

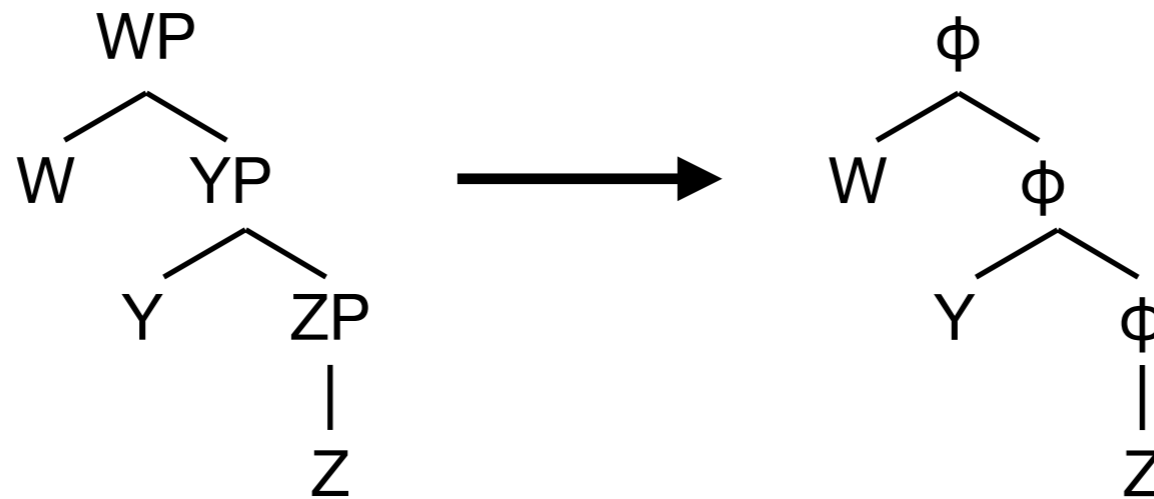
# Constraining subcategory-sensitive MATCH constraints

Nicholas Van Handel  
LSA 2021



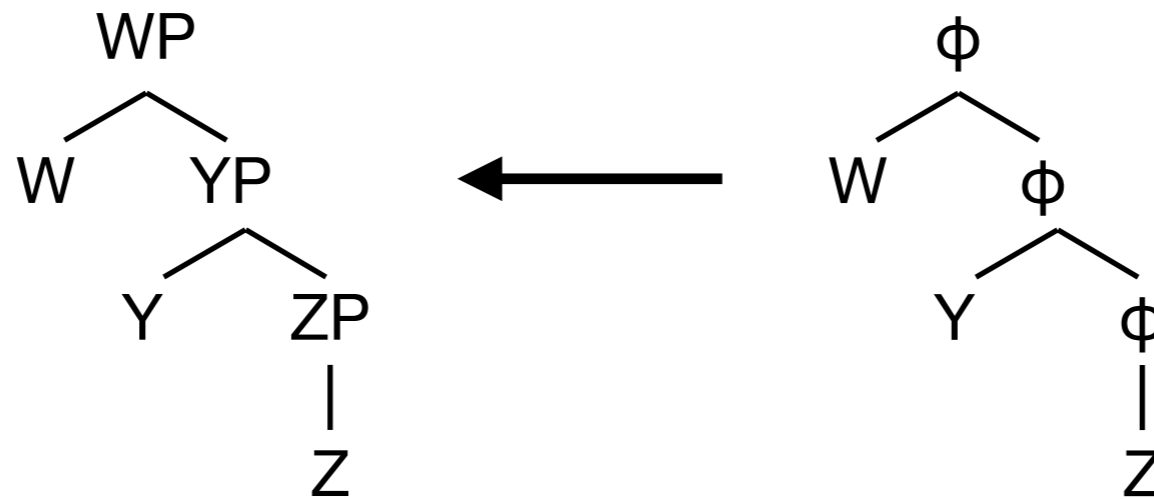
# Match constraints

- Syntax-to-prosody: Match-SP(XP,  $\phi$ ); “MatchXP”
  - “Assign a violation for each XP not matched by a  $\phi$ ”



# Match constraints

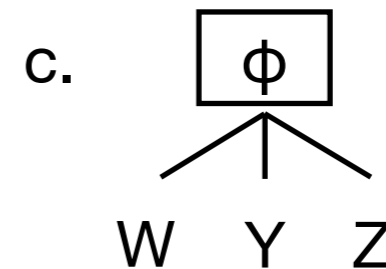
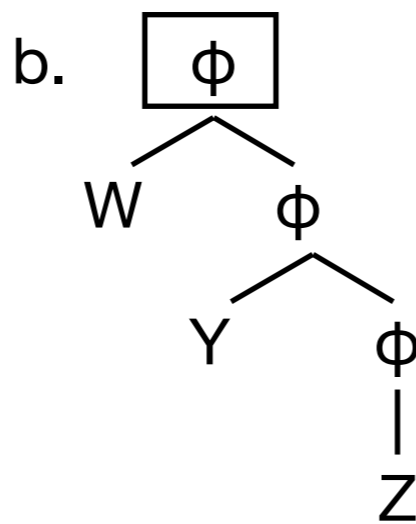
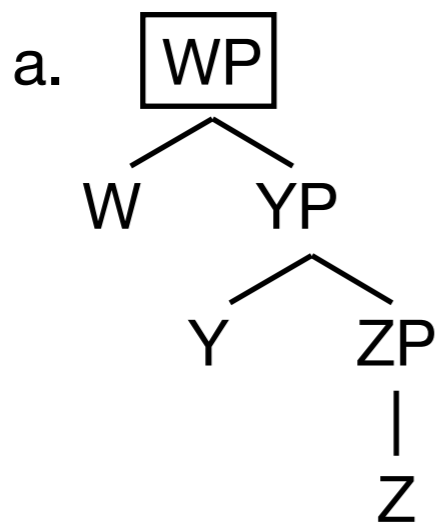
- Prosody-to-syntax: Match-PS( $\phi$ , XP); “Match- $\phi$ ”
  - “Assign a violation for each  $\phi$  not matched by an XP”





