Proceedings of the Society for Computation in Linguistics

Volume 1 Article 16

2018

Imdlawn Tashlhiyt Berber Syllabification is Quantifier-Free

Kristina Strother-Garcia
University of Delaware, kstrothergarcia@gmail.com

Recommended Citation

Strother-Garcia, Kristina (2018) "Imdlawn Tashlhiyt Berber Syllabification is Quantifier-Free," *Proceedings of the Society for Computation in Linguistics*: Vol. 1 , Article 16.
DOI: https://doi.org/10.7275/R5J67F4D
Available at: https://scholarworks.umass.edu/scil/vol1/iss1/16

Imdlawn Tashlhiyt Berber Syllabification is Quantifier-Free[*]

Kristina Strother-Garcia
University of Delaware
kmsg@udel.edu

Abstract

Imdlawn Tashlhiyt Berber (ITB) is unusual due to its tolerance of non-vocalic syllabic nuclei. Rule-based and constraint-based accounts of ITB syllabification do not directly address the question of how complex the process is. Model theory and formal logic allow for comparison of complexity across different theories of phonology by identifying the computational power (or expressivity) of linguistic formalisms in a grammar-independent way. With these tools, I develop a mathematical formalism for representing ITB syllabification using Quantifier-Free (QF) logic, one of the least powerful logics known. This result indicates that ITB syllabification is relatively simple from a computational standpoint and that grammatical formalisms could succeed with even less powerful mechanisms than are currently accepted.

1 Introduction

Accounting for syllabification in ITB has become a sort of litmus test for phonological frameworks handling syllable theory. Any segment can be nucleic in ITB in some environment, making words like [tX.zNt] 'you store' commonplace.[1] Despite the seemingly bizarre syllables reported in these words, careful study shows that syllabification in ITB is predictable and follows the Sonority Sequencing Principle (SSP)[2] almost perfectly (Dell and Elmedlaoui, 1985; Frampton, 2011; Prince and Smolensky, 1993). In this paper, I develop a QF transduction that maps underlying representations (URs) to syllabified surface representations (SRs) in ITB. This result establishes that ITB syllabification is computationally simple and local in a strict sense, a fact which is not immediately evident from previous analyses.

1.1 Motivation

Phonological processes can be thought of as functions or maps (as in Tesar, 2014) from URs to SRs. For example, the underlying string /ovɚ/ (*over*) maps to the syllabified SR [o.vɚ] in English.

A crucial question is then: what is the nature of this map? One way to characterize a map or function is to examine the kind of logic needed to express it as a transduction. Regular functions are exactly those realized by transducers in Monadic Second-Order (MSO) logic (Engelfriet and Hoogeboom, 2001; Filiot and Reynier, 2016). A strict subset of these functions correspond to transductions definable in First-Order (FO) logic, of which a strict subset are QF-definable.

Many regular functions correspond to hypothetical UR-to-SR maps not observed in the phonology of natural languages. Under certain assumptions, both rule-based theories and Optimality Theory (OT) overgenerate by allowing grammars for such unattested maps (Frank and Satta, 1998; Graf, 2010; Heinz, 2011b; Heinz and Idsardi, 2013; John-

[*] I thank Jeff Heinz, Adam Jardine, Mai Ha Vu, and Hossep Dolatian for insightful feedback on an earlier draft of this paper.

[1] Periods signify syllable boundaries and capital letters indicate nucleic consonants, as in Prince and Smolensky (1993).

[2] This principle has been reiterated in one form or another by Sievers (1881), Saussure (1916), Harris (1969), Hooper (1976), Selkirk (1984), and others.

Proceedings of the Society for Computation in Linguistics (SCiL) 2018, pages 145-153.
Salt Lake City, Utah, January 4-7, 2018

son, 1972; Kaplan and Kay, 1994; Karttunen, 1993; Karttunen, 1998). OT grammars cannot express *all* regular functions, so they also undergenerate in this sense (Buccola, 2013; Idsardi, 2000). In contrast, Declarative Phonology (DP) grammars are FO by definition (Coleman, 1998; Scobbie, 1991). This will be discussed further in §2.2.

While no previous theories of phonology restrict grammars to functions realized by QF maps, certain classes of QF-definable transductions are already known to characterize a variety of phonological processes (Chandlee, 2014; Chandlee et al., 2014; Chandlee et al., 2015; Chandlee and Lindell, 2016). The question of whether all UR-to-SR maps in phonology are QF-definable has strong implications for typology and learnability, as it places an upper limit on the logical complexity of phonological maps. In this paper, I develop a QF transduction for ITB syllabification, showing that this phonological function is computationally local in a strict sense.

1.2 Organization of the Paper

The remainder of the paper is organized as follows. In §2 I give a brief overview of the surface facts in ITB and previous accounts in rule-based and constraint-based frameworks. I then introduce formal word models, transductions, and logics in §3, highlighting the implications of prohibiting quantification. I develop shorthand logical predicates to characterize the input to the ITB syllabification transduction in §4. The transduction itself is defined in §5. Finally, in §6 I conclude.

2 Background on ITB

In this section I review the well-formedness principles evident from SRs in ITB and briefly summarize previous approaches to characterizing them.

