Process-specific constraint effects in BMRS

DRAFT (November 15, 2022)

Adam Jardine and Christopher Oakden

Rutgers University   Berkeley Global

1 Introduction

This squib demonstrates how process-specific constraint (henceforth PSC) phenomena can be captured with phonological theories written in *boolean monadic recursive schemes* (BMRS; Bhaskar et al., 2020; Chandlee and Jardine, 2021), a logical formalism for analyzing phonological maps that is grounded in the computational nature of phonological generalizations. As a case study, in the pharyngeal harmony system of Palestinian Arabic (Davis, 1995; McCarthy, 1997), rightward pharyngeal harmony can be blocked by high, front segments, but leftward pharyngeal harmony proceeds unimpeded. Below, as throughout the squib, pharyngeal segments are underlined.

(1)
a. *Leftward harmony*: /xayyaat̲/ → [x̲a̲y̲y̲a̲a̲t̲] ‘tailor’
b. *Rightward harmony*: /t̲uubak/ → [t̲u̲u̲b̲a̲k̲] ‘your blocks’
c. *Blocking of rightward harmony*: /s̲ayyad/ → [s̲a̲yyad] ‘hunter’

Thus, blocking is a PSC specific to rightward spread. Representing pharyngeality with the feature RTR (a convention we follow in this squib), Davis (1995)’s rule-based analysis achieves the PSC effect by tagging the rightward spread rule with the target condition "RTR/HI and RTR/FR" such that segments with these features block rightward spread but not leftward spread. He claims OT does not predict these effects. McCarthy (1997) responds by showing that OT not only captures PSCs as a direct result of constraint ranking, but also provides a more restrictive theory of PCSs. The transitive nature of ranking predicts that if some crucial ranking between markedness constraints produces a blocking effect for one process, the same effect will be observed for any other process compelled by a markedness constraint lower in the hierarchy. This principle is termed the *Subset Criterion*, and no such prediction is made by PSCs tagged on individual rules.

As shown below, analyses in BMRS capture phonological generalizations through ordered hierarchies of *licensing* and *blocking structures* (Chandlee and Jardine, 2021). Following McCarthy (1997)'s generalization about rankings for PSC interactions in OT grammars, the analysis presented here shows that BMRS analyses produce the same effects. It also means that the Subset Criterion is describable by this basic mechanism in BMRS in much the same way as it is describable in OT as a result of ranking transitivity. Thus, to the extent that—as Davis and McCarthy argue—PSC effects are a legitimate aspect of phonological grammars that should be captured by a theory of phonology, this is point in favor of BMRS analyses. BMRS grammars can not only capture these effects, but their restrictive computational properties avoid the typological overgeneration of OT analyses of spreading (see Chandlee and Jardine, 2021).

2 Spreading in BMRS

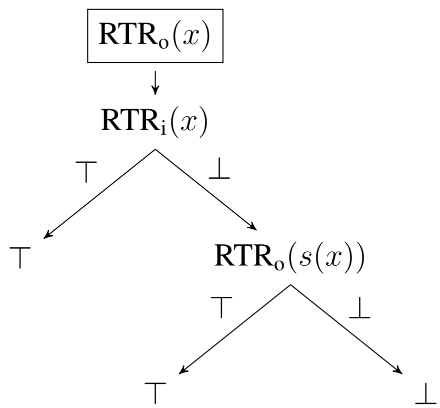
The BMRS formalism describes underlying and surface structures in terms of monadic (=unary), boolean functions that map segments to truth values. For example, $\text{RTR}_\text{i}(x)$ is a function that returns $\top$ (true) when x is an RTR segment in the input, $\perp$ (false) otherwise. The 'i' subscript indicates that it holds for the input. To refer to the local environment of a word, the formalism also adopts segment-valued functions $s(x)$ and $p(x)$ that return the *successor* (i.e., immediately following) and *predecessor* (i.e., immediately preceding) segments. For example, $\text{RTR}_\text{i}(s(x))$ is $\top$ iff the immediately following segment is RTR. We adopt the (innocuous) semantic convention that, for any boolean function B, $B(s(x))$ returns $\perp$ when $s(x)$ is undefined—that is, x has no successor (i.e., it is word-final).

