Crash course in Optimality Theory

Asimov's three laws of robot ethics

[1] A robot may not injure a human being or, through inaction, allow a human being to come to harm.

[2] A robot must obey the orders given it by human beings, except where such orders would conflict with the First Law.

[3] A robot must protect its own existence, as long as such protection does not conflict with the First or Second Law.

This example is taken from Jochen Trommer's EGG 2008 lecture slides

Crash course in Optimality Theory

Asimov's three laws of robot ethics

[1] A robot may not injure a human being or, through inaction, allow a human being to come to harm.

[2] A robot must obey the orders given it by human beings, **except where such orders would conflict with the First Law**.

[3] A robot must protect its own existence, **as long as such protection does not conflict with the First or Second Law**.

Instead of rules that refer to other rules, let's…

(i) formulate these rules as constraints

(ii) Rank these constraints

Crash course in Optimality Theory

Asimov's three laws of robot ethics as constraints

INJURE HUMAN A robot may not injure a human being or, through inaction, allow a human being to come to harm.

OBEY ORDER A robot must obey the orders given it by human beings.

PROTECT EXISTENCE A robot must protect its own existence

INJURE HUMAN $\gg$ OBEY ORDER $\gg$ PROTECT EXISTENCE

Crash course in Optimality Theory

Scenario 1:

H says to a **R**obot: Kill my boss!

Potential outcomes

1. **R** kills **H**'s boss
2. **R** kills **H** (who gave it the order)
3. **R** doesn't kill anyone
4. **R** kills itself

Crash course in Optimality Theory

Step 1: Create a tableau for the scenario

Input: H says to R: Kill my boss!

	INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
R kills **H**’s boss			
R kills **H**			
R doesn’t kill anyone			
R kills itself			

NB: The higher-ranked the constraint, the further leftwards it appears in the tableau

Crash course in Optimality Theory

Step 2: Assign violations for each candidate output

Input: H says to R: Kill my boss!

	INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
R kills **H**’s boss			
R kills **H**			
R doesn’t kill anyone			
R kills itself			

Crash course in Optimality Theory

Step 2: Assign violations for each candidate output

Input: H says to R: Kill my boss!

	INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
R kills **H**'s boss	*		
R kills **H**			
R doesn't kill anyone			
R kills itself			

Crash course in Optimality Theory

Step 2: Assign violations for each candidate output

Input: H says to R: Kill my boss!

	INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
R kills **H**'s boss	*		
R kills **H**	*	*	
R doesn't kill anyone			
R kills itself			

Crash course in Optimality Theory

Step 2: Assign violations for each candidate output

Input: H says to R: Kill my boss!

	INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
R kills **H**'s boss	*		
R kills **H**	*	*	
R doesn't kill anyone		*	
R kills itself			

Crash course in Optimality Theory

Step 2: Assign violations for each candidate output

Input: H says to R: Kill my boss!

	INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
R kills **H**'s boss	*		
R kills **H**	*	*	
R doesn't kill anyone		*	
R kills itself		*	*

Crash course in Optimality Theory

Step 3: Eliminate suboptimal candidate outputs

Input: H says to R: Kill my boss!

	INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
R kills **H**'s boss	*		
R kills **H**	*	*	
R doesn't kill anyone		*	
R kills itself		*	*

Crash course in Optimality Theory

Step 3: Eliminate suboptimal candidate outputs

Input: H says to R: Kill my boss!

	INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
R kills **H**'s boss	*		
R kills **H**	*	*	
R doesn't kill anyone		*	
R kills itself		*	*!

Crash course in Optimality Theory

Step 3: Eliminate suboptimal candidate outputs

Input: H says to R: Kill my boss!

	INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
R kills **H**'s boss	*!		
R kills **H**	*	*	
R doesn't kill anyone		*	
R kills itself		*	*!

Crash course in Optimality Theory

Step 3: Eliminate suboptimal candidate outputs

Input: H says to R: Kill my boss!

	INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
R kills **H**'s boss	*!		
R kills **H**	*!	*	
☞ **R** doesn't kill anyone		*	
R kills itself		*	*!

__NB__: Multiple asterisks are assigned if a constraint is violated multiple times

An candidate that violates a lower-ranked constraint once is "wins" over a candidate that violates a higher-ranked constraint only once

Crash course in Optimality Theory

Task 1: A different ranking of constraints for Scenario 1

Input: H says to R: Kill my boss!

	OBEY ORDER	**INJURE HUMAN**	**PROTECT EXISTENCE**
R kills **H**'s boss			
R kills **H**			
R doesn't kill anyone			
R kills itself			

Crash course in Optimality Theory

Task 1: A different ranking of constraints for Scenario 1

Input: H says to R: Kill my boss!

	OBEY ORDER	INJURE HUMAN	PROTECT EXISTENCE
☞ **R** kills **H**'s boss		*	
R kills **H**	*!	*	
R doesn't kill anyone	*!		
R kills itself	*!		*

Crash course in Optimality Theory

Task 2: A different Scenario (i.e. a different input)

Input: H says to R: Kill my boss! If you don't, I will kill him!

	INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
R kills **H**'s boss			
R kills **H**			
R doesn't kill anyone			
R kills itself			

Crash course in Optimality Theory

Task 2: A different Scenario (i.e. a different input)

Input: H says to R: Kill my boss! If you don't, I will kill him!

	INJURE HUMAN	OBEY ORDER	PROTECT EXISTENCE
☞ **R** kills **H**'s boss	*		
R kills **H**	*	*!	
R doesn't kill anyone	*	*!	
R kills itself	*	*!	*

Quick summary

Optimality theory (OT)

Input → multiple candidate outputs → actual output(s)

- ❖ Most candidate outputs will be filtered out as **suboptimal**
- ❖ Optimal candidates don’t have to perfect, but incur fewer violations of constraints
- ❖ Constraints are **ranked**: multiple violations of lower-ranked constraints are preferred to single violations of higher-ranked ones

Crash course in Prosodic Theory

Segmental phonology =

Phonological theory at the level of phonemes (vowels and consonants) and how they interact

e.g. *Insertion and assimilation of voiceless plosives in many English dialects*

/dænØs/	[dænts]	*dance*
/strɛŋØɵ/	[strɛŋkɵ]	*strength*
/hæmØstər/	[hæmpstər]	*hamster*

Insertion: $/\text{Ø}/ \leftrightarrow \text{C}_{i\,[\text{+plosive, -voice}]} \;/\; \text{C}_{[\text{+nasal}]} \,\_\_\_\_\, \text{C}_{[\text{+fricative, -voice}]}$

Assimilation: $\text{C}_i \leftrightarrow \text{C}_{k\,[\text{POA}]} \;/\; \text{C}_k \,\_\_\_\_$

Crash course in Prosodic Theory

Suprasegmental phonology =

Phonological theory above the level of phonemes

Syllables = σ σ σ σ

ʌn . bə . liːv . əbl

Feet = $(\sigma_w \ \sigma_s)_{Ft}$ $(\sigma_w \ \sigma_s)_{Ft}$ $(\sigma_w \ \sigma_s)_{Ft}$ $(\sigma_w \ \sigma_s)_{Ft}$ $(\sigma_w \ \sigma_s)_{Ft}$

if . mu . sic . be . the . food . of . love . play . on

- A foot must contain at least one strong (stressed) syllable

Phonological word =$($ˈ**mon**keys$)_\omega$ $($ˈ**eat**$)_\omega$ $($**ba**ˈnanas$)_\omega$

- ωs must contain at least one foot
- Seeing as they are not usually feet, function words are not usually ωs

Crash course in Prosodic Theory

Suprasegmental phonology =

Phonological theory above the level of phonemes

Phonological phrase = (good novels)$_{\phi}$ (read easier)$_{\phi}$

- As a default, ϕs correspond to syntactic phrases

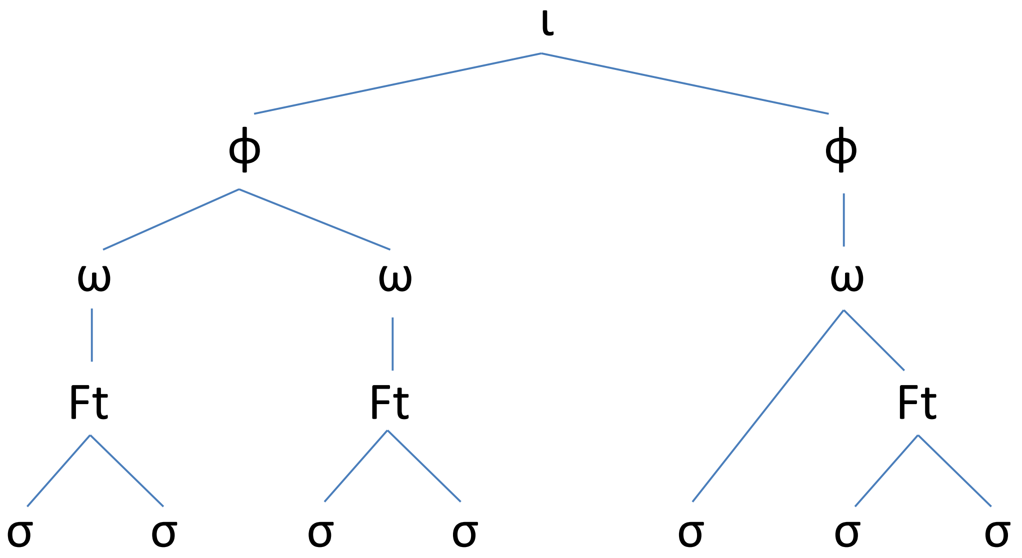
Intonational domain = (who did you see?)ι

- As a default, ιs correspond to syntactic clauses (CPs)
- A pause between two ιs is typically longer than a pause between two ϕs

Crash course in Prosodic Theory

The prosodic hierarchy

Intonational phrase (ι)
Phonological phrase (ϕ)
Prosodic word (ω)
Foot (Ft)
Syllable (σ)

