

- **Use factorial typology to predict possible languages**
- **Consider implications of syllable-structure analysis in OT**

Background preparation:

- *Exercise - Constraint rankings & their predictions*

0. Today's plan

- OT check-in
- Factorial typology:
The predictions of a constraint *set*
- Factorial typology of basic syllable structure
- Comparing models:
Syllable structure with OT vs. rules

1. OT check-in

- How do we use an OT tableau to **figure out the constraint ranking** in a particular language?
 - What do we know?
 - What do we need to figure out?

1. OT check-in

- How do we use an OT tableau to **figure out the constraint ranking** in a particular language?
 - What do we know?
 - the **winning output**; the **input** (proposed)
 - What do we need to figure out?
 - how the constraints are **ranked**
 - How do we do this?

1. OT check-in

- How do we use an OT tableau to **figure out the constraint ranking** in a particular language?
 - What do we know?
 - the **winning output**; the **input** (proposed)
 - What do we need to figure out?
 - how the constraints are **ranked**
 - How do we do this?
 - find informative losers
 - make valid ranking arguments

1. OT check-in

- How do we use an OT tableau to predict **which output candidate wins** for a certain input in a certain language?
 - What do we know?
 - What do we need to figure out?

1. OT check-in

- How do we use an OT tableau to predict **which output candidate wins** for a certain input in a certain language?
 - What do we know?
 - the constraint **ranking**; the **input** of interest
 - What do we need to figure out?
 - which **output candidate** is optimal (“the best”) according to the ranking
 - How do we do this?

1. OT check-in

- How do we use a tableau to **predict** the output, given the ranking and the input?
- Which output wins, if $A \gg B \gg C$?

/input/	A	B	C
(a) output1	*		
(b) output2		*	*
(c) output3		*	

1. OT check-in

- How do we use a tableau to **predict** the output, given the ranking and the input?
- Which output wins, if $A \gg B \gg C$?

/input/	A	B	C
(a) output1	*!		
(b) output2		*	*!
→ (c) output3		*	

- Optional: Use ‘!’ when the ranking is **known**, to show what violation **makes** a candidate lose

2. Factorial typology

- In OT, because the **constraints are universal...**
 - whenever we propose a constraint in the analysis of one language,
 - we are automatically proposing that this constraint is found in all languages
- Consider the **typological implications** of a constraint we are using in an analysis:
 - If this constraint is found in all languages,
 - and could potentially be ranked high or low,
 - are we making inaccurate **predictions?**

2. Factorial typology

- How do we **test** the typological implications (=predictions) of a proposed set of constraints?
 - Consider **all possible rankings**
 - Consider *what kind* of language each ranking would predict
 - Evaluate whether such languages are plausible
- This is obviously a very large undertaking
 - Typically, we consider the predictions of a small set of relevant constraints at a time
 - (There is also OT analysis software to help with this!)

2. Factorial typology

- How many **possible rankings** are there for a set of 3 constraints?

2. Factorial typology

- How many **possible rankings** are there for a set of 3 constraints?

A B C

B C A

C A B

A C B

B A C

C B A

- $3 * 2 * 1 = 6$ rankings ('3 factorial')
- $n * (n-1) * (n-2) * \dots * 1 = \mathbf{n!}$ ('n factorial')
- The **list of all possible rankings** of a set of constraints is called the '**factorial typology**' of that set of constraints

3. Factorial typology of basic syllable structure

- Prep qns: Constraint rankings and their predictions
 - What are **all the rankings** of the mini constraint set NoEPENTHESIS, ONSET, NoCODA?

3. Factorial typology of basic syllable structure

- Prep qns: Constraint rankings and their predictions

- **All the rankings** of NoEPENTHESIS, ONSET, NoCODA?

1 ONSET » NoCODA » NoEPENTHESIS

2 NoCODA » ONSET » NoEPENTHESIS

3 NoEPENTHESIS » ONSET » NoCODA

4 NoEPENTHESIS » NoCODA » ONSET

5 ONSET » NoEPENTHESIS » NoCODA

6 NoCODA » NoEPENTHESIS » ONSET

Group discussion | For your ranking:

- Which **output** wins for /abik/? /kamo/?
- Which **σ types** (CV, CVC, V, CV) are **allowed**?