# Subcategory-sensitive constraints

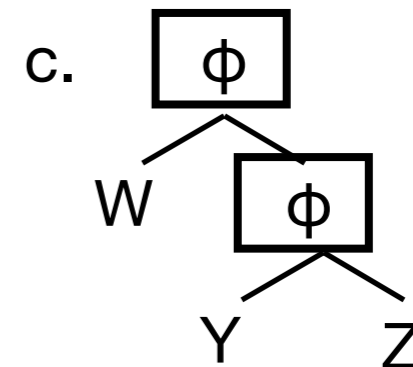
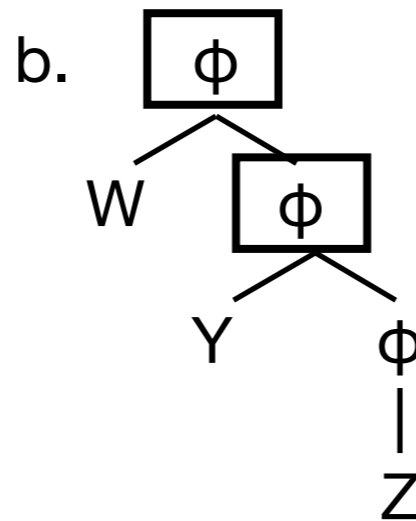
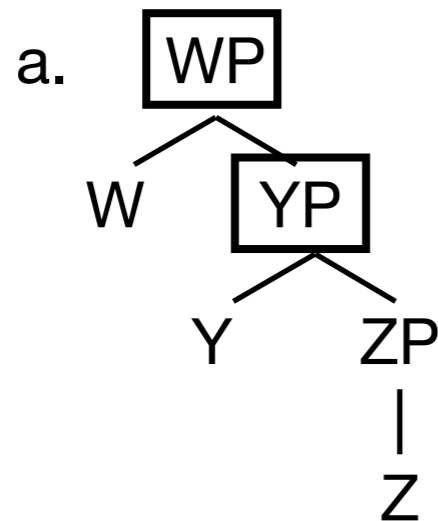
- Ishihara (2014):  $\text{MatchSP}(XP^{[+Max]}, \phi^{[+Max]})$ 
  - $XP^{[+Max]}$ : each maximal XP needs a correspondent
  - $\phi^{[+Max]}$ : each correspondent must be a maximal  $\phi$
- Prioritize matching a particular subset of XPs



(Ito and Mester 2017, Kalivoda 2018, Bellik et al 2020, i.a.)

# Subcategory-sensitive constraints

- Ito & Mester (2013): MatchSP( $XP^{[-Min]}$ ,  $\phi$ )
  - $XP^{[-Min]}$ : each non-minimal XP needs a correspondent
  - $\phi$ : each correspondent must be a  $\phi$  (of any subcategory)



# The Problem

- Without limits, this theory predicts a proliferation of Match constraints
- XP (Argument 1):
  - Maximal: [+ , - , unspecified]
  - Minimal: [+ , - , unspecified]
- $\phi$  (Argument 2):
  - Maximal: [+ , - , unspecified]
  - Minimal: [+ , - , unspecified]
- $3^4 = 81$  MatchSP constraints!

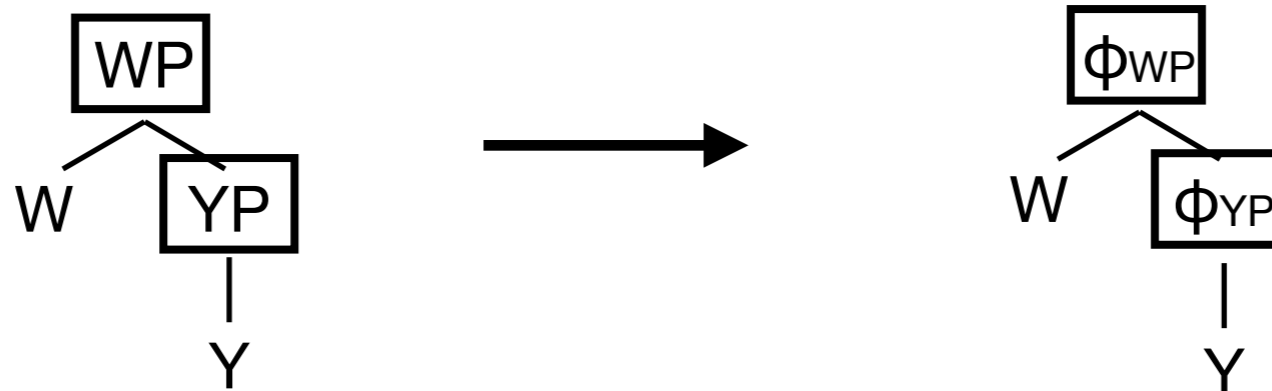
# The Problem

- Many logically possible constraints are suspect
  - MatchSP( $XP^{[-Min]}$ ,  $\phi^{[+Min]}$ )
    - “Assign one violation for each *non-minimal* XP that is not matched by a *minimal*  $\phi$ ”
- Certain combinations enforce deviations from the syntax
  - “Anti-Match”
  - It is more harmonic to be less faithful to syntax!