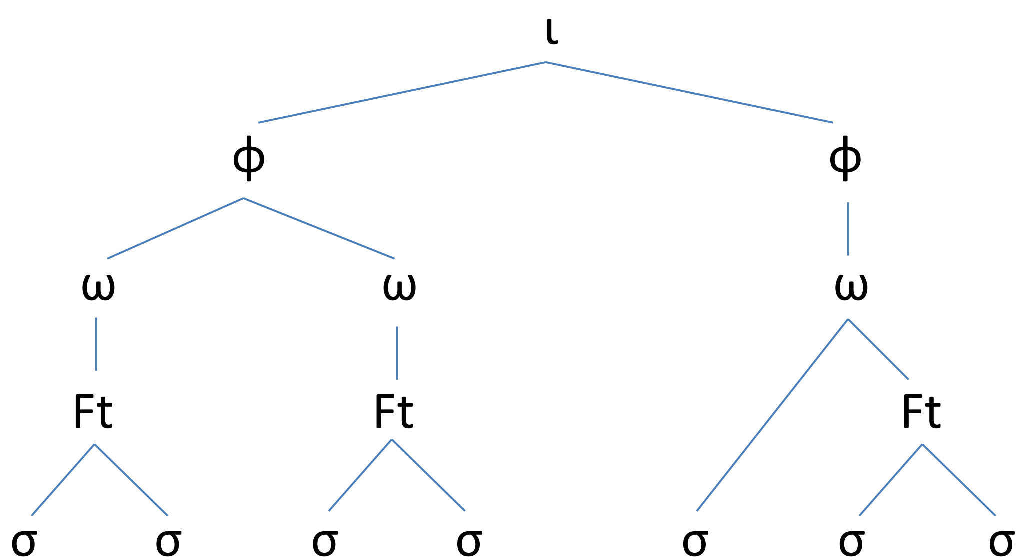


Crash course in Prosodic Theory

The prosodic hierarchy

- Constraint on hierarchical organisation:

  Proper headedness (adapted from Itō & Mester 2003)
  Every nonterminal prosodic category of level *L* must immediately dominate a category of level L^{-1}

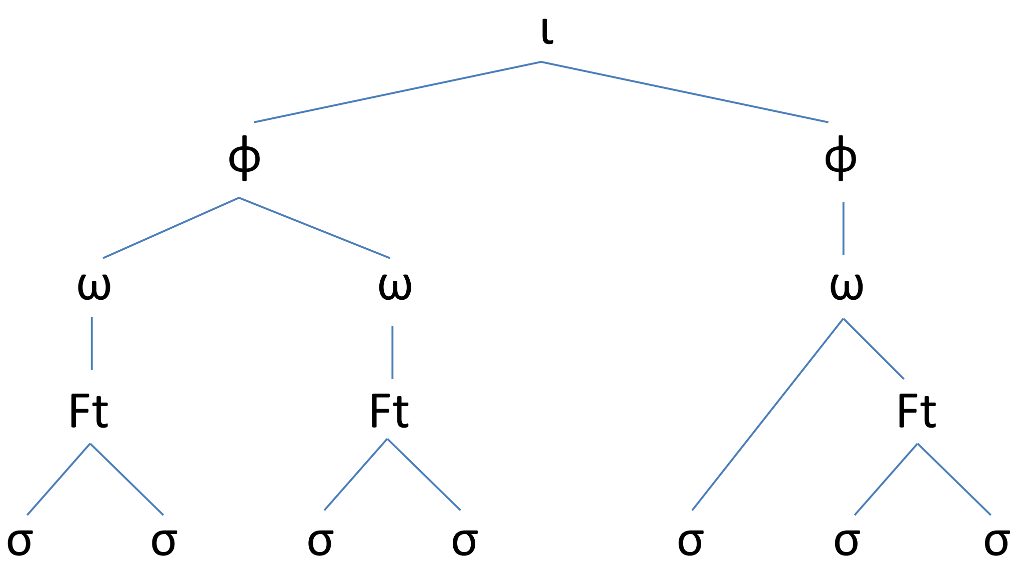


Crash course in Prosodic Theory

The prosodic hierarchy

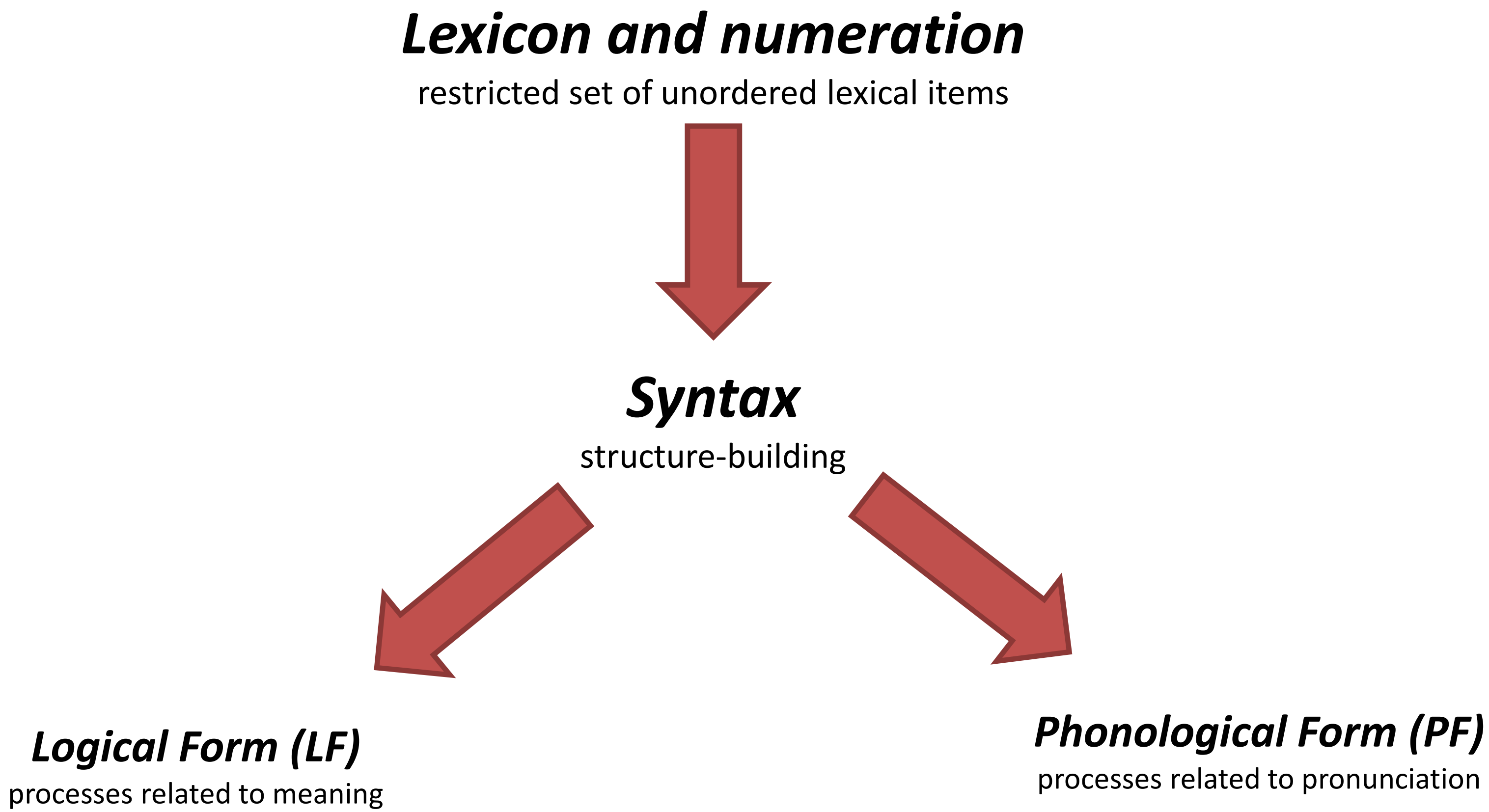
Intonational phrase (ι)
Phonological phrase (ϕ)
Prosodic word (ω)
Foot (Ft)
Syllable (σ)