3. Factorial typology of basic syllable structure

Debriefing

Questions we will now consider:

- **Which output** wins for **/abik/** under each ranking?
What about **/kamo/**?
 - What “kind of language” does each ranking produce? (generalizations about σ structure)
- Are all the **predicted language types** plausible?
 - What are the actual cross-linguistic facts about onsets and codas?

3. Factorial typology of basic syllable structure

- What violations are assigned to each of these candidates? (dashed lines = constraints not ranked)
 - Do we predict different outcomes in different languages?

/abik/	NoEPENTH	ONSET	NoCODA
(a) a.bik			
(b) a.bi.kə			
(c) ta.bik			
(d) ta.bi.kə			

3. Factorial typology of basic syllable structure

- What violations are assigned to each of these candidates? (dashed lines = constraints not ranked)
 - Do we predict different outcomes in different languages?

/abik/	NoEPENTH	ONSET	NoCODA
(a) a.bik		*	*
(b) a.bi.kə	*	*	
(c) ta.bik	*		*
(d) ta.bi.kə	**		

3. Factorial typology of basic syllable structure

- What violations are assigned to each of these candidates? (dashed lines = constraints not ranked)
 - Do we predict different outcomes in different languages?

/kamo/	NoEPENTH	ONSET	NoCODA
(a) ka.mo			
(b) kam.o			
(c) ə.ka.mo			
(d) ka.mot			

3. Factorial typology of basic syllable structure

- What violations are assigned to each of these candidates? (dashed lines = constraints not ranked)
 - Do we predict different outcomes in different languages?

/kamo/	NoEPENTH	ONSET	NoCODA
(a) ka.mo			
(b) kam.o		*	*
(c) ə.ka.mo	*	*	
(d) ka.mot	*		*

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under this ranking?

1) ONSET » NoCODA » NoEPENTHESIS

/abik/	ONSET	NoCODA	NoEPENTH
(a) a.bik	*	*	
(b) a.bi.kə	*		*
(c) ta.bik		*	*
(d) ta.bi.kə			**

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under this ranking?

1) $O_{NSET} \gg NoC_{ODA} \gg NoE_{PENTHESIS}$ /abik/ → [ta.bi.kə]

/abik/	O_{NSET}	NoC_{ODA}	NoE_{PENTH}
(a) a.bik	*!	*	
(b) a.bi.kə	*!		*
(c) ta.bik		*!	*
→ (d) ta.bi.kə			**

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under this ranking?

2) NoCODA » ONSET » NoEPENTHESIS

/abik/	NoCODA	ONSET	NoEPENTH
(a) a.bik	*	*	
(b) a.bi.kə		*	*
(c) ta.bik	*		*
(d) ta.bi.kə			**

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under this ranking?

2) NoCODA » ONSET » NoEPENTHESIS

/abik/ → [ta.bi.kə]

/abik/	NoCODA	ONSET	NoEPENTH
(a) a.bik	*!	*	
(b) a.bi.kə		*!	*
(c) ta.bik	*!		*
→ (d) ta.bi.kə			**

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under this ranking?

3) NoEPENTHESIS » ONSET » NoCODA

/abik/	NoEPENTH	ONSET	NoCODA
(a) a.bik		*	*
(b) a.bi.kə	*	*	
(c) ta.bik	*		*
(d) ta.bi.kə	**		

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under this ranking?

3) NoEPENTHESIS » ONSET » NoCODA

/abik/ → [a.bik]

/abik/	NoEPENTH	ONSET	NoCODA
→ (a) a.bik		*	*
(b) a.bi.kə	*!	*	
(c) ta.bik	*!		*
(d) ta.bi.kə	*!*		

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under this ranking?

