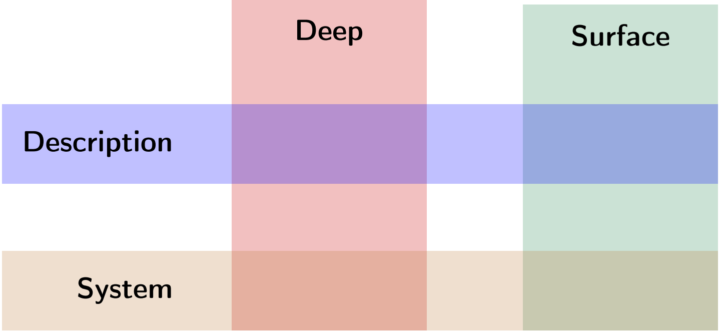
Do we Need Features for Morphosyntax?

Thomas Graf

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

ZAS
Jun 26, 2017

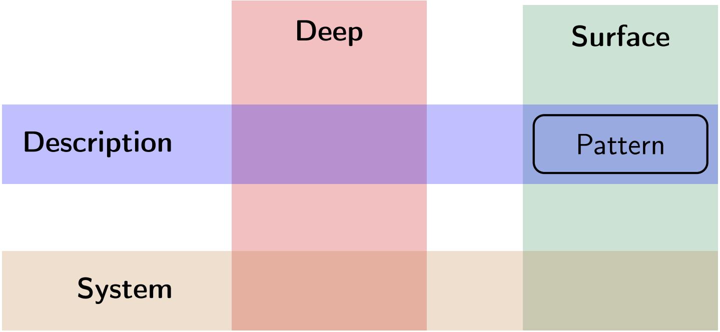
Two Routes Towards Generalizations



Why Route 2?

- Many Surface to Deep mappings
- Systematize first, then implement at Deep level

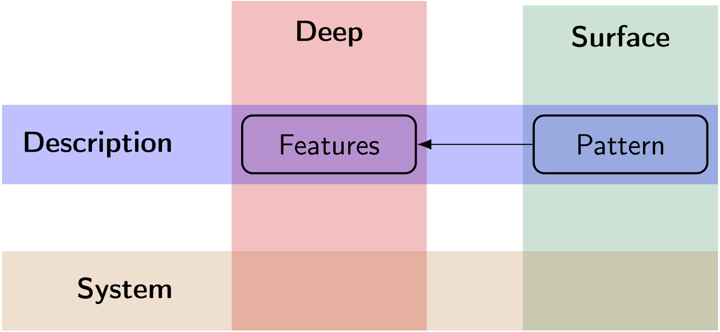
Two Routes Towards Generalizations



Why Route 2?

- Many Surface to Deep mappings
- Systematize first, then implement at Deep level

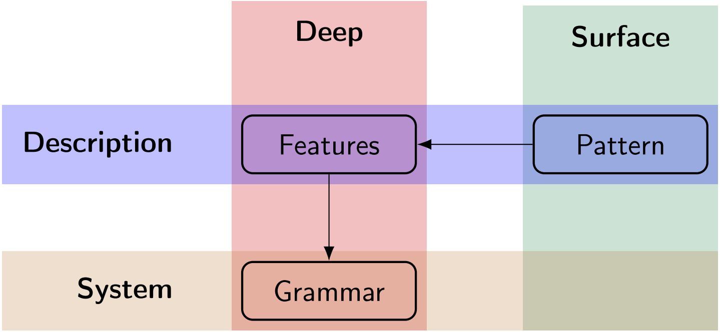
Two Routes Towards Generalizations



Why Route 2?