largely determined by syntactic structure

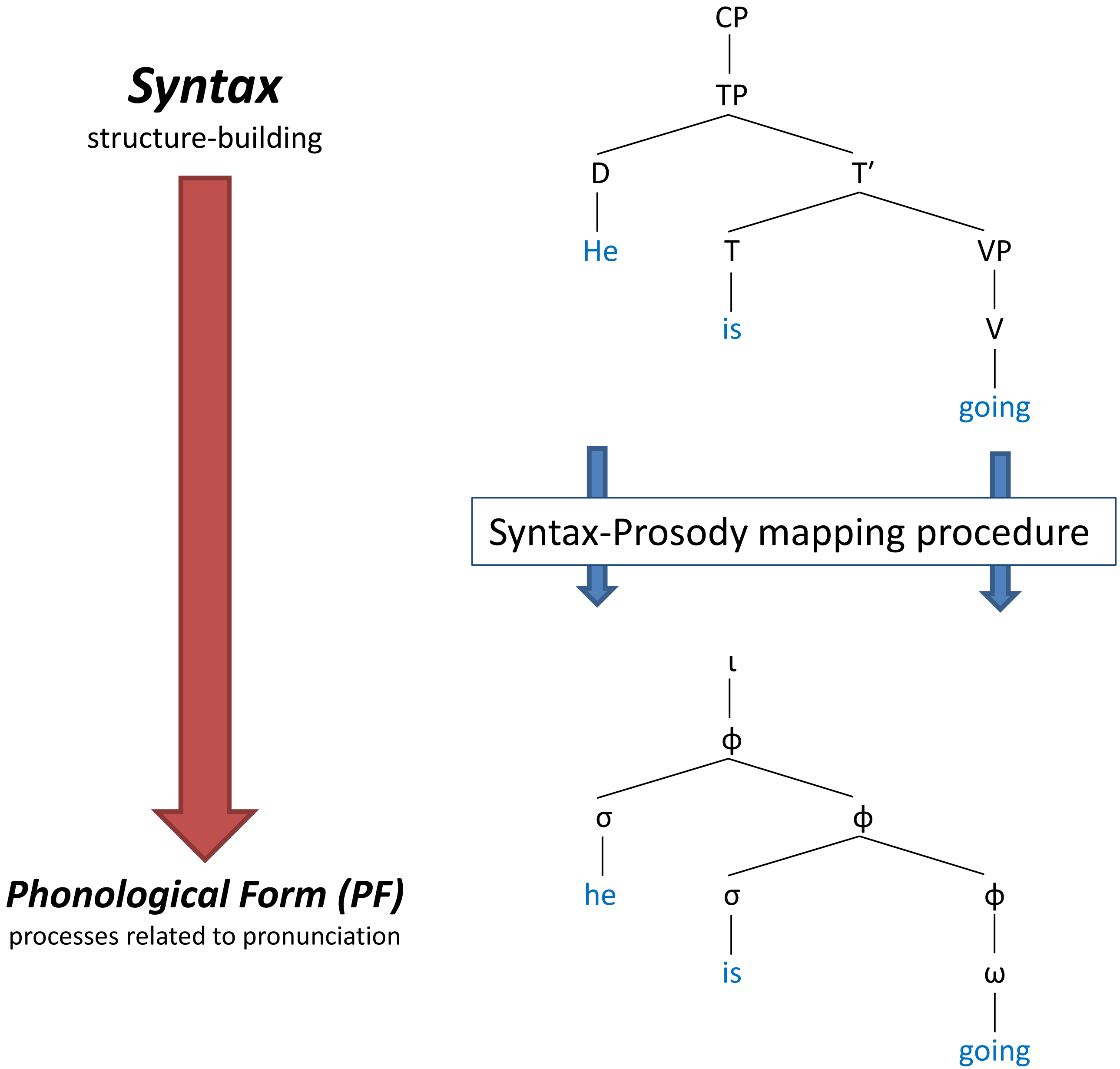


Crash course in Prosodic Theory

Recall: the inverted Y-model of Grammar



Crash course in Prosodic Theory



Crash course in Prosodic Theory

❖ Syntax-Prosody mapping is regulated by the MATCH rules (cf. Selkirk 2011)

[1] **Match(CP, ι)**

Map a CP node to an intonation phrase (ι)

[2] **Match(XP, φ)**

Putting CP aside, map a non-terminal syntactic node (XP, X′) to a phonological phrase (φ)

[3] **Match(X, ω)**

Map a terminal syntactic node corresponding to a **lexical** category to a prosodic word (ω)

Lexical = N, A, V

Functional = D, T, P, C, weak pronouns, cliticisable verbs

Crash course in Prosodic Theory

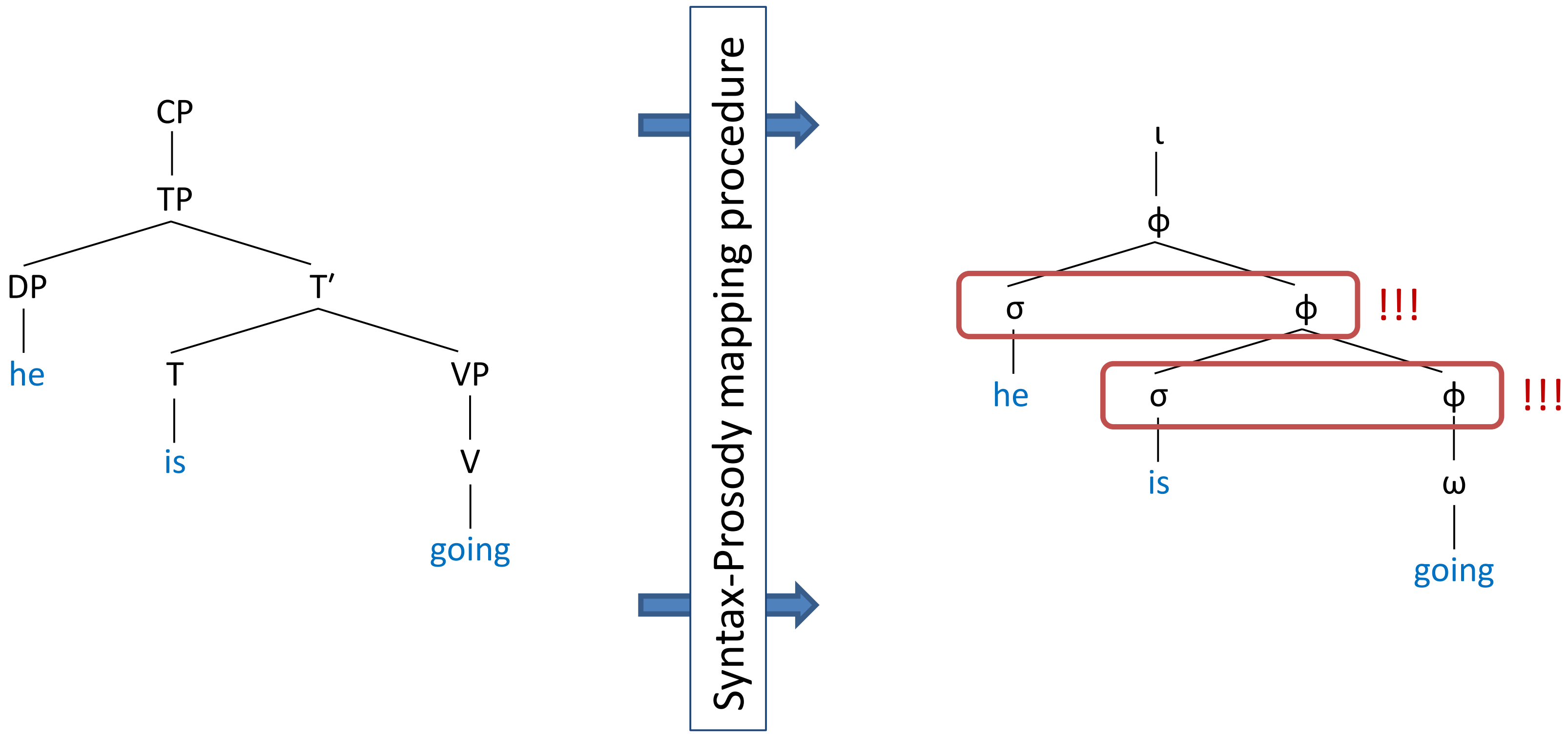
- The prosodic structure obtained from applying the MATCH rules is a **faithful** structure, as it doesn't deviate from its input (namely, a syntactic tree)

- But there are often independent phonological demands that require mapping to be **unfaithful**

Crash course in Prosodic Theory

STRONGSTART

A phonological constituent optimally begins with a leftmost daughter constituent which is not lower in the prosodic hierarchy than the constituent that immediately follows it.



Crash course in Prosodic Theory

MATCH(ω, Lex)

Every ω must contain an instance of a lexical word.

Isn't this the same as our previous rule, **MATCH(X, ω)**?

Not quite. MATCH(ω, Lex) prevents "overmatching", whereas the main Match rules are violated when "undermatching" occurs.

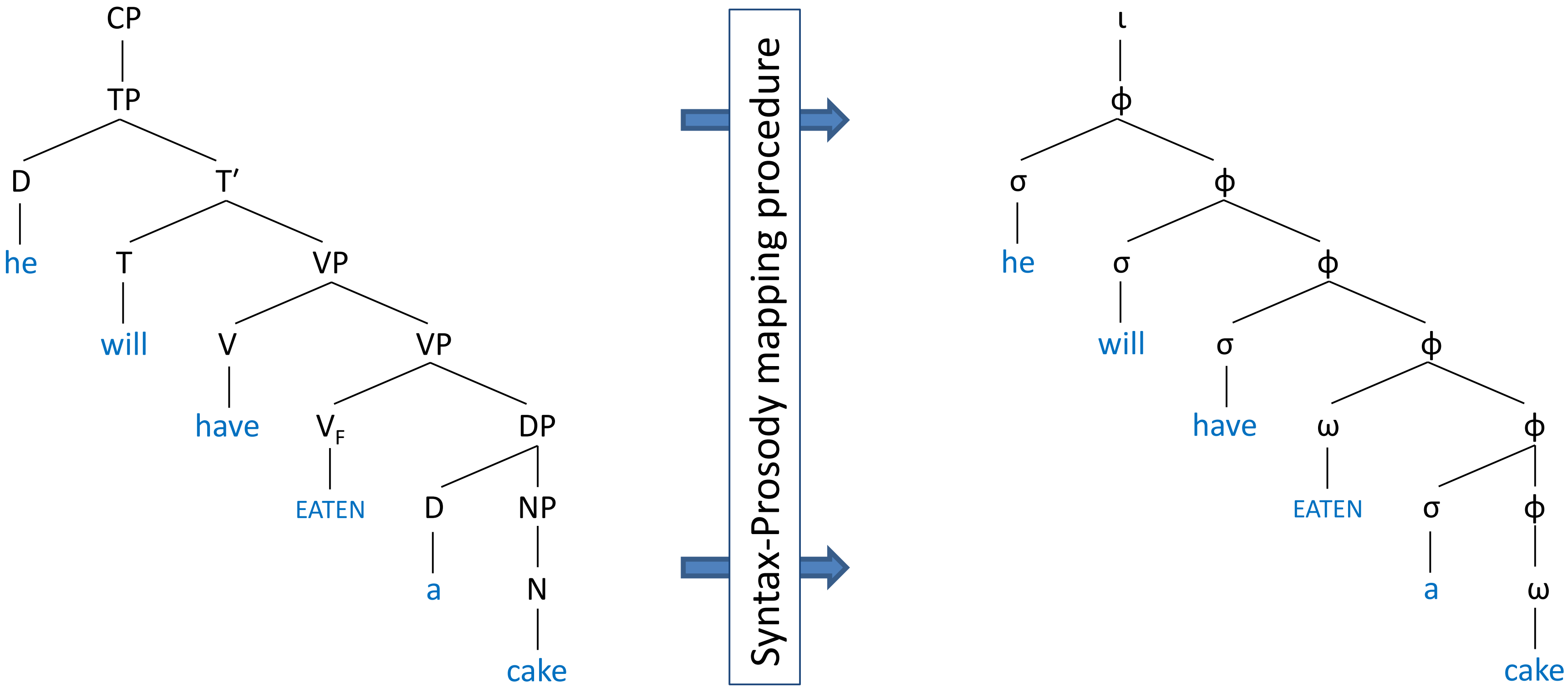
(*this will make more sense very shortly…*)

ALIGN(FOC, ϕ)

Align the left and right edges of a focused element with the left and right edges of a ϕ