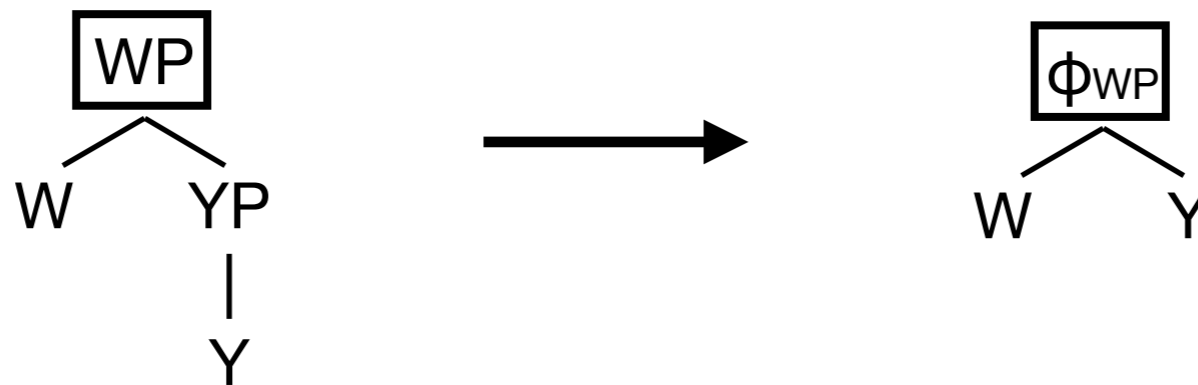
# The Problem

- Compare  $\text{MatchSP}(XP, \phi)$  and  $\text{MatchSP}(XP^{[-\text{Min}]}, \phi^{[+\text{Min}]})$
- $\text{MatchSP}(XP, \phi)$  prefers mapping each  $XP$  onto its own  $\phi$



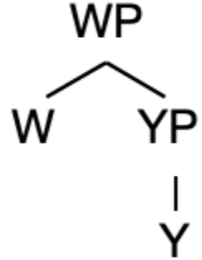
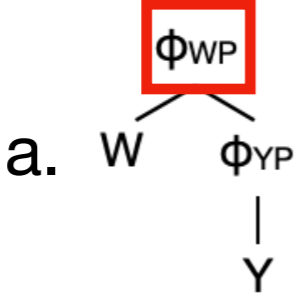

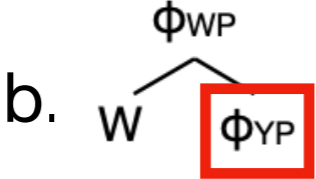
# The Problem

- Compare  $\text{MatchSP}(XP, \phi)$  and  $\text{MatchSP}(XP^{[-\text{Min}]}, \phi^{[+\text{Min}]})$
- $\text{MatchSP}(XP^{[-\text{Min}]}, \phi^{[+\text{Min}]})$  cares only about  $[-\text{Min}]$  WP
- Failing to map YP to a  $\phi$  ensures that  $\phi_{\text{WP}}$  is minimal



# The Problem

- Anti-Match: **flattening** constraint
- [-Min] XPs map to [+Min]  $\phi$  by ignoring bottom layer of structure
  - Flattening constraints prefer candidates that violate MatchSP

	MatchSP (XP <sup>[-Min]</sup> , $\phi$ <sup>[+Min]</sup> )	MatchSP (XP, $\phi$ )	MatchPS (XP, $\phi$ )
a. 	* (WP)		
 b. 		* (YP)	



# The Problem

- Match constraints are supposed to enforce syntax-prosody correspondence
  - Isomorphic structures are more marked according to Anti-Match constraints
- How widespread is Anti-Match behavior?
  - Which feature specifications cause Anti-Match behavior?
  - Can we find any generalizations such that we can exclude these specifications from our theory?

# Match constraints in SPOT

- Recall:  $3^4 = 81$  MatchSP constraints
- Large constraint space + large candidate set: not feasible by hand
- Use SPOT to determine when subcategory-sensitive Match conflicts with MatchSP and MatchPS

Custom Match(Syntax→Prosody) (i)  
Create your own custom Match constraint.

---

CP     XP     X<sup>0</sup>

Enforce Match only for syntactic nodes that are...

lexical (i)

overtly headed (i)

+  maximal (i)

-  minimal (i)

Prosodic categories must be...

+  maximal (i)

Any  minimal (i)

# Preview of results

- Two types of Anti-Match:

## 1. Flattening

- Favor *ignoring* a level in the syntax
- Conflict with MatchSP(XP,  $\phi$ )

## 2. Expansion

- Favor *adding* levels not present in the syntax
- Conflict with MatchPS(XP,  $\phi$ )

# Preview of results

- Two combinations of specifications cause Anti-Match:
  - **Conflicting:**  $\text{MatchSP}(XP^{[-\text{Min}]}, \phi^{[+\text{Min}]})$
  - **Only on  $\phi$ :**  $\text{MatchSP}(XP, \phi^{[+\text{Min}]})$
- Two configurations avoid Anti-Match
  - **Identical:**  $\text{MatchSP}(XP^{[+\text{Min}]}, \phi^{[+\text{Min}]})$
  - **Only on  $XP$ :**  $\text{MatchSP}(XP^{[+\text{Min}]}, \phi)$

# CON(s)

- Permuted CONs to test predictions of different Match constraints
- Generated typology of each CON
- Three constraints per typology:
  - 1 subcategory-sensitive MatchSP
  - General MatchSP( $XP, \phi$ )
  - General MatchPS( $\phi, XP$ )
- Restricted to constraints specified for [Max] or [Min], not both



# GEN: Inputs

- Automatically generated 1-4 word inputs in SPOT
- All logically possible recursive nestings
  - No unary XPs

1.	{X}
----	-----

1.	{X Y}
2.	{[X Y]}

1.	{X [Y Z]}
2.	{[X [Y Z]]}

1.	{W [X [Y Z]]}
2.	{W [[X Y] Z]}
3.	{[W X] [Y Z]}
4.	{[W [X [Y Z]]]}
5.	{[W [[X Y] Z]]}
6.	{[[W X] [Y Z]]}

# GEN: Output

- Weak layering
  - Allow recursion
  - Allow non-exhaustive parsing

## GEN: Output parameters

No prosodic recursion (Non-Recursivity) [i](#)

Enforce headedness [i](#)

No level-skipping (Exhaustivity) [i](#)

All intermediate nodes are branching [i](#)

Restrict maximum number of branches [i](#)

Allow movement [i](#)

## Prosodic categories ^

Root prosodic tree in [i](#)

I   $\varphi$    $\omega$

Intermediate nodes are [i](#)

I   $\varphi$    $\omega$

Prosodic terminals are [i](#)