We can then describe a phonological process affecting the RTR feature by creating a definition for an *output* function $\text{RTR}_\text{o}(x)$ that describes the conditions under which x is RTR in the output (i.e., when $\text{RTR}_\text{o}(x)$ evaluates to $\top$) and when it is not ($\text{RTR}_\text{o}(x)$ evaluates to $\perp$). That it is an output property is denoted by the subscript 'o'. For example, the BMRS definition of $\text{RTR}_\text{o}(x)$ below describes leftward spread of an RTR feature.

(2) $$\begin{aligned} \text{RTR}_\text{o}(x) \quad = \quad & \texttt{if } \text{RTR}_\text{i}(x) \texttt{ then } \top \texttt{ else} \\ & \texttt{if } \text{RTR}_\text{o}(s(x)) \texttt{ then } \top \texttt{ else } \perp \end{aligned}$$

The right-hand side of (2) reads, "if x satisfies $\text{RTR}_\text{i}(x)$, then return $\top$. If not, then evaluate $\text{RTR}_\text{o}(s(x))$: if it is satisfied, then return $\top$, else return $\perp$." More intuitively, first the definition checks to see if x is RTR in the input. If so, then the entire statement returns true, terminating evaluation. If this fails, then the definition next checks to see if the sucessor of x is RTR in the *output*. In other words, $\text{RTR}_\text{i}(x)$ and $\text{RTR}_\text{o}(s(x))$ licence a true value for $\text{RTR}_\text{o}(x)$—if either returns true, then $\text{RTR}_\text{o}(x)$ also returns true. If both fail, then the $\perp$ at the end of the definition means that $\text{RTR}_\text{o}(x)$ will return false. The logic of this computation is schematicized in (3).

(3) Flow of computation of (2)



An example evaluation of (2) is given below in (4) for (1a) /balla$\underline{\text{s}}$/ $\rightarrow$ [$\underline{\text{ballas}}$]. In (4), as in the derivations given throughout this paper, the indices of input segments are given explicitly (/b/ is 1, /a/ is 2, etc). The first two rows below the input representation give the values computed for the predicates $\text{RTR}_\text{i}(x)$ and $\text{RTR}_\text{o}(s(x))$ in (2); the third row for the output value for RTR (i.e., $\text{RTR}_\text{o}(x)$) . Predicates that are not evaluated according to the order of evaluation suggested by the BMRS syntax are left blank.

(4)

	b	a	l	l	a	$\underline{\text{s}}$
	1	2	3	4	5	6
RTR$_\text{i}(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$
RTR$_\text{o}(s(x))$	$\top$	$\top$	$\top$	$\top$	$\top$	
RTR$_\text{o}(x)$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
	$\underline{\text{b}}$	$\underline{\text{a}}$	$\underline{\text{l}}$	$\underline{\text{l}}$	$\underline{\text{a}}$	$\underline{\text{s}}$

The truth values for RTR$_\text{o}(x)$ are computed in the following way. Take, for example, element 1 (/b/). Replacing x with 1, we have

$$\begin{aligned}\text{RTR}_\text{o}(1) &= \texttt{if}\ \text{RTR}_\text{i}(1)\ \texttt{then}\ \top\ \texttt{else} \\ &\quad\ \texttt{if}\ \text{RTR}_\text{o}(s(1))\ \texttt{then}\ \top\ \texttt{else}\ \bot\end{aligned}$$

First, we check RTR$_\text{i}(1)$. This is false—/b/ is not RTR—so we move to the `else` condition. We then check the value of RTR$_\text{o}(s(1))$; that is, RTR$_\text{o}(2)$. Recursively, then, the definition of RTR$_\text{o}(x)$ is then evaluated with x replaced with 2; that is:

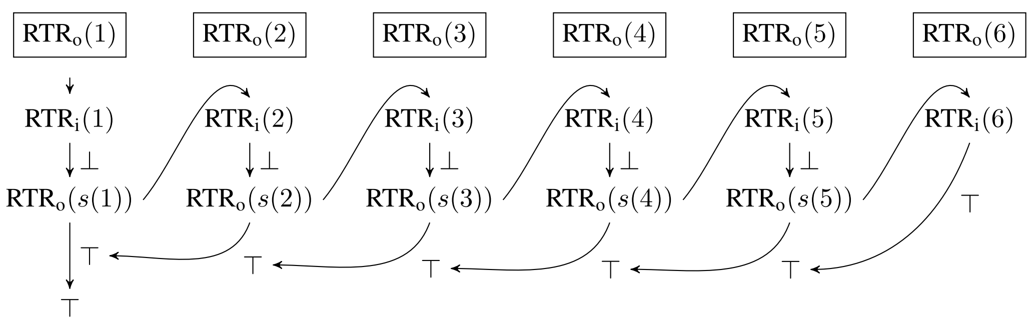
$$\begin{aligned}\text{RTR}_\text{o}(2) &= \texttt{if}\ \text{RTR}_\text{i}(2)\ \texttt{then}\ \top\ \texttt{else} \\ &\quad\ \texttt{if}\ \text{RTR}_\text{o}(s(2))\ \texttt{then}\ \top\ \texttt{else}\ \bot\end{aligned}$$

Again, the first `if` condition—RTR$_\text{i}(2)$—is false for 2, as /a/ is not underlyingly RTR. We again move to the next `if` condition, RTR$_\text{o}(s(2))$. This in turn requires evaluating RTR$_\text{o}(x)$ for the following segment—i.e., 3 /l/.

It is perhaps now obvious that this recursive evaluation of the successor of x will repeat rightward as long as x is not an RTR segment. This eventually will evaluate $\text{RTR}_\text{o}(x)$ for segment 6 /s̱/. Here, as segment 6 /s̱/ is RTR, the first `if` condition, $\text{RTR}_\text{i}(6)$, will evaluate to true. As shown in (4), this means that it is unnecessary to check $\text{RTR}_\text{o}(s(6))$.

Thus, we now have a value of $\top$ for $\text{RTR}_\text{o}(6)$. This allows $\text{RTR}_\text{o}(s(5))$ to return $\top$, which by the definition of $\text{RTR}_\text{o}(x)$ means the entire statement evaluates to $\top$. This returns a value of $\top$ for $\text{RTR}_\text{o}(5)$, which in the exact same way allows $\text{RTR}_\text{o}(4)$ to return $\top$. This continues leftward, returning $\top$ for $\text{RTR}_\text{o}(x)$ for each preceding segment. A flow diagram illustrating how these computations result in an output $\perp$, following the schematic diagram in (3), is given below in (5).

(5) Flow diagram of evaluation of computation of 1 /b/ in (4)



For contrast, (6) gives an evaluation for /falg-a/ ‘piece’ (Davis, 1995, p. 480, (19b)), in which no segments are underlyingly RTR, and so every segment returns $\perp$ for $\text{RTR}_\text{o}(x)$.

(6)

	f	a	l	g	a
	1	2	3	4	5
$\text{RTR}_\text{i}(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$\text{RTR}_\text{o}(s(x))$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$\text{RTR}_\text{o}(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
	f	a	l	g	a

Here, evaluation procedes as in the above evaluation, except for the final segment 5. Here, as 5 is /a/, it returns $\bot$ for RTR(x). Furthermore, by our semantic convention mentioned above, as $s(5)$ is undefined, $\text{RTR}_\text{o}(s(5))$ returns $\bot$. As both licensing structures are not satisfied, $\text{RTR}_\text{o}(5)$ returns $\bot$. This causes the recursive evaluation of $\text{RTR}_\text{o}(x)$ to return $\bot$ for each preceding segment.