Implicit rule (adopted by Weir): No prosodic candidate can show "redundant" ϕs, i.e. $(_{\phi} (_{\phi}$ blah$))$

Task: drawing the corresponding prosodic tree



$[_\iota\ [_\phi\ (_\sigma$ he$)\ [_\phi\ (_\sigma$ will$)\ [_\phi\ (_\sigma$ have$)\ [_\phi\ [_\omega$ eaten$]\ [_\phi\ (_\sigma$ a$)\ [_\phi\ [_\omega$ cake $]]]]]]]$

Input: $[_{CP}\ [_{TP}\ [_{D}$ he$]\ [_{T'}\ [_{T}$ will$]\ [_{VP}\ [_{V}$ have$]\ [_{VP}\ [_{V}$ EATEN$]_{F}\ [_{DP}\ [_{D}$ a$]\ [_{NP}\ [_{N}$ cake$]]]]]]]]$

	ALIGN(Foc, Φ)	MATCH(ω, Lex)	STRONGSTART	MATCH(S,P)
$[_{\iota}\ [_{\phi}\ (_{\sigma}$ he$)\ [_{\phi}\ (_{\sigma}$ will$)\ [_{\phi}\ (_{\sigma}$ have$)\ [_{\phi}\ [_{\omega}$ eaten$]\ [_{\phi}\ (_{\sigma}$ a$)\ [_{\phi}\ [_{\omega}$ cake$]]]]]]]]$				
$[_{\iota}\ [_{\phi}\ (_{\sigma}$ he$)\ [_{\phi}\ (_{\sigma}$ will$)\ [_{\phi}\ (_{\sigma}$ have$)\ [_{\phi}\ (_{\sigma}$ eaten$)\ [_{\phi}\ (_{\sigma}$ a$)\ [_{\phi}\ (_{\sigma}$ cake$)]]]]]]]$				
$[_{\iota}\ [_{\phi}\ [_{\phi}\ [_{\omega}$ he$]]\ [_{\phi}\ [_{\phi}\ [_{\omega}$ will$]]\ [_{\phi}\ [_{\phi}\ [_{\omega}$ have$]]\ [_{\phi}\ [_{\phi}\ [_{\omega}$ eaten$]]\ [_{\phi}\ [_{\phi}\ [_{\omega}$ a$]]\ [_{\phi}\ [_{\omega}$ cake$]]]]]]]]$				
$[_{\iota}\ [_{\phi}\ (_{\sigma}$ he$)\ [_{\phi}\ (_{\sigma}$ will$)\ [_{\phi}\ (_{\sigma}$ have$)\ [_{\phi}\ [_{\phi}\ [_{\omega}$ eaten$]]\ [_{\phi}\ (_{\sigma}$ a$)\ [_{\phi}\ [_{\omega}$ cake$]]]]]]]]$				
$[_{\iota}\ [_{\phi}\ (_{\sigma}$ he'll$)\ [_{\phi}\ (_{\sigma}$ have$)\ [_{\phi}\ [_{\phi}\ [_{\omega}$ eaten$]]\ [_{\phi}\ (_{\sigma}$ a$)\ [_{\phi}\ [_{\omega}$ cake$]]]]]]]$				
$[_{\iota}\ [_{\phi}\ (_{\sigma}$ *he'll've*$)\ [_{\phi}\ [_{\phi}\ [_{\omega}$ eaten$]]\ [_{\phi}\ (_{\sigma}$ a$)\ [_{\phi}\ [_{\omega}$ cake$]]]]]]$				

STRONGSTART

A phonological constituent optimally begins with a leftmost daughter constituent which is not lower in the prosodic hierarchy than the constituent that immediately follows it.

MATCH(ω, Lex)

Every ω must contain an instance of a lexical word.

ALIGN(Foc, ϕ)

Align the left and right edges of a focused element with the left and right edges of a ϕ

Input: $[_{CP}\ [_{TP}\ [_{D}$ he$]\ [_{T'}\ [_{T}$ will$]\ [_{VP}\ [_{V}$ have$]\ [_{VP}\ [_{V}$ EATEN$]_{F}\ [_{DP}\ [_{D}$ a$]\ [_{NP}\ [_{N}$ cake$]]]]]]]]$

	ALIGN(Foc, Φ)	MATCH(ω, Lex)	STRONGSTART	MATCH(S,P)
$[_\iota\ [_\phi\ (_\sigma$ he$)\ [_\phi\ (_\sigma$ will$)\ [_\phi\ (_\sigma$ have$)\ [_\phi\ [_\omega$ eaten$]\ [_\phi\ (_\sigma$ a$)\ [_\phi\ [_\omega$ cake$]]]]]]]]$	*		*****	
$[_\iota\ [_\phi\ (_\sigma$ he$)\ [_\phi\ (_\sigma$ will$)\ [_\phi\ (_\sigma$ have$)\ [_\phi\ (_\sigma$ eaten$)\ [_\phi\ (_\sigma$ a$)\ [_\phi\ (_\sigma$ cake$)]]]]]]]$				
$[_\iota\ [_\phi\ [_\phi\ [_\omega$ he$]]\ [_\phi\ [_\phi\ [_\omega$ will$]]\ [_\phi\ [_\phi\ [_\omega$ have$]]\ [_\phi\ [_\phi\ [_\omega$ eaten$]]\ [_\phi\ [_\phi\ [_\omega$ a$]]\ [_\phi\ [_\omega$ cake$]]]]]]]]$				
$[_\iota\ [_\phi\ (_\sigma$ he$)\ [_\phi\ (_\sigma$ will$)\ [_\phi\ (_\sigma$ have$)\ [_\phi\ [_\phi\ [_\omega$ eaten$]]\ [_\phi\ (_\sigma$ a$)\ [_\phi\ [_\omega$ cake$]]]]]]]]$				
$[_\iota\ [_\phi\ (_\sigma$ he’ll$)\ [_\phi\ (_\sigma$ have$)\ [_\phi\ [_\phi\ [_\omega$ eaten$]]\ [_\phi\ (_\sigma$ a$)\ [_\phi\ [_\omega$ cake$]]]]]]]$				
$[_\iota\ [_\phi\ (_\sigma$ *he’ll’ve*$)\ [_\phi\ [_\phi\ [_\omega$ eaten$]]\ [_\phi\ (_\sigma$ a$)\ [_\phi\ [_\omega$ cake$]]]]]]$				

STRONGSTART
A phonological constituent optimally begins with a leftmost daughter constituent which is not lower in the prosodic hierarchy than the constituent that immediately follows it.

MATCH(ω, Lex)
Every ω must contain an instance of a lexical word.

ALIGN(Foc, ϕ)
Align the left and right edges of a focused element with the left and right edges of a ϕ

Input: [$_{CP}$ [$_{TP}$ [$_{D}$ he] [$_{T'}$ [$_{T}$ will] [$_{VP}$ [$_{V}$ have] [$_{VP}$ [$_{V}$ EATEN]$_{F}$ [$_{DP}$ [$_{D}$ a] [$_{NP}$ [$_{N}$ cake]]]]]]]]

	Align(Foc, Φ)	Match(ω, Lex)	StrongStart	Match(S,P)
[$_{\iota}$ [$_{\phi}$ ($_{\sigma}$ he) [$_{\phi}$ ($_{\sigma}$ will) [$_{\phi}$ ($_{\sigma}$ have) [$_{\phi}$ [$_{\omega}$ eaten] [$_{\phi}$ ($_{\sigma}$ a) [$_{\phi}$ [$_{\omega}$ cake]]]]]]]]	*		*****	
[$_{\iota}$ [$_{\phi}$ ($_{\sigma}$ he) [$_{\phi}$ ($_{\sigma}$ will) [$_{\phi}$ ($_{\sigma}$ have) [$_{\phi}$ ($_{\sigma}$ eaten) [$_{\phi}$ ($_{\sigma}$ a) [$_{\phi}$ ($_{\sigma}$ cake)]]]]]]]	*		****	**
[$_{\iota}$ [$_{\phi}$ [$_{\phi}$ [$_{\omega}$ he]] [$_{\phi}$ [$_{\phi}$ [$_{\omega}$ will]] [$_{\phi}$ [$_{\phi}$ [$_{\omega}$ have]] [$_{\phi}$ [$_{\phi}$ [$_{\omega}$ eaten]] [$_{\phi}$ [$_{\phi}$ [$_{\omega}$ a]] [$_{\phi}$ [$_{\omega}$ cake]]]]]]]]				
[$_{\iota}$ [$_{\phi}$ ($_{\sigma}$ he) [$_{\phi}$ ($_{\sigma}$ will) [$_{\phi}$ ($_{\sigma}$ have) [$_{\phi}$ [$_{\phi}$ [$_{\omega}$ eaten]] [$_{\phi}$ ($_{\sigma}$ a) [$_{\phi}$ [$_{\omega}$ cake]]]]]]]]				
[$_{\iota}$ [$_{\phi}$ ($_{\sigma}$ he'll) [$_{\phi}$ ($_{\sigma}$ have) [$_{\phi}$ [$_{\phi}$ [$_{\omega}$ eaten]] [$_{\phi}$ ($_{\sigma}$ a) [$_{\phi}$ [$_{\omega}$ cake]]]]]]]				
[$_{\iota}$ [$_{\phi}$ ($_{\sigma}$ *he'll've*) [$_{\phi}$ [$_{\phi}$ [$_{\omega}$ eaten]] [$_{\phi}$ ($_{\sigma}$ a) [$_{\phi}$ [$_{\omega}$ cake]]]]]]				

StrongStart

A phonological constituent optimally begins with a leftmost daughter constituent which is <u>not lower</u> in the prosodic hierarchy than the constituent that immediately follows it.

Match(ω, Lex)

Every ω must contain an instance of a lexical word.