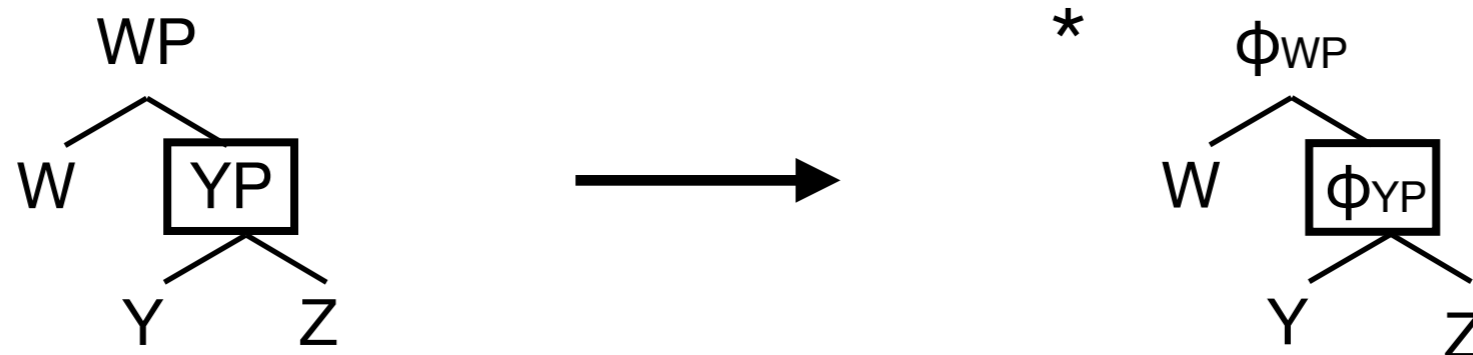
$\varphi$    $\omega$   Ft

# 1. Conflicting Specifications

- Constraints with opposite [Min]/[Max] values, e.g.,
  - Match( $XP^{[-Min]}$ ,  $\phi^{[+Min]}$ ): Flattening
  - **Match( $XP^{[+Min]}$ ,  $\phi^{[-Min]}$ )**
  - Match( $XP^{[-Max]}$ ,  $\phi^{[+Max]}$ )
  - Match( $XP^{[+Max]}$ ,  $\phi^{[-Max]}$ )

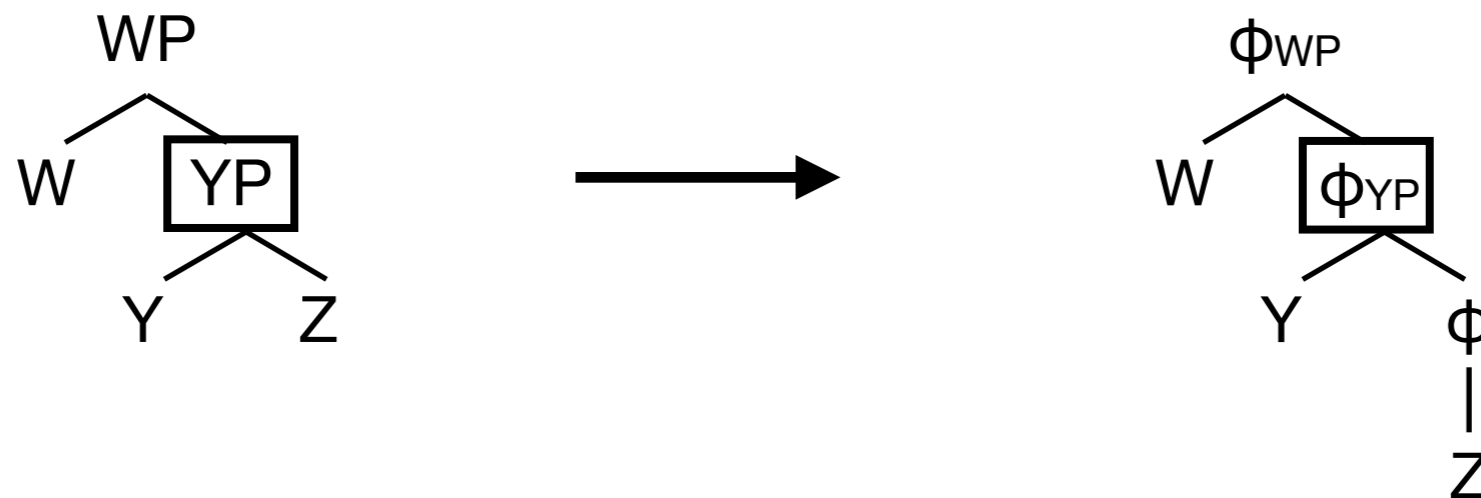
# MatchSP( $XP^{[+Min]}$ , $\phi^{[-Min]}$ )

- Isomorphic mapping violates MatchSP( $XP^{[+Min]}$ ,  $\phi^{[-Min]}$ )
- $[+Min]$  YP's correspondent,  $\phi_{YP}$ , is also  $[+Min]$




# MatchSP( $XP^{[+Min]}$ , $\phi^{[-Min]}$ )

- MatchSP( $XP^{[+Min]}$ ,  $\phi^{[-Min]}$ )
  - $[+Min]$  YP can be mapped to a  $[-Min]$   $\phi$  by placing a  $\phi$  around Z



# MatchSP( $XP^{[+Min]}$ , $\phi^{[-Min]}$ )

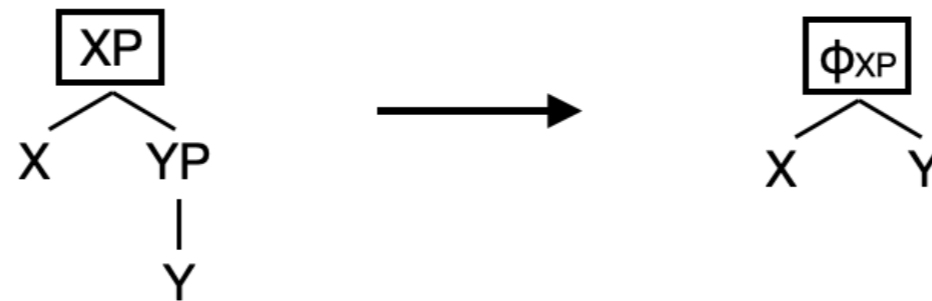
- **Expansion constraint**
- $[+Min]$  XPs map to  $[-Min]$   $\phi$  by adding another layer of structure
  - Expansion constraints prefer candidates that violate MatchPS

