Sonority and Syllabification in a Connectionist Network: An Analysis of BrbrNet

Paul Tupper and Michael Fry

Abstract
This work explores the connectionist network BrbrNet and gives a comprehensive formal analysis of its operation. BrbrNet (Legendre, Sorace, and Smolensky 2006) is a connectionist implementation of the syllabification of words in Imdlawn Tashylhiyt Berber (ITB). Numerical simulations suggested that BrbrNet correctly syllabified most input strings of segments, given special initial conditions for the network, but there was no proof offered of the formal soundness of it. Indeed, an example of an input where BrbrNet converges to the wrong syllabification was subsequently identified (Christo Kirov, personal communication). We provide an alternate set of parameters for which BrbrNet converges for all allowed inputs. The proof of soundness for these parameters highlights the similiarity between the dynamics of BrbrNet and the originally proposed serial syllabification algorithm for ITB (Dell and Elmedlaoui 1985). We go on to discuss the feasibility of BrbrNet and similar models as a realistic component of language production.

1. Introduction

One outstanding challenge in linguistics and cognitive science is determining how language is implemented in the brain. At the level of linguistic analysis, language appears to be discrete and symbolic; at the level of neurology, the brain consists of a network of billions of neurons with continuously varying activations. One particular proposal for the unification of these two levels is the Integrated Connectionist/Symbolic (ICS) Cognitive Architecture, as presented in *The Harmonic Mind* (Smolensky and Legendre 2006). A test case for which Smolensky and Legendre's proposal is well-developed is the syllabification of Imdlawn Tashlhiyt Berber (ITB), a system first analysed in Dell and Elmedlaoui (1985). In Legendre, Sorace, and Smolenksy (2006) a connectionist network BrbrNet is described that implements at the level of connectionist units the phenomena of ITB syllabification. Moreover, the workings of BrbrNet can simultaneously be understood at both the symbolic and the connectionist level: either as the finding of an optimal output for satisfying a ranked list of constraints, or as the changing of the activations in a network of nodes. What unifies the two levels of analysis is the idea of *Harmony*. The correct output of BrbrNet for a given input is the one that maximizes harmony. At the symbolic level, the most harmonic candidate is the one that best satisfies the hierarchy of ranked constraints; at the connectionist level, it is defined as a mathematical function of the activations of the nodes in the network.

Syllabification of a word in ITB depends only upon the sonority of the segments in the word. Dell and Elmedlaoui (1985) model the relative sonority of segments by locating them on an 8-point scale which we show in Table 1. Accordingly, their work and all subsequent work on ITB syllabification following it, including ours, is completely dependent on the concept of sonority. However, we remain agnostic on whether or not sonority is an innate part of the language facility. For us, the sonority scale is "a statement of the relative inclination of the segments of a language to be the nucleus of their particular syllable" (Goldsmith and Larson 1990).

Legendre, Sorace, and Smolensky (2006) provide extensive evidence that BrbrNet correctly implements ITB syllabification as described by Dell and Elmedlaoui (1985). However, a problem remains in the analysis of BrbrNet: the winning output must be the most harmonic of all candidates in order to reproduce the correct linguistic behavior. But in a harmony-maximising connectionist network such as BrbrNet, the system goes to the most harmonic among the candidates accessible from the initial conditions and not the most harmonic overall, as we shall see below. In other words, the symbolic nature of language requires outputs that are *global* maxima of harmony, while connectionist systems in general only output *local* maxima of harmony. This mismatch leads to a challenge for BrbrNet and other connectionist implementations of symbolic systems.

To demonstrate this problem, we start with a simple example of syllabificiation in ITB: we consider a word with two segments /gl/ 'bust' (Lahrouchi 2010). Syllabification in this case consists in assigning each segment in the word the role of nucleus or the role of margin, where we use the term margin to denote either an onset or a coda. The constraints relevant for an OT analysis (Prince and Smolensky 1993/2004) are ONSET and HNUC.

(1) ONSET: A non-initial syllable must have an onset.

(2) HNUC: A higher sonority nucleus is more harmonic than one of lower sonority.

(Note that following Prince and Smolensky (1993/2004), we use a non-standard version of the constraint ONSET, which allows onset-free initial syllables. See McCarthy and Prince (1993) for a treatment of the same phenomena with standard constraints.) ONSET is ranked higher than HNUC, which prevents both segments of /gl/ from being nuclei at the same time, regardless of the input. According to the sonority hierarchy introduced in Dell and Elmedlaoui (1985), which we show in Table 1, the segment /l/ is more sonorous than /g/. So the effect of HNUC for this particular input is to make the nucleus be /l/ and not /g/, resulting in /g/ forming the onset. Following the notation of Dell and Elmedlaoui (1985) and Smolensky and Legendre (2006), we will denote this syllabification by [gL], using a capital letter to indicate a syllable nucleus and a lower-case letter to indicate a syllable margin.

Table 1. Sonority index for the segments of ITB

Sonority Class	Segments	Sonority Index
voiceless stops	t, k	1
voiced stops	b, d, g	2
voiceless fricatives	s, f, x	3
voiced fricatives	z, γ	4
nasals	m, n	5
liquids	l, r	6
high vocoids	i/y, u/w	7
low vowel	a	8

In contrast to the static OT analysis of the syllabification of /gl/, BrbrNet is a connectionist network whose dynamics compute the correct syllabification. The syllabification is computed using two coupled nodes, each corresponding to one of the input segments. Each node can take on any activation between 0 and 1. The final states of the nodes determine the syllabification: a node having activation 0 indicates a margin and a node having activation 1 indicates a nucleus. Let x_1 be the activation of the /g/

node and let x_2 be the activation of the /l/ node at a given time. The activations are restricted to lie between 0 and 1, so that the state of the system at any time can be viewed as a point within a unit square, as shown in Figure 1. The effect of the dynamics specified above is for the system to maximize the harmony

(3) $$H(x_1,x_2) = -256x_1x_2 + x_1 + 7x_2.$$

This Harmony function consists of three terms, each of which has a phonological interpretation. The first term $-256x_1x_2$ is never positive, and is maximized when it has value zero. This can only occur when either x_1 or x_2 is zero, which corresponds to the constraint ONSET. The second term, x_1, is maximized when x_1 is 1, and the third term, $7x_2$, is maximized when x_2 is one. The fact that the coefficient of the third term is larger than the coefficient of the second term corresponds to the constraint HNUC.

Once initial activations are selected, the activations of the nodes will change in time to increase H until it reaches a state where H cannot increase further without leaving the square $0 \le x_1,x_2 \le 1$. Figure 1 shows the space of possible states for BrbrNet together with trajectories of the system for different initial conditions.

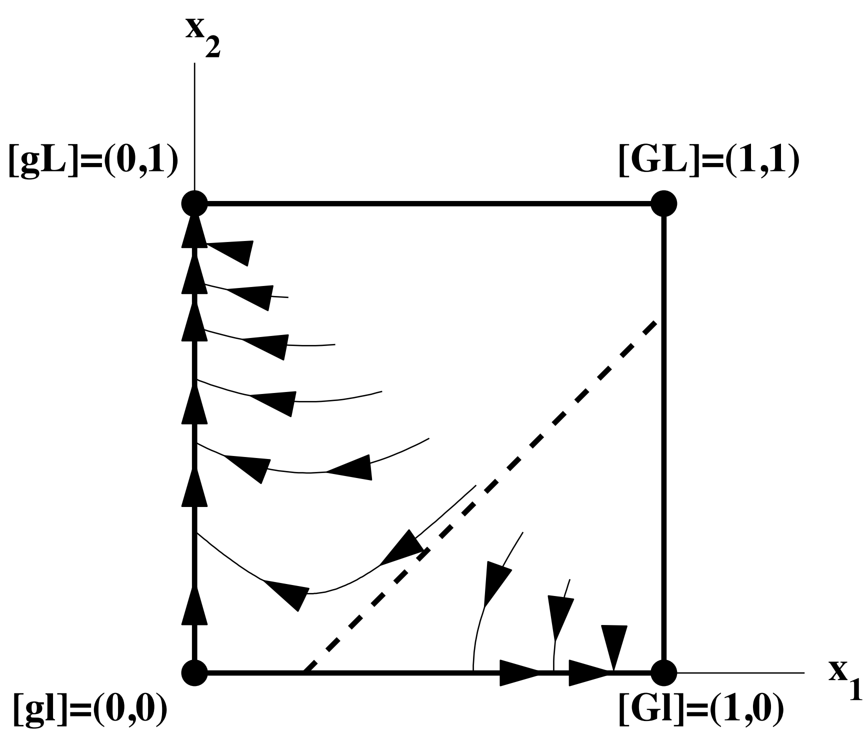

Figure 1. The state space of BrbrNet for the input /gl/.