Align(Foc, ϕ)

Align the left and right edges of a focused element with the left and right edges of a ϕ

Input: [$_{CP}$ [$_{TP}$ [$_{D}$ he] [$_{T'}$ [$_{T}$ will] [$_{VP}$ [$_{V}$ have] [$_{VP}$ [$_{V}$ EATEN]$_{F}$ [$_{DP}$ [$_{D}$ a] [$_{NP}$ [$_{N}$ cake]]]]]]]]

	ALIGN(FOC, Φ)	MATCH(ω, Lex)	STRONGSTART	MATCH(S,P)
[$_\iota$ [$_\phi$ ($_\sigma$ he) [$_\phi$ ($_\sigma$ will) [$_\phi$ ($_\sigma$ have) [$_\phi$ [$_\omega$ eaten] [$_\phi$ ($_\sigma$ a) [$_\phi$ [$_\omega$ cake]]]]]]]]	*		*****	
[$_\iota$ [$_\phi$ ($_\sigma$ he) [$_\phi$ ($_\sigma$ will) [$_\phi$ ($_\sigma$ have) [$_\phi$ ($_\sigma$ eaten) [$_\phi$ ($_\sigma$ a) [$_\phi$ ($_\sigma$ cake)]]]]]]]	*		****	**
[$_\iota$ [$_\phi$ [$_\phi$ [$_\omega$ he]] [$_\phi$ [$_\phi$ [$_\omega$ will]] [$_\phi$ [$_\phi$ [$_\omega$ have]] [$_\phi$ [$_\phi$ [$_\omega$ eaten]] [$_\phi$ [$_\phi$ [$_\omega$ a]] [$_\phi$ [$_\omega$ cake]]]]]]]		****		
[$_\iota$ [$_\phi$ ($_\sigma$ he) [$_\phi$ ($_\sigma$ will) [$_\phi$ ($_\sigma$ have) [$_\phi$ [$_\phi$ [$_\omega$ eaten]] [$_\phi$ ($_\sigma$ a) [$_\phi$ [$_\omega$ cake]]]]]]]				
[$_\iota$ [$_\phi$ ($_\sigma$ he'll) [$_\phi$ ($_\sigma$ have) [$_\phi$ [$_\phi$ [$_\omega$ eaten]] [$_\phi$ ($_\sigma$ a) [$_\phi$ [$_\omega$ cake]]]]]]				
[$_\iota$ [$_\phi$ ($_\sigma$ *he'll've*) [$_\phi$ [$_\phi$ [$_\omega$ eaten]] [$_\phi$ ($_\sigma$ a) [$_\phi$ [$_\omega$ cake]]]]]				

STRONGSTART

A phonological constituent optimally begins with a leftmost daughter constituent which is not lower in the prosodic hierarchy than the constituent that immediately follows it.

MATCH(ω, Lex)

Every ω must contain an instance of a lexical word.

ALIGN(FOC, ϕ)

Align the left and right edges of a focused element with the left and right edges of a ϕ

Input: $[_{CP}\ [_{TP}\ [_D$ he$]\ [_{T'}\ [_T$ will$]\ [_{VP}\ [_V$ have$]\ [_{VP}\ [_V$ EATEN$]_F\ [_{DP}\ [_D$ a$]\ [_{NP}\ [_N$ cake$]]]]]]]]$

	ALIGN(FOC, Φ)	MATCH(ω, LEX)	STRONGSTART	MATCH(S,P)
$[_\iota\ [_\phi\ (_\sigma$ he$)\ [_\phi\ (_\sigma$ will$)\ [_\phi\ (_\sigma$ have$)\ [_\phi\ [_\omega$ eaten$]\ [_\phi\ (_\sigma$ a$)\ [_\phi\ [_\omega$ cake$]]]]]]]]$	*		*****	
$[_\iota\ [_\phi\ (_\sigma$ he$)\ [_\phi\ (_\sigma$ will$)\ [_\phi\ (_\sigma$ have$)\ [_\phi\ (_\sigma$ eaten$)\ [_\phi\ (_\sigma$ a$)\ [_\phi\ (_\sigma$ cake$)]]]]]]]$	*		****	**
$[_\iota\ [_\phi\ [_\phi\ [_\omega$ he$]]\ [_\phi\ [_\phi\ [_\omega$ will$]]\ [_\phi\ [_\phi\ [_\omega$ have$]]\ [_\phi\ [_\phi\ [_\omega$ eaten$]]\ [_\phi\ [_\phi\ [_\omega$ a$]]\ [_\phi\ [_\omega$ cake$]]]]]]]]$		****		
$[_\iota\ [_\phi\ (_\sigma$ he$)\ [_\phi\ (_\sigma$ will$)\ [_\phi\ (_\sigma$ have$)\ [_\phi\ [_\phi\ [_\omega$ eaten$]]\ [_\phi\ (_\sigma$ a$)\ [_\phi\ [_\omega$ cake$]]]]]]]]$			****	
$[_\iota\ [_\phi\ (_\sigma$ he'll$)\ [_\phi\ (_\sigma$ have$)\ [_\phi\ [_\phi\ [_\omega$ eaten$]]\ [_\phi\ (_\sigma$ a$)\ [_\phi\ [_\omega$ cake$]]]]]]]$				
$[_\iota\ [_\phi\ (_\sigma$ *he'll've*$)\ [_\phi\ [_\phi\ [_\omega$ eaten$]]\ [_\phi\ (_\sigma$ a$)\ [_\phi\ [_\omega$ cake$]]]]]]$				

STRONGSTART
A phonological constituent optimally begins with a leftmost daughter constituent which is <u>not lower</u> in the prosodic hierarchy than the constituent that immediately follows it.

MATCH(ω, LEX)
Every ω must contain an instance of a lexical word.

ALIGN(FOC, ϕ)
Align the left and right edges of a focused element with the left and right edges of a ϕ

Input: [$_{CP}$ [$_{TP}$ [$_{D}$ he] [$_{T'}$ [$_{T}$ will] [$_{VP}$ [$_{V}$ have] [$_{VP}$ [$_{V}$ EATEN]$_{F}$ [$_{DP}$ [$_{D}$ a] [$_{NP}$ [$_{N}$ cake]]]]]]]]

	ALIGN(FOC, Φ)	MATCH(ω, Lex)	STRONGSTART	MATCH(S,P)
[$_ι$ [$_φ$ ($_σ$ he) [$_φ$ ($_σ$ will) [$_φ$ ($_σ$ have) [$_φ$ [$_ω$ eaten] [$_φ$ ($_σ$ a) [$_φ$ [$_ω$ cake]]]]]]]	*		*****	
[$_ι$ [$_φ$ ($_σ$ he) [$_φ$ ($_σ$ will) [$_φ$ ($_σ$ have) [$_φ$ ($_σ$ eaten) [$_φ$ ($_σ$ a) [$_φ$ ($_σ$ cake)]]]]]]]	*		****	**
[$_ι$ [$_φ$ [$_φ$ [$_ω$ he]] [$_φ$ [$_φ$ [$_ω$ will]] [$_φ$ [$_φ$ [$_ω$ have]] [$_φ$ [$_φ$ [$_ω$ eaten]] [$_φ$ [$_φ$ [$_ω$ a]] [$_φ$ [$_ω$ cake]]]]]]]		****		
[$_ι$ [$_φ$ ($_σ$ he) [$_φ$ ($_σ$ will) [$_φ$ ($_σ$ have) [$_φ$ [$_φ$ [$_ω$ eaten]] [$_φ$ ($_σ$ a) [$_φ$ [$_ω$ cake]]]]]]]]			****	
[$_ι$ [$_φ$ ($_σ$ he'll) [$_φ$ ($_σ$ have) [$_φ$ [$_φ$ [$_ω$ eaten]] [$_φ$ ($_σ$ a) [$_φ$ [$_ω$ cake]]]]]]]			***	*
[$_ι$ [$_φ$ ($_σ$ *he'll've*) [$_φ$ [$_φ$ [$_ω$ eaten]] [$_φ$ ($_σ$ a) [$_φ$ [$_ω$ cake]]]]]]				

STRONGSTART
A phonological constituent optimally begins with a leftmost daughter constituent which is <u>not lower</u> in the prosodic hierarchy than the constituent that immediately follows it.

MATCH(ω, Lex)
Every ω must contain an instance of a lexical word.

ALIGN(FOC, ϕ)
Align the left and right edges of a focused element with the left and right edges of a ϕ

Input: $[_{CP}\ [_{TP}\ [_{D}\ \text{he}]\ [_{T'}\ [_{T}\ \text{will}]\ [_{VP}\ [_{V}\ \text{have}]\ [_{VP}\ [_{V}\ \text{EATEN}]_{F}\ [_{DP}\ [_{D}\ \text{a}]\ [_{NP}\ [_{N}\ \text{cake}]]]]]]]]$

	ALIGN(FOC, Φ)	MATCH(ω, LEX)	STRONGSTART	MATCH(S,P)
$[_\iota\ [_\phi\ (_\sigma\ \text{he})\ [_\phi\ (_\sigma\ \text{will})\ [_\phi\ (_\sigma\ \text{have})\ [_\phi\ [_\omega\ \text{eaten}]\ [_\phi\ (_\sigma\ \text{a})\ [_\phi\ [_\omega\ \text{cake}]]]]]]]]$	*		*****	
$[_\iota\ [_\phi\ (_\sigma\ \text{he})\ [_\phi\ (_\sigma\ \text{will})\ [_\phi\ (_\sigma\ \text{have})\ [_\phi\ (_\sigma\ \text{eaten})\ [_\phi\ (_\sigma\ \text{a})\ [_\phi\ (_\sigma\ \text{cake})]]]]]]]$	*		****	**
$[_\iota\ [_\phi\ [_\phi\ [_\omega\ \text{he}]]\ [_\phi\ [_\phi\ [_\omega\ \text{will}]]\ [_\phi\ [_\phi\ [_\omega\ \text{have}]]\ [_\phi\ [_\phi\ [_\omega\ \text{eaten}]]\ [_\phi\ [_\phi\ [_\omega\ \text{a}]]\ [_\phi\ [_\omega\ \text{cake}]]]]]]]]$		****		
$[_\iota\ [_\phi\ (_\sigma\ \text{he})\ [_\phi\ (_\sigma\ \text{will})\ [_\phi\ (_\sigma\ \text{have})\ [_\phi\ [_\phi\ [_\omega\ \text{eaten}]]\ [_\phi\ (_\sigma\ \text{a})\ [_\phi\ [_\omega\ \text{cake}]]]]]]]]$			****	
$[_\iota\ [_\phi\ (_\sigma\ \text{he'll})\ [_\phi\ (_\sigma\ \text{have})\ [_\phi\ [_\phi\ [_\omega\ \text{eaten}]]\ [_\phi\ (_\sigma\ \text{a})\ [_\phi\ [_\omega\ \text{cake}]]]]]]]$			***	*
$[_\iota\ [_\phi\ (_\sigma\ \textit{he'll've})\ [_\phi\ [_\phi\ [_\omega\ \text{eaten}]]\ [_\phi\ (_\sigma\ \text{a})\ [_\phi\ [_\omega\ \text{cake}]]]]]]$			**	**

STRONGSTART

A phonological constituent optimally begins with a leftmost daughter constituent which is <u>not lower</u> in the prosodic hierarchy than the constituent that immediately follows it.