	$\begin{array}{c} WP \\ \swarrow \searrow \\ W \quad YP \\ \quad \swarrow \searrow \\ \quad Y \quad Z \end{array}$	MatchSP ( $XP^{[+Min]}$ , $\phi^{[-Min]}$ )	MatchSP ( $XP$ , $\phi$ )	MatchPS ( $XP$ , $\phi$ )
a.	$\begin{array}{c} \phi_{WP} \\ \swarrow \searrow \\ W \quad \phi_{YP} \\ \quad \swarrow \searrow \\ \quad Y \quad Z \end{array}$	*		
 b.	$\begin{array}{c} \phi_{WP} \\ \swarrow \searrow \\ W \quad \phi_{YP} \\ \quad \swarrow \searrow \\ \quad Y \quad \phi \\ \quad \quad   \\ \quad \quad Z \end{array}$			*

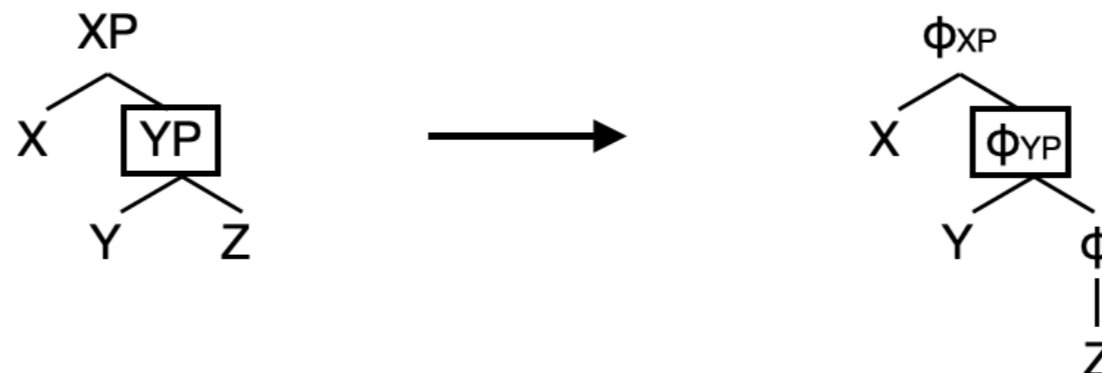


# Summary

- Anti-Match arises when a constraint calls for a change in dominance relations
- MatchSP( $XP^{[-Min]}$ ,  $\phi^{[+Min]}$ ): **flattening**
  - To go from  $[-Min]$  to  $[+Min]$ : ignore structure!



- MatchSP( $XP^{[+Min]}$ ,  $\phi^{[-Min]}$ ): **expansion**
  - To go from  $[+Min]$  to  $[-Min]$ : add structure!



# 1. Conflicting Specifications



- General problem: similar patterns seen with  $[\pm\text{Max}]$ , e.g.,
  - MatchSP( $\text{XP}^{[-\text{Min}]}$ ,  $\phi^{[+\text{Min}]}$ ): Flattening
  - MatchSP( $\text{XP}^{[+\text{Min}]}$ ,  $\phi^{[-\text{Min}]}$ ): Expansion
  - MatchSP( $\text{XP}^{[-\text{Max}]}$ ,  $\phi^{[+\text{Max}]}$ ): Flattening
  - MatchSP( $\text{XP}^{[+\text{Max}]}$ ,  $\phi^{[-\text{Max}]}$ ): Expansion

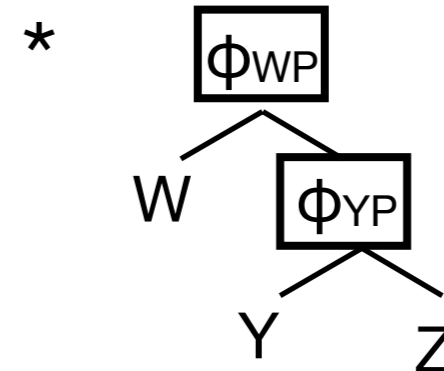
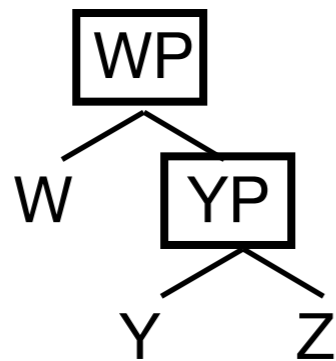


# 2. Specifications on $\phi$

- Problem: Anti-Match generalizes beyond obvious conflicts in specifications, e.g.,
  - **MatchSP(XP,  $\phi^{[-Min]}$ )**
  - MatchSP(XP,  $\phi^{[+Min]}$ )
  - MatchSP(XP,  $\phi^{[-Max]}$ )
  - MatchSP(XP,  $\phi^{[+Max]}$ )

