

## Today's objectives:

- **Describing syllable “options”**
- **Applying syllable-building rules**

---

### *Background preparation:*

- *Handout: “...Describing syllabification options”*
- *Handout: “Syllable-building rules”*

# 0. Today's plan

- Review and context
  - Our approach to syllable structure
  - Check-in on prep questions: Finding examples
- Example: Cairene Arabic
  - Describing  $\sigma$  options (making generalizations)
  - Applying  $\sigma$ -building rules (tools of the model)
- Some points to note about  $\sigma$ -building rules
- Analysis practice: Korean loanwords

# 1. Review and context

- Many aspects of syllable structure are either:
  - the same in all languages
  - chosen from a very small range of possibilities
- This is unlike segmental rules, which seem to differ widely from language to language
- Therefore, our approach to modeling syllable-building rules will look a little different from our approach to modeling segmental rules:
  - **Universal** rules that **build** structure
  - **Language-specific** limits on rule **application**

# 1. Review and context

- Our first step in analyzing syllable structure:  
**Observe** and **describe** what syllable-structure patterns are possible out there in the world
  - Handout - "[...Describing syllabification options](#)"
- **In practical terms**, when working with a data set:
  - a) Use the available phonological **evidence** to determine how segments are assigned to syllables in the language
  - b) Make **generalizations** about legal **nuclei**, **onsets**, or **codas** in the language

# 1. Review and context

- Checking in on prep questions:

Onsetless syllable?	Onset cluster?	Coda?	Coda cluster?
------------------------	-------------------	-------	------------------

*herb*

*string*

*match*

# 1. Review and context

- Checking in on prep questions:

	Onsetless syllable?	Onset cluster?	Coda?	Coda cluster?
<i>herb</i>	<b>Y</b>	<b>N</b>	<b>Y</b>	<b>N</b>
<i>string</i>	<b>N</b>	<b>Y</b>	<b>Y</b>	<b>(?)</b>
<i>match</i>	<b>N</b>	<b>N</b>	<b>Y</b>	<b>N</b>

- What's a good way to **avoid** being fooled by spelling?

# 1. Review and context

- Our next step in analyzing syllable structure:  
What kind of **syllable-building rules** can we propose to enforce both universal patterns and options in individual languages?
  - Handout - "[Syllable-building rules](#)"
- **In practical terms**, when working with a data set:
  - a) **Identify** the syllable-building **rules** that are universal to all languages
  - b) **Identify** the **limits** on legal onsets, nuclei, and codas in the specific language in question
  - c) **Apply** the universal syllable-building **rules**, as restricted by the **limits** on legal onsets, nuclei, and codas in the specific language in question

## 2. Example: Cairene Arabic

- Data set - [Cairene Arabic](#)
  - a) Use the available phonological **evidence** to determine how segments are assigned to syllables in the language
    - We did this in previous class discussions, using evidence from pharyngealization spread and vowel epenthesis (insertion)

/faʂlu/ → [**FAS**.lu]

‘his term’

(*not* \*[fa.SLU])

/ʔul-t-l-u/ → [ʔul.t**i**.lu]

‘I said to him’

(*not* \*[ʔult.lu], \*[ʔul.tlu])



## 2. Example: Cairene Arabic

- Data set - [Cairene Arabic](#)
  - b) Make **generalizations** about legal **nuclei**, **onsets**, or **codas** in the language
    - Use the **summary question list** at the end of the handout “[Syllables: Overview / Describing syllabification options](#)” and see how many questions you can answer

## 2. Example: Cairene Arabic

- Data set - [Cairene Arabic](#)
  - b) Make **generalizations** about legal **nuclei**, **onsets**, or **codas** in the language

### *Nuclei:*

- Nucleus is universally mandatory
- Vowel as nucleus is universally allowed
- Are diphthongs allowed?
- Other natural classes/possible nuclei?

## 2. Example: Cairene Arabic

- Data set - [Cairene Arabic](#)
  - b) Make **generalizations** about legal **nuclei**, **onsets**, or **codas** in the language

### *Onsets:*

- Are onsetless syllables allowed?
- Are onset clusters allowed?
  - If yes, any restrictions?