The horizontal axis corresponds to the activation of the first node x_1 and the vertical axis corresponds to the activation of the second node x_2. There are two local harmony maxima within the square. One at (0,1) corresponding to the syllabification [gL] has harmony 7, and the other at (1,0) corresponding to the syllabification [Gl] has harmony 1. The (0,1) state has the highest harmony of all states in the square, and is therefore the correct output. Indeed, if the system starts out in state (0,0), as is intended for BrbrNet, the system will converge to (0,1). This is shown by the arrows on the left side of the square. (The trajectory does not deviate sufficiently from the line $x_1 = 0$ for it to be visible on this plot.) However, depending on where the system is started initially, different outcomes are possible. The thin lines with arrows in the box show the trajectory of the system from different starting points. Whenever the initial condition is above the dashed line in the figure, the system will converge to (0,1), which is the correct syllabification for the input /gl/. Whenever the initial condition is below the dashed line, it will converge to (1,0), which is the incorrect syllabification.

If we think of syllabification as the process of selecting nuclei for syllables to form around, then it is natural to start with no nuclei, just margins, which corresponds to the initial activations (0,0). When we assume that the system starts at (0,0), BrbrNet computes the correct output for this input. Indeed, BrbrNet, with the original parameters and an initial condition of all activations 0, produces the correct output for all input strings of 4 or fewer segments. But as we shall see, there are strings with more segments in which erroneous output is generated. It turns out that for these problematic sequences, trajectories started at the state of all activations 0 converge to a *local* maximum of harmony (as is (0,1) in this example) instead of the *global* maximum of harmony ((1,0) in this example).

In what follows we will explain how local harmony maxima lead to problems for BrbrNet with its original parameters. We have identified new parameters for which BrbrNet always outputs the correct syllabification and we will outline the proof of the soundness of BrbrNet with these new parameter settings.

In Section 2 we review the basic facts of syllabificiation in ITB and present how the system is modeled in both the Optimality Theory and Harmonic Grammar frameworks. In Section 3 we give a detailed review of BrbrNet and then show how it breaks down on particular examples for the original parameters. In Section 4 we provide the proof of soundness of BrbrNet for a different set of parameters. In Section 5 we discuss the neurological feasibility of BrbrNet with these new parameters, as well as the relation between BrbrNet and the next generation of ICS models (Smolensky, Goldrick, and Mathis 2010).

2. Syllabification in Imdlawn Tashlhiyt Berber

2.1 The Dell-Elmedlaoui syllabification algorithm

We begin by describing in more detail the basic phenomena of syllabification in ITB as reported by Dell and Elmedlaoui (1985, 1988). ITB is a variety of the Tashylhiyt dialect of Berber spoken in the Imdlawn valley in the Western Higher Atlas (Dell and Elmedlaoui 1985). Tashylhiyt (ISO 939-3: shi), also known as Shilha or Tachelhit, is a dialect of Berber spoken by about 3 million speakers in Morocco and Algeria (Lewis 2009).

In Table 2 below we show the syllabification of several words in ITB (Dell and Elmedlaoui 1985). The input to syllabification is a string of segments from the ITB segmental inventory. The second column shows the sonority profile of the input strings, determined using Table 1. The third column of Table 2 shows the syllabification of the input segment strings: dots indicate syllable breaks, capital letters indicate syllable nuclei, and lower-case letters indicate onsets or codas.

Table 2. Imdlawn Tashlhiyt dialect of Berber

Segment String	Sonority	Output	Gloss

	Profile	Syllabification	
ildi	7627	Il.dI	‘pull’ 3 m. sg.
tldi	1627	tL.dI	‘pull’ 3 f. sg.
trglt	16261	tR.gLt	‘lock’ 2 sg. perfective
trglas	162683	tRg.lAs	‘lock’ 3 f. sg. perfective, with dat. 3 m. sg. object
tsti	1317	tS.tI	‘select’ 3 f. sg.

The syllabification of the input string of segments can be determined by applying the Dell-Elmedlaoui algorithm, which we show here. In the input string segments are unsyllabified. As the algorithm proceeds, segments are assigned the role of nucleus, onset, or coda.

Dell-Elmedlaoui Syllabification Algorithm (1985)

1. Input string of segments. Segments are initially not assigned a syllabification.
2. Let the index k equal 8.
3. Going from left to right, find the first segment with sonority index k from among the unsyllabified segments that are not preceded by a syllabified segment. Assign this segment the role nucleus.
4. If this segment is not the first in the string, assign the segment before it the role onset.
5. Repeat steps 3 and 4 until all unsyllabified segments of index k are preceded by a syllabified segment.
6. If k is not already 1, decrease k by 1 and return to step 3.
7. Assign all remaining unsyllabified segments the role of coda.

The algorithm starts with the highest sonority value, 8, and scans from left to right in the string until it finds a segment of that sonority. Each time a segment of sonority 8 is identified, as long as it is not preceded by a syllabified segment, it is assigned the role of nucleus. The segment before it, if there is one, is then assigned the role of onset. When all segments with sonority 8 have been syllabified, the procedure is repeated with sonority level 7 (excluding segments that have already been syllabified), and then with sonority level 6, and so forth down to sonority level 1. When this is finished, all that remains in the string are isolated unsyllabified segments that are preceded by nuclei. These are syllabified as codas. Below we provide an example of the workings of the algorithm taken from Legendre, Sorace, and Smolenksy (2006).

Table 3. Dell-Elmedlaoui algorithm syllabifying /txznt/ ‘you (sg.) stored’

a.	$t_1\ x_3\ z_4\ n_5\ t_1$	(input)
b.	$t_1\ x_3\ z_4\ n_5\ t_1$	no segments at sonority level 8
c.	$t_1\ x_3\ z_4\ n_5\ t_1$	no segments at sonority level 7
d.	$t_1\ x_3\ z_4\ n_5\ t_1$	no segments at sonority level 6
e.	$t_1\ x_3\ \mathbf{z_4}\ \mathbf{n_5}\ t_1$ .z N	on scan at sonority level 5 (steps 3 and 4) current syllabification
f.	$t_1\ x_3\ .z_4\ n_5\ t_1$	no eligible segments at sonority level 4
g.	$\mathbf{t_1}\ \mathbf{x_3}\ .z_4\ n_5\ t_1$ .t X .z N	on scan at sonority level 3 (steps 3 and 4) current syllabification
h.	$.t_1\ x_3\ .z_4\ n_5\ t_1$	no segments at sonority level 2
i.	$.t_1\ x_3\ .z_4\ n_5\ t_1$	no eligible segments at sonority level 1
j.	$.t_1\ x_3\ .z_4\ n_5\ \mathbf{t_1}.$ .t X .z N t .	parse codas

		output

2.2 *Optimality Theory*

Prince and Smolensky (1993/2004: Ch 2) show how to analyze Dell and Elmedlaoui's description of ITB's syllabification using Optimality Theory (OT). As we describe in the introduction, their initial analysis uses just two constraints: ONSET and HNUC, where ONSET is ranked higher than HNUC. Subsequently Prince and Smolenksy (1993/2004: Ch 8) show how HNUC can be decomposed into a family of related constraints on a fixed scale: HNuc is subsumed by a family of constraints $*C/v_i$ for i running from 1 through 8. The constraint $*C/v_i$ indicates that segments of sonority level i are not to be assigned onset or coda (i.e. *C*=consonant).

(4) $*C/v_i$: A segment of sonority i is not parsed as an onset or coda.

Ranking the constraints in the following order describes ITB:

(5) ONSET >> $*C/v_8$>>$*C/v_7$ >> … >> $*C/v_2$>>$*C/v_1$

The constraints can be seen interacting in the following tableau taken from Legendre, Sorace, and Smolensky. (2006). The input is /txznt/ 'you (sg.) stored' (Dell and Eldemlaoui 1985), which corresponds to sonority profile [13451].