2.1 The Basic Facts

Unlike most languages, ITB allows any phonetic segment to be a syllabic nucleus. The main principle driving syllable well-formedness in ITB is the SSP, which states that sonority rises monotonically from a given segment to the sonority peak of its syllable (Selkirk, 1984). Dell and Elmedlaoui (1985) report the following sonority hierarchy for ITB[3]:

$$\text{vcl. stops} <_s \text{vcd. stops} <_s \text{vcl. fric} <_s \text{vcd. fric} <_s \text{nas} <_s \text{liq} <_s \text{HV} <_s \text{[a]} \quad (1)$$

The high vocoids (HVs) are [i,j,u,w]. The symbol $<_s$ denotes lesser sonority. As with the traditional notion of lesser sonority, I assume that the binary relation $<_s$ is irreflexive, asymmetric, and transitive. It is then simple to define relations $=_s$ and $\leq_s$, as in (2-3). These will be of use later.

$$=_s(x,y) \overset{def}{=} \neg <_s(x,y) \wedge \neg <_s(y,x) \quad (2)$$

$$\leq_s(x,y) \overset{def}{=} <_s(x,y) \vee =_s(y,x) \quad (3)$$

In addition to the SSP, there are four other principles of syllable well-formedness to note in ITB. First, all non-initial syllables must have an onset. Second, initial stops and final obstruents are forbidden from being nucleic. Third, with the exception of a small class of morphemes, the glide/vowel distinction among the HVs [i∼j] and [u∼w] is predictable based on syllable position (Dell and Elmedlaoui, 1985). That is, a nucleic HV is vocalic, as in [tag.r**u̲**rt] 'stable,' while a non-nucleic HV is a glide, as in [sa.**w̲**Lx] 'I spoke.' The latter example also illustrates the fourth well-formedness principle in ITB syllabification: the SSP is violated in glide-sonorant (GR) syllables.[4] Following another vowel, the HV in a GR syllable surfaces as a glide and forms the onset to a nucleic sonorant, preventing hiatus. Consider the UR /saulx/. The /a/ must be nucleic because it is the most sonorous possible segment. If the /u/ were also nucleic, it could have no onset. Instead, the /l/ becomes a nucleus and the /u/ becomes its onset, surfacing as the glide [w].

2.2 Previous Approaches

Dell and Elmedlaoui (1985) propose an ordered set of iterative rules to identify syllabic nuclei, each referring to a certain natural class (e.g., voiceless stops). They assign nucleic status first to instances of [a], then to HVs, then to liquids, etc., with the restriction that every non-initial syllable must have an onset. Their rules do not reference sonority directly, but are clearly applied so as to pick out the most

[3] vcl. = voiceless; vcd. = voiced; fric = fricative(s); nas = nasal(s); liq = liquid(s).

[4] As in 'glide-resonant' from Dell and Elmedlaoui (1985).

sonorous segments first and step down in sonority at every subsequent rule application. Any remaining segments are later assigned to coda positions. Frampton (2011) simplifies Dell and Elmedlaoui's (1985) treatment by introducing a way to simultaneously identify all points of application, making explicit reference to notions of "more sonorous" and "more left."

Prince and Smolensky (1993) offer an OT account of ITB syllabification, where GEN produces every possible syllabification of a given input form. The two main OT constraints involved in "core syllabification" penalize non-initial onsetless syllables and syllables whose nuclei are not the most sonorous segment in the syllable. The greater the absolute difference in sonority, the more the low-sonority nucleus is penalized. Additional constraints enforce the remaining well-formedness principles described in §2.1. Importantly, these constraints are all violable and the correct surface form is the one that violates the fewest highly ranked constraints. Global evaluation is required because penalties are summed over the entire candidate.

Rather than allowing SRs to violate "soft" constraints like in OT, DP grammars simply reject any SR that violates any constraint (Bird et al., 1992; Scobbie, 1991, 1996). DP relies on the Elsewhere Condition, which stipulates that, if a sequence of segments is targeted by multiple constraints, the more specific constraint applies. Lexical entries themselves are viewed as constraints, so DP does not formalize phonological grammars as UR-to-SR maps (Scobbie, 1991, 1993). There is only one level of representation and all constraints must be satisfied simultaneously. Scobbie (1993) sketches a DP treatment of ITB syllabification using constraints similar in spirit to those proposed by Prince and Smolensky (1993), but crucially differ in that they are inviolable, unranked, and defined explicitly in FO logic.

In the remaining sections, I develop a QF transduction and illustrate how ITB syllabification can be computed without recourse to global evaluation. My approach diverges from DP in three ways: i) I do not make use of lexical constraints or the Elsewhere Condition; ii) I represent syllabification as a map, requiring two levels of representation; and iii) I use strictly QF logic. This paper is therefore better situated with recent model-theoretic approaches that explicitly examine the computational characteristics of phonological generalizations independent of grammatical formalisms (Chandlee, 2014; Graf, 2010; Heinz, 2011a,b; Jardine, 2016; Rogers and Pullum, 2011).

3 Formal Background

Here I offer formal definitions of model theories, word models, and transductions, as well as an informal explanation of the differences among three logics: MSO, FO, and QF. I focus on the successor model theory, as this will be used to represent the input to the ITB syllabification transduction.