## 2. Example: Cairene Arabic

- Data set - [Cairene Arabic](#)
  - b) Make **generalizations** about legal **nuclei**, **onsets**, or **codas** in the language

### *Codas:*

- Are codas allowed?
  - If yes, any restrictions?
- Are coda clusters allowed?
  - If yes, any restrictions?

## 2. Example: Cairene Arabic

- Data set - [Cairene Arabic](#)
  - c) **Apply** the universal syllable-building **rules**, as restricted by the **limits** on legal onsets, nuclei, and codas in the specific language in question
    - Handout: “[Syllable-building rules](#)”
      - Nucleus Rule
      - Onset Rule
      - Coda Rule
- Proposal: Syllable-building rules tell the grammar how to **associate** segments with syllables

## 2. Example: Cairene Arabic

- Data set - [Cairene Arabic](#)
  - a) **Apply** the universal syllable-building **rules**, as restricted by the **limits** on legal onsets, nuclei, and codas in the specific language in question
    - i) **Nucleus Rule**

Every unsyllabified segment that is a legal nucleus (V<sup>n</sup>) projects (=creates and associates to) a syllable

      - *Limit: Only vowels are legal nuclei*

b  
---  
V<sup>n</sup>

## 2. Example: Cairene Arabic

- Data set - [Cairene Arabic](#)
  - a) **Apply** the universal syllable-building **rules**, as restricted by the **limits** on legal onsets, nuclei, and codas in the specific language in question

- **Nucleus Rule**

*Limit: Only vowels are legal nuclei*

/faʃlu/

/ʔul-t-l-u/

[faʃlu]

[ʔultlu]

## 2. Example: Cairene Arabic

- Data set - [Cairene Arabic](#)
  - a) **Identify** the syllable boundaries in the words
  - b) **Describe** the syllable structure of the words
  - c) **Apply** the universal syllable-building **rules**, as restricted by the **limits** on legal onsets, nuclei, and codas in the specific language in question

### - **Nucleus Rule**

*Limit: Only vowels are legal nuclei*

/faSlu/

/ʔul-t-l-u/

σ     σ  
|     |  
[ f a S l u ]

σ     σ  
|     |  
[ ʔ u l t l u ]



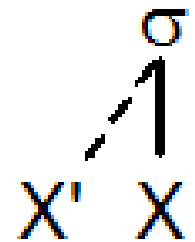
## 2. Example: Cairene Arabic

- Data set - [Cairene Arabic](#)
  - a) **Identify** the syllable boundaries in the given word
  - b) **Describe** the syllable structure of the given word
  - c) **Apply** the universal syllable-building **rules**, as restricted by the **limits** on legal onsets, nuclei, and codas in the specific language in question

### - Onset Rule

Every unsyllabified segment (X') that immediately precedes a syllabified segment is added to that syllable, as long as a **legal onset** is produced

- *Limit: Applies only once; no clusters*



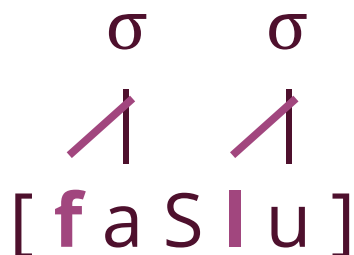
## 2. Example: Cairene Arabic

- Data set - [Cairene Arabic](#)
  - c) **Apply** the universal syllable-building **rules**, as restricted by the **limits** on legal onsets, nuclei, and codas in the specific language in question

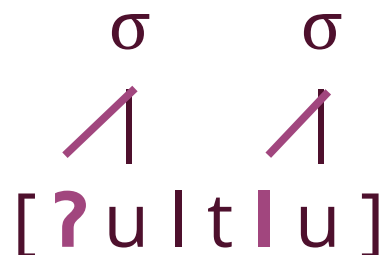
### - Onset Rule

*Limit: Applies only once; no clusters*

/faʃlu/



/ʔul-t-l-u/



## 2. Example: Cairene Arabic

- Data set - [Cairene Arabic](#)
  - a) **Apply** the universal syllable-building **rules**, as restricted by the **limits** on legal onsets, nuclei, and codas in the specific language in question

- **Coda Rule**

Every unsyllabified segment (X') that immediately follows a syllabified segment is added to that syllable, as long as a **legal coda** is produced