- Many Surface to Deep mappings
- Systematize first, then implement at Deep level

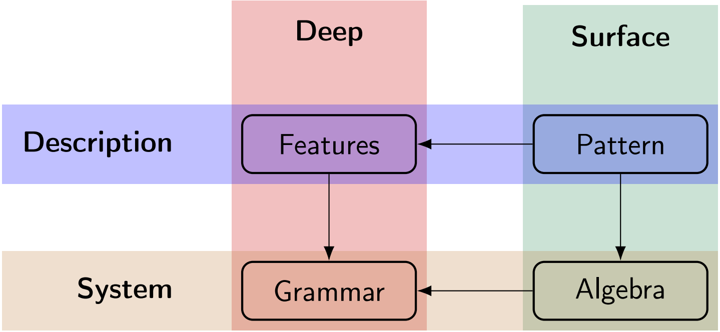
Two Routes Towards Generalizations



Why Route 2?

- Many Surface to Deep mappings
- Systematize first, then implement at Deep level

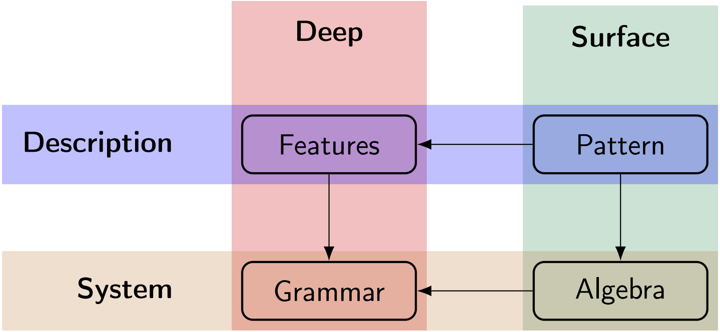
Two Routes Towards Generalizations



Why Route 2?

- Many Surface to Deep mappings
- Systematize first, then implement at Deep level

Two Routes Towards Generalizations



Why Route 2?

- Many Surface to Deep mappings
- Systematize first, then implement at Deep level

A Case Study: *ABA and PCC

*ABA Generalization (Bobaljik 2012)

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

(1) a. smart, smarter, smartest (AAA)
b. good, better, best (ABB)
c. * good, better, goodest (ABA)

Person Case Constraint (PCC; Bonet 1994; Walkow 2012)

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

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

A Case Study: *ABA and PCC

*ABA Generalization (Bobaljik 2012)

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

(1) a. smart, smarter, smartest (AAA)
b. good, better, best (ABB)
c. * good, better, goodest (ABA)

Person Case Constraint (PCC; Bonet 1994; Walkow 2012)

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

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

Outline

1 The *ABA Generalization: Monotonicity

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

3 Person Case Constraint

*ABA: A First Account

- **Syncretism:** multiple cells mapped to the same output
- 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 structure of $\{A, B, C\}$.

Illustrating Monotonicity

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

1 2 3

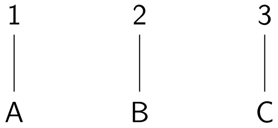
A B C

Patterns:

- But why should spell-out functions be monotonic?

Illustrating Monotonicity

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

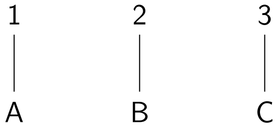


Patterns:

- But why should spell-out functions be monotonic?

Illustrating Monotonicity

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

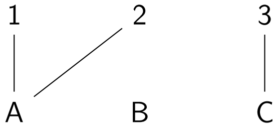


Patterns: ABC,

- But why should spell-out functions be monotonic?

Illustrating Monotonicity

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

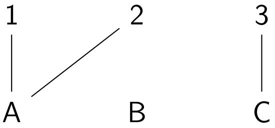


Patterns: ABC,

- But why should spell-out functions be monotonic?

Illustrating Monotonicity

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

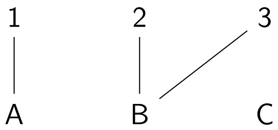


Patterns: ABC, AAB = AAC,

- But why should spell-out functions be monotonic?

Illustrating Monotonicity

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

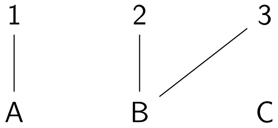


Patterns: ABC, AAB = AAC,

- But why should spell-out functions be monotonic?