(6) OT: Optimality Theory Grammar for Berber Syllabification

/txznt/		HNUC					
/13451/	ONSET	...	*$*C/v_5$*	$*C/v_4$	$*C/v_3$	$*C/v_2$	$*C/v_1$
a. ➔ .tX.zNt.				*			**
b. .T.xZ.Nt.	*!				*		*
c. .T.xZ.nT.			*!		*		
d. .Tx.zNt.				*	*!		*

Recall that the version of ONSET we use allows initial syllables without onsets. Hence the only candidate form shown that violates ONSET is b. Since there are no segments in the input string of sonority 6, 7, or 8, the corresponding constraints $*C/v_6$, etc. do not play a role. The candidate c. is ruled out since it violates $*C/v_5$ by not having /n/ form a nucleus. Finally, although both candidates a. and d. violate $*C/v_4$ by not having /z/ form a nucleus, candidate d. loses by violating $*C/v_3$ by having /t/ instead of /x/ as a nucleus in the first syllable.

2.3 *Harmonic Grammar*

We briefly introduce Harmonic Grammar (HG), which provides the bridge between Optimality Theory (OT) and connectionist models in the ICS Cognitive Architecture. HG is another constraint-based theory of grammar that is closely related to OT (Legendre, Miyata, and Smolenksy 1990; Pater 2009). As in OT, phonological patterns are governed by the interaction of violable constraints. Unlike in OT, where constraints are ranked and *strict domination* indicates that satisfying a higher-ranked constraint is always preferred to satisfying any number of lower-ranked constraints, in HG each constraint is given a weight and several constraints with lower weights may "gang up" and overrule a constraint with a higher weight.

In HG (Legendre, Sorace, and Smolensky 2006), for each selection of constraints c_i and weights w_i we define the harmony of a potential output x to be

$$H(x) = \sum_i w_i c_i(x).$$

In the above expression, $c_i(x)$ is defined to be -1 when x violates the constraint i once and 0 if it does not violate it. (In general, if constraint i is violated m times, then $c_i(x) = -m$.) If $w_i > w_j$ then violating constraint i incurs a greater penalty than violating constraint j. The output of HG for a given input is the string x that maximizes the harmony $H(x)$. An example of this is the string [0 1] which maximized the harmony $H(x) = -256x_1x_2 + x_1 + 7x_2$ in the introduction.

Any phonological system that can be modeled by OT can also be modeled by HG (Prince and Smolenksy 1993/2004; Prince 2002), given a bound on the number of times any form can violate each constraint. All that is necessary is to assign each constraint a weight in the appropriate manner. It is sufficient that the weight of each constraint be assigned so that the weight of a higher ranked constraint exceeds the sum of the weights of all the constraints ranked lower than it, factoring in multiple violations of a constraint by a form. This typically involves weights that grow exponentially in magnitude with the number of constraints in the system. We show here the HG implementation of syllabification in ITB (Legendre, Sorace, and Smolenksy 2006).

(7) HG: Harmonic Grammar for Berber Syllabification

$$H(x) = w_{ONSET}c_{ONSET}(x) + \sum_{k=1,\ldots,8} p_k c_k(x)$$

a. ONSET: A non-initial syllable must have an onset. Weight= $w_{ONSET} = 2^8$.
b. $*C/v_k$: Segment with sonority k must not be a syllable margin. Weight= $2^k - 1$.

/txznt/		HNUC						
/13451/	ONSET	…	*C/v$_5$	*C/v$_4$	*C/v$_3$	*C/v$_2$	*C/v$_1$	
Weight	2^8		2^5-1	2^4-1	2^3-1	2^2-1	2^1-1	**Harmony**
a. ➔ .tX.zNt.				-1			-2	-17
b. .T.xZ.Nt	-1				-1		-1	-264
c. .T.xZ.nT.			-1		-1			-38
d. .Tx.zNt.				-1	-1		-1	-23

As with the OT analysis in the previous subsection, candidate a. is the winner since it has the highest value for the harmony. The weights have been chosen to yield the same outputs as the OT constraint ranking. In particular, after p_1 is chosen arbitrarily to be 1, p_{k+1} is chosen to be 1 greater than twice p_k. This ensures that the Harmony gained by assigning a more sonorant segment to the role of nucleus is greater than that of assigning any two less sonorant segments that role. The weight w_{ONSET} only needs to be greater than p_8 to guarantee that ONSET is ranked higher than the other constraints (Legendre, Sorace, and Smolensky 2006: 381). The ratio of the largest weight to the smallest weight is 2^8. An analogous HG analysis of a similar system with K sonority levels would require a ratio of 2^K.

3. BrbrNet

BrbrNet is a connectionist implementation of syllabification in ITB that builds on the HG implementation shown above (Legendre, Sorace, and Smolensky 2006). Connectionism is a broad computational paradigm for the simulation of cognitive processes (Rummelhart, McClelland, and the PDP Research Group 1986). Connectionist models consist of networks (sometimes called *artificial neural networks*) that are comprised of *nodes* and *connections*. At any given point in time, each node has an *activation* value associated with it. The connections are links between pairs of nodes, and each connection has a specified *weight* that may be positive or negative. The state of the network is determined by the activation of each of the nodes. The activation of each node changes over time as determined by the activation of all nodes connected to it and the weights of those connections. If node A is connected to node B by an excitatory connection (meaning one with positive weight), then greater activation of A leads to greater activation of B. If node A is connected to node B by an inhibitory node (meaning one with negative weight), then greater activation of A leads to less activation of B.

Within the ICS program, Smolensky and Legendre (2006) demonstrate how, just as OT can be implemented by HG, HG can be implemented in a connectionist network. This latter step is illustrated in detail using the example of BrbrNet. An important precursor to BrbrNet is described in Goldsmith and Larson (1990); see Legendre, Sorace, and Smolensky (2006: 389) for a discussion.

In Figure 2 below we show a schematic picture of BrbrNet. BrbrNet consists of a sequence of nodes, each node connected to the adjacent nodes by an inhibitory connection. Initially the activation of all the nodes is set to 0. The input string is fed into BrbrNet through input connections to each of the nodes. If the ith segment of the input string has sonority k then p_k is the input to node i. The state of the system evolves over time: the activation of each node changes in response to the excitatory signal from the input and the inhibitory signal from adjacent nodes. Eventually the system reaches equilibrium and each node has either activation 0 or activation 1. An activation of 1 for the ith node indicates that the ith segment is parsed as a nucleus. An activation of 0 for the ith node indicates that the ith segment is parsed as an onset or a coda.

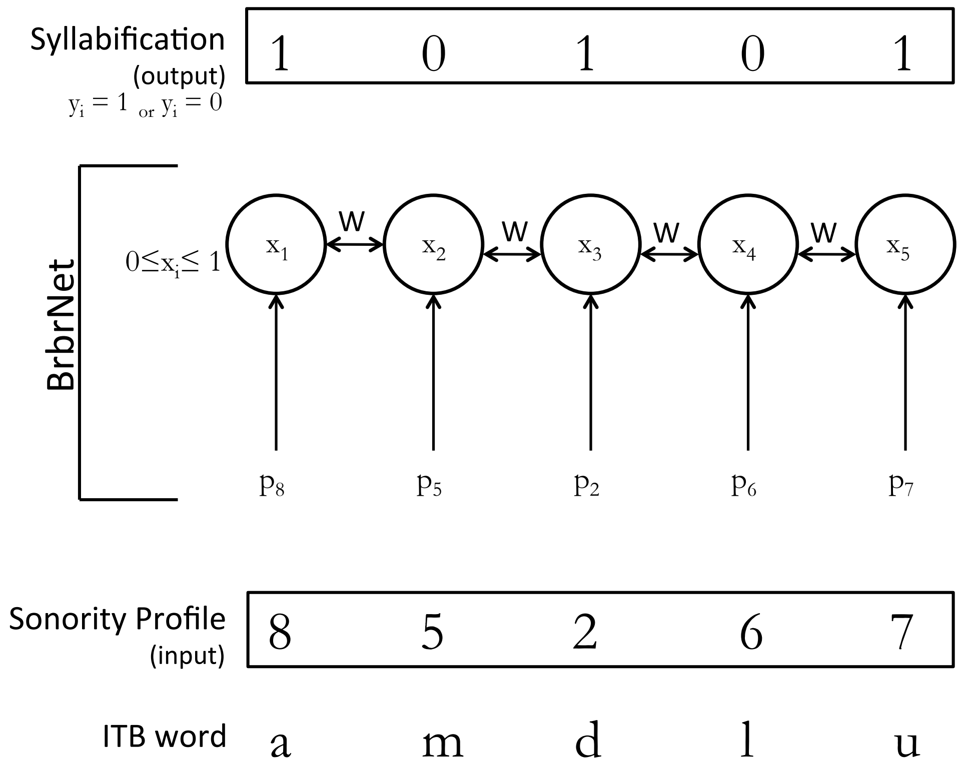

Figure 2. Schematic diagram of BrbrNet

As an example, in Figure 2 we show the syllabification of the word /amdlu/ 'cloud' (provided by Abdelkrim Jebbour, personal communication). The word has sonority profile /85267/. The string of inputs to the nodes is then [$p_8\ p_5\ p_2\ p_6\ p_7$]. If the network correctly syllabifies this input, the equilibrium values of the nodes at equilibrium will be [1 0 1 0 1], which implies the syllabification [A.mD.lU].

