Optimality in Grammar: core syllabification in Imdlawn Tashlhiyt Berber

Alan Price & Paul Smolesnsky

Syllabification of ITB

The Imdlawn Tashlhiyt dialect of Berber (ITB) exhibits an interesting case of syllabification.

In ITB, any segment – consonant or vowel, obstruent of sonorant – can form a syllable nucleus.

For instance, there are syllables such as tK, rB, xZ, wL where the capital letter represents a syllable nucleus.

Dell and Elmedlaoui who reported this phenomenon draw evidence supporting such syllabification form various linguistic aspects including native speakers intuition and phonological processes.

Syllabification of ITB

In ITB, the syllabification domain is the phonological phrase and all syllables must have an onset except when they occur in phrase-initial position.

Since any segment can for a syllable nucleus, there is a lot of ambiguity in syllabification.

Dell and Elmedlaoui noticed that assignment of nucleus is determined by the relative sonority of the elements in the string.

Hence , in a string such as /tzmt/ the most sonorant segment is /m/; thus, it becomes the nucleus [tzMt].

Syllabification Algorithm

Sonority scale in ITB:

Low V > High V > Liquid > Nasal > Voiced Fric. > Voiceless Fric. > Voiced Stop > Voiceless Stop

In ITB, the syllable nucleus is first assigned to the most sonorant segment based on the scale above.

Based on the sonority scale, Dell and Elmedlaoui developed the following algorithm that is designed to select the correct syllable nucleus.

Syllabification Algorithm

Dell-Elmedlaoui Algorithm for core syllabification (DEA)

Build a core (CV) over each substring of the form XY, where

X is any segment (except [a], and

Y is a matrix of features describing a step of the sonority scale.

The algorithm states that Y starts at the top of the sonority scale and replace it successively with the matrix of features appropriate to the next lower step of the scale.

The algorithm iterates from left to right for each fixing of the nucleus.

Syllabification Algorithm

It is important to mention that the algorithm is subject to Free Element Condition (FEC) which states that the algorithm will only apply to elements that are not supplied with the relevant structure.

Hence, in each iteration, the algorithm apples to fewer elements.

Also, this means that any element seized as an onset in a previous iteration is no longer eligible to be a syllable nucleus.

Lastly, the DEA is a rule-based analysis that relies on rule ordering. Each iteration represents a rule and the first iteration represents the highest ranked rule and the second iteration represents the second highest ranked rule, and so on.

DEA in Action

Steps of the DEA	/txznas/ ‘she stored for him’
Seek [X] [+low,-cns] (low vowel)	txz(**na**)s
Seek [X] [-low, -cns] (high vowel)	_____
Seek [X] [+cns, +son, -nas] (liquid)	_____
Seek [X] [+cns, +son, +nas] (nasal)	Blocked by FEC
Seek [X] [-son, +cnt, +voi] (voiced fricative)	t(**xZ**)(na)s
Seek [X] [-son,+cnt, -voi] (-voice fricative)	Blocked by FEC
Seek [X] [-son, -cnt, +voi] (voiced stop)	_____
Seek [X] [-son, -cnt, -voi] (-voice fricative)	Blocked by FEC

Issues with the DEA

1: The DEA fails to answer the following questions:

a. Why start at the top?
b. Why descend the scale?
c. Why apply the scale to the nucleus rather rather than the onset?

The DEA fails to account for such concerns because it does not include syllable structure markedness in its scheme.

All the DEA does is scanning the input for certain specific forms and acts when it finds them.

Hence, the syllabification case in ITB is better understood when viewed in terms of OT as shown in the next section.