MATCH(ω, LEX)

Every ω must contain an instance of a lexical word.

ALIGN(FOC, ϕ)

Align the left and right edges of a focused element with the left and right edges of a ϕ

Input: $[_{CP}$ $[_{TP}$ $[_{D}$ he] $[_{T'}$ $[_{T}$ will] $[_{VP}$ $[_{V}$ have] $[_{VP}$ $[_{V}$ EATEN]$_{F}$ $[_{DP}$ $[_{D}$ a] $[_{NP}$ $[_{N}$ cake]]]]]]]

	ALIGN(FOC, Φ)	MATCH(ω, LEX)	STRONGSTART	MATCH(S,P)
$[_ι$ $[_φ$ $(_σ$ he) $[_φ$ $(_σ$ will) $[_φ$ $(_σ$ have) $[_φ$ $[_ω$ eaten] $[_φ$ $(_σ$ a) $[_φ$ $[_ω$ cake]]]]]]]]	*		***!**	
$[_ι$ $[_φ$ $(_σ$ he) $[_φ$ $(_σ$ will) $[_φ$ $(_σ$ have) $[_φ$ $(_σ$ eaten) $[_φ$ $(_σ$ a) $[_φ$ $(_σ$ cake)]]]]]]]]	*		***!*	**
$[_ι$ $[_φ$ $[_φ$ $[_ω$ he]] $[_φ$ $[_φ$ $[_ω$ will]] $[_φ$ $[_φ$ $[_ω$ have]] $[_φ$ $[_φ$ $[_ω$ eaten]] $[_φ$ $[_φ$ $[_ω$ a]] $[_φ$ $[_ω$ cake]]]]]]]]		*!***		
$[_ι$ $[_φ$ $(_σ$ he) $[_φ$ $(_σ$ will) $[_φ$ $(_σ$ have) $[_φ$ $[_φ$ $[_ω$ eaten]] $[_φ$ $(_σ$ a) $[_φ$ $[_ω$ cake]]]]]]]]			***!*	
$[_ι$ $[_φ$ $(_σ$ he'll) $[_φ$ $(_σ$ have) $[_φ$ $[_φ$ $[_ω$ eaten]] $[_φ$ $(_σ$ a) $[_φ$ $[_ω$ cake]]]]]]]			***!	*
☞ $[_ι$ $[_φ$ $(_σ$ *he'll've*) $[_φ$ $[_φ$ $[_ω$ eaten]] $[_φ$ $(_σ$ a) $[_φ$ $[_ω$ cake]]]]]]			**	**

STRONGSTART
A phonological constituent optimally begins with a leftmost daughter constituent which is <u>not lower</u> in the prosodic hierarchy than the constituent that immediately follows it.

MATCH(ω, LEX)
Every ω must contain an instance of a lexical word.

ALIGN(FOC, φ)
Align the left and right edges of a focused element with the left and right edges of a φ

Back to Weir's analysis of subject drop…

Recall:

Subject drop is permitted with cliticisable auxiliary verbs only if:

(i) the verb is realised in its clitic form

(ii) the full form of the verb has the clitic form of negation (-*n′t*) attached to it

(iii) the full form of the verb is contrastively focused

Back to Weir's analysis of subject drop…

Recall:

Subject drop is permitted with cliticisable auxiliary verbs only if:

(i) the verb is realised in its clitic form

(ii) the full form of the verb has the clitic form of negation (-*n'**t*) attached to it

(iii) the full form of the verb is contrastively focused

One more prosodic constraint is required:

MAX: Don't delete elements.

Input: $[_{CP}\ [_{TP}\ [_{DP}$ he$]\ [_{T'}\ [_{T}$ is$]\ [_{VP}\ [_{V}$ going$]]]]]$

		StrongStart	Match(ω, Lex)	Max	Match(S, P)
a.	$[_{\iota}\ [_{\phi}\ (_{\sigma}$ he$)\ [_{\phi}\ (_{\sigma}$ is$)\ [_{\phi}\ [_{\omega}$ going$]]]]]$	**			
b.	$[_{\iota}\ [_{\phi}\ [_{\phi}\ [_{w}$ he$]]\ [_{\phi}\ [_{\phi}\ [_{w}$ is$]]\ [_{\phi}\ [_{\omega}$ going$]]]]]$		**		
c.	$[_{\iota}\ [_{\phi}\ (_{\sigma}$ he's$)\ [_{\phi}\ [_{\omega}$ going$]]]]$	*			*
d.	$[_{\iota}\ [_{\phi}\ [_{\omega}$ he's$_{\sigma}$ going$_{Ft}]]]$	*			**
e.	$[_{\iota}\ [_{\phi}\ [_{\omega}$ is$_{\sigma}$ going$_{Ft}]]]$	*		*	*
f.	$[_{\iota}\ [_{\phi}\ [_{\omega}$ 'sgoing$]]]$			*	*
g.	$[_{\iota}\ [_{\phi}\ [_{\omega}$ going$]]]$			**	

The dashed line means two constraints are equally-ranked

StrongStart

A phonological constituent optimally begins with a leftmost daughter constituent which is <u>not lower</u> in the prosodic hierarchy than the constituent that immediately follows it.

Match(ω, Lex)

Every ω must contain an instance of a lexical word.

Max: don't delete elements

Input: $[_{CP}\ [_{TP}\ [_{DP}\ \text{he}]\ [_{T'}\ [_{T}\ \text{is}]\ [_{VP}\ [_{V}\ \text{going}]]]]]$

	StrongStart	Match(ω, Lex)	Max	Match(S, P)
a. $[_{\iota}\ [_{\phi}\ (_{\sigma}\ \text{he})\ [_{\phi}\ (_{\sigma}\ \text{is})\ [_{\phi}\ [_{\omega}\ \text{going}]]]]]$				
b. $[_{\iota}\ [_{\phi}\ [_{\phi}\ [_{w}\ \text{he}]]\ [_{\phi}\ [_{\phi}\ [_{w}\ \text{is}]]\ [_{\phi}\ [_{\omega}\ \text{going}]]]]]$				
c. $[_{\iota}\ [_{\phi}\ (_{\sigma}\ \text{he's})\ [_{\phi}\ [_{\omega}\ \text{going}]]]]$				
d. $[_{\iota}\ [_{\phi}\ [_{\omega}\ \text{he's}_{\sigma}\ \text{going}_{Ft}]]]$				
e. $[_{\iota}\ [_{\phi}\ [_{\omega}\ \text{is}_{\sigma}\ \text{going}_{Ft}]]]$				
f. $[_{\iota}\ [_{\phi}\ [_{\omega}\ \text{'sgoing}]]]$				
g. $[_{\iota}\ [_{\phi}\ [_{\omega}\ \text{going}]]]$				

The dashed line means two constraints are equally-ranked

StrongStart
A phonological constituent optimally begins with a leftmost daughter constituent which is <u>not lower</u> in the prosodic hierarchy than the constituent that immediately follows it.

Match(ω, Lex)
Every ω must contain an instance of a lexical word.

Max: don't delete elements

Input: $[_{CP}\ [_{TP}\ [_{DP}\ \text{he}]\ [_{T'}\ [_{T}\ \text{is}]\ [_{VP}\ [_{V}\ \text{going}]]]]]$

		StrongStart	Match(ω, Lex)	Max	Match(S, P)
a.	$[_{\iota}\ [_{\phi}\ (_{\sigma}\ \text{he})\ [_{\phi}\ (_{\sigma}\ \text{is})\ [_{\phi}\ [_{\omega}\ \text{going}]]]]]$	**			
b.	$[_{\iota}\ [_{\phi}\ [_{\phi}\ [_{w}\ \text{he}]]\ [_{\phi}\ [_{\phi}\ [_{w}\ \text{is}]]\ [_{\phi}\ [_{\omega}\ \text{going}]]]]]$		**		
c.	$[_{\iota}\ [_{\phi}\ (_{\sigma}\ \text{he's})\ [_{\phi}\ [_{\omega}\ \text{going}]]]]$	*			*
d.	$[_{\iota}\ [_{\phi}\ [_{\omega}\ \text{he's}_{\sigma}\ \text{going}_{Ft}]]]$	*			**
e.	$[_{\iota}\ [_{\phi}\ [_{\omega}\ \text{is}_{\sigma}\ \text{going}_{Ft}]]]$	*		*	**
f.	$[_{\iota}\ [_{\phi}\ [_{\omega}\ \text{'sgoing}]]]$			*	**
g.	$[_{\iota}\ [_{\phi}\ [_{\omega}\ \text{going}]]]$			**	**

Input: $[_{CP}\ [_{TP}\ [_{DP}\ \text{he}]\ [_{T'}\ [_{T}\ \text{is}]\ [_{VP}\ [_{V}\ \text{going}]]]]]$

	StrongStart	Match(ω, Lex)	Max	Match(S, P)
a. $[_{\iota}\ [_{\phi}\ (_{\sigma}\ \text{he})\ [_{\phi}\ (_{\sigma}\ \text{is})\ [_{\phi}\ [_{\omega}\ \text{going}]]]]]$	*!*			
b. $[_{\iota}\ [_{\phi}\ [_{\phi}\ [_{w}\ \text{he}]]\ [_{\phi}\ [_{\phi}\ [_{w}\ \text{is}]]\ [_{\phi}\ [_{\omega}\ \text{going}]]]]]$		*!*		
c. $[_{\iota}\ [_{\phi}\ (_{\sigma}\ \text{he's})\ [_{\phi}\ [_{\omega}\ \text{going}]]]]$	*!			*
d. $[_{\iota}\ [_{\phi}\ [_{\omega}\ \text{he's}_{\sigma}\ \text{going}_{Ft}]]]$	*!			**
e. $[_{\iota}\ [_{\phi}\ [_{\omega}\ \text{is}_{\sigma}\ \text{going}_{Ft}]]]$	*!		*	**
☞ f. $[_{\iota}\ [_{\phi}\ [_{\omega}\ \text{'sgoing}]]]$			*	**
g. $[_{\iota}\ [_{\phi}\ [_{\omega}\ \text{going}]]]$			**!	**