The details of how BrbrNet works are as follows. We denote the activation of the ith node at time t by $x_i(t)$. We describe the dynamics of the node activations $x_1,\ldots,x_N$, where N is the number of segments in the input string. At time $t=0$ we have $x_i(0)=0$ for all i, which indicates that none of the segments are syllabified yet. Suppose segment i has sonority k. x_i is pushed upwards by the input p_k and pushed downwards by inhibitory connections with the other nodes. Furthermore, each node's activation is constrained to be between 0 and 1. We can write the update rules for the nodes as

$$x^* = x_i(t) + \tau(p_k - wx_{i-1}(t) - wx_{i+1}(t))$$

$$x_i(t+\tau) = \begin{cases} x^*, & \text{if } 0 \le x^* \le 1, \\ 1, & \text{if } x^* > 1, \\ 0, & \text{if } x^* < 0. \end{cases}$$

Here k is the sonority of the ith input segment and x^* is an intermediate value of x_i used in the calculation. The first equation updates the activation of x_i by adding a contribution due to the input and subtracting a contribution due to competition with other nodes. The second equation shows how the activation is adjusted so that it remains within [0,1]. We have adopted the convention that $x_0 = 0$ and $x_{N+1} = 0$ for all time, where N is the length of the input string, which allows us to treat all nodes in a uniform manner. The true dynamics of BrbrNet are obtained by taking the limit as τ goes to 0. BrbrNet is an example of *Brain-State-in-a-Box* (BSB) dynamics (Anderson et al. 1977; Golden 1986, Cohen and Grossberg 1983). The "box" in its name refers to the constraint that, for all i, x_i must remain between 0 and 1 for all time. The space of possible states for the system corresponds to a box in N dimensional space.

We have now laid out the architecture of the BrbNet. What remains to be set are the values of the parameters w and $p_1,\ldots, p_8$. w is the strength with which adjacent nodes are prevented from both taking activation 1, and so it corresponds to the constraint ONSET. The parameter p_k is the force with which the activation of a node with sonority k is pushed up, and thus corresponds to the constraint *C/v$_k$. The parameter values used in Legendre, Sorace, and Smolensky (2006) are $w = 2^8$, $p_k = 2^k - 1$ for $k = 1,\ldots,8$, which corresponds to the weights of the constraints which we stated for the HG grammar of ITB syllabification. A consequence of this is that if we define the Harmony of the system to be

$$H(x) = \sum_{i=1}^{N-1} w x_i x_{i+1} + \sum_{i=1}^{N} p_i x_i$$

then for any syllabification of the input string corresponding to an x consisting of 0s and 1s, the Harmony value is identical to the value given by the Harmony of the HG grammar given in Subsection 2.3.

An important feature of BrbrNet is that, starting from any initial condition, BrbrNet is guaranteed to increase the harmony. Since the dynamics of BrbrNet are confined to a box, the system will therefore converge to a local maximum of harmony. Since the harmony is a convex function of the activations, any local minima are located at the corners of the box. The final values of $[x_1, x_2 \ldots, x_N]$ will always be a string of 0s and 1s and so can be interpreted as a candidate syllabification of the input. In other words, it is an assignment of the role of nucleus or margin to each segment. However, the *correct* syllabification of any ITB word depends on converging to a *global* maximum of harmony. It is not at all true that Harmony-maximizing systems always converge to the global maximum of the harmony. In fact, starting a system sufficiently close to any local maximum will result in the system converging to that local maximum. In order to demonstrate that BrbrNet converges correctly for all inputs, it is necessary to show that it always converges to the global maximum assuming an initial state of all zero activation.

Legendre, Sorace, Smolensky (2006) conjectured that, with these parameter values, BrbrNet would syllabify all strings of segments correctly when starting at zero activation. They performed computer simulations demonstrating that this was so for all strings of length 4 or less. However, subsequently Christo Kirov (personal communication) discovered a string of length 5 that BrbrNet did not syllabify correctly: [34787], corresponding to the input string /fzway/. (/fzway/ is not an attested word of ITB to the best of our knowledge.) More generally, some longer strings containing this string as a substring are also not syllabified correctly. In Figure 3 we show the activation values versus time for this input. The numbers in the figure indicate the sonority of the segment whose activation is plotted. The two nodes corresponding to segments with sonority 7 have nearly indistinguishable activations over time so we have plotted only one of them.

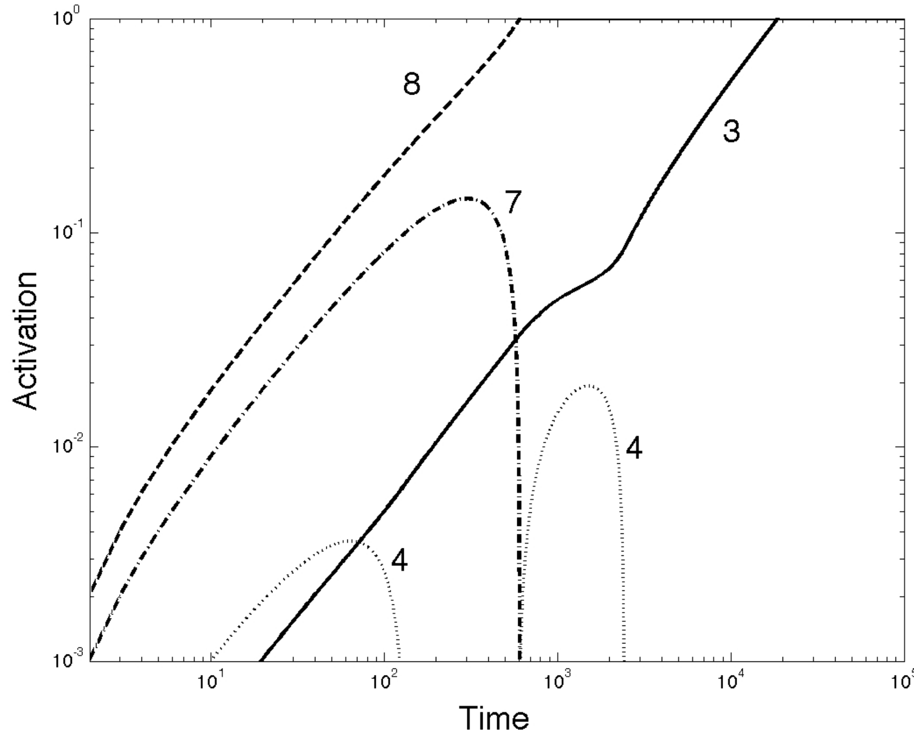

Figure 3 Dynamics of syllabification of input string /fzway/ with sonority profile [34787] by BrbrNet with the original parameters.