Illustrating Monotonicity

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

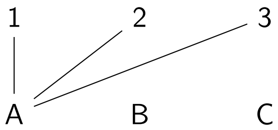


Patterns: ABC, AAB = AAC, ABB = ACC,

- But why should spell-out functions be monotonic?

Illustrating Monotonicity

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

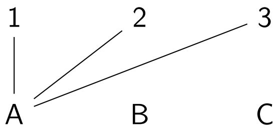


Patterns: ABC, AAB = AAC, ABB = ACC,

- But why should spell-out functions be monotonic?

Illustrating Monotonicity

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

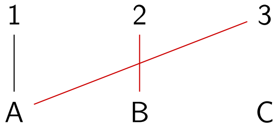


Patterns: ABC, AAB = AAC, ABB = ACC, ABC

- But why should spell-out functions be monotonic?

Illustrating Monotonicity

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



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

- But why should spell-out functions be monotonic?

A More General View: Graph Structure Preservation

The General Idea

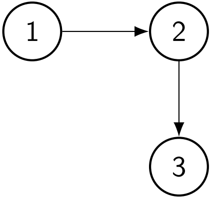
- *ABA is about structure preservation.
- Syncretism is modification of a base graph.
- Modification must not contradict orderings of base graph.

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 $x \triangleleft y \wedge y \blacktriangleleft x \rightarrow x \blacktriangleleft y$

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:
  $x \triangleleft y \wedge y \blacktriangleleft x \to x \blacktriangleleft y$

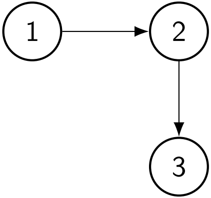


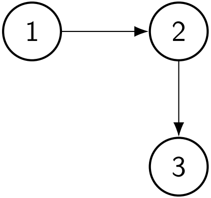
Weakly Non-Inverting Graph Mappings

- 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:
  $x \lhd y \wedge y \blacktriangleleft x \rightarrow x \blacktriangleleft y$

Weakly Non-Inverting Graph Mappings

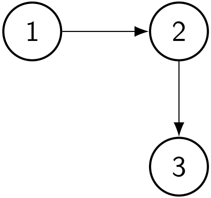
- Since we want graphs to encode hierarchies, they must be *weakly connected*: ignoring the direction of arrows, all nodes are mutually reachable.
- And the mapping must be weakly non-inverting:
  $x \lhd y \land y \blacktriangleleft x \to x \blacktriangleleft y$

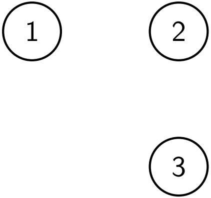




Weakly Non-Inverting Graph Mappings

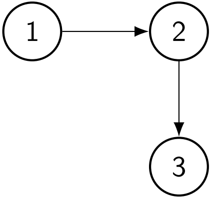
- Since we want graphs to encode hierarchies, they must be *weakly connected*: ignoring the direction of arrows, all nodes are mutually reachable.
- And the mapping must be weakly non-inverting:
  $x \triangleleft y \wedge y \blacktriangleleft x \rightarrow x \blacktriangleleft y$

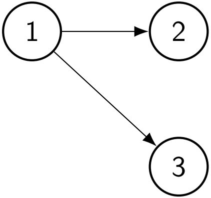




Weakly Non-Inverting Graph Mappings

- 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: $x \lhd y \wedge y \blacktriangleleft x \rightarrow x \blacktriangleleft y$



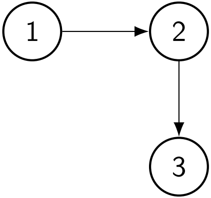


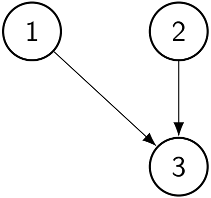
Weakly Non-Inverting Graph Mappings

- 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:
  $x \triangleleft y \wedge y \blacktriangleleft x \rightarrow x \blacktriangleleft y$