3.1 Model Theories for Words

A word model is a type of graph useful for representing relational structures. Classes of word models are defined by model theories. Given an alphabet Σ, a model theory $\mathfrak{M}$ has the signature $\langle \mathfrak{D}; \mathfrak{R}; \mathfrak{F} \rangle$ where $\mathfrak{D}$ is a domain, $\mathfrak{R}$ is a a set of relations among domain elements (nodes), and $\mathfrak{F}$ is a set of functions. For every σ in Σ there is a unary relation R_σ in $\mathfrak{R}$ that can be thought of as a labeling relation. For example, let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. Then $\mathfrak{R}$ includes the unary relations R_a, R_b, R_c. $\mathfrak{R}$ may also contain additional relations of higher arity. The following example will help to make these definitions clear.

3.2 The Successor Model Theory

The successor model theory $\mathfrak{M}^{\triangleleft}$ is defined in (4).

$$\begin{aligned} \mathfrak{M}^{\triangleleft} \overset{\text{def}}{=} \langle \mathfrak{D}; \{R_\sigma \mid \sigma \in \Sigma\}; \\ \{\mathtt{pred}(x), \mathtt{succ}(x)\}\rangle \end{aligned} \tag{4}$$

The unary functions $\mathtt{pred}(x)$ and $\mathtt{succ}(x)$ pick out the immediate predecessor and successor of a given position, respectively.[5] In the general case, $\mathtt{succ}(x) = x + 1$ and $\mathtt{pred}(x) = x - 1$. To ensure the predecessor function is total, it is defined so that the initial position is its own predecessor, i.e. $\mathtt{pred}(0) = 0$. Similarly, the final position is its own successor, making the successor function total. Then in a string of n positions, $\mathtt{succ}(n) = n$.

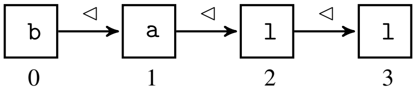
The model for the string *ball* under this theory is denoted $\mathcal{M}^{\triangleleft}_{ball}$. Taking the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{l}\}$,

[5] The use of unary functions (rather than binary relations) for predecessor and successor is due to Chandlee and Lindell (2016).

$\mathcal{M}^{\triangleleft}_{ball}$ is defined in (5) and represented visually in Figure 1.

$$\begin{aligned}
&\mathcal{M}^{\triangleleft}_{ball}\\
&\mathcal{D}=\{0,1,2,3\}\\
&R_a=\{1\}\\
&R_b=\{0\}\\
&R_l=\{2,3\}\\
&\texttt{succ}(x)=\begin{cases}1 & x=0\\ 2 & x=1\\ 3 & x=2\\ 4 & x\in\{3,4\}\end{cases}\\
&\texttt{pred}(x)=\begin{cases}0 & x\in\{0,1\}\\ 1 & x=2\\ 2 & x=3\\ 3 & x=4\end{cases}
\end{aligned} \tag{5}$$

Figure 1: A visual representation of $\mathcal{M}^{\triangleleft}_{ball}$.



Its domain $\mathcal{D}$ consists of four nodes, each represented as a rectangle with an index below it. Unary relations are illustrated as node labels. For example, node 1 is labeled a. This is denoted $1\in R_a$, $R_a(1)=\texttt{TRUE}$, or, equivalently, $\texttt{a}(1)=\texttt{TRUE}$. The successor function is illustrated by directed edges (arrows) with the $\triangleleft$ label. Thus $1\triangleleft 2$ is equivalent to $\texttt{succ}(1)=2$.

3.3 A Modified Successor Model Theory

The remainder of this paper uses a slight variant of the traditional successor model theory. The key difference lies in the definition of the alphabet and the use of non-mutually exclusive position labels.

The alphabet can be conceptualized as a set of primitives – labels defined outside of the model theory itself. In traditional word models (as in Büchi, 1960), each position has exactly one label (i.e., it belongs to a single unary relation). Additionally, traditional labels are simply letters of the alphabet. The unary relations in the word models to follow are untraditional with respect to both of these conventions. In line with previous work in computational phonology (Daland et al., 2011; Heinz and Strother-Garcia, to appear; Strother-Garcia et al., 2017), I permit each position to have more than one label. This allows us to represent phonological segments as bundles of information like phonological features, as well as syllable position (onset, nucleus, or coda), rather than disparate symbols.

Let $\mathcal{F}$ be a set of primitive phonological features. I adopt the features given in (6) for ITB.[6]

$$\begin{aligned}\mathcal{F}\overset{\text{def}}{=}\{&\texttt{voice},\texttt{vocoid},\texttt{high},\texttt{lab},\texttt{alv},\texttt{post},\\ &\texttt{pal},\texttt{vel},\texttt{uv},\texttt{phar},\texttt{glot},\texttt{stop},\texttt{fric},\\ &\texttt{nas},\texttt{approx},\texttt{lat}\}\end{aligned} \tag{6}$$

Then the alphabet is simply $\Sigma=\mathcal{F}$. For each feature $f\in\mathcal{F}$, there is a unary relation $R_f\in\mathfrak{R}$ that represents a particular position being labeled with that feature. Let $\mathcal{R}_f$ be the set of such relations, defined in (7). As with alphabet primitives in traditional word models, $R_f(x)$ can also be written as $f(x)$ for any primitive $f\in\mathcal{F}$. For example, $R_{\texttt{voice}}(x)$ is equivalent to $\texttt{voice}(x)$.