4) NoEPENTHESIS » NoCODA » ONSET

/abik/	NoEPENTH	NoCODA	ONSET
(a) a.bik		*	*
(b) a.bi.kə	*		*
(c) ta.bik	*	*	
(d) ta.bi.kə	**		

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under this ranking?

4) NoEPENTHESIS » NoCODA » ONSET /abik/ → [a.bik]

/abik/	NoEPENTH	NoCODA	ONSET
→ (a) a.bik		*	*
(b) a.bi.kə	*!		*
(c) ta.bik	*!	*	
(d) ta.bi.kə	*!*		

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under this ranking?

5) ONSET » NOEPENTHESIS » NOCODA

/abik/	ONSET	NOEPENTH	NOCODA
(a) a.bik	*		*
(b) a.bi.kə	*	*	
(c) ta.bik		*	*
(d) ta.bi.kə		**	

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under this ranking?

5) $O_{\text{NSET}} \gg \text{NoE}_{\text{PENTHESIS}} \gg \text{NoC}_{\text{ODA}}$ /abik/ → [ta.bik]

/abik/	O_{NSET}	$\text{NoE}_{\text{PENTH}}$	NoC_{ODA}
(a) a.bik	*!		*
(b) a.bi.kə	*!	*	
→ (c) ta.bik		*	*
(d) ta.bi.kə		**!	

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under this ranking?

6) NoCODA » NoEPENTHESIS » ONSET

/abik/	NoCODA	NoEPENTH	ONSET
(a) a.bik	*		*
(b) a.bi.kə		*	*
(c) ta.bik	*	*	
(d) ta.bi.kə		**	

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under this ranking?

6) NoCODA » NoEPENTHESIS » ONSET /abik/ → [a.bi.kə]

/abik/	NoCODA	NoEPENTH	ONSET
(a) a.bik	*!		*
→ (b) a.bi.kə		*	*
(c) ta.bik	*!	*	
(d) ta.bi.kə		**!	

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under each ranking?

1 ONSET » NoCODA » NoEPENTH /abik/ → [ta.bi.kə]

2 NoCODA » ONSET » NoEPENTH /abik/ → [ta.bi.kə]

3 NoEPENTH » ONSET » NoCODA /abik/ → [a.bik]

4 NoEPENTH » NoCODA » ONSET /abik/ → [a.bik]

5 ONSET » NoEPENTH » NoCODA /abik/ → [ta.bik]

6 NoCODA » NoEPENTH » ONSET /abik/ → [a.bi.kə]

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under these rankings?

1 ONSET » NoCODA » NoEPENTH /abik/ → [ta.bi.kə]

2 NoCODA » ONSET » NoEPENTH /abik/ → [ta.bi.kə]

- What generalizations can we make about syllable structure options for a language with this pattern?

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under these rankings?

1 ONSET » NoCODA » NoEPENTH /abik/ → [ta.bi.kə]

2 NoCODA » ONSET » NoEPENTH /abik/ → [ta.bi.kə]

- What generalizations can we make about syllable structure options for a language with this pattern?
 - Onsetless syllables are forbidden
 - Codas are forbidden

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under these rankings?

3 NoEPENTH » ONSET » NoCODA /abik/ → [a.bik]

4 NoEPENTH » NoCODA » ONSET /abik/ → [a.bik]

- What generalizations can we make about syllable structure options for a language with this pattern?

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under these rankings?

3 NoEPENTH » ONSET » NoCODA /abik/ → [a.bik]

4 NoEPENTH » NoCODA » ONSET /abik/ → [a.bik]

- What generalizations can we make about syllable structure options for a language with this pattern?
 - Onsetless syllables are allowed
 - Codas are allowed

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under this ranking?

5 ONSET » NoEPENTH » NoCODA /abik/ → [ta.bik]

- What generalizations can we make about syllable structure options for a language with this pattern?

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under this ranking?

5 ONSET » NOEPENTH » NOCODA /abik/ → [ta.bik]

- What generalizations can we make about syllable structure options for a language with this pattern?
 - Onsetless syllables are forbidden
 - Codas are allowed

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under each ranking?