The correct syllabification of the string is [0 1 0 1 0]. However, as can be seen in the Figure 3, the syllabification that results from the original parameter values is [1 0 0 1 0], which violates HNUC. As desired, since the segment 8 has the highest sonority, its activation remains highest for all time, and it converges to 1. Once the activation of the 8 segment reaches 1, the activation of the two 7 segments rapidly converges to zero. However, there is an intermediate time at which the activations of the 7 nodes are still relatively large. Normally the 4 node would always have greater activation than the 3 node, and so the 4 node would form the nucleus. But because the activation of the 7 node is so high during intermediate times, the 4 node is inhibited for long enough that the 3 node is able to grow large. The result is that by the time the activation of the 7 node goes to zero, the activation of the 3 node is so far ahead of the activation of the 4 node that the inhibitory connections (BrbrNet equivalent of ONSET) forces the 4 node's activation to 0, overcoming the effect of the input p_4. The 3 node's activation then goes to 1, giving the erroneous activation pattern.

We propose an alternative set of parameters for BrbrNet and present a proof that BrbrNet computes the correct syllabification for all input strings with these parameters. The values are shown in Table 4 where they can be compared to the original values of Legendre, Sorace, and Smolensky (2006). Figure 4 shows how the activations of the nodes change over time for these values for the same input string as we considered above. As before, the activation of the 8 node converges to 1, and the activation of each 7 node converges to 0. Again there is a period of time in which the 3 node has greater activation than the 4 nodes. However, in contrast to the previous case the 3 node only reaches an activation of 10^{-5} by the time the first 7 node is forced to 0. Though greater than the 4 node's activation, the activation of the 3 node is so small that the 4 node is able to surpass it and form the nucleus.

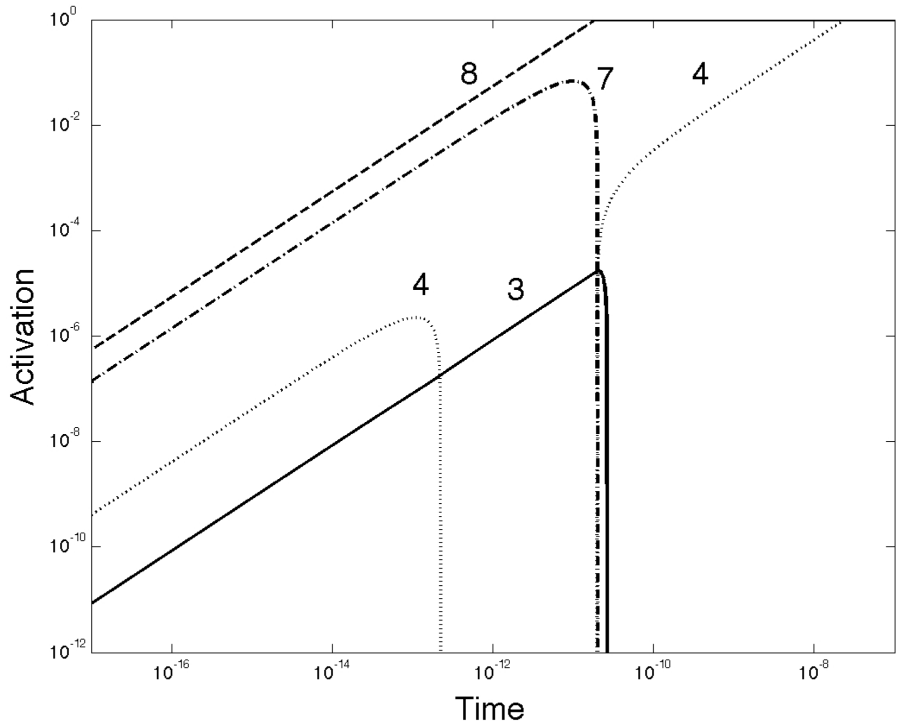

Figure 4. Dynamics of syllabification of input string [34787] by BrbrNet (new parameters.)

Table 4. Parameter values for BrbrNet

Parameter	Legendre et al.'s values	Our values
w	256	27,329,871,550
p_8	255	54,659,743,099

p_7	127	13,664,935,775
p_6	63	3,416,233,944
p_5	31	465,701,011
p_4	15	41,104,127
p_3	7	850,506
p_2	3	2,953
p_1	1	1

As can be seen from Table 4, our parameters take a much larger range of values than the original parameters for BrbrNet. One way to get a feeling for what these parameter values mean dynamically is to consider what happens if a word is inputted to the system with a single segment of sonority k. Since there is no inhibition between adjacent nodes, the activation x of the node grows with rate p_k. This gives that $x(t) = p_k t$, and $x = 1$ at time $1/p_k$ and later. Given our values of the p_k, segments of sonority 8 will syllabify on the order of 55 billion times faster than segments of sonority 1.

We discuss the feasibility of such time scales in Section 5, but in the meantime comment that some sort of discrepancy in time scales is necessary for BrbrNet to correctly syllabify all inputs, though probably not as great as a discrepancy as we have used. As we detail in the discussion of our main result below, for the string /34787/ it is necessary that segment /3/ not grow too much before /787/ have completely syllabified. In general, segments with a given sonority must move very little during the period of time in which all the higher sonority segments syllabify. For this reason we argue that the operation BrbrNet is quite similar to the original Dell-Elmedlaoui algorithm, in that segments of higher sonority are syllabified long before the activations of lower sonority segments move very far from 0.

4. Soundness of BrbrNet for New Parameters

The theorem below gives conditions on the parameters w and $p_1,\ldots,p_8$ so that BrbrNet outputs the correct syllabification when initialized with all activations 0. The result is more general in that applies to any syllabification system similar to ITB's that has K levels of sonority. As in Legendre, Sorace, and Smolensky (2006), we only consider BrbrNet applied to input strings with no *sonority plateaus*, that is, no two adjacent segments with the same sonority. We define the correct output to be the one generated by the Dell-Elmedlaoui algorithm, or equivalently by the OT or HG analyses presented earlier.

Main Theorem. *Let $s_1,\ldots,s_N$ be a string of sonority values between 1 and K. Let the dynamics of the corresponding node activations $x_1,\ldots,x_N$ be given by*

$$\frac{dx_i}{dt} = p_{s_i} - w(x_{i-1} + x_{i+1}),$$

subject to the constraints $0 \le x_i \le 1$, for $i = 1,\ldots,N$, where $x_0 = x_{N+1} = 0$.

Let $x_i(0) = 0$ for all i.

Let $w > 0$ be arbitrary. Let $p_1,\ldots,p_K$ satisfy $p_K \le 2w$,

$p_k \leq p_{k+1}/4$ *for* $k = 1,2,\ldots K-1,$

and

$$p_{k-1} \leq \frac{p_k}{2wt_{k+2}+1} \text{ for } k = 2,\ldots K-2,$$

where

$$t_K = \frac{4}{3p_K} + \frac{5}{2w},\ t_{K-1} = \frac{4}{3p_{K-1}} + \frac{5}{2w}$$

and

$$t_k = t_{k+2} + \frac{2}{p_k} + \frac{5}{2w}, \text{ for } k = 1,\ldots,K-2.$$

Then BrbrNet converges to the correct output for all input strings without a sonority plateau.

The auxiliary parameters t_k have an important interpretation. We say that a segment i is correctly syllabified by time t if, at time t and all times later, x_i takes the correct value of either 0 or 1. In the proof of the Main Theorem below we will see that t_k is the time by which all segments of sonority k or higher and all their neighbors are correctly syllabified.

Taking $K = 8$, the parameters we give above in Table 4 satisfy the conditions of this theorem, thus demonstrating that BrbrNet with these parameters computes the correct syllabification of all input strings in ITB without sonority plateaus.

The proof of the Main Theorem follows four steps which we detail in the four subsections below. In Step 1 we show that any substring of length three of the input string, /qrs /, where $r > q,s$ syllabifies correctly to [0 1 0] within a certain amount of time depending on p_r and w. In Step 2 we show that any substring /qr / with $r > q$ whose neighbor to the right is a segment with higher sonority that has already correctly syllabified to 0 will syllabify to [0 1] within a certain amount of time also depending on p_r and w. The same result holds for substrings /qr/ with $q > r$ whose neighbor to the left is a segment with higher sonority that has already been correctly syllabified to 0. In Step 3 we show that certain isolated segments, which we call *trough segments*, not handled in steps 1 and 2 are also syllabified correctly. In Step 4 we put these results together with the bounds on how fast the syllabification of different segments occur to show that the entire string syllabifies correctly, thus proving the Main Theorem.

4.1 *Step 1: Correct syllabification of peak triples*.