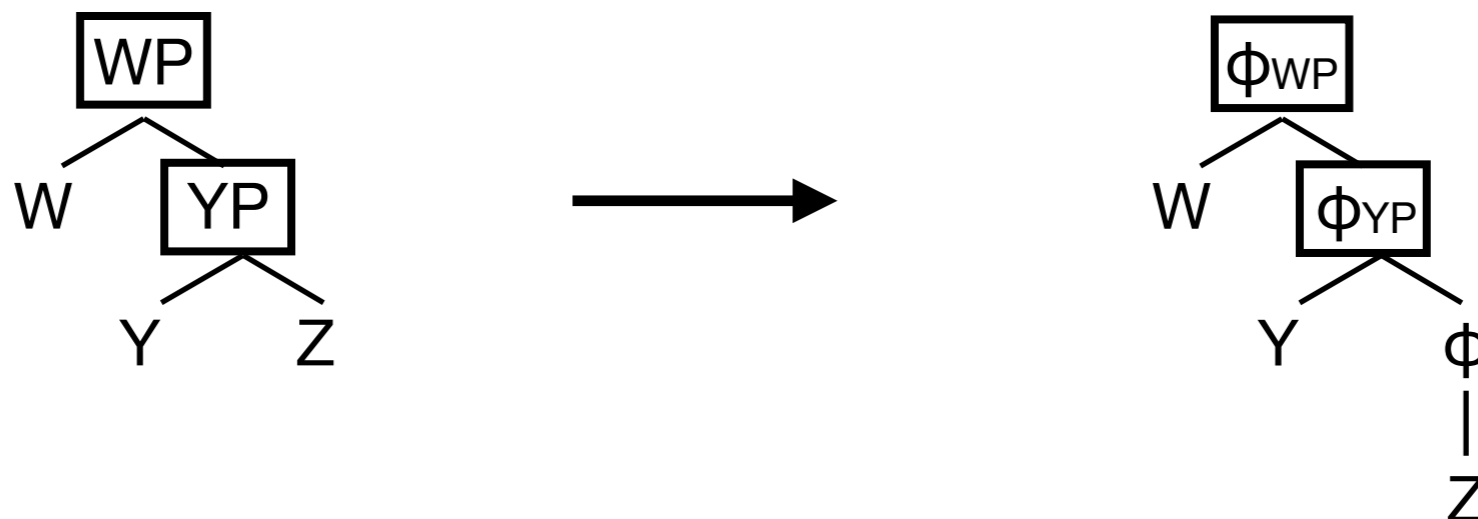
# MatchSP(XP, $\phi[-\text{Min}]$ )

- Isomorphic mapping violates MatchSP(XP,  $\phi[-\text{Min}]$ )
- WP's correspondent,  $\phi_{\text{WP}}$ , is [-Min]: 
- YP's correspondent,  $\phi_{\text{YP}}$ , is [+Min]: 



# MatchSP(XP, $\phi$ [-Min])

- Again, **expansion** preferred over isomorphy to make  $\phi_{YP}$  [-Min]
- Implicit call for a reversal in dominance relations
  - [+Min] YP is included in the set of all XPs
- Anti-Match arises even when specifications aren't in apparent conflict



# 2. Specifications on $\phi$

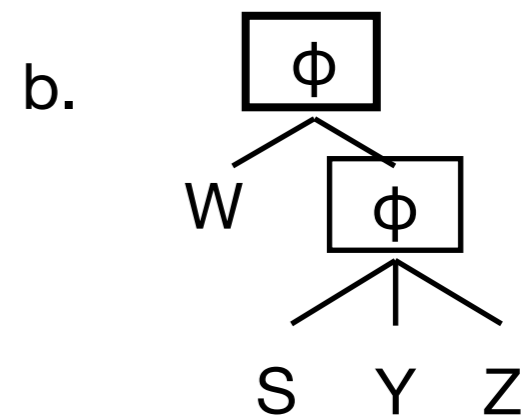
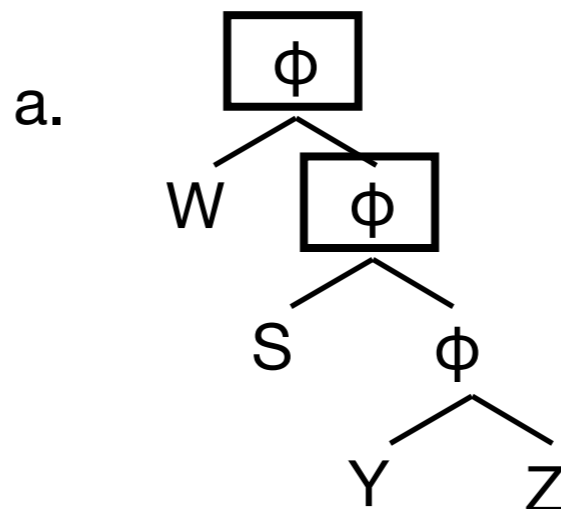
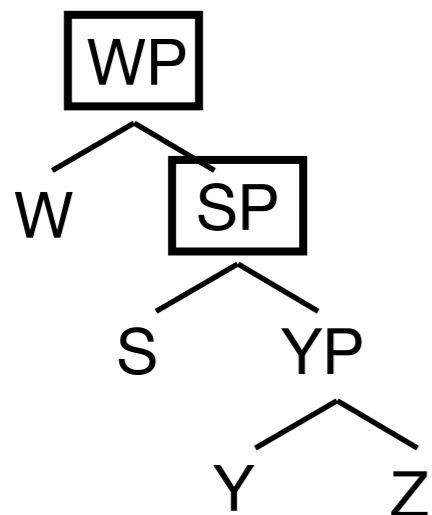
- Conclusion: Anti-Match behavior generalizes across constraints with specifications on  $\phi$ , e.g.,
  - Match(XP,  $\phi^{[-Min]}$ ): Expansion
  - Match(XP,  $\phi^{[-Max]}$ ): Expansion
  - Match(XP,  $\phi^{[+Max]}$ ): Flattening

# 3. Specifications on XP

- No Anti-Match when specifications are only on XP, e.g.,
  - **Match( $XP^{[-Min]}$ ,  $\phi$ )**
  - Match( $XP^{[+Min]}$ ,  $\phi$ )
  - Match( $XP^{[-Max]}$ ,  $\phi$ )
  - Match( $XP^{[+Max]}$ ,  $\phi$ )


# MatchSP( $XP^{[-Min]}$ , $\phi$ )

- MatchSP( $XP^{[-Min]}$ ,  $\phi$ ) (Ito and Mester, 2013)
  - Satisfied by isomorphic parse (a)
  - Also satisfied by non-isomorphic parses like (b) , as long as non-minimal XPs have a corresponding  $\phi$
- Crucially, this constraint does not *prefer* non-isomorphic (b)