Subject drop is permitted with cliticisable auxiliary verbs only if:

(i) the verb is realised in its clitic form ✓

(ii) the full form of the verb has the clitic form of negation (-*n't*) attached to it

(iii) the full form of the verb is contrastively focused

Input: $[_{CP}\ [_{TP}\ [_{DP}\ \text{he}]\ [_{T'}\ [_{T}\ \text{is}]\ [_{VP}\ [_{V}\ \text{going}]]]]]$

	StrongStart	Match(ω, Lex)	Max	Match(S, P)
a. $[_{\iota}\ [_{\phi}\ (_{\sigma}\ \text{he})\ [_{\phi}\ (_{\sigma}\ \text{is})\ [_{\phi}\ [_{\omega}\ \text{going}]]]]]$	*!*			
b. $[_{\iota}\ [_{\phi}\ [_{\phi}\ [_{w}\ \text{he}]]\ [_{\phi}\ [_{\phi}\ [_{w}\ \text{is}]]\ [_{\phi}\ [_{\omega}\ \text{going}]]]]]$		*!*		
c. $[_{\iota}\ [_{\phi}\ (_{\sigma}\ \text{he's})\ [_{\phi}\ [_{\omega}\ \text{going}]]]]$	*!			*
d. $[_{\iota}\ [_{\phi}\ [_{\omega}\ \text{he's}_{\sigma}\ \text{going}_{Ft}]]]$	*!			**
e. $[_{\iota}\ [_{\phi}\ [_{\omega}\ \text{is}_{\sigma}\ \text{going}_{Ft}]]]$	*!		*	**
☞ f. $[_{\iota}\ [_{\phi}\ [_{\omega}\ \text{'sgoing}]]]$			*	**
g. $[_{\iota}\ [_{\phi}\ [_{\omega}\ \text{going}]]]$			**!	**

Subject drop is permitted with cliticisable auxiliary verbs only if:

(i) the verb is realised in its clitic form ✓

(ii) the full form of the verb has the clitic form of negation (-*n't*) attached to it

(iii) the full form of the verb is contrastively focused

Input: $[_{CP}\ [_{TP}\ [_{DP}\ \text{it}]\ [_{T'}\ [_{T}\ \text{will}]\ [_{NegP}\ [_{Neg}\ \text{not}]\ [_{VP}\ [_{V}\ \text{rain}]]]]]$

	STRONGSTART	MATCH(ω, Lex)	MAX	MATCH(S, P)
a. $[_\iota\ [_\phi\ (_\sigma\ \text{it})\ [_\phi\ (_\sigma\ \text{will})\ [_\phi\ (_\sigma\ \text{not})\ [_\phi\ [_\omega\ \text{rain}]]]]]]$				
b. $[_\iota\ [_\phi\ [_\phi\ [_\omega\ \text{it}]]\ [_\phi\ [_\phi\ [_\omega\ \text{will}]]\ [_\phi\ [_\phi\ [_\omega\ \text{not}]]\ [_\phi\ [_\omega\ \text{rain}]]]]]]$				
c. $[_\iota\ [_\phi\ (_\sigma\ \text{will})\ [_\phi\ (_\sigma\ \text{not})\ [_\phi\ [_\omega\ \text{rain}]]]]]$				
d. $[_\iota\ [_\phi\ (_\sigma\ \text{it})\ [_\phi\ (_\sigma\ \text{won't})\ [_\phi\ [_\omega\ \text{rain}]]]]]$				
e. $[_\iota\ [_\phi\ (_\sigma\ \text{won't})\ [_\phi\ [_\omega\ \text{rain}]]]]$				
f. $[_\iota\ [_\phi\ (_\sigma\ \text{not})\ [_\phi\ [_\omega\ \text{rain}]]]]$				
g. $[_\iota\ [_\phi\ (_\sigma\ \text{it})\ [_\phi\ [_\omega\ \text{won't}_\sigma\ \text{rain}_{Ft}\ ]]]]$				
h. $[_\iota\ [_\phi\ [_\omega\ \text{won't}_\sigma\ \text{rain}_{Ft}\ ]]]]]$				
i. $[_\iota\ [_\phi\ [_\phi\ [_\omega\ \text{won't}]]\ [_\phi\ [_\omega\ \text{rain}]]]]$				

STRONGSTART
A phonological constituent optimally begins with a leftmost daughter constituent which is <u>not lower</u> in the prosodic hierarchy than the constituent that immediately follows it.

MATCH(ω, Lex)
Every ω must contain an instance of a lexical word.

MAX: don't delete elements

Input: $[_{CP}\ [_{TP}\ [_{DP}\ \text{it}]\ [_{T'}\ [_{T}\ \text{will}]\ [_{NegP}\ [_{Neg}\ \text{not}]\ [_{VP}\ [_{V}\ \text{rain}]]]]]]$

		StrongStart	Match(ω, Lex)	Max	Match(S, P)
a.	$[_\iota\ [_\phi\ (_\sigma\ \text{it})\ [_\phi\ (_\sigma\ \text{will})\ [_\phi\ (_\sigma\ \text{not})\ [_\phi\ [_\omega\ \text{rain}]]]]]]$	***			
b.	$[_\iota\ [_\phi\ [_\phi\ [_\omega\ \text{it}]]\ [_\phi\ [_\phi\ [_\omega\ \text{will}]]\ [_\phi\ [_\phi\ [_\omega\ \text{not}]]\ [_\phi\ [_\omega\ \text{rain}]]]]]]$		***		
c.	$[_\iota\ [_\phi\ (_\sigma\ \text{will})\ [_\phi\ (_\sigma\ \text{not})\ [_\phi\ [_\omega\ \text{rain}]]]]]$	**		*	*
d.	$[_\iota\ [_\phi\ (_\sigma\ \text{it})\ [_\phi\ (_\sigma\ \text{won't})\ [_\phi\ [_\omega\ \text{rain}]]]]]$	**			*
e.	$[_\iota\ [_\phi\ (_\sigma\ \text{won't})\ [_\phi\ [_\omega\ \text{rain}]]]]$	*		*	**
f.	$[_\iota\ [_\phi\ (_\sigma\ \text{not})\ [_\phi\ [_\omega\ \text{rain}]]]]$	*		**	**
g.	$[_\iota\ [_\phi\ (_\sigma\ \text{it})\ [_\phi\ [_\omega\ \text{won't}_\sigma\ \text{rain}_{Ft}\ ]]]]$	**			**
h.	$[_\iota\ [_\phi\ [_\omega\ \text{won't}_\sigma\ \text{rain}_{Ft}\ ]]]]]$	*		*	**
i.	$[_\iota\ [_\phi\ [_\phi\ [_\omega\ \text{won't}]]\ [_\phi\ [_\omega\ \text{rain}]]]]$		*	*	**

Input: $[_{CP}\ [_{TP}\ [_{DP}$ it$]\ [_{T'}\ [_{T}$ will$]\ [_{NegP}\ [_{Neg}$ not$]\ [_{VP}\ [_{V}$ rain$]]]]]$

		StrongStart	Match(ω, Lex)	Max	Match(S, P)
a.	$[_\iota\ [_\phi\ (_\sigma$ it$)\ [_\phi\ (_\sigma$ will$)\ [_\phi\ (_\sigma$ not$)\ [_\phi\ [_\omega$ rain$]]]]]]$	*!**			
b.	$[_\iota\ [_\phi\ [_\phi\ [_\omega$ it$]]\ [_\phi\ [_\phi\ [_\omega$ will$]]\ [_\phi\ [_\phi\ [_\omega$ not$]]\ [_\phi\ [_\omega$ rain$]]]]]]$		**!*		
c.	$[_\iota\ [_\phi\ (_\sigma$ will$)\ [_\phi\ (_\sigma$ not$)\ [_\phi\ [_\omega$ rain$]]]]]$	*!*		*	*
d.	$[_\iota\ [_\phi\ (_\sigma$ it$)\ [_\phi\ (_\sigma$ won't$)\ [_\phi\ [_\omega$ rain$]]]]]$	*!*			*
e.	$[_\iota\ [_\phi\ (_\sigma$ won't$)\ [_\phi\ [_\omega$ rain$]]]]$	*!		*	**
f.	$[_\iota\ [_\phi\ (_\sigma$ not$)\ [_\phi\ [_\omega$ rain$]]]]$	*!		**	**
g.	$[_\iota\ [_\phi\ (_\sigma$ it$)\ [_\phi\ [_\omega$ won't$_\sigma$ rain$_{Ft}$ $]]]]$	*!*			**
h.	$[_\iota\ [_\phi\ [_\omega$ won't$_\sigma$ rain$_{Ft}$ $]]]]]$	*!		*	**
☞ i.	$[_\iota\ [_\phi\ [_\phi\ [_\omega$ won't$]]\ [_\phi\ [_\omega$ rain$]]]]$		*	*	**

Input: $[_{CP}\ [_{TP}\ [_{DP}$ it$]$ $[_{T'}\ [_T$ will$]$ $[_{NegP}\ [_{Neg}$ not$]$ $[_{VP}\ [_V$ rain$]]]]]$