Weakly Non-Inverting Graph Mappings

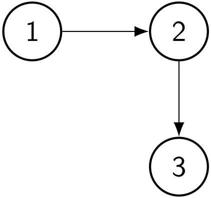
- Since we want graphs to encode hierarchies, they must be *weakly connected*: ignoring the direction of arrows, all nodes are mutually reachable.
- And the mapping must be weakly non-inverting:
  $x \triangleleft y \wedge y \blacktriangleleft x \rightarrow x \blacktriangleleft y$

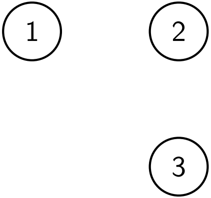




Weakly Non-Inverting Graph Mappings

- 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: $x \lhd y \wedge y \blacktriangleleft x \rightarrow x \blacktriangleleft y$





Weakly Non-Inverting Graph Mappings

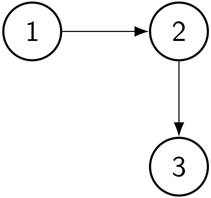
- Since we want graphs to encode hierarchies, they must be *weakly connected*: ignoring the direction of arrows, all nodes are mutually reachable.
- And the mapping must be weakly non-inverting:
  $x \lhd y \wedge y \blacktriangleleft x \rightarrow x \blacktriangleleft y$

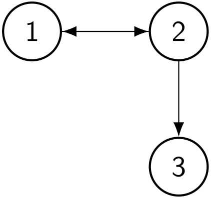
Weakly Non-Inverting Graph Mappings

- 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:
  $x \lhd y \land y \blacktriangleleft x \rightarrow x \blacktriangleleft y$

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:
  $x \lhd y \wedge y \blacktriangleleft x \rightarrow x \blacktriangleleft y$



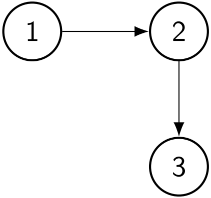


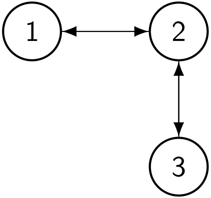
Weakly Non-Inverting Graph Mappings

- 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: $x \triangleleft y \land y \blacktriangleleft x \rightarrow x \blacktriangleleft y$

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:
  $\mathbf{x} \triangleleft \mathbf{y} \wedge \mathbf{y} \blacktriangleleft \mathbf{x} \rightarrow \mathbf{x} \blacktriangleleft \mathbf{y}$



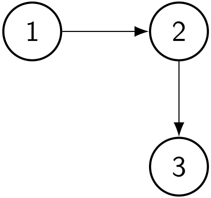


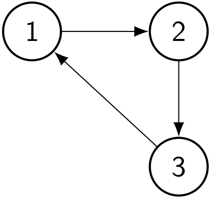
Weakly Non-Inverting Graph Mappings

- 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:
  $x \triangleleft y \wedge y \blacktriangleleft x \rightarrow x \blacktriangleleft y$

Weakly Non-Inverting Graph Mappings

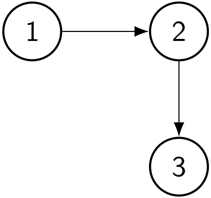
- Since we want graphs to encode hierarchies, they must be *weakly connected*: ignoring the direction of arrows, all nodes are mutually reachable.
- And the mapping must be weakly non-inverting:
  $x \triangleleft y \wedge y \blacktriangleleft x \rightarrow x \blacktriangleleft y$