In this way, the BMRS formalism captures changes to the input by evaluation of boolean properties of segments, which potentially can be recursively evaluated. Following Chandlee and Jardine (2021) we call a property $\mathbb{P}(x)$ a *licensing structure* if, as with $\text{RTR}_\text{o}(s(x))$, it is in the configuration '`if` $\mathbb{P}(x)$ `then` $\top$', as it causes the statement to be returned true. In contrast, we call a property $\mathbb{P}(x)$ a *blocking structure* if it is in the configuration `if` $\mathbb{P}(x)$ `then` $\bot$, as it causes the statement to be returned false.

The descriptive utility of the full BMRS formalism, then, lies in the interaction of these licensing and blocking structures. It should be noted that technically, in the evaluation of a logical statements, the order of evaluation of expressions is not fixed. However, the BMRS syntax explicitly suggests a route of evaluation: for `if` P `then` Q `else` R, first evaluate P, then evaluate Q or R (but not both) depending on the result. In the below discussion of the calculation of BMRS hierarchies, we assume this route of evaluation.

3 PSC effects in a BMRS grammar

In McCarthy (1997)'s OT analysis, the Arabic data are captured by the following ranking, where RTR-LEFT and RTR-RIGHT are ALIGN constraints that trigger leftward and rightward spread, respectively, of the RTR feature. The markedness constraint RTR/HI&FR (that is, *[high, front, RTR]) captures the blocking condition.

(7) RTR-LEFT $\gg$ RTR/HI&FR $\gg$ RTR-RIGHT $\gg$ IDENT-RTR

Sandwiching RTR/HI&FR between RTR-LEFT and RTR-RIGHT thus causes the markedness constraint to block rightward, but not leftward, spread, as it is overridden by the high-ranked RTR-LEFT constraint. In general, a PSC interaction obtains when some constraint $\mathbb{C}$ ranks between two markedness constraints $\mathbb{M}_i$ and $\mathbb{M}_j$, each of which outrank some faithfulness constraint $\mathbb{F}$. Given $\mathbb{M}_i \gg \mathbb{C} \gg \mathbb{M}_j \gg \mathbb{F}$, the effect of $\mathbb{C}$ is 'specific' to the process triggered by $\mathbb{M}_j \gg \mathbb{F}$ and not to the one triggered by $\mathbb{M}_i$.

The PSC effects observed for Palestinian Arabic also arise in a BMRS definition for the Arabic grammar. First, to capture the blocking condition, the following expression defines what it means to be a high, front vowel (in the input).

(8) $\textsc{Hi\&Fr}(x) \;=\; \texttt{if}\ \mathrm{high_i}(x)\ \texttt{then}\ \mathrm{front_i}(x)\ \texttt{else}\ \bot$

This statement evaluates to true if and only if x is a segment that is both high and front.

We can then add HI&FR as a blocking structure for RTR in the output as follows.

(9)
$$
\begin{aligned}
\mathrm{RTR}_o(x) \; = \; & \texttt{if } \mathrm{RTR}_i(x) \texttt{ then } \top \texttt{ else} \\
& \texttt{if } \mathrm{RTR}_o(s(x)) \texttt{ then } \top \texttt{ else} \\
& \texttt{if } \textsc{Hi\&Fr}(x) \texttt{ then } \bot \texttt{ else} \\
& \texttt{if } \mathrm{RTR}_o(p(x)) \texttt{ then } \top \texttt{ else } \bot
\end{aligned}
$$

As above in 4, rightward spread obtains via initial satisfaction of RTR(x) followed by iterative evaluation of RTR$_o(p(x))$. However, since the blocking structure HI&FR(x) comes before RTR$_o(p(x))$ in the order, spreading can only proceed provided the current input symbol does not return a 'true' value for HI&FR(x). This is shown below with (1c) /$\underline{\text{s}}$ayyad/→[$\underline{\text{sa}}$yyad] 'hunter'.