- *Limit: Applies, but only once; no legal clusters*

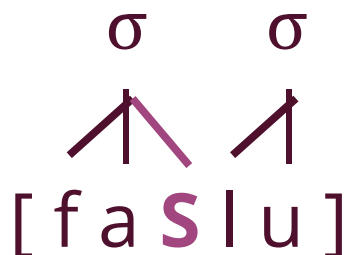
## 2. Example: Cairene Arabic

- Data set - [Cairene Arabic](#)
  - c) **Apply** the universal syllable-building **rules**, as restricted by the **limits** on legal onsets, nuclei, and codas in the specific language in question

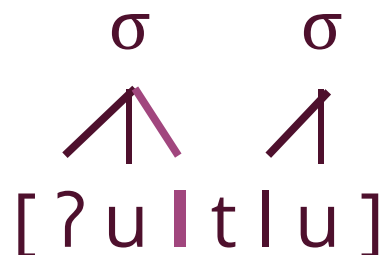
- **Coda Rule**

*Limit: Applies, but only once; no legal clusters*

/faʂlu/



/ʔul-t-l-u/



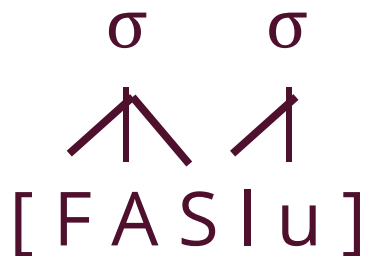
## 2. Example: Cairene Arabic

- Data set - [Cairene Arabic](#)

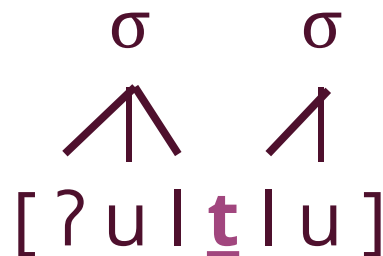
Results:

- Pharyngealization in [ **FAS**.lu ] correctly predicted
- Remaining unsyllabified consonant in [ ?ul.**{t}**.lu ] correctly predicts the location of epenthesis according to the rule  $\emptyset \rightarrow [-bk, +hi] / C' \_$

/faʃlu/



/ʔul-t-lu/



### 3. More about syllable-building rules

- How do these rules account for...
  - **Universal** pattern: Every syllable has a nucleus
  - **Universal** pattern: Consonants that immediately precede nuclei are *onsets*, not codas
  - Language-particular **limits** on syllabification
- See handout: ["Syllable-building rules"](#)

### 3. More about syllable-building rules

- How do these rules account for...
  - **Universal** pattern: Every syllable has a nucleus
    - Only create  $\sigma$  if there is V (legal nucleus)
  - **Universal** pattern: Consonants that immediately precede nuclei are *onsets*, not codas
    - Onset Rule universally **precedes** Coda Rule
    - Universally specified rule order—surprising?
  - Language-particular **limits** on syllabification
- See handout: [“Syllable-building rules”](#)

### 3. More about syllable-building rules

- Language-particular **limits** on syllabification
  - How does our current model account for this?
  - What are some alternatives?
  - Case study for discussion:  
Coda cluster only if [+nas][−nas, −cont]  
Some URs: /ompta/ /ensta/



### 3. More about syllable-building rules

- Given our analysis of Cairene Arabic, what do we have to assume about the ordering of the syllable-building rules and the epenthesis rule?
  - Epenthesis rule:  $\emptyset \rightarrow [-bk, +hi] / C' \_$
  - Consider this example:  
/katab-t-l-ha/ → [katabt<sub>i</sub>lha] 'I wrote to her'  
\*[katabt<sub>i</sub>l<sub>i</sub>ha]

### 3. More about syllable-building rules

- Conclusion: Syllable-building rules are **persistent**

It always seems to be the case that *after* epenthesis or deletion rules apply:

- Syllable structure is deleted
  - Syllable-building rules start over again
- 
- By comparison...  
(Most) segmental rules only apply once!

## 4. Analysis practice: Korean loanwords

- Data set: [Korean loanwords](#)
  - What **settings** can we determine for the syllable-structure **options** in Korean?
  - How do the **syllable-building rules** apply to Korean words?

**(prep questions for next time)**