We define a *peak triple* to be a substring of segments $i-1,\ i,\ i+1$, with sonorities /qrs / such that $r > q,s$. We show that the output of BrbrNet for such substrings is [0 1 0], yielding the correct syllabification, independent of what the neighboring segments may be.

Our argument uses the concept of a *trapping region* (Strogatz 1994). A trapping region is a portion of state space such that if the system enters there, then the system will inevitably converge to the desired final state, in this case [0 1 0]. Below we will show that the set of all x_{i-1}, x_i, x_{i+1} such that

$$x_{i-1}, x_{i+1} \le \frac{p_r}{8w},\; x_i \ge \frac{p_r}{4w}$$

is a trapping region, and that this region can be used to prove convergence for peak triples.

The principal result for peak triples is Theorem 4 below. Its proof requires three lemmas: Lemma 1 uses the Taylor expansion of the activations about $t=0$ to obtain upper bounds on x_{i-1} and x_{i+1} and lower bounds on x_i. In Lemma 2 we use Lemma 1 to show that $x_{i-1},\ x_i,\ x_{i+1}$ enters the trapping region within time $1/(2w)$. In Lemma 3 we show that once $x_{i-1},\ x_i,\ x_{i+1}$ enters the trapping region, it will soon converge to the correct syllabification of the substring.

Lemma 1. *Let the dynamics of the x_i be as given in the Main Theorem. Then for all $\tau_0, t \ge 0$,*

$$x_{i-1}(\tau_0+t) \le x_{i-1}(\tau_0) + tp_q$$

$$x_i(\tau_0+t) \ge t[-wx_{i-1}(\tau_0) - wx_{i+1}(\tau_0) + p_r] - \frac{1}{2}t^2 w(p_q+p_s)$$

$$x_{i+1}(\tau_0+t) \le x_{i+1}(\tau_0) + tp_s.$$

Proof: From the equation for x_i we have

$$\frac{dx_{i-1}}{dt} = -wx_{i-2} - wx_i + p_q,$$
$$\frac{dx_i}{dt} = -wx_{i-1} - wx_{i+1} + p_r,$$
$$\frac{dx_{i+1}}{dt} = -wx_i - wx_{i+2} + p_s.$$

By Taylor's theorem, for some $s \in [0,t]$ we have

$$x_{i-1}(\tau_0+t) = x_{i-1}(\tau_0) + t\frac{dx_{i-1}}{dt}(\tau_0+s) \le x_{i-1}(\tau_0) + tp_q.$$

Likewise

$$x_{i+1}(\tau_0+t) \le x_{i+1}(\tau_0) + tp_s.$$

Now

$$\frac{d^2x_i}{dt^2} = -w\frac{dx_{i-1}}{dt} - w\frac{dx_{i+1}}{dt} = w^2[x_{i-2} + 2x_i + x_{i+2}] - wp_q - wp_s \ge -w(p_q+p_s),$$

and so, using Taylor's Theorem again, for some $s \in [0,t]$

$$x_i(\tau_0+t) = x_i(\tau_0) + t\frac{dx_i}{dt}(\tau_0) + \frac{1}{2}t^2\frac{d^2x_i}{dt^2}(\tau_0+s)$$
$$\ge t[-wx_{i-1}(\tau_0) - wx_{i+1}(\tau_0) + p_r] - \frac{1}{2}t^2 w(p_q+p_s),$$

as required, where we have used the fact that $x_i(\tau_0) \geq 0$.

Lemma 2. *Let $p_r \geq 4\max(p_q, p_s)$. Then x_{i-1}, x_i, x_{i+1} is in the trapping region at time $t = 1/(2w)$.*

Proof: The result follows by using Lemma 1 with $t = 1/(2w)$ and $\tau_0 = 0$.

Lemma 3. *Let $2w \geq p_r$. Let $p_r \geq 4\max(p_q, p_s)$. If (x_{i-1}, x_i, x_{i+1}) is in the trapping region at time τ_1 then*

$$x_{i-1}(t) = x_{i+1}(t) = 0,$$

$$x_i(t) = 1,$$

for all $t \geq \tau_1 + \dfrac{4}{3p_r} + \dfrac{2}{w}$.

Proof: For all x_{i-1}, x_i, x_{i+1}, in the trapping region we have

$$\frac{dx_i}{dt} = -wx_{i-1} - wx_{i+1} + p_r \leq -w\frac{p_r}{8w} - w\frac{p_r}{8w} + p_r$$
$$= -\frac{1}{4}p_r + p_r = \frac{3}{4}p_r > 0,$$

and

$$\frac{dx_{i-1}}{dt} = -wx_{i-2} - wx_i + p_q$$
$$\leq -w\frac{p_r}{4w} + p_q \leq -\frac{p_r}{4} + \frac{p_r}{4} = 0,$$

and likewise

$$\frac{dx_{i+1}}{dt} \leq 0.$$

So when the system enters the trapping region it stays in the trapping region. The bound on the derivative of x_i above shows that it takes at most time $4/(3p_r)$ to reach 1 once it is in the trapping region. Once x_i has reached 1 we have

$$\frac{dx_{i-1}}{dt} \leq p_q - w \leq \frac{1}{2}w - w = -\frac{w}{2},$$

so x_{i-1} goes to zero within time $2/w$, and the same goes for x_{i+1}.

Theorem 4. *Let segments $i-1,\ i,\ i+1$ have sonorities /qrs/ with $r > q,s$. Suppose $2w \geq p_r$ $p_r \geq 4\max(p_q, p_s)$. Then for all $t \geq \dfrac{4}{3p_r} + \dfrac{5}{2w}$ we have $x_{i-1}(t) = x_{i+1}(t) = 0$ and $x_i(t) = 1$.*

Proof: This follows from Lemmas 2 and 3.

4.2 *Step 2: Correct syllabification of shoulder pairs*.

We define a *shoulder pair* to be a pair of adjacent segments that are (1) adjacent to a segment that is already correctly syllabified to 0, and (2) form a monotonically rising or falling sonority sequence with the already syllabified segment. The two possibilities for a shoulder pair $/qr/$ are either that it is part of a larger substring $/qrs/$ where $q < r < s$ and the segment with sonority s is already syllabified to 0, or that it is part of a larger substring $/pqr/$ where $p < q < r$ and the segment with sonority p is already syllabified to 0. We consider the first case since the second case follows by symmetry. We assume the shoulder pair segments have indices $i-1,\ i$, so that we need to show that eventually $x_{i-1}(t) = 0$ and $x_i(t) = 1$.

The argument here is very similar to the argument in Step 1, but with an added wrinkle: by the time the segment to the right of the shoulder pair has syllabified, x_{i-1} and x_i may no longer be 0. This is what caused a problem for the string /34787/ in the original version of BrbrNet. In order for an argument like that in Step 1 to work for a shoulder pair we need to be able to show that x_{i-1} has not become too large by the time the segment to the right has syllabified to zero. This puts a constraint on the p_q that is satisfied by the requirement $p_{k-1} \le \dfrac{p_k}{2wt_{k+2}+1}$ in the statement of the Main Theorem.

We define the trapping region for a shoulder pair to be all $x_{i-1},\ x_i$ such that

$$x_{i-1} \le \frac{p_r}{2w},\ x_i \ge \frac{p_r}{4w}.$$

The principal result for shoulder pairs is Theorem 7 below. Its proof uses Lemmas 5 and 6 which are analogous to Lemmas 2 and 3 above. Lemma 5 shows that when the shoulder pair enters the trapping region it will correctly syllabify within a given time. Lemma 6 shows that the shoulder pair will eventually enter the trapping region if the adjacent segment to the right has syllabified quickly enough.

Lemma 5. *Let* $2w \ge p_r$ *and* $p_r \ge 4p_q$. *If* x_{i-1}, x_i *is in the trapping region at time* τ_1 *then* $y_{i-1}(t) = 0$, $y_i(t) = 0$, *for all* $t \ge \tau_1 + \tau$ *where* $\tau = \dfrac{2}{p_r} + \dfrac{2}{w}$.