(10)

	$\underline{\text{s}}$	a	y	y	a	d
	1	2	3	4	5	6
RTR$_i(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
RTR$_o(s(x))$		$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
HI&FR(x)		$\bot$	$\top$	$\top$	$\bot$	$\bot$
RTR$_o(p(x))$		$\top$			$\bot$	$\bot$
RTR$_o(x)$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$
	$\underline{\text{s}}$	$\underline{\text{a}}$	y	y	a	d

The trigger /$\underline{\text{s}}$/ returns 'true' for RTR$_o(x)$ by virtue of evaluating $\top$ for RTR(x). The RTR feature then 'spreads' to segment 2 /a/ as this segment then satisfies RTR$_o(p(x))$.

However, the following segment 3 /y/ satisfies the higher-ranked blocking structure, it returns 'false' for RTR$_o(x)$—regardless of the fact that its predecessor is RTR on the surface and thus would evaluate to true for RTR$_o(p(x))$. As segment 3 thus is false for RTR$_o(x)$, it blocks further rightward spread.

Importantly, RTR$_o(s(x))$ ranks *above* this blocking structure Hi&Fr(x) in (10), this same grammar that blocks rightward spread also permits leftward spread over high, front segments, as shown below with /xayya$\underline{t}$/→[$\underline{\text{xayyat}}$].

(11)

	x	a	y	y	a	$\underline{t}$
	1	2	3	4	5	6
RTR$_i(x)$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\top$
RTR$_o(s(x))$	$\top$	$\top$	$\top$	$\top$	$\top$	
Hi&Fr(x)						
RTR$_o(p(x))$						
RTR$_o(x)$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
	$\underline{x}$	$\underline{a}$	$\underline{y}$	$\underline{y}$	$\underline{a}$	$\underline{t}$

In spite of satisfying Hi&Fr(x), segments 3 and 4 (i.e., the two /y/s) surface with an RTR feature by virtue of satisfying the licensing structure RTR$_o(s(x))$ higher in the order, allowing the span to spread to the beginning of the word. As highlighted by the derivation table in (11), the order of evaluation suggested by the BMRS syntax does not even allow Hi&Fr(x) to be evaluated.

Thus, the order in (10) exhibits a PSC effect: as the blocking constraint $\textsc{Hi\&Fr}(x)$ is sandwiched between the two licensing predicates $\text{RTR}_{\text{o}}(s(x))$ and $\text{RTR}_{\text{o}}(p(x))$, it only affects the latter. This mirrors the constraint ranking posited by McCarthy.

4 BMRS preserves the Subset Criterion

One direct consequence of hierarchical relations in BMRS systems of equations is that the formal motivation for PSC effects, McCarthy's Subset Criterion (which he attributes to personal communcation with Alan Prince) falls out automatically from the evaluation of the grammar, just as in OT. McCarthy presents a general schema, which we give in 12, for constraint interaction of the kind seen in the ranking for Arabic in 7.

(12) $\mathbb{L} \gg \mathbb{M}_i \gg \mathbb{C} \gg \mathbb{M}_j \gg \mathbb{F}$

The Subset Criterion is thus derived from the transitive ranking of constraints: "if $\mathbb{M}_i \gg \mathbb{M}_j \gg \mathbb{F}$, then the set of constraints that can, in principle, impinge on $\mathbb{M}_i$ is a subset of the set of constraints that can, in principle, impinge on $\mathbb{M}_j$" (239). In other words, when higher-ranked $\mathbb{M}_i$ is subject to a PSC induced by $\mathbb{L}$, then lower-ranked $\mathbb{M}_j$ may also be subject to that constraint.