$$\mathcal{R}_f\overset{\text{def}}{=}\{R_f\mid f\in\mathcal{F}\} \tag{7}$$

In addition to this set of unary relations, I will make use of the binary sonority relations defined in §2.1. Let the sonority relations be members of the set $\mathcal{R}_s$, as defined in (8). Then the modified successor model theory $\mathfrak{M}$ is defined in (9).

$$\mathcal{R}_s\overset{\text{def}}{=}\{<_s,=_s,\leq_s\} \tag{8}$$

$$\mathfrak{M}\overset{\text{def}}{=}\langle\mathfrak{D};\{\mathcal{R}_f\cup\mathcal{R}_s\};\{\texttt{pred}(x),\texttt{succ}(x)\}\rangle \tag{9}$$

3.4 Graph Transductions

As word models are a type of graph, graph transductions can be used to represent input-output maps from one word model $\mathfrak{M}^A$ to another $\mathfrak{M}^B$. A transduction is defined with a set of formulas, one for each relation R and function F in $\mathfrak{M}^B$. These formulas are interpreted with respect the input structure in $\mathfrak{M}^A$. See (Courcelle, 1994; Engelfriet and Hoogeboom, 2001) for details.

For example, consider a transduction Γ_{ba} that changes all bs in a word model to as. Here $\mathfrak{M}^A=\mathfrak{M}^B=\mathfrak{M}^{\triangleleft}$. Then given $\Sigma=\{\texttt{a},\texttt{b},\texttt{l}\}$, the transduc-

[6] lab = labial; alv = alveolar; post = postalveolar; pal = palatal; vel = velar; uv = uvular; phar = pharyngeal; glot = glottal; approx = approximant; lat = lateral.

tion Γ_{ba} is the set of predicates (10-14) where the superscript ω indicates the relations over the output.

When applied to the input $\mathcal{M}^{\lhd}_{ball}$, the transduction changes the label of the first position from b to a and leaves the remaining positions unchanged, as illustrated in Figure 2.

$$R^{\omega}_{a}(x) \overset{\text{def}}{=} R_a(x) \vee R_b(x) \tag{10}$$

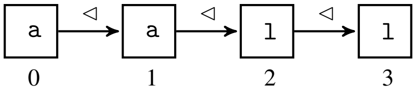
$$R^{\omega}_{b}(x) \overset{\text{def}}{=} \texttt{FALSE} \tag{11}$$

$$R^{\omega}_{l}(x) \overset{\text{def}}{=} R_l(x) \tag{12}$$

$$\texttt{succ}^{\omega}(x) \overset{\text{def}}{=} \texttt{succ}(x) \tag{13}$$

$$\texttt{pred}^{\omega}(x) \overset{\text{def}}{=} \texttt{pred}(x) \tag{14}$$

Figure 2: A visual representation of $\Gamma_{ba}(\mathcal{M}^{\lhd}_{ball})$.



In this example, the input and output share the same model theory, but this need not be the case. As will be seen in §5, new relations may be added to the output model theory.

3.5 Logics and Locality

Statements in FO logic can use universal and existential quantifiers to quantify over elements of the domain. MSO statements can also quantify over *sets* of domain elements. For example, consider the following definitions from Jardine and Heinz (2015). First, set closure under successor is defined in (15). Then (16) defines the general precedence relation $\prec$. Captial X denotes a set, while lowercase x and y denote elements of the domain.

$$\begin{aligned}\mathsf{closed}(X) \overset{\text{def}}{=} & (\forall x, y)(x \in X \wedge x \lhd y) \\ & \Rightarrow y \in X\end{aligned} \tag{15}$$

$$\begin{aligned}x \prec y \overset{\text{def}}{=} & (\forall X)(x \in X \wedge \mathsf{closed}(X) \\ & \Rightarrow y \in X\end{aligned} \tag{16}$$

Because (15) involves universal quantification over some set X, it is strictly MSO and not FO. Sentences of MSO without quantification over sets, like (16), are FO. Sentences of FO with no quantification are QF.[7]

To see why quantification is important, compare (17) to (10), which is reproduced in (18). The former states that an output position x will be labeled a if the corresponding input position is an a *or* if there is a position labeled b somewhere in the input. Checking whether $R^{\omega\prime}_{a}(x)$ is true requires global evaluation of the string to see if any position is labeled b. This is due to the existential quantifier $\exists$, which makes (17) strictly FO. In contrast, (18) lacks any quantification. $R^{\omega}_{a}(x)$ can be evaluated independently at every position in the string.

$$R^{\omega\prime}_{a}(x) \overset{\text{def}}{=} R_a(x) \vee (\exists y)[R_b(y)] \tag{17}$$

$$R^{\omega}_{a}(x) \overset{\text{def}}{=} R_a(x) \vee R_b(x) \tag{18}$$

This example illustrates the relationship between quantification and locality. If a predicate is stated with quantification, computing its truth value requires global evaluation of the string. If the predicate is QF, truth evaluation must be possible over a substring of bounded size. A transduction defined entirely by QF predicates is a QF transduction. Thus a QF transduction amounts to a constraint-checking function that operates locally, within a bounded window of evaluation. Note that Γ_{ba} is QF, with all predicates referring to a single position.