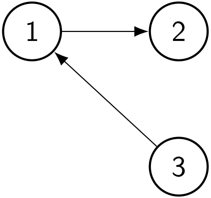




Weakly Non-Inverting Graph Mappings

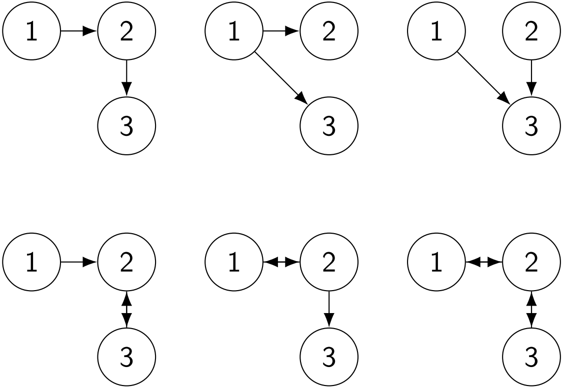
- Since we want graphs to encode hierarchies, they must be *weakly connected*: ignoring the direction of arrows, all nodes are mutually reachable.
- And the mapping must be weakly non-inverting:
  $x \lhd y \wedge y \blacktriangleleft x \rightarrow x \blacktriangleleft y$





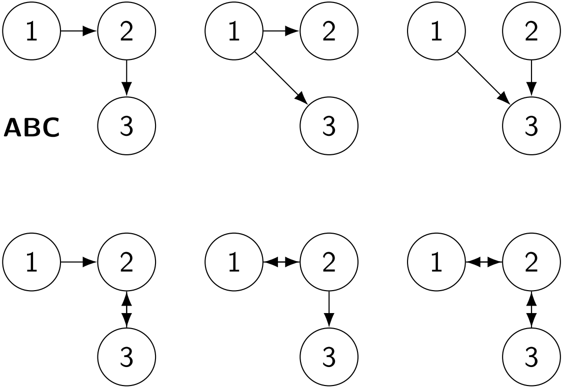
Graphs and Syncretism

- 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**.



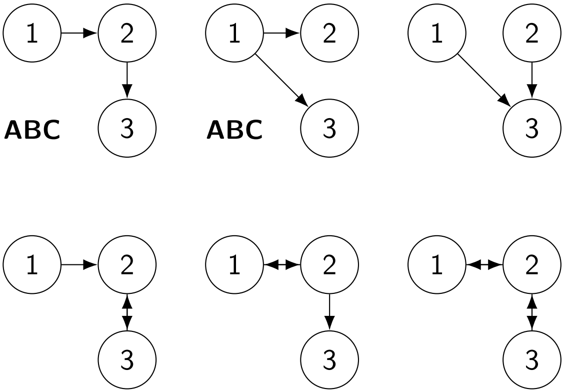
Graphs and Syncretism

- 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**.



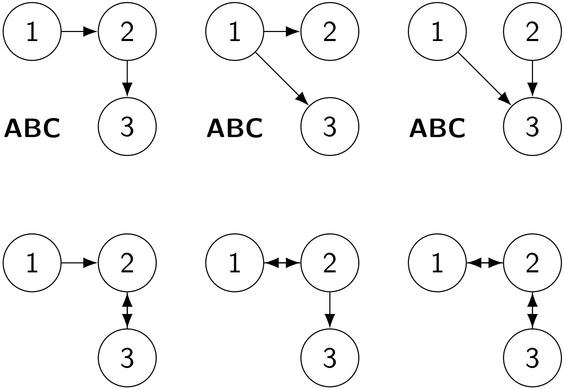
Graphs and Syncretism

- 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**.



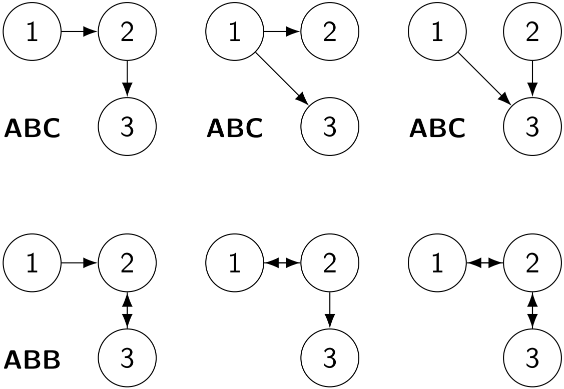
Graphs and Syncretism

- 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**.