Hierarchies of licensing and blocking structures in the BMRS formalism yield the exact same effect. A generalized example using McCarthy's notation demonstrates this fact. Let P_{o} be some output property that is subject to two processes, one that is licensed by $\mathbb{M}_i$ but subject to a blocking condition $\mathbb{L}$, and one that is licensed by $\mathbb{M}_j$ but subject to a blocking condition $\mathbb{C}$. If $\mathbb{M}_i$ takes precedence over $\mathbb{M}_j$, then the BMRS definition for $\mathrm{P_o}$ would look as below in (13).

(13) $$\begin{aligned}\mathrm{P_o}(x) \;=\; & \texttt{if } \mathbb{L}(x) \texttt{ then } \bot \texttt{ else} \\ & \texttt{if } \mathbb{M}_i(x) \texttt{ then } \top \texttt{ else} \\ & \texttt{if } \mathbb{C}(x) \texttt{ then } \bot \texttt{ else} \\ & \texttt{if } \mathbb{M}_j(x) \texttt{ then } \top \texttt{ else } \mathrm{P_i}(x)\end{aligned}$$

In (13), the PSC $\mathbb{L}$ for $\mathbb{M}_i$ is implemented as a blocking structure $\mathbb{L}(x)$ ordered before $\mathbb{M}_i(x)$; likewise $\mathbb{C}(x)$ and $\mathbb{M}_j(x)$. This reflects precisely the total order in (12), and produces the same effects. This is because is that the PSC limitation $\mathbb{L}(x)$ is calculated before $\mathbb{M}_i(x)$ given the hierarchy, and necessarily before $\mathbb{M}_j(x)$. Thus, the hierarchical relations in BMRS follows the same transitivity of the strict ordering relation over OT constraints, a property not derived by PSC tags on individual rules.

5 BMRS avoids pathological PSC effects

By McCarthy's account, the Subset Criterion—driven by the basic mechanism of constraint interaction—results in a more restrictive theory of PSC than is available to the rule-based formalism. Davis (1995) posits a hypothetical harmony system where rightward spread is subject to one condition and leftward spread is subject to a different, non-overlapping condition. Such a case is predicted to be impossible in OT because it would require a circular ranking. McCarthy illustrates with a toy example using the de-conjoined RTR/HI and RTR/FR as separate conditions on rightward and leftward pharyngeal spread. The required rankings are thus (McCarthy, 1997, p. 240):

(14)	Ranking	Interpretation
a.	RTR/HI $\gg$ RTR-RIGHT	High segments block rightward harmony.
b.	RTR-RIGHT $\gg$ RTR/FR	Front segments don't block rightward harmony.
c.	RTR/FR $\gg$ RTR-LEFT	Front segments block leftward harmony.
d.	RTR-LEFT $\gg$ RTR/HI	High segments don't block leftward harmony.

A total order maintaining these sub-rankings is impossible; RTR/HI cannot rank above RTR-RIGHT *and* below RTR-LEFT when RTR-RIGHT $\gg$ RTR-LEFT via transitivity.

Given the same set of constraints, a BMRS systems of equations make the same predictions about the hypothetical case above and thus align with the restrictions on PSC imposed by the Subset Criterion. The following shows this with McCarthy's example. To implement the individual blocking markedness constraints in (14), we replace the single blocking structure HI&FR(x) from (9) with individual blocking structures HI(x) (true iff x is a high segment) and FR(x) (true iff x is a front segment).

Next, recall that in (9), leftward harmony is motivated by the licensing structure $\text{RTR}_\text{o}(s(x))$ and rightward harmony is motivated by the licensing structure $\text{RTR}_\text{o}(p(x))$. In the Davis-McCarthy pathology described above, high segments block rightward harmony *but not* leftward harmony, and conversely front segments block leftward harmony *but not* rightward harmony. To capture the former, the blocking structure HI(x) must be ordered *after* $\text{RTR}_\text{o}(s(x))$ so it does not block leftward harmony, but *before* $\text{RTR}_\text{o}(p(x))$, so it does block rightward harmony. This is shown schematically in (15a).