4 User-Defined Predicates

Given a model theory $\mathfrak{M}$ and an alphabet Σ, logical predicates can be defined to make it easier to refer to certain types of information in the input word model. For example, writing $\texttt{voice}(x) \wedge \texttt{lab}(x) \wedge \texttt{stop}(x)$ to refer to a [b] is cumbersome. Instead, I use the unary predicate $\mathsf{b}(x)$, defined in (19). HVs must also be defined, as in (20).

$$\mathsf{b}(x) \overset{\text{def}}{=} \texttt{voice}(x) \wedge \texttt{lab}(x) \wedge \texttt{stop}(x) \tag{19}$$

$$\mathsf{HV}(x) \overset{\text{def}}{=} \texttt{high}(x) \wedge \texttt{vocoid}(x) \tag{20}$$

Although I write $\texttt{stop}(x)$ and $\mathsf{b}(x)$ similarly, note that the former is the labeling relation for a primitive of $\mathfrak{M}$ while the latter is a predicate derived from such primitives. I use `typewriter` font for primitives and sans serif font for user-defined predicates.

Note that whether a formula is MSO, FO, or QF is determined by its interpretation in terms of primitives. For example, the statement $\mathsf{HV}(x) \vee \mathsf{b}(x)$ is

[7] For formal definitions of MSO and FO, I refer the reader to Enderton (2001), Fagin et al. (1995), and Shoenfield (1967).

QF because the predicates $\mathsf{HV}(x)$ and $\mathsf{b}(x)$ are both QF. User-defined predicates are not meant to obscure the logical nature of the description; they are just well-defined abbreviations.

4.1 Natural Classes and Word Position

Natural classes can be defined similarly to $\mathsf{b}(x)$. Given the primitives in $\mathcal{R}_f$, I define obs and son in (21-22).[8]

$$\mathsf{obs}(x) \overset{\text{def}}{=} \mathtt{stop}(x) \vee \mathtt{fric}(x) \tag{21}$$

$$\mathsf{son}(x) \overset{\text{def}}{=} \neg\mathsf{obs}(x) \tag{22}$$

Unary predicates can also pick out a segment's position in the word. Initial and final positions are defined as in §3.2. Then a medial position is one that is neither initial nor final. These definitions are formalized in (23-25).

$$\mathsf{init}(x) \overset{\text{def}}{=} \mathtt{pred}(x) = x \tag{23}$$

$$\mathsf{fin}(x) \overset{\text{def}}{=} \mathtt{succ}(x) = x \tag{24}$$

$$\mathsf{med}(x) \overset{\text{def}}{=} \neg(\mathsf{init}(x) \vee \mathsf{fin}(x)) \tag{25}$$

4.2 Sonority and Other Considerations

To determine syllable constituency, it is first necessary to identify sonority peaks, other positions that may be nucleic, and marked positions prohibited from being nucleic.

A word-medial sonority peak is simply a segment that is more sonorous than both its neighboring segments, as defined in (26). To be exhaustive, I also define word-initial and word-final 'peaks' in (27) and (28), respectively. Then a sonority peak (29) is any of these three.

$$\begin{aligned}\mathsf{med\_pk}(x) \overset{\text{def}}{=} &\ \mathsf{med}(x) \wedge \mathtt{pred}(x) <_s x \\ &\wedge \mathtt{succ}(x) <_s x\end{aligned} \tag{26}$$

$$\mathsf{init\_pk}(x) \overset{\text{def}}{=} \mathsf{init}(x) \wedge \mathtt{succ}(x) <_s x \tag{27}$$

$$\mathsf{fin\_pk}(x) \overset{\text{def}}{=} \mathsf{fin}(x) \wedge \mathtt{pred}(x) <_s x \tag{28}$$

$$\begin{aligned}\mathsf{son\_pk}(x) \overset{\text{def}}{=} &\ \mathsf{med\_pk}(x) \vee \mathsf{init\_pk}(x) \\ &\vee \mathsf{fin\_pk}(x)\end{aligned} \tag{29}$$

Frampton (2011) observes that "a slot x is 'more prominent' than an adjacent slot y ...if they are equally sonorous and x is to the left of y, unless x is initial." In other words, the leftmost segment of a sonority plateau takes prominence when it comes to assigning nucleic status, unless it is word-initial. This configuration is captured by the predicate left_prom, defined in (30). Note that word-final positions are explicitly excluded. Were this left out of the definition, every final position would satisfy left_prom due to it being its own successor and, therefore, equally sonorous to its successor (itself). Then a prominence peak is a position that satisfies either son_pk or left_prom, as in (31).

$$\mathsf{left\_prom}(x) \overset{\text{def}}{=} x =_s \mathtt{succ}(x) \wedge \mathsf{med}(x) \tag{30}$$

$$\mathsf{prom\_pk}(x) \overset{\text{def}}{=} \mathsf{son\_pk}(x) \vee \mathsf{left\_prom}(x) \tag{31}$$

Prominence peaks are typically syllabic nuclei, with three exceptions. The first two exceptions are simple: neither initial stops nor final obstruents may be nucleic. I refer to both as 'marked,' represented by the shorthand predicate mrkd in (32). The third exception occurs when a HV is preceded by another vowel, resulting in a GR syllable. The shorthand predicate GR_nuc (33) picks out the sonorant in a GR syllable, which is always nucleic.