6 NoCODA » NoEPENTH » ONSET /abik/ → [a.bi.kə]

- What generalizations can we make about syllable structure options?

3. Factorial typology of basic syllable structure

- Which output wins for /abik/ under each ranking?

6 NoCODA » NoEPENTH » ONSET /abik/ → [a.bi.kə]

- What generalizations can we make about syllable structure options?
 - Onsetless syllables are allowed
 - Codas are forbidden

3. Factorial typology of basic syllable structure

- Are the predictions of our constraint set plausible?

<i>Ranking</i>	<i>Onsets?</i>	<i>Codas?</i>
1 ONSET » NoCODA » NoEP 2 NoCODA » ONSET » NoEP	required	forbidden
3 NoEP » ONSET » NoCODA 4 NoEP » NoCODA » ONSET	not required	not forbidden
5 ONSET » NoEP » NoCODA	required	not forbidden
6 NoCODA » NoEP » ONSET	not required	forbidden

3. Factorial typology of basic syllable structure

- Are the predictions of our constraint set plausible?

<i>Ranking</i>	<i>Onsets?</i>	<i>Codas?</i>
1 ONSET » NoCODA » NoEP 2 NoCODA » ONSET » NoEP	required	forbidden
3 NoEP » ONSET » NoCODA 4 NoEP » NoCODA » ONSET	not required	not forbidden
5 ONSET » NoEP » NoCODA	required	not forbidden
6 NoCODA » NoEP » ONSET	not required	forbidden

- Yes! Onsets can be required; codas can be banned; no language *bans* onsets, *requires* codas

3. Factorial typology of basic syllable structure

- Are the predictions of our constraint set plausible?
 - Yes! Onsets can be required; codas can be banned; no language *bans* onsets, *requires* codas
- Of course, we have to take into account NoDELETION as well — some languages that enforce onsets or ban codas do so via deletion instead of epenthesis
 - Checking that by hand would require that we look at $4! = 24$ grammars, not just 6

4. Factorial typology—conclusions, implications

- Ways to do an approximate check on the factorial-typology predictions of a **larger** constraint set
 - Consider what would happen *if each constraint were the highest ranked*
 - Example: If we propose a “HAVECODA” constraint, it will make bad predictions
 - Consider what would happen we switch the rankings of *just the faithfulness constraints* or *just the markedness constraints*
 - Example: Our 6 grammars, but with NoDELETION *ranked below* NoEPENTHESIS

4. Factorial typology—conclusions, implications

- One more complication
 - Sometimes there is a ranking predicted by factorial typology that no known language **actually uses**
 - This can often be explained by factors related to **phonetics, child language acquisition, and historical change**
 - A ranking that children would never see evidence to learn (for phonetic reasons) would never arise in any language

4. Factorial typology—conclusions, implications

- A key point of this discussion
 - **Every OT analysis** is also inherently **making typological predictions**
 - This makes the theory more interesting, and **places more restrictions on it**, compared to rule-based phonology

5. Comparing models— σ with OT vs. rules

- When we analyzed syllable structure before, there were two steps in our procedure:

Step 1. Determine **what syllable structure is like** in the language we are analyzing.

- How did we do this before?
- Does this need to change when we use OT?

5. Comparing models— σ with OT vs. rules

- When we analyzed syllable structure before, there were two steps in our procedure:

Step 1. Determine **what syllable structure is like** in the language we are analyzing.

- We used **phonological evidence** from the data set to determine the “syllable-structure options”
 - What is a legal nucleus? Are onsets required? Are codas allowed? Are there onset or coda clusters?
- Does this need to change when we use OT?

5. Comparing models— σ with OT vs. rules

- When we analyzed syllable structure before, there were two steps in our procedure:
 - Step 1.** Determine **what syllable structure is like** in the language we are analyzing.
 - We used **phonological evidence** from the data set to determine the “syllable-structure options”
 - What is a legal nucleus? Are onsets required? Are codas allowed? Are there onset or coda clusters?
 - > **Step 1 does not change in OT.** We still need to use phonological evidence to make proposals about syllable structure in each language.