(15) a. `if` $\mathrm{RTR_o}(s(x))$ `then` $\top$ `else`

...

`if` $\textsc{Hi}(x)$ `then` $\perp$ `else`

...

`if` $\mathrm{RTR_o}(p(x))$ `then` $\top$ `else`

b. `if` $\mathrm{RTR_o}(p(x))$ `then` $\top$ `else`

...

`if` $\textsc{Fr}(x)$ `then` $\perp$ `else`

...

`if` $\mathrm{RTR_o}(s(x))$ `then` $\top$ `else`

However, to capture that leftward harmony, but not rightward harmony, is blocked by front segments, we would need to order $\mathrm{RTR_o}(p(x))$ before $\textsc{Fr}(x)$, which would in turn be ordered before $\mathrm{RTR_o}(s(x))$, as given above in (15b).

Clearly this requires contradictory orderings of $\mathrm{RTR_o}(p(x))$ and $\mathrm{RTR_o}(s(x))$: (15a) requires $\mathrm{RTR_o}(s(x))$ to be ordered before $\mathrm{RTR_o}(p(x))$, but (15b) requires $\mathrm{RTR_o}(p(x))$ to be ordered before $\mathrm{RTR_o}(s(x))$. (Inserting two instances of $\mathrm{RTR_o}(s(x))$, for example, would not work either—the first would pre-empt the second, rendering it meaningless.)

Thus, given a standard, phonologically motivated set of blocking and licensing structures, it is impossible in BMRS to construct the Davis-McCarthy pathology, due to the ordering of licensing and blocking structures in BMRS preserves the Subset Criterion.

Of course, as a computational formalism, BMRS is fairly expressive (see Bhaskar et al. 2020)—in particular, we have shown here that it can capture bidirectional spreading, which is in the relatively complex Weakly Deterministic class of functions (Heinz and Lai, 2013). Indeed, as a reviewer points out, the pathology pointed out by Davis and McCarthy is indeed describable by BMRS, if one is able to define arbitrary licensing and blocking structures. However, this is also true for OT and arbitrary constraints, as has been pointed out many times before (by, e.g., Eisner 1997). Thus, just as in OT, in order to make the correct predictions, the BMRS formalism needs to be constrained by a theory of phonologically-motivated licensing and blocking constraints. Articulating such a theory falls beyond the purview of this squib, but we can expect that, like OT, such a theory will appeal to the usual principles of general, simple constraints.

As the reviewer further points out, it is an open empirical question whether this prediction holds true in the typology beyond Arabic. *A priori*, we should prefer a theory that makes more restrictive predictions, and test them against observation. While a full typological survey is beyond the purview of this squib, for the reasons presented here, we argue that BMRS is a promising tool to state such a theory.

References

Bhaskar, S., Chandlee, J., Jardine, A., and Oakden, C. (2020). Boolean monadic recursive schemes as a logical characterization of the subsequential functions. In Leporati, A., Martín-Vide, C., Shapira, D., and Zandron, C., editors, *Language and Automata Theory and Applications - LATA 2020*, Lecture Notes in Computer Science, pages 157–169. Springer.

Chandlee, J. and Jardine, A. (2021). Computational universals in linguistic theory: Using recursive programs for phonological analysis. *Language*, 93:485–519.

Davis, S. (1995). Emphasis spread in Arabic and grounded phonology. *Linguistic Inquiry*, 26:465–498.

Eisner, J. (1997). What constraints should OT allow? Talk handout, Linguistic Society of America, Chicago. ROA#204-0797. Available at http://roa.rutgers.edu/.

Heinz, J. and Lai, R. (2013). Vowel harmony and subsequentiality. In *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*, pages 52–63.

McCarthy, J. J. (1997). Process-specific constraints in Optimality Theory. *Linguistic Inquiry*, 28:231–251.