Proof: The proof is very similar to that of Lemma 3. For all $x_{i-1},\ x_i$, in the trapping region we have

$$\frac{dx_i}{dt} = -wx_{i-1} + p_r \le -w\frac{p_r}{2w} + p_r = \frac{1}{2}p_r > 0,$$

and

$$\frac{dx_{i-1}}{dt} = -wx_{i-2} - wx_i + p_q \le -w\frac{p_r}{4w} + p_q \le -\frac{p_r}{4} + \frac{p_r}{4} = 0.$$

So when the system enters the trapping region it stays in the trapping region. The bound on the derivative of x_i above shows that it takes at most time $2/p_r$ to reach 1 once it is in the trapping region. Once x_i has reached 1 we have

$$\frac{dx_{i-1}}{dt} \leq p_q - w \leq \frac{1}{2}w - w = -\frac{w}{2},$$

so x_{i-1} goes to zero within time $2/w$, and the same goes for x_{i+1}.

Lemma 6. *Suppose that* $p_k \geq p_{k-1}$ *for all* k*. Suppose* $x_{i-1}(\tau_0) \leq \frac{p_r - p_{r-1}}{2w}$*. Then* x_{i-1}, x_i is in the trapping region at time $t = \tau_0 + \tau$ where $\tau = \frac{1}{2w}$.

Proof: The proof is very similar to that of Lemma 2. Using Lemma 1 we have

$$x_{i-1}(\tau_0 + \tau) \leq x_{i-1}(\tau_0) + \tau p_q \leq \frac{p_r - p_{r-1}}{2w} + \frac{p_q}{2w}$$

$$\leq \frac{p_r}{2w} - \frac{p_{r-1}}{2w} + \frac{p_{r-1}}{2w} = \frac{p_r}{2w}.$$

On the other hand

$$x_i(\tau_0 + \tau) \geq \tau[-wx_{i-1}(\tau_0) + p_r] - \frac{1}{2}\tau^2 w p_q \geq \tau\left[-\frac{p_r}{2} + \frac{p_{r-1}}{2} + p_r\right] - \frac{1}{2}\tau^2 w p_{r-1}$$

$$= \frac{1}{2}\tau p_r + \left[\frac{1}{2}\tau - \frac{1}{2}\tau^2 w\right]p_{r-1} \geq \frac{p_r}{4w} + \left[\frac{1}{4w} - \frac{1}{8w}\right]p_{r-1} \geq \frac{p_r}{4w},$$

as required.

Theorem 7. (a) *Let segment* i *have sonority* r*, and let segment* $i-1$ *have sonority* q*, where* $r > q$*. Suppose* $x_{i+1}(s) = 0$ *for all* $s \geq \tau_0$*. Suppose (1)* $2w \geq p_r$ *(2)* $p_r \geq 4p_q$ *(3)* $\tau_0 \leq \frac{p_r - p_{r-1}}{2wp_{r-1}}$ *(4)* $p_k \geq p_{k-1}$ *for all* k*. Then for all* $t \geq \tau_0 + \frac{2}{p_r} + \frac{5}{2w}$ *we have* $x_{i-1}(t) = 0$ *and* $x_i(t) = 1$*.*

(b) *Let segment* i *have sonority* q*, and let segment* $i+1$ *have sonority* r*, where* $q > r$*. Suppose* $x_{i-1}(s) = 0$ *for all* $s \geq \tau_0$*. Suppose (1)* $2w \geq p_r$ *(2)* $p_r \geq 4p_q$ *(3)* $\tau_0 \leq \frac{p_r - p_{r-1}}{2wp_{r-1}}$ *(4)* $p_k \geq p_{k-1}$ *for all* k*. Then for all* $t \geq \tau_0 + \frac{2}{p_r} + \frac{5}{2w}$ *we have* $x_{i+1}(t) = 0$ *and* $x_i(t) = 1$*.*

Proof: Part (a) follows from Lemmas 5 and 6. Part (b) follows by symmetry.

4.3 *Step 3: Correct syllabification of a trough segment*.

We define a *trough segment* to be an unsyllabified segment which is of lower sonority than both its neighbors and such that both its neighbors have already syllabified in Step 1 or Step 2. Its neighbors must have both been syllabified to 0, or else the trough segment would already have been syllabified.

Hence the trough segment node eventually receives no inhibition from its neighbors and its activation is just pushed up to 1, as required.

Lemma 8. *Let segments $i-1$, i, $i+1$ have sonorities /qrs/, where $r < q,s$. Suppose segments $i-1$ and $i+1$ are correctly syllabified to 0 by time τ. Then segment i will correctly syllabify to 1 by time $\tau + 1/p_r$.*

Proof: When x_{i-1} and x_{i+1} are 0 we have $dx_i/dt = p_r$, and so the result follows.

4.4 *Step 4: Correct syllabification of the entire string.*

Now we consider the input string as a whole. Using induction we show that the whole string syllabifies correctly. Step 1 shows that any peak triples syllabify correctly. Step 2 shows that any shoulder pairs syllabify correctly whether they are adjacent to a peak triple or another shoulder pair. These two steps show the correct syllabification of the majority of the input string; all that remains is isolated trough segments that are syllabified by Step 3.

Lemma 9. *Let t_k for $k = 1,\ldots,K$ be defined as in the Main Theorem. Then for $k = 1,\ldots,K$, all segments of sonority k or greater and all their neighbors are correctly syllabified by time t_k.*

Proof. We use induction to show that the claim is true for all k, starting with the base case $k = K$. Since there are no sonority plateaus, any segment with sonority K forms a peak triple with its two neighbors. Therefore, by Theorem 4, all segments with sonority K and their neighbors are correctly syllabified by time $t_K = \dfrac{4}{3p_K} + \dfrac{5}{2w}$.

Now suppose that the claim is true for a given $k+1$. We will show that it follows for k.

Consider a given segment of sonority k. If it has already been syllabified by time t_{k+1} then we are done, since $t_k > t_{k+1}$. So we assume the segment has not been syllabified by time t_{k+1}. There are three cases: (1) The segment has two unsyllabified neighbors in which case the segment and its neighbors form a peak triple. (2) The segment has one unsyllabified neighbors in which case the segment forms a shoulder pair with its other neighbor. (3) The segment has no unsyllabified neighbors in which case it's a trough segment.

In case (1), Theorem 4 shows immediately that the segment will syllabified by time $\dfrac{4}{3p_k} + \dfrac{5}{2w}$ which is less than or equal to t_k for all k. This follows from the definition of t_k in the Main Theorem and

$$t_k = t_{k+2} + \frac{2}{p_k} + \frac{5}{2w} \geq \frac{4}{3p_k} + \frac{5}{2w}.$$

In case (2), we need to apply Theorem 7 with $\tau_0 = t_{k+2}$. Confirming condition (3) of Theorem 7 requires

$$t_{k+2} \leq \frac{p_k - p_{k-1}}{2wp_{k-1}},$$

which follows from the conditions of the Main Theorem. Hence Theorem 7 shows that the segment will be syllabified by time

$$t_{k+2} + \frac{2}{p_k} + \frac{5}{2w} = t_k,$$

as required.

In case (3), we apply Lemma 8, which states that once the neighbors of the segment are correctly syllabified to 0, it will take at most an additional time of $1/p_k$ for the segment to syllabify as a 1. This makes for an upper bound of $t_{k+2} + 1/p_k < t_k$ for the time until a trough segment with sonority k to syllabify.

This completes the inductive step, showing that all segments of sonority k or higher, and their neighbors, are syllabified by time t_k.

Proof of the Main Theorem: This is now a straightforward consequence of Lemma 9.

We comment here that Lemma 7 and its proof show that the process of BrbrNet syllabifying an input is very similar to the Dell-Elmedlaoui algorithm. In both processes, segments with the highest sonority and their neighbors are dealt with first before lower sonority segments are given an opportunity to form a nucleus. In BrbrNet, when the activations of higher sonority segments are going to 1, the activations of lower sonority segments are changing, but move only negligibly far from zero. A parallel computational procedure in which all peak triples are syllabified simultaneously is conceivable, but this is not what BrbrNet does.

4. Discussion

As discussed in Section 3, our parameter values for BrbrNet imply that the segment /a/, with sonority index 8, syllabifies 55 billion times faster than /t/ with a sonority index of 1. We discuss here why we believe this is not neurologically plausible.

To get an estimate of how fast a node corresponding to the segment /a/ would have to go from an activation of 0 to an activation of 1, we need to estimate at what rate speakers must syllabify words. Pellegrino, Coupé, and Marsico (2011) estimate rates of syllable production in speech for various languages to vary roughly between 5 and 8 syllables per second. Assuming that speech in ITB occurs within that range, and that few words in ITB contain more than 5 syllables, we make the conservative estimate that a speaker is able to produce a word in about 1 second. If speakers are performing the syllabification task on a word-by-word basis, then they must be able to syllabify a word within one second.

