Locality is epiphenomenal: adjacency is opaqueness*

Rim Dabbous[a], *Marjorie Leduc*[b],
Charles Reiss[a] *& David Ta-Chun Shen*[a]
[a]*Concordia University &* [b]*Rutgers University*

We suggest thinking about phonological rule environments in terms of a parameterized algorithm—target segments (INR) *initiate* a SEARCH for segments (TRM) that *terminate* the SEARCH. If independent conditions (CON) on the terminator are met, a CHANGE in the target potentially occurs. We show how this *scoping* works, and also show that adjacency is just a special case of a long distance search. Segment transparency and opaqueness are derived without contrast, markedness considerations or other functionalist notions. This paper develops previous work on the use of search algorithms to model phonological environments (Dabbous et al. 2021; Shen 2016; Samuels 2011; Mailhot and Reiss 2007, 2004).

1. Introduction

Long-distance phonological processes have long been a central focus of phonological research. Approaches have ranged from using formalisms like parenthesis-star to characterize intervening material between target and trigger (Chomsky and Halle 1968) to recasting long-distance processes as local through tier-based representations (e.g., Heinz 2010, 2018; Chandlee and Heinz 2018; Burness et al. 2021). The question of how best to capture long-distance phonological dependencies remains a topic of lively debate.

In this paper, we propose a formalization of long-distance processes through the SEARCH and CHANGE model, a framework that prioritizes non-locality while treating segment adjacency, i.e., locality as a special case of a long-distance relation.

Within the SEARCH and CHANGE framework, phenomena traditionally viewed as distinct, such as segment opaqueness, transparency, and locality, are shown to be closely interrelated. These patterns can be explained as variations arising from minor adjustments to the parameters of two fundamental operations: a SEARCH procedure and a CHANGE procedure.[1] Consequently, phonological phenomena previously considered distinct can be parsimoniously unified, requiring only slight modifications to a constrained set of parameters.

---

[*] Thanks to Lee Bickmore for confirming that our schematic Bantu data is grounded in reality.

[1] This paper focuses primarily on the computational formalization of SEARCH, with limited discussion of the CHANGE function. For more on the set-theoretic basis of CHANGE, readers can refer to Bale and Reiss (2018) or Reiss (2021).

Actes du congrès annuel de l'Association canadienne de linguistique 2024.
Proceedings of the 2024 annual conference of the Canadian Linguistic Association.

1.1 Two Basic Assumptions

The descriptive power of the SEARCH and CHANGE model rests on two fundamental assumptions: the *Non-Locality* assumption and the *Substance-Freeness of Structural Changes* (SFST) assumption. While many frameworks take adjacency to be the primitive in phonological structures and derive long-distance environments via tiers or extra computational machinery (Goldsmith (1979), Clements (1980), Heinz et al. (2011), Burness et al. (2021)), SEARCH and CHANGE flips this assumption by considering that phonological rules are in fact built on unbounded search procedures, with locality emerging as a special case.

The second assumption–SFSC–diminishes the role that notions like assimilation and feature copying have been given in phonological theory by proposing that the content of the structural change of a rule need not be mirrored in the conditioning environment. For example, vowels may be nasalized before nasal consonants, but they might instead be raised; and vowel raising can also occur at