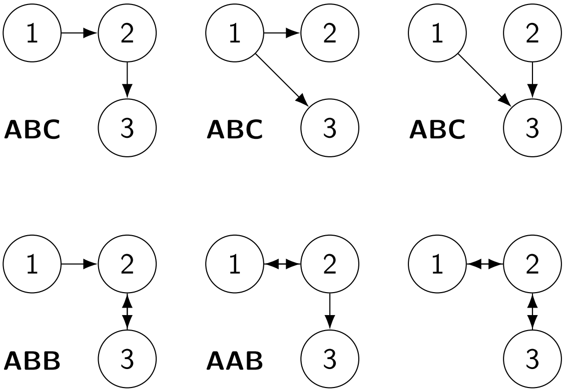
Graphs and Syncretism

- 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**.



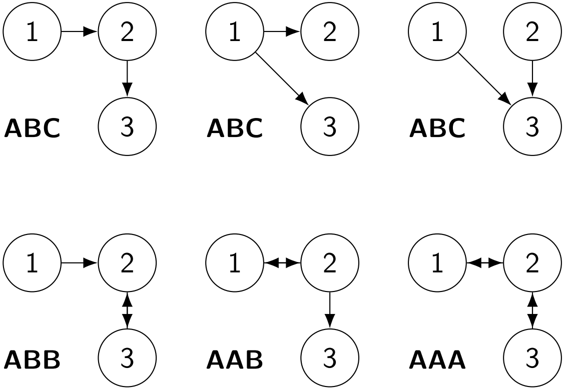
Graphs and Syncretism

- 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

- 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**.



Why Weakly Non-Inverting Maps?

- The restriction to weakly non-inverting maps reduces computational complexity.
- These graph mappings correspond to **strictly 1-local string mappings**.
- Those are the **weakest class of mappings**.
- So the *ABA generalization has a third-factor explanation: (Chomsky 2005)
  - independent base hierarchy of cells
  - computationally limited changes to hierarchy

Scaling to Larger Systems

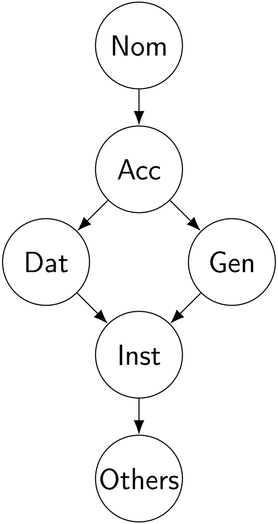
- Some morphosyntactic phenomena have many different cells.
  case syncretism, noun stem allomorphy
- Those do not scale well for feature combinatorics.
- Weakly non-inverting maps still obey *ABA
  if output graphs must be **connected**:

$$\forall x, y[x \blacktriangleleft y \vee y \blacktriangleleft x]$$

- Weakly non-inverting + strong connectedness =
  base arrows must not be removed

Case Syncretism

- Modified case hierarchy as base (Blake 2001)
- Allows syncretism of both Acc & Dat and Acc & Gen (Harðarson 2016)



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 syntactic mechanisms, which the graph-theoretic view abstracts away from.

Phenomenon	Target graph	Constraints
Pronoun allomorphy	(weakly) connected	none
Adjectival gradation	(weakly) connected	$2 \blacktriangleleft 1 \rightarrow 3 \blacktriangleleft 1$
Case syncretism	connected	none
Noun stem suppletion	connected	$\neg\exists z[z \triangleleft x] \rightarrow (y \blacktriangleleft x \rightarrow x \blacktriangleleft y)$ $\exists z[z \triangleleft x] \rightarrow (x \blacktriangleleft y \leftrightarrow y \blacktriangleleft x)$

The Graph-Theoretic View of the Person Case Constraint

- There are four attested variants of the PCC:

  S(trong)-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)

IO↓/DO→	1	2	3
1	NA	✓	✓
2	*	NA	✓
3	*	*	NA

U-PCC

IO↓/DO→	1	2	3
1	NA	*	✓
2	*	NA	✓
3	*	*	NA

S-PCC

IO↓/DO→	1	2	3
1	NA	✓	✓
2	✓	NA	✓
3	*	*	NA

W-PCC

IO↓/DO→	1	2	3
1	NA	✓	✓
2	*	NA	✓
3	*	✓	NA

M1-PCC

Graph-Theoretic Unification

Generalized PCC

y must not be reachable from x.