5. Comparing models— σ with OT vs. rules

- When we analyzed syllable structure before, there were two steps in our procedure:

Step 2. Make our **model of the mental grammar** produce the appropriate syllable structure for the language we are analyzing.

- How did we do this before?
- Does this need to change when we use OT?

5. Comparing models— σ with OT vs. rules

- When we analyzed syllable structure before, there were two steps in our procedure:
 - Step 2.** Make our **model of the mental grammar** produce the appropriate syllable structure for the language we are analyzing.
 - Originally, we took care of Step 2 by means of three **syllable-building rules**: the Nucleus Rule, the Onset Rule, and the Coda Rule
 - Does this need to change when we use OT?

5. Comparing models— σ with OT vs. rules

- When we analyzed syllable structure before, there were two steps in our procedure:
 - Step 2.** Make our **model of the mental grammar** produce the appropriate syllable structure for the language we are analyzing.
 - Originally, we took care of Step 2 by means of three **syllable-building rules**: the Nucleus Rule, the Onset Rule, and the Coda Rule
 - > **OT does not use rules**→we don't use these now!
We **rank constraints** such as ONSET, NoCODA, etc., to **choose the right output** candidate

5. Comparing models— σ with OT vs. rules

- What is the status of the “syllable-structure options” in the two models?
 - “Syllable-structure options” = What is a legal nucleus? Are onsets required? Are codas allowed? Are there onset or coda clusters?

5. Comparing models— σ with OT vs. rules

- What is the status of the “syllable-structure options” in the two models?
 - “Syllable-structure options” = What is a legal nucleus? Are onsets required? Are codas allowed? Are there onset or coda clusters?
- Rule-based model: These “options” needed to be in the model, to stop syllable-building rules from applying (but how did that actually work?!)
- OT model: **Now these “options” are not themselves in the model!** (Whew.) They just help us summarize the facts about the world

5. Comparing models— σ with OT vs. rules

- Languages always syllabify [V.CV] and never [VC.V]
 - How did we make our rule-based model do this?
 - How do we make this happen in OT?

5. Comparing models— σ with OT vs. rules

- Languages always syllabify [V.CV] and never [VC.V]
 - How did we make our rule-based model do this?
 - > We said that the Onset Rule had to apply *earlier than* the Coda Rule **in all languages**
 - This is unusual — rule ordering is usually language-specific
 - How do we make this happen in OT?
 - > The answer is a little surprising!

5. Comparing models— σ with OT vs. rules

- Languages always syllabify [V.CV] and never [VC.V]
 - How do we make this happen in OT?

/kamo/	ONSET	NoCODA
(a) [ka.mo]		
(b) [kam.o]	*	*

- Does ONSET have to be **ranked above** NoCODA for [ka.mo] to win?

5. Syllable-structure analysis—OT vs. rules

- Languages always syllabify [V.CV] and never [VC.V]
 - How do we make this happen in OT?

/kamo/	ONSET	NoCODA
→ (a) [ka.mo]		
(b) [kam.o]	*	*

- Does ONSET have to be **ranked above** NoCODA for [ka.mo] to win? No! If we have a constraint set that **has** ONSET and NoCODA — and not the reverse constraints — then [V.CV] **always** beats [VC.V]

5. Syllable-structure analysis—OT vs. rules

Summary: Model comparisons

- The role of the syllable-structure “options”
 - Rule-based model: needed to formalize them in order to have universal syllable-building rules, but unclear how!
 - OT model: doesn't need to formalize them; different rankings of universal constraints get the right patterns
- Universal choice of [V.CV] over *[VC.V]
 - Rule-based model: Arbitrary stipulation that Onset Rule precedes Coda Rule
 - OT model: No arbitrary ranking between O_{NSET} , No_{CODA}
 - Deeper qn: Why have O_{NSET} , No_{CODA} as constraints?

6. General questions or concerns about OT?

- Any thoughts, questions, or concerns about this new view of the phonological mental grammar?