"Optimality" in DEA

1. **Cores of DEA (two rules)**:
   1.1. All non-phrase-initial syllables must have onesets
   1.2. Iterative search for nucleus along the sonority scale

2. **Sequential nature of DEA**:
   2.1. Rule ordering: rule (1.1) "overrides" rule (1.2)
   2.2. Inside the sonority scale: a segment class that is more sonorous take precedence over the less sonorous one in DEA's iterative implementation

3. **Optimal output**: syllables with *most sonorous possible* nuclei *given the two rules*

Rules → Constraints

1. **"Equivalence" between rule and constraint**:
   1.1. A rule to obey ⟺ a constraint not to break
   1.2. A rule ($p \rightarrow q$) ⟺ a constraint ($\neg(p \,\&\, \neg q)$)

2. **Two constraints**:
   2.1. The Onset Constraint (ONS). Syllables must have onsets (except phrase initially).
   2.2. The Nuclear Harmony Constraint (HNUC). A higher sonority nucleus is more harmonic than one of lower sonority. *I.e.* If $|x| > |y|$ then $Nuc/x \succ Nuc/y$.

3. **Harmony**: a well-formedness scale along which a maximal-Harmony structure is well-formed and all other structures are ill-formed.

Rules → Constraints

1. **Difference in reaching the optimal output**:
   1.1. Rule-based: sequential application of rules to generate an output
   1.2. Constraint-based: hierarchical evaluation of constraints to select an optimal output among possible candidate outputs (generated by **Gen**)

2. **Rule ordering → constraint ranking (interaction)**: ONS >> HNUC (both violable)

Candidates	ONS	HNUC
☞ ~.wL.~		|l|
~.ul.~	* !	|u|

3. Iterative search for nucleus along the sonority scale → (iterative) selection of candidates with most harmonic nuclei
   3.1. The former only requires one pass of the sonority scale, but the latter requires multiple lookups

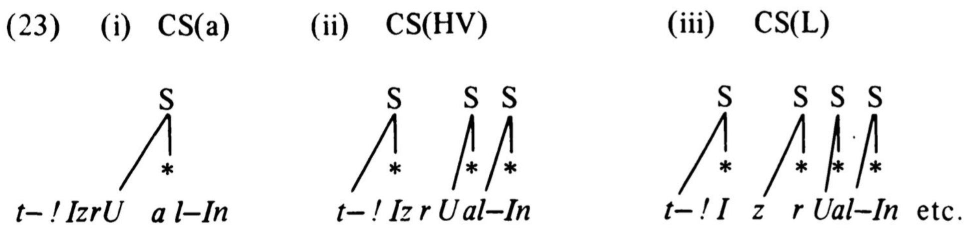
OT in practice

Harmonic Serialism (serial and iterative analysis):

- Gen (*input*$_i$): the set of (partial) syllabifications of *input*$_i$ which differ from *input*$_i$ in no more than one syllabic adjunction.

Candidates	ONS	HNUC	Comments
☞ tx(z**N**)t		n	optimal: onsetted, best available nucleus
txz(**N**)t	* !	n	no onset, HNUC irrelevant
t(x**Z**)nt		z !	|z| < |n|
(t**X**)znt		x !	|x| < |n|
txz(n**T**)		t !	|t| < |n|

How does this work for the following (no winner at ii)?



OT in practice

Classic OT (parallel and recursive analysis):

- Gen ($input_i$): no restriction in the difference between the $input_i$ and the set of (partial) syllabifications of $input_i$

Candidates	ONS	HNUC		Comments
☞ .tX.zNt.		n	x	optimal
.Tx.zNt.		n	t !	|n| = |n|, |t| < |x|
.tXz.nT.		x !	t	|x| < |n|, t irrelevant
.txZ.Nt.	* !	n	z	HNUC irrelevant
.T.X.Z.N.T.	* ! ***	n z x t t		HNUC irrelevant

Summary

Rule-based constructive approach

- suffers from the formal arbitrariness of rewrite rules (appears to be an arbitrary choice among the universe of possibilities)
- is doomed to severe explanatory shortcomings
- fails to offer a uniform treatment for different phonenomena governed by some principles

Constraint-based evaluative approach

- arbitrariness of constructions disappears because all possibilities are evaluated
- “[t]he main explanatory burden falls the constraints themselves, and on the apparatus that governs their interactions.”
- provides a more principled and simplifying way to approach linguistic phonenomena