		STRONGSTART	MATCH(ω, Lex)	MAX	MATCH(S, P)
	a. $[_\iota\ [_\phi\ (_\sigma$ it$)$ $[_\phi\ (_\sigma$ will$)$ $[_\phi\ (_\sigma$ not$)$ $[_\phi\ [_\omega$ rain$]]]]]]$	*!**			
	b. $[_\iota\ [_\phi\ [_\phi\ [_\omega$ it$]]$ $[_\phi\ [_\phi\ [_\omega$ will$]]$ $[_\phi\ [_\phi\ [_\omega$ not$]]$ $[_\phi\ [_\omega$ rain$]]]]]]$		**!*		
	c. $[_\iota\ [_\phi\ (_\sigma$ will$)$ $[_\phi\ (_\sigma$ not$)$ $[_\phi\ [_\omega$ rain$]]]]]$	*!*		*	*
	d. $[_\iota\ [_\phi\ (_\sigma$ it$)$ $[_\phi\ (_\sigma$ won't$)$ $[_\phi\ [_\omega$ rain$]]]]]$	*!*			*
	e. $[_\iota\ [_\phi\ (_\sigma$ won't$)$ $[_\phi\ [_\omega$ rain$]]]]$	*!		*	**
	f. $[_\iota\ [_\phi\ (_\sigma$ not$)$ $[_\phi\ [_\omega$ rain$]]]]$	*!		**	**
	g. $[_\iota\ [_\phi\ (_\sigma$ it$)$ $[_\phi\ [_\omega$ won't$_\sigma$ rain$_{Ft}$ $]]]]$	*!*			**
	h. $[_\iota\ [_\phi\ [_\omega$ won't$_\sigma$ rain$_{Ft}$ $]]]]]$	*!		*	**
☞	i. $[_\iota\ [_\phi\ [_\phi\ [_\omega$ won't$]]$ $[_\phi\ [_\omega$ rain$]]]]$		*	*	**

Subject drop is permitted with cliticisable auxiliary verbs only if:

(i) the verb is realised in its clitic form ✓

(ii) the full form of the verb has the clitic form of negation (-*n't*) attached to it ✓

(iii) the full form of the verb is contrastively focused

Input: $[_{CP}\ [_{TP}\ [_{DP}$ it$]\ [_{T'}\ [_{T}$ will$]\ [_{NegP}\ [_{Neg}$ not$]\ [_{VP}\ [_{V}$ rain$]]]]]$

	StrongStart	Match(ω, Lex)	Max	Match(S, P)
a. $[_{\iota}\ [_{\phi}\ (_{\sigma}$ it$)\ [_{\phi}\ (_{\sigma}$ will$)\ [_{\phi}\ (_{\sigma}$ not$)\ [_{\phi}\ [_{\omega}$ rain$]]]]]]$	*!**			
b. $[_{\iota}\ [_{\phi}\ [_{\phi}\ [_{\omega}$ it$]]\ [_{\phi}\ [_{\phi}\ [_{\omega}$ will$]]\ [_{\phi}\ [_{\phi}\ [_{\omega}$ not$]]\ [_{\phi}\ [_{\omega}$ rain$]]]]]]$		**!*		
c. $[_{\iota}\ [_{\phi}\ (_{\sigma}$ will$)\ [_{\phi}\ (_{\sigma}$ not$)\ [_{\phi}\ [_{\omega}$ rain$]]]]]$	*!*		*	*
d. $[_{\iota}\ [_{\phi}\ (_{\sigma}$ it$)\ [_{\phi}\ (_{\sigma}$ won't$)\ [_{\phi}\ [_{\omega}$ rain$]]]]]$	*!*			*
e. $[_{\iota}\ [_{\phi}\ (_{\sigma}$ won't$)\ [_{\phi}\ [_{\omega}$ rain$]]]]$	*!		*	**
f. $[_{\iota}\ [_{\phi}\ (_{\sigma}$ not$)\ [_{\phi}\ [_{\omega}$ rain$]]]]$	*!		**	**
g. $[_{\iota}\ [_{\phi}\ (_{\sigma}$ it$)\ [_{\phi}\ [_{\omega}$ won't$_{\sigma}$ rain$_{Ft}$ $]]]]$	*!*			**
h. $[_{\iota}\ [_{\phi}\ [_{\omega}$ won't$_{\sigma}$ rain$_{Ft}$ $]]]]]$	*!		*	**
☞ i. $[_{\iota}\ [_{\phi}\ [_{\phi}\ [_{\omega}$ won't$]]\ [_{\phi}\ [_{\omega}$ rain$]]]]$		*	*	**

Subject drop is permitted with cliticisable auxiliary verbs only if:

(i) the verb is realised in its clitic form ✓

(ii) the full form of the verb has the clitic form of negation (-*n't*) attached to it ✓

(iii) the full form of the verb is contrastively focused

Input: $[_{CP}\ [_{TP}\ [_{DP}$ it$]\ [_{T'}\ [_T$ IS$]_F\ [_{VP}\ [_V$ working$]]]]]$

	ALIGN(FOC, ϕ)	STRONGSTART	MATCH(ω, Lex)	MAX	MATCH(S, P)
a. $[_ι\ [_ϕ\ (_σ$ it$)\ [_ϕ\ (_σ$ is$)\ [_ϕ\ [_ω$ working$]]]]]$					
b. $[_ι\ [_ϕ\ [_ϕ\ [_ω$ it$]]\ [_ϕ\ [_ϕ\ [_ω$ is$]]\ [_ϕ\ [_ω$ working$]]]]]$					
c. $[_ι\ [_ϕ\ (_σ$ it's$)\ [_ϕ\ [_ω$ working$]]]]$					
d. $[_ι\ [_ϕ\ [_ω$'s$_σ$ working$_{Ft}]]]$					
e. $[_ι\ [_ϕ\ [_ϕ\ [_ω$ is$]]\ [_ϕ\ [_ω$ working$]]]]]$					

STRONGSTART
A phonological constituent optimally begins with a leftmost daughter constituent which is <u>not lower</u> in the prosodic hierarchy than the constituent that immediately follows it.

MATCH(ω, Lex)
Every ω must contain an instance of a lexical word.

MAX: don't delete elements

ALIGN(FOC, ϕ)
Align the left and right edges of a focused element with the left and right edges of a ϕ

Input: $[_{CP}\ [_{TP}\ [_{DP}$ it$]\ [_{T'}\ [_{T}$ IS$]_{F}\ [_{VP}\ [_{V}$ working$]]]]]$

	ALIGN(Foc, Φ)	STRONGSTART	MATCH(ω, Lex)	MAX	MATCH(S, P)
a. $[_{\iota}\ [_{\phi}\ (_{\sigma}$ it$)\ [_{\phi}\ (_{\sigma}$ is$)\ [_{\phi}\ [_{\omega}$ working$]]]]]$	*	**			
b. $[_{\iota}\ [_{\phi}\ [_{\phi}\ [_{\omega}$ it$]]\ [_{\phi}\ [_{\phi}\ [_{\omega}$ is$]]\ [_{\phi}\ [_{\omega}$ working$]]]]]$			**		
c. $[_{\iota}\ [_{\phi}\ (_{\sigma}$ it's$)\ [_{\phi}\ [_{\omega}$ working$]]]]$	*	*			*
d. $[_{\iota}\ [_{\phi}\ [_{\omega}$'s$_{\sigma}$ working$_{Ft}]]]$	*	*		*	**
e. $[_{\iota}\ [_{\phi}\ [_{\phi}\ [_{\omega}$ is$]]\ [_{\phi}\ [_{\omega}$ working$]]]]]$			*	*	*

Input: $[_{CP}\ [_{TP}\ [_{DP}$ it$]$ $[_{T'}\ [_{T}$ IS$]_{F}$ $[_{VP}\ [_{V}$ working$]]]]]$

		ALIGN(FOC, Φ)	STRONGSTART	MATCH(ω, Lex)	MAX	MATCH(S, P)
	a. $[_{\iota}\ [_{\phi}\ (_{\sigma}$ it$)$ $[_{\phi}\ (_{\sigma}$ is$)$ $[_{\phi}\ [_{\omega}$ working$]]]]]$	*	*!*			
	b. $[_{\iota}\ [_{\phi}\ [_{\phi}\ [_{\omega}$ it$]]$ $[_{\phi}\ [_{\phi}\ [_{\omega}$ is$]]$ $[_{\phi}\ [_{\omega}$ working$]]]]]$			**!		
	c. $[_{\iota}\ [_{\phi}\ (_{\sigma}$ it's$)$ $[_{\phi}\ [_{\omega}$ working$]]]]$	*	*!			*
	d. $[_{\iota}\ [_{\phi}\ [_{\omega}$'s$_{\sigma}$ working$_{Ft}]]]$	*	*!		*	**
☞	e. $[_{\iota}\ [_{\phi}\ [_{\phi}\ [_{\omega}$ is$]]$ $[_{\phi}\ [_{\omega}$ working$]]]]]$			*	*	*

Subject drop is permitted with cliticisable auxiliary verbs only if:

(i) the verb is realised in its clitic form ✓

(ii) the full form of the verb has the clitic form of negation (-*n't*) attached to it ✓

(iii) the full form of the verb is contrastively focused ✓

Another constraint:

- Subject drop only permitted if utterance-initial

* I don't think ~~he~~ should go.

Related observation: cliticisation of embedded subjects is preferred to deletion of them

a. * ~~He~~ thinks you left.
b. ~~He~~ thinks ya left.
c. * ~~He~~ thinks ~~ya~~ left.

Why? Cliticisation is always less costly than deletion (due to the **MAX** constraint)

Cliticisation of the subject is only available in embedded cases, as the subject cliticises leftwards

Input: $[_{CP}$ $[_{TP}$ $[_{DP}$ he] $[_{T'}$ $[_{VP}$ $[_{V}$ thinks] $[_{CP}$ $[_{TP}$ $[_{DP}$ you] $[_{T'}$ $[_{VP}$ $[_{V}$ left]]]]]]]]]

	STRONGSTART	MATCH(ω, Lex)	MAX	MATCH(S, P)
a. $[_{\iota}$ $[_{\phi}$ $[_{\omega}$ thinks] $[_{\iota}$ $[_{\phi}$ $(_{\sigma}$ you) $[_{\phi}$ $[_{\omega}$ left]]]]]]	**!		*	**
b. $[_{\iota}$ $[_{\phi}$ $[_{\omega}$ thinks] $[_{\iota}$ $[_{\phi}$ $[_{\phi}$ $[_{\omega}$ you]] $[_{\phi}$ $[_{\omega}$ left]]]]]]	*	*!	*	*
☞ c. $[_{\iota}$ $[_{\phi}$ $[_{\omega}$ thinks ya] $[_{\iota}$ $[_{\phi}$ $[_{\omega}$ left]]]]]	*		*	***
d. $[_{\iota}$ $[_{\phi}$ $[_{\omega}$ thinks] $[_{\iota}$ $[_{\phi}$ $[_{\omega}$ left]]]]]	*		**!	***