The slowest part of BrbrNet is the syllabification of segments such as /t/ and /k/ with sonority index 1. Assuming this takes one second, then the segment /a/ must syllabify within 2×10^{-11} seconds, which is much smaller than the time scales at which neurological processes occur. For example, Newell (1989) provides estimates of various neurological time scales: organelles in cells operate on the time scale of 0.1ms, neurons on the scale of 1ms, and neural circuits on the scale of 10ms. Newell uses these figures to put constraints on how many serial operations may be performed within one second, but we may use them to constrain the speed of one component of a parallel process. If we take the 1ms time scale of a neuron to be the fastest that a connectionist unit can go from 0 activation to full activation, then this can only happen 1000 times faster than the slowest process in a calculation. Thus our parameters are completely unreasonable.

In contrast, with Legendre, Sorace, and Smolensky's parameters, /a/ only needs to syllabify 255 times faster than /t/. So /a/ only needs to be syllabified within about 4ms, which is on the border of what is possible with Newell's estimates of timescales. If we assume that making the parameter values only moderately more disparate than Legendre, Sorace, and Smolensky's is sufficient and that calculations are performed on the scale that neurons operate, then BrbrNet becomes a plausible mechanism for the implementation of ITB syllabification.

BrbrNet is no longer a neurologically plausible model, though, if we consider phonological systems with many more constraints. Even with Legendre et al.'s more moderately growing parameters, the range of scale of the parameter weights grows exponentially with the number of constraints. If there were another two levels of sonority added to BrbrNet, the most sonorous nodes would have to syllabify over a 1000 times faster than the least sonorous nodes, which is no longer possible with neurons. The problem becomes worse with more realistic phonological systems encompassing dozens or even hundreds of constraints (Ashley et al. 2010). The architecture of BrbrNet cannot be generalized in a neurologically plausible way to systems containing more than a dozen constraints.

However, BrbrNet is not the current state of the art in the implementation of phonological systems in neural networks. The next generation of ICS model as given in Smolensky, Goldrick, and Mathis (2010), called *Gradient Symbol Processing,* shares only some aspects of architecture with BrbrNet. Gradient Symbol Processing, like BrbrNet, can be viewed as a connectionist implementation of HG or OT. Unlike BrbrNet, in which information about the syllabification of a segment is stored in a single node, distributed representations of discrete variables are used. Distributed representations are desirable and more realistic in connectionist models for various reasons (Smolensky, Goldrick, and Mathis 2010: 10). The architecture of Gradient Symbol Processing has two components: Optimization and Quantization. The Optimization aspect roughly corresponds to the type of Harmony maximizing dynamics we see in BrbrNet. The Quantization aspect is what enforces the output of discrete variables despite the use of distributed representations. A further important difference of Gradient Symbol Processing with BrbrNet is that stochastic forcing, rather than special initial conditions, is used to ensure that the system enters a state of maximum Harmony. In the light of our analysis and discussion of BrbrNet, this next development in the ICS Cognitive Architecture poses two natural challenges (1) Obtain a proof of soundness for Gradient Symbol Processing, (2) Investigate whether Gradient Symbol Processing can be developed for systems with many constraints without encountering the timescale problems we have described here.

Acknowledgements: The authors thank John Alderete and Steve Parker for many helpful comments on earlier drafts. The authors were supported by a Canadian National Science and Engineering Research Council Discovery Grant.

List of Abbrevations

ITB	Imdlawn Tashlhiyt Berber
HG	Harmonic Grammar
OT	Optimality Theory
ICS	Integrated Connectionist/Symbolic

Bibliography

Anderson, James A., Jack W. Silverstein, Stephen A. Ritz and Randall S. Jones. 1977. Distinctive features, categorical perception, and probability learning: Some applications of a neural model. *Psychological Review* 84: 413-451.

Ashley, Karen C., Lynette Disch, David C. Ford, Erin MacSaveny, Steve Parker, Carla Unseth, Angela M. Williams, Rachel Wong, and Brendon Yoder. 2010. How many constraints are there? A preliminary inventory of OT phonological constraints. *Occasional Papers in Applied Linguistics, No. 9.* Dallas: GIAL. http:/www.gial.edu/opal/index.html.

Cohen, Michael A. and Stephen Grossberg. 1983. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks*. IEEE Transactions on Systems on Man and Cybernetics*, 13: 815-826.

Dell, François and Mohamed Elmedlaoui. 1985. Syllabic consonants in Berber: Some New Evidence. *J of African Languages and Linguistics.* Vol 7(2), 105-130.

Dell, François and Mohamed Elmedlaoui, M. 1988. Syllabic consonants and syllabification in Imdlawn Tashlhiyt Berber. *J of African Languages and Linguistics.* Vol 10(1),1-17.

Golden, Richard M. 1986. The brain-state-in-a-box neural model is a gradient descent algorithm. *Journal of Mathematical Psychology*. 30: 73-80.

Goldsmith, John and Gary Larson. 1990. Local modeling and syllabification. In Michael Ziolkowski, Manuela Noske, and Karen Deaton, (eds.), *Papers from the 26th Annual Regional Meeting of the Chicago Linguistic Society,* vol. 2: *Parasession on the syllable in phonetics and phonology,* pp. 129-42. Chicago: Chicago Linguistic Society.

Lahrouchi, Mohamed. 2010. On the Internal Structure of Tashlhiyt Berber Triconsonantal Roots*. Linguistic Inquiry.* Vol 41(2), 255-285.

Legendre, Géraldine, Yoshiro Miyata and Paul Smolensky. 1990. Can connectionism contribute to syntax? Harmonic Grammar, with an application. In Michael Ziolkowski, Manuela Noske and Karen Deaton (eds.), *Proceedings of the 26th regional meeting of the Chicago Linguistic Society* (pp. 237–252). Chicago: Chicago Linguistic Society.

Legendre, Géraldine, Antonella Sorace and Paul Smolensky. 2006. The optimality theory–harmonic grammar connection. In *The Harmonic Mind.* Smolensky and Legendre, editors.

Lewis, M. Paul (ed.), 2009. Ethnologue: Languages of the World, Sixteenth edition. Dallas, Tex.: SIL International. Online version: http://www.ethnologue.com/.

McCarthy, John J. and Alan Prince. 1993. *Prosodic Morphology I: constraint interaction and satisfaction.* University of Massachusetts, Amherst, and Rutgers University, New Brunswick, NJ. RuCCS-TR-3. ROA-482.

Newell, Allen. 1989. Putting it all together. In David Klahr & Kenneth Kotovsky (Eds.), *Complex information processing: the impact of Herbert A. Simon* (pp. 399-440). Hillsdale NJ: Erlbaum.

Pater, Joe. 2009. Weighted Constraints in Generative Linguistics. *Cognitive Science.* Vol 33(6), 999-1035.

Pellegrino, François, Christophe Coupé, and Egidio Marsico. 2011. Across-Language Perspective on Speech Information Rate. *Language* 87: 539-558.

Prince, Alan. 2002. Anything goes*.* In Takeru Honma, Masao Okazaki, Toshiyuki Tabata, & Shin-ichi Tanaka (Eds.), *New century of phonology and phonological theory* (pp. 66-90). Tokyo: Kaitakusha.

Prince, Alan and Paul Smolensky. 1993/2004. *Optimality Theory: Constraint interaction in generative grammar.* Blackwell Publishing.

Rummelhart, David E., James L. McClelland, and the PDP Research Group. 1986. *Parallel distributed processing: Explorations in the Microstructure of Cognition.* Vol 1, *Foundations.* MIT Press.

Smolensky, Paul, Matthew Goldrick and Donald Mathis. 2010. Optimization and Quantization in Gradient Symbol Systems: A Framework for Integrating the Continuous and the Discrete in Cognition. Rutgers Optimality Archive. ROA-1103.

Smolensky, Paul and Géraldine Legendre. 2006. *The Harmonic Mind.* MIT Press.

Strogatz, Steven. 1994. *Nonlinear Dynamics and Chaos.* Cambridge MA: Perseus Books.