$$\mathsf{mrkd}(x) \overset{\text{def}}{=} \mathsf{init\_stop}(x) \vee \mathsf{fin\_obs}(x) \tag{32}$$

$$\begin{aligned}\mathsf{GR\_nuc}(x) \overset{\text{def}}{=} &\ \mathtt{vocoid}(\mathtt{pred}(x)) \wedge \mathsf{son}(x) \\ &\wedge \mathsf{prom\_pk}(\mathtt{pred}(\mathtt{pred}(x)))\end{aligned} \tag{33}$$

4.3 Syllable Constituency

Now it is easy to identify the syllable constituent for any given input segment. Predicate (34) states that a segment is nucleic if it is **a)** an unmarked prominence peak or **b)** the sonorant in a GR sequence.[9] A segment is an onset if it is not nucleic, but its successor is; this type of onset segment satisfies ons_1 (35). Additionally, a word-initial obstruent satisfies ons_2 (36) if its successor satisfies ons_1, as in the first syllable of [txZ.nas] 'store (3rd sg. fem.)'. In either case, the segment is part of an onset, thereby satisfying ons (37). Finally, a segment is a coda if it is neither an onset nor a nucleus, as in (38).

$$\begin{aligned}\mathsf{nuc}(x) \overset{\text{def}}{=} &\ (\mathsf{prom\_pk}(x) \wedge \neg\mathsf{mrkd}(x)) \\ &\vee \mathsf{GR\_nuc}(x)\end{aligned} \tag{34}$$

[8] obs = obstruent; son = sonorant; init = initial; fin = final; med = medial; pk = peak; prom = prominence; mrkd = marked.

[9] This insight is due to Frampton (2011).

$$\mathsf{ons_1}(x) \overset{\text{def}}{=} \neg \mathsf{nuc}(x) \wedge \mathsf{nuc}(\mathtt{succ}(x)) \tag{35}$$

$$\mathsf{ons_2}(x) \overset{\text{def}}{=} \mathsf{init\_obs} \wedge \mathsf{ons_1}(\mathtt{succ}(x)) \tag{36}$$

$$\mathsf{ons}(x) \overset{\text{def}}{=} \mathsf{ons_1}(x) \vee \mathsf{ons_2}(x) \tag{37}$$

$$\mathsf{cod}(x) \overset{\text{def}}{=} \neg \mathsf{nuc}(x) \wedge \neg \mathsf{ons}(x) \tag{38}$$

5 The ITB Syllabification Transduction

In addition to predicates corresponding to the relations in $\mathfrak{M}$, I will also define relations over the output to indicate a segment's position in the syllable (ons, nuc, cod), which is not explicit in the input. Let $\mathcal{R}_\sigma$ be the set of these three syllable constituent labels, as in (39). Then the model theory of the output will be $\mathfrak{M}'$, defined in (40).

$$\mathcal{R}_\sigma \overset{\text{def}}{=} \{\mathsf{ons},\ \mathsf{nuc},\ \mathsf{cod}\} \tag{39}$$

$$\mathfrak{M}' \overset{\text{def}}{=} \langle \mathfrak{D}'; \{\mathcal{R}_f \cup \mathcal{R}_s \cup \mathcal{R}_\sigma\}; \{\mathtt{pred}(x), \mathtt{succ}(x)\}\rangle \tag{40}$$

Armed with the predicates defined in the previous section, Γ itself is now simple to define. The transduction is completely defined with predicates (42-50). Crucially, all predicates are QF, showing that ITB syllabification is fundamentally local in nature.

5.1 Unary Relations

Because I am concerned with syllabification and not unrelated segmental processes, I assume all feature labels are preserved under Γ. Recall that $\mathcal{R}_f$ is the set of unary relations for phonological features defined over the domain of the transduction. Let $\mathcal{R}_f^\omega$ (41) be the corresponding set of unary relations over the codomain. For each feature f in $\mathcal{F}$, there is one such predicate R_f^ω, as defined in (42).

$$\mathcal{R}_f^\omega \overset{\text{def}}{=} \{R_f^\omega \mid f \in \mathcal{F}\} \tag{41}$$

$$R_f^\omega(x) \overset{\text{def}}{=} R_f(x) \tag{42}$$

5.2 Binary Relations

In the absence of any changes to segmental feature specification, the binary sonority relations (42-45) are also preserved from the input.

$$<_s^\omega (x,y) \overset{\text{def}}{=} <_s (x,y) \tag{43}$$

$$=_s^\omega (x,y) \overset{\text{def}}{=} =_s (x,y) \tag{44}$$

$$\leq_s^\omega (x,y) \overset{\text{def}}{=} \leq_s (x,y) \tag{45}$$

5.3 Functions

The ordering of domain elements does not change, so the output functions $\mathtt{succ}^\omega(x)$ and $\mathtt{pred}^\omega(x)$ are similarly preserved, as in (46-47).

