph.
- Then the previous set of graphs describes the **class of attested syncretisms**.
Graphs and Syncretism

- Suppose two cells may be syncretic iff they are mutually reachable in a graph.
- Then the previous set of graphs describes the **class of attested syncretisms**.

![Graphs showing syncretic relationships](image-url)
Graphs and Syncretism

- Suppose two cells may be syncretic iff they are mutually reachable in a graph.
- Then the previous set of graphs describes the **class of attested syncretisms**.
Graphs and Syncretism

- Suppose two cells may be syncretic iff they are mutually reachable in a graph.
- Then the previous set of graphs describes the class of attested syncretisms.
Graphs and Syncretism

- Suppose two cells may be syncretic iff they are mutually reachable in a graph.
- Then the previous set of graphs describes the class of attested syncretisms.
Graphs and Syncretism

- Suppose two cells may be syncretic iff they are mutually reachable in a graph.
- Then the previous set of graphs describes the **class of attested syncretisms**.

![Graphs and Syncretism Diagram](image)
Graphs and Syncretism

- Suppose two cells may be syncretic iff they are mutually reachable in a graph.
- Then the previous set of graphs describes the class of attested syncretisms.
Scaling to Larger Systems

- The previous account works for any 3-cell paradigm.
- Some morphosyntactic phenomena have many different cells.
  - case syncretism, noun stem allomorphy
- For those, weakly non-inverting maps incorrectly allow ABA!
Scaling to Larger Systems

- The previous account works for any 3-cell paradigm.
- Some morphosyntactic phenomena have many different cells. case syncretism, noun stem allomorphy
- For those, weakly non-inverting maps **incorrectly allow ABA!**

![Graph showing the 3-cell paradigm with labels Nom, Acc, Gen, Dat, Inst, Misc connected in a cycle.

Nom -> Acc -> Gen -> Dat -> Inst -> Misc]
Scaling to Larger Systems

- The previous account works for any 3-cell paradigm.
- Some morphosyntactic phenomena have many different cells. case syncretism, noun stem allomorphy
- For those, weakly non-inverting maps **incorrectly allow ABA!**
Scaling to Larger Systems

- The previous account works for any 3-cell paradigm.
- Some morphosyntactic phenomena have many different cells. 
  case syncretism, noun stem allomorphy
- For those, weakly non-inverting maps *incorrectly allow ABA*!
The Fix: A Stronger Connectivity Requirement

- Weakly non-inverting maps still obey \( *ABA \) if output graphs must be connected:

\[
\forall x, y [ x \triangleright y \lor y \triangleright x ]
\]

- We can also assume this for 3-cell paradigms.
The Fix: A Stronger Connectivity Requirement

- Weakly non-inverting maps still obey $^\ast$ABA if output graphs must be connected:
  \[ \forall x, y [ x \triangleright y \lor y \triangleright x ] \]

- We can also assume this for 3-cell paradigms.
The Fix: A Stronger Connectivity Requirement

- Weakly non-inverting maps still obey $^\star$ABA if output graphs must be connected:

$$\forall x, y [x \triangleright y \lor y \triangleright x]$$

- We can also assume this for 3-cell paradigms.
A Note on Case Syncretism

- Attested syncretisms of **Acc & Dat and Acc & Gen** in Icelandic (Harðarson 2016)

**Example**

- drottning-∅/-u/-ar/-u ‘daughter’
- arm-ar/-a/-a/-um ‘arm’

- Modified case hierarchy as base (Blake 2001)

- **Prediction**: some language has Acc & Dat and Gen & Inst, or Acc & Gen and Dat & Inst
**Interim Summary**

- Weakly non-inverting graph mappings preserve aspects of the base order.
- This structure preservation derives the *ABA generalization.
- Some ad hoc stipulations are still needed in certain cases.
- Those reflect aspects of the grammatical machinery, which the graph-theoretic view abstracts away from.