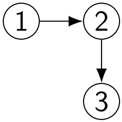
Standard PCCs:

y = IO, x = DO

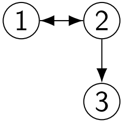
Symmetric PCCs:

y = DO, x = IO

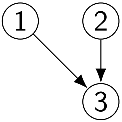
U	1	2	3
1	NA	✓	✓
2	*	NA	✓
3	*	*	NA



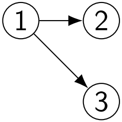
S	1	2	3
1	NA	*	✓
2	*	NA	✓
3	*	*	NA



W	1	2	3
1	NA	✓	✓
2	✓	NA	✓
3	*	*	NA



M1	1	2	3
1	NA	✓	✓
2	*	NA	✓
3	*	✓	NA



Overview of Relevant Graph Classes

Phenomenon	Target graph	Constraints
Pronoun allomorphy	(w-)connected	none
Adjectival gradation	(w-)connected	$2 \blacktriangleleft 1 \rightarrow 3 \blacktriangleleft 1$
Case syncretism	connected	none
Noun stem suppletion	connected	$\neg\exists z[z \triangleleft x] \rightarrow (y \blacktriangleleft x \rightarrow x \blacktriangleleft y)$
		$\exists z[z \triangleleft x] \rightarrow (x \blacktriangleleft y \leftrightarrow y \blacktriangleleft x)$
PCC	w-connected	$\neg\exists z[z \triangleleft x] \rightarrow (y \blacktriangleleft x \rightarrow x \blacktriangleleft y)$
		$\neg\exists z[x \triangleleft z] \rightarrow \neg\exists z[x \blacktriangleleft z]$

Conclusion

- Graphs generalize across domains of morphosyntax
- No need for features, **talk directly about cells**
- Scales better than combinatorics
- Can be a theory of markedness rather than well-formedness
- **But:** a lot of work still to be done
  Gender Case Constraint, inverse marking, resolved agreement, ...

References

Blake, Barry J. 2001. *Case*. Cambridge: Cambridge University Press.

Bobaljik, Jonathan D. 2012. *Universals in comparative morphology: Suppletion, superlatives, and the structure of words*. Cambridge, MA: MIT Press.

Bonet, Eulàlia. 1994. The Person-Case Constraint: A morphological approach. In *The Morphology-Syntax Connection*, number 22 in MIT Working Papers in Linguistics, 33–52.

Chomsky, Noam. 2005. Three factors in language design. *Linguistic Inquiry* 36:1–22. URL `http://dx.doi.org/10.1162/0024389052993655`.

Goldsmith, John. 1976. *Autosegmental phonology*. Doctoral Dissertation, MIT.

Harðarson, Gísli Rúnar. 2016. A case for a Weak Case Contiguity hypothesis — a reply to Caha. *Natural Language and Linguistic Theory* 34:1329–1343. URL `https://doi.org/10.1007/s11049-016-9328-x`.

Nevins, Andrew. 2007. The representation of third person and its consequences for person-case effects. *Natural Language and Linguistic Theory* 25:273–313. URL `https://doi.org/10.1007/s11049-006-9017-2`.

Stegovec, Adrian. 2016. Personality disorders and missing persons: Deriving the Person-Case Constraint without Case. URL `http://ling.auf.net/lingbuzz/002632`, ms., University of Connecticut.

Walkow, Martin. 2012. *Goals, big and small*. Doctoral Dissertation, University of Massachusetts Amherst.

Zwicky, Arnold. 1977. Hierarchies of person. In *Chicago Linguistic Society*, volume 13, 714–733.