$$\mathtt{succ}^\omega(x) \overset{\text{def}}{=} \mathtt{succ}(x) \tag{46}$$

$$\mathtt{pred}^\omega(x) \overset{\text{def}}{=} \mathtt{pred}(x) \tag{47}$$

5.4 Syllable Constituents

The work of identifying onsets, nuclei, and codas in the input form is essentially already done. All that remains is to formalize the predicates that label the syllable constituents in the *output* form. These are given in (48-50).

$$\mathsf{nuc}^\omega(x) \overset{\text{def}}{=} \mathsf{nuc}(x) \tag{48}$$

$$\mathsf{ons}^\omega(x) \overset{\text{def}}{=} \mathsf{ons}(x) \tag{49}$$

$$\mathsf{cod}^\omega(x) \overset{\text{def}}{=} \mathsf{cod}(x) \tag{50}$$

5.5 Example

To illustrate how the transduction works, consider the underlying form /saulx/ 'I spoke.' Its word model consists of five positions, 0 through 4. Each position has a set of feature labels which I abbreviate with the shorthand segment predicates s, a, u, l, and x, defined analogously to b (19). Table 1 gives the truth values for the predicates relevant to syllable structure in the word *saulx*.

The first position is less sonorous than the second, with sonority falling monotonically after that. There are no sonority plateaus, so no position may satisfy left_prom. The only prominence peak is then the single sonority peak, position 1. Because position 4 is a final obstruent, it is marked. Note that position 1 is a vowel, position 2 is a glide, and position 3 is a sonorant. This configuration means that position 3 satisfies GR_nuc and therefore satisfies nuc, even though it is not a prominence peak. Position 1 also satisfies nuc by virtue of satisfying prom_pk. Positions 0 and 2 satisfy $\mathsf{ons_1}$ because their successors are both nucleic. Finally, position 4 satisfies cod because it satisfies neither ons nor nuc.

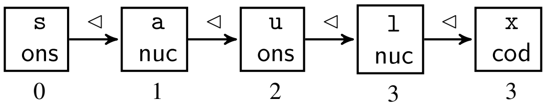
Figure 3 illustrates the resulting output form $\Gamma(\mathcal{M}_{saulx})$, also denoted $\mathcal{M}'_{saulx}$. Recall that the vowel-glide distinction is predictable from syllable constituency. Because position 2 satisfies u and ons,

Table 1: Truth table for $\mathcal{M}_{saulx}$).

x	0	1	2	3	4
s(x)	✓	.	.	.	.
a(x)	.	✓	.	.	.
u(x)	.	.	✓	.	.
l(x)	.	.	.	✓	.
x(x)	.	.	.	.	✓
$x <_s \texttt{succ}(x)$	✓	.	.	.	.
$x =_s \texttt{succ}(x)$	.	.	.	.	.
son_pk(x)	.	✓	.	.	.
left_prom(x)	.	.	.	.	.
prom_pk(x)	.	✓	.	.	.
fin_obs(x)	.	.	.	.	✓
mrkd(x)	.	.	.	.	✓
GR_nuc(x)	.	.	.	✓	.
nuc(x)	.	✓	.	✓	.
ons$_1$(x)	✓	.	✓	.	.
ons(x)	✓	.	✓	.	.
cod(x)	.	.	.	.	✓

it surfaces as the glide [w]. Thus the surface form is pronounced [sa.wLx].

Figure 3: A visual representation of $\mathcal{M}'_{saulx}$.



6 Conclusion

I have shown that syllabification in ITB can be represented by a QF graph transduction, a formalism restricted to substantially lower computational complexity than proposed phonological grammars. Unlike these grammatical formalisms, the logical formalism makes no commitment to the implementation of the UR-to-SR map.

So which aspects of complexity are genuinely linguistic and which ones are by-products of the chosen formalism? This paper demonstrates how model theory and logic provide a foundation for studying this question. The minimal power of the logic needed to define a word model transduction is a measure of the complexity of the UR-to-SR map. Establishing that ITB syllabification is QF highlights an insight not apparent from grammatical formalisms: the local nature of computing syllable constituency in ITB is exactly what precludes the need for quantification.

This result, along with others cited previously, suggests that a fruitful avenue of future research in rule-based frameworks and OT would be to identify which properties of their machinery are responsible for their relatively high logical power. It is possible that careful modifications may increase restrictiveness in a way that makes these grammars computationally equivalent to QF transductions.

References

Steven Bird, John S Coleman, Janet Pierrehumbert, and James M Scobbie. 1992. Declarative Phonology. In *Proceedings of the XVth International Congress of Linguists. Université Laval, Québec*. Université Laval, Québec.

Brian Buccola. 2013. On the expressivity of Optimality Theory versus ordered rewrite rules. In Glyn Morrill and MarkJan Nederhof, editors, *Proceedings of Formal Grammar 2012 and 2013*, volume 8306 of *Lecture Notes in Computer Science*, pages 142–158, Berlin Heidelberg. Springer-Verlag.

J. Richard Büchi. 1960. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92.

Jane Chandlee and Steven Lindell. 2016. Local languages. Paper presented at the 4th Workshop on Natural Language and Computer Science, in affiliation with LICS at Columbia University, NY.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–503.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2015. Output strictly local functions. In *Proceedings of the 14th Meeting on the Mathematics of Language*, pages 112–125. Association for Computational Linguistics.