<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>Target graph</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pronoun allomorphy</td>
<td>(weakly) connected</td>
<td>none</td>
</tr>
<tr>
<td>Adjectival gradation</td>
<td>(weakly) connected</td>
<td>2 ▶ 1 → 3 ▶ 1</td>
</tr>
<tr>
<td>Case syncretism</td>
<td>connected</td>
<td>none</td>
</tr>
<tr>
<td>Noun stem suppletion</td>
<td>connected</td>
<td>∃z[z ◀ x] → (x ◀ y ↔ y ◀ x)</td>
</tr>
</tbody>
</table>
The Graph-Theoretic View of the Person Case Constraint

- There are **four attested variants** of the PCC:
  - **S(strong)-PCC** DO must be 3.
    - (Bonet 1994)
  - **U(ltrastrong)-PCC** DO is less prominent than IO, where 3 is less prominent than 2, and 2 is less prominent than 1.
    - (Nevins 2007)
  - **W(eak)-PCC** 3IO combines only with 3DO.
    - (Bonet 1994)
  - **M(e first)-PCC** If IO is 2 or 3, then DO is not 1.
    - (Nevins 2007)

- But symmetric variants have been discovered.
  - (Stegovec 2016)

- This looks like a mess!
A More Systematic Perspective (Walkow 2012)

<table>
<thead>
<tr>
<th>IO↓/DO→</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NA</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>*</td>
<td>NA</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>*</td>
<td>*</td>
<td>NA</td>
</tr>
</tbody>
</table>

**U-PCC**

<table>
<thead>
<tr>
<th>IO↓/DO→</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NA</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>✓</td>
<td>NA</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>*</td>
<td>*</td>
<td>NA</td>
</tr>
</tbody>
</table>

**W-PCC**

<table>
<thead>
<tr>
<th>IO↓/DO→</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NA</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>*</td>
<td>NA</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>*</td>
<td>*</td>
<td>NA</td>
</tr>
</tbody>
</table>

**S-PCC**

<table>
<thead>
<tr>
<th>IO↓/DO→</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NA</td>
<td>*</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>*</td>
<td>NA</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>*</td>
<td>*</td>
<td>NA</td>
</tr>
</tbody>
</table>

**M1-PCC**
Graph-Theoretic Unification

**Generalized PCC**

\( y \) must not be reachable from \( x \).

**Standard PCCs:**

\( y = IO, \ x = DO \)

**Symmetric PCCs:**

\( y = DO, \ x = IO \)
Extending the PCC

- What about the other two graphs?

![Graphs](https://via.placeholder.com/150)

- The first is currently unattested.
- The second blocks all clitic combinations, as in Cairene Arabic. (Shlonsky 1997:207, Walkow p.c.)
- So 5 out of 6 graphs are attested PCCs.
Summary of Relevant Graph Classes

<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>Target graph</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pronoun allomorphy</td>
<td>(w-)connected</td>
<td>none</td>
</tr>
<tr>
<td>Adjectival gradation</td>
<td>(w-)connected</td>
<td>$2 \nrightarrow 1 \rightarrow 3 \nrightarrow 1$</td>
</tr>
<tr>
<td>Case syncretism</td>
<td>connected</td>
<td>none</td>
</tr>
<tr>
<td>Noun stem suppletion</td>
<td>connected</td>
<td>$\exists z [z \nleftarrow x] \rightarrow (x \nleftrightarrow y \nleftrightarrow y \nleftrightarrow x)$</td>
</tr>
<tr>
<td>PCC</td>
<td>w-connected</td>
<td>$3 \nrightarrow 2 \rightarrow 3 \nrightarrow 1$</td>
</tr>
</tbody>
</table>

- This is a fairly natural characterization.
- Generative accounts are too fine-grained, only mathematics allows for this unification.
### Summary of Relevant Graph Classes