# MatchSP( $XP[-Min]$ , $\phi$ )

- **Special-general relationship:**
  - MatchSP( $XP[-Min]$ ,  $\phi$ ) assigns a subset of violations assigned by MatchSP( $XP$ ,  $\phi$ )
- First argument delimits the set of XPs the Match constraint cares about
  - MatchSP( $XP$ ,  $\phi$ ) cares about all XPs: WP, SP, YP
  - MatchSP( $XP[-Min]$ ,  $\phi$ ) cares about  $XP[-Min]$ , a subset: WP, SP

	$WP[W \ SP[S \ YP[Y \ Z]]]$	MatchSP ( $XP[-Min]$ , $\phi$ )	MatchSP ( $XP$ , $\phi$ )	MatchPS ( $XP$ , $\phi$ )
 a.	(W (S (Y Z))) Isomorphic			
b.	(W (S Y Z)) Partial Flattening		* (YP)	
c.	(W S Y Z) Flattened	* (SP)	** (SP, YP)	

# 3. Specifications on XP

- All constraints with specifications on XP are in this special-general relationship with MatchSP
- Favor isomorphism, e.g.
  - $\text{Match}(\text{XP}^{[-\text{Min}]}, \phi)$
  - $\text{Match}(\text{XP}^{[+\text{Min}]}, \phi)$
  - $\text{Match}(\text{XP}^{[-\text{Max}]}, \phi)$
  - $\text{Match}(\text{XP}^{[+\text{Max}]}, \phi)$

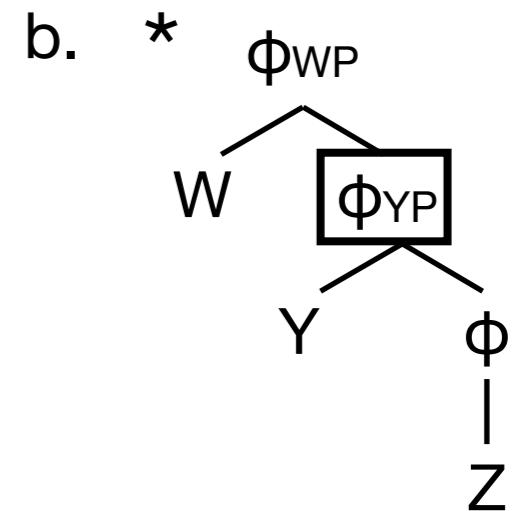
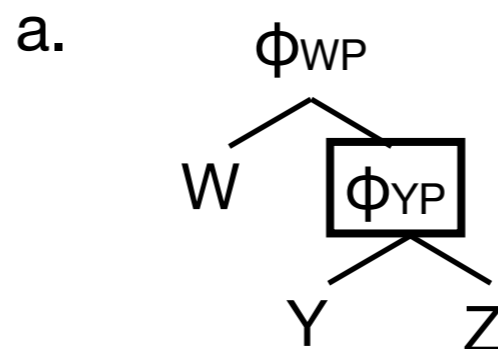
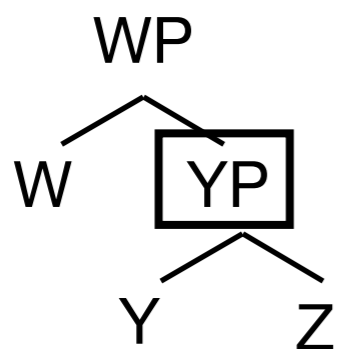


# 4. Identical specifications

- Avoid Anti-Match, but not in a special-general relationship with MatchSP, e.g.,
  - **Match( $XP^{[+Min]}$ ,  $\phi^{[+Min]}$ )**
  - Match( $XP^{[-Min]}$ ,  $\phi^{[-Min]}$ )
  - Match( $XP^{[+Max]}$ ,  $\phi^{[+Max]}$ )
  - Match( $XP^{[-Max]}$ ,  $\phi^{[-Max]}$ )


# MatchSP( $XP^{[+Min]}$ , $\phi^{[+Min]}$ )

- MatchSP( $XP^{[+Min]}$ ,  $\phi^{[+Min]}$ )
  - Satisfied by isomorphic parse (a)
  - NOT satisfied by (b), because  $\phi_{YP}$  is [-Min]
- MatchSP( $XP^{[+Min]}$ ,  $\phi^{[+Min]}$ ) works to preserve dominance relations



# MatchSP( $XP^{[+Min]}$ , $\phi^{[+Min]}$ )

- No **special-general relationship**
- Cand C only violates MatchSP( $XP^{[+Min]}$ ,  $\phi^{[+Min]}$ )
  - MatchSP( $XP^{[+Min]}$ ,  $\phi^{[+Min]}$ ) requires  $[+Min]$   $\phi_{YP}$
  - MatchSP( $XP$ ,  $\phi$ ) is happy to have *any*  $\phi_{YP}$
- Identical specifications are **dominance-preserving**

	$WP[W_{YP}[Y Z]]$	MatchSP ( $XP^{[+Min]}$ , $\phi^{[+Min]}$ )	MatchSP ( $XP$ , $\phi$ )	MatchPS ( $XP$ , $\phi$ )
 a.	(W (Y Z)) Isomorphic			
b.	(W Y Z) Flattened	* (YP)	* (YP)	
c.	(W (Y (Z))) Expanded	* (YP)		

# 4. Identical specifications

- Avoid Anti-Match *and* preserve dominance relations, e.g.,
  - Match( $XP^{[-Min]}$ ,  $\phi^{[-Min]}$ )
  - Match( $XP^{[+Min]}$ ,  $\phi^{[+Min]}$ )
  - Match( $XP^{[-Max]}$ ,  $\phi^{[-Max]}$ )
  - Match( $XP^{[+Max]}$ ,  $\phi^{[+Max]}$ )