Jane Chandlee. 2014. *Strictly Local Phonological Processes*. Ph.D. thesis, University of Delaware.

John Coleman. 1998. *Phonological representations: their names, forms and powers*. Cambridge University Press.

Bruno Courcelle. 1994. Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science*, 126(1):53–75.

Robert Daland, Bruce Hayes, James White, Marc Garellek, Andreas Davis, and Ingrid Normann. 2011. Explaining sonority projection effects. *Phonology*, 28(197):197–234.

François Dell and Mohamed Elmedlaoui. 1985. Syllabic consonants and syllabification in Imdlawn Tashlhiyt

Berber. *Journal of African Languages and Linguistics*, 7:105–130.

Herbert B. Enderton. 2001. *A Mathematical Introduction to Logic*. Academic Press, 2nd edition.

Joost Engelfriet and Hendrik Jan Hoogeboom. 2001. MSO definable string transductions and two-way finite-state transducers. *ACM Transactions on Computational Logic (TOCL)*, 2(2):216–254.

R. Fagin, L.J. Stockmeyer, and M.Y. Vardi. 1995. On monadic NP vs monadic co-NP. *Information and Computation*, 120(1):78 – 92.

Emmanuel Filiot and Pierre-Alain Reynier. 2016. Transducers, logic and algebra for functions of finite words. *ACM SIGLOG News*, 3(3):4–19.

John Frampton. 2011. GDE syllabification: A generalization of Dell and Elmedlaoui's syllabification algorithm. *The Linguistic Review*, 28(3):241–279.

Robert Frank and Giorgio Satta. 1998. Optimality Theory and the generative complexity of constraint violability. *Computational Linguistics*, 24(2):307–315.

Thomas Graf. 2010. Logics of phonological reasoning. Master's thesis, University of California, Los Angeles.

James W Harris. 1969. *Spanish phonology*. Cambridge, Mass.: Massachusetts Institute of Technology Press.

Jeffrey Heinz and William Idsardi. 2013. What complexity differences reveal about domains in language. *Topics in cognitive science*, 5(1):111–131.

Jeffrey Heinz and Kristina Strother-Garcia. to appear. Cluster reduction in Tibetan. In Jeffrey Heinz, editor, *Doing Computational Phonology*. Oxford: Oxford University Press.

Jeffrey Heinz. 2011a. Computational phonology part I: Foundations. *Language and Linguistics Compass*, 5(4):140–152.

Jeffrey Heinz. 2011b. Computational phonology part II: Grammars, learning, and the future. *Language and Linguistics Compass*, 5(4):153–168.

Joan B Hooper. 1976. *An introduction to natural generative phonology*. New York: Academic Press.

William J Idsardi. 2000. Clarifying opacity. *Linguistic review*, 17(2):337–350.

Adam Jardine and Jeffrey Heinz. 2015. A concatenation operation to derive autosegmental graphs. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, pages 139–151, Chicago, USA, July.

Adam Jardine. 2016. *Locality and non-linear representations in tonal phonology*. Ph.D. thesis, University of Delaware.

C Douglas Johnson. 1972. *Formal aspects of phonological description*. The Hague: Mouton.

Ronald M Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.

Lauri Karttunen. 1993. Finite-state constraints. *The last phonological rule*, pages 173–194.

Lauri Karttunen. 1998. The proper treatment of optimality in computational phonology: plenary talk. In *Proceedings of the International Workshop on Finite State Methods in Natural Language Processing*, pages 1–12. Association for Computational Linguistics.

Alan Prince and Paul Smolensky. 1993. *Optimality Theory: Constraint interaction in Generative Grammar*. Blackwell, Malden, Mass.

James Rogers and Geoffrey K. Pullum. 2011. Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*, 20:329–342.

Ferdinand de Saussure. 1916. *Cours de linguistique générale*. Paris: Payot.

James M Scobbie, John S Coleman, and Steven Bird. 1996. *Key aspects of Declarative Phonology*. European Studies Research Institute, Salford.

James M Scobbie. 1991. Towards Declarative Phonology. *Edinburgh Working Papers in Cognitive Science*, 7:1–27.

James M Scobbie. 1993. Constraint violation and conflict from the perspective of Declarative Phonology. *Canadian Journal of Linguistics/Revue canadienne de linguistique*, 38(2):155–167.

Elisabeth O Selkirk. 1984. On the major class features and syllable theory. In M. Halle, M. Aronoff, and R. T. Oehrle, editors, *Language sound structure: Studies in phonology*. MIT Press, Cambridge, Mass.

Joseph R Shoenfield. 1967. *Mathematical logic*, volume 21. Reading: Addison-Wesley.

Eduard Sievers. 1881. *Grundz uge der Phonetik*. Leipzig: Breitkopf and Hartel.

Kristina Strother-Garcia, Jeffrey Heinz, and Hyun Jin Hwangbo. 2017. Using model theory for grammatical inference: A case study from phonology. In Sicco Verwer, Menno van Zaanen, and Rick Smetsers, editors, *Proceedings of The 13th International Conference on Grammatical Inference*, volume 57 of *Proceedings of Machine Learning Research*, pages 66–78, Delft, The Netherlands, 05–07 Oct. PMLR.

Bruce Tesar. 2014. *Output-driven phonology: Theory and learning*. Number 139. Cambridge University Press.