<table>
<thead>
<tr>
<th>Phenomenon</th>
<th>Target graph</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pronoun allomorphy</td>
<td>(w-)connected</td>
<td>none</td>
</tr>
<tr>
<td>Adjectival gradation</td>
<td>(w-)connected</td>
<td>$2 \sqsupset 1 \rightarrow 3 \sqsupset 1$</td>
</tr>
<tr>
<td>Case syncretism</td>
<td>connected</td>
<td>none</td>
</tr>
<tr>
<td>Noun stem suppletion</td>
<td>connected</td>
<td>$\exists z [z \lessdot x] \rightarrow (x \sqsupset y \leftrightarrow y \sqsupset x)$</td>
</tr>
<tr>
<td>PCC</td>
<td>w-connected</td>
<td>$3 \sqsupset 2 \rightarrow 3 \sqsupset 1$</td>
</tr>
</tbody>
</table>

- This is a fairly natural characterization.
- Generative accounts are too fine-grained, only mathematics allows for this unification.
Why Weakly Non-Inverting Maps?

- From a certain perspective, being weakly non-inverting is computationally simple.
- All the required graphs can be represented as strings.
Why Weakly Non-Inverting Maps?

- From a certain perspective, being weakly non-inverting is computationally simple.
- All the required graphs can be represented as strings.
Why Weakly Non-Inverting Maps?

- From a certain perspective, being weakly non-inverting is computationally simple.
- All the required graphs can be represented as strings.
Why Weakly Non-Inverting Maps?

- From a certain perspective, being weakly non-inverting is computationally simple.
- All the required graphs can be represented as strings.
Why Weakly Non-Inverting Maps?

- From a certain perspective, being weakly non-inverting is computationally simple.
- All the required graphs can be represented as strings.
Why Weakly Non-Inverting Maps?

- From a certain perspective, being weakly non-inverting is computationally simple.
- All the required graphs can be represented as strings.
Why Weakly Non-Inverting Maps?

- From a certain perspective, being weakly non-inverting is computationally simple.
- All the required graphs can be represented as strings.
Subregular String Mappings

For weak mappings, we look at subregular string transductions.

Diagram:

- NFST
- MSO
- DFST
- weakly deterministic
- left-subsequential
  - L-OSL
  - 1-SL
- right-subsequential
  - ISL
  - R-OSL
1-SL Mappings

- 1-SL relations/maps = state-free N/DFST transductions
- This is sufficient to compute weakly non-inverting maps over the string representations.

\[ \sigma : \sigma \]
\[ x : y \quad x, y \in \{-, =, |\} \]

- Switching the order of \( ab \) requires memorizing \( a \Rightarrow \) not 1-SL
1-SL Mappings

- 1-SL relations/maps = state-free N/DFST transductions
- This is sufficient to compute weakly non-inverting maps over the string representations.

\[ \sigma : \sigma \quad \text{x : y} \quad \text{x, y} \in \{-, =, |}\]

- Switching the order of \( ab \) requires memorizing \( a \Rightarrow \) not 1-SL

\[ \text{a : } \varepsilon \quad \text{b : } ba \]
Extrapolating to Graph Mappings

- Of course 1-SL could reverse direction with a symbol for inverse order (←) in the string representations.
- But strings capture the idea that reversal is costly, cf.:
  - impossibility of local rotations with LBUTTs
  - markedness of metathesis in phonology
- Current graph transductions don’t capture this, deleting and adding edges is cheap.
- Maybe we need a different view of graph transductions, or a more restricted transduction class (DAG, tree, string).
- **Bottom line:** class of attested patterns should reduce to computational simplicity
Conclusion

- Graphs generalize across domains of morphosyntax
  - Base hierarchy
  - Maximally simple transduction (1-SL)
- Approach could be about markedness rather than well-formedness (weaker typological claim)
- **But:** a lot of work still to be done
  Gender Case Constraint, inverse marking, resolved agreement, . . .

Two General Points

- More work on subregular graph transductions, please!
- Mathematical view also useful for “boring” linguistic problems
Conclusion

- Graphs generalize across domains of morphosyntax
  - Base hierarchy
  - Maximally simple transduction (1-SL)
- Approach could be about markedness rather than well-formedness (weaker typological claim)
- **But:** a lot of work still to be done
  Gender Case Constraint, inverse marking, resolved agreement, . . .

Two General Points

- More work on subregular graph transductions, please!
- Mathematical view also useful for “boring” linguistic problems
References