# Conclusion

- Anti-Match: specifications **conflict** or **only on  $\phi$** 
  - **Flattening**: [+Max] or [+Min] on  $\phi$
  - **Expansion**: [-Max] or [-Min] on  $\phi$
- Lawful Match:
  - **Specialized**: specification only on XP
  - **Dominance-preserving**: identical specifications

# Conclusion

- Anti-Match: when specifications **conflict** or are **only on  $\phi$**
- Should be excluded from the theory
  - Mapping should *enforce* isomorphism, not prevent it
  - Weird theory of markedness: penalize being too faithful to the syntactic input
  - Deviations should be driven instead by well known prosodic constraints (e.g., binarity, sisterhood)

# Conclusion

- Limit subcategory-sensitive constraints to:
  - **Specialized**, e.g.,  $\text{MatchSP}(\text{XP}^{[+\text{Min}]}, \phi)$
  - **Dominance-preserving**, e.g.,  $\text{MatchSP}(\text{XP}^{[+\text{Min}]}, \phi^{[+\text{Min}]})$
- Restriction takes us from 81 to 17 MatchSP constraints
- Whether all 17 are still needed is an open question, but we've narrowed the space of constraints considerably

# Conclusion

- Theories with prosodic recursion and subcategories have a large search space for both candidates and constraints
- SPOT is particularly well-suited to developing these theories
  - Theory comparison would be labor-intensive without automatic generation and evaluation of candidates
  - We can easily test different constraint definitions



# Thank you!

Junko Ito, Armin Mester

Ryan Bennett, Amanda Rysling

Jenny Bellik, Nick Kalivoda, Richard Bibbs

This research was supported by the National Science Foundation, Award #1749368.

# Appendix

# Binary vs. privative features

- Greater restrictions with privative features
  - [Min] vs. unspecified
  - [Max] vs. unspecified
- $2^4 = 16$  constraints, not 81
- No need to stipulate that [+Max] can't be paired with [-Max]
- But, still need to stipulate that you can't have specifications only on  $\phi$

# Binary vs. privative features

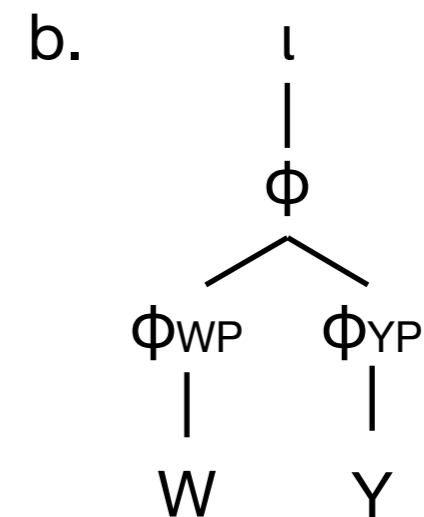
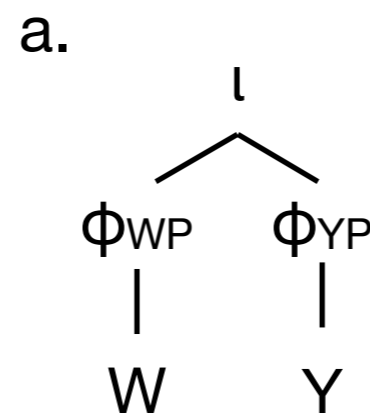
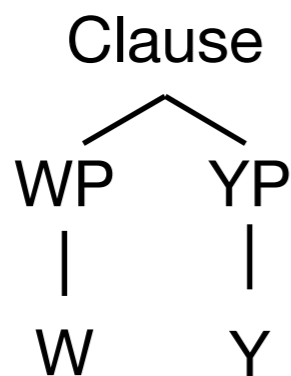
- Potential difficulty: previous analyses using [-Min] / [-Max]
  - Irish: LH phrase accent at left edge of  $\phi^{[-\text{Min}]}$  (Elfner, 2015)
  - Basque:  $\phi^{[-\text{Min}]}$  is domain of pitch reset (Elordieta, 2015)
  - BinMaxHead( $\omega^{[+\text{Max}, -\text{Min}]}$ ) (Ito and Mester, 2021)

# Eliminate subcategory-sensitive Match?

- Subcategory-sensitive constraints can sometimes be replaced by other constraints
- Ito and Mester (2013, 2016) on Japanese
  - $\text{MatchSP}(\text{XP}^{[-\text{Min}]}, \phi) \rightarrow \text{MatchPS}(\phi, \text{XP})$
- Bellik and Kalivoda (2020) on Irish
  - $\text{MatchSP}(\text{XP}^{[-\text{Min}]}, \phi) \rightarrow \text{MatchSP}(\text{XP}_{\text{OverHead}}, \phi)$
- Van Handel (ms) on Italian
  - $\text{MatchSP}(\text{XP}^{[+\text{Max}]}, \phi^{[+\text{Max}]}) \rightarrow \text{MatchPS}(\phi, \text{XP})$

# Specialized vs. Dominance-preserving

- Is it crucial that subcategory-sensitive constraints preserve dominance relations?
- E.g.,  $\text{MatchSP}(XP^{[+Max]}, \phi^{[+Max]})$  vs.  $\text{MatchSP}(XP^{[+Max]}, \phi)$
- Only  $\text{MatchSP}(XP^{[+Max]}, \phi^{[+Max]})$  prefers (a) to (b)



# Specialized vs. Dominance-preserving

- Empirical work necessary to know whether both **Specialized** and **Dominance-preserving** are necessary
- Allowing for only **Specialized** or only **Dominance-preserving** reduces the set to 9 constraints

# Two-Word XP Inputs

- Inputs with two-word XPs (rather than unary XPs) ensure we include the configuration in which a [+Min]  $\phi$  can become [-Min]
- Below, if YP consisted solely of Y, it wouldn't be possible to create a [-Min]  $\phi_{YP}$ 
  - Assuming non-vacuous recursion, i.e., \*((Y))
- Two-word [+Min] YP can become [-Min] by putting either Y or Z into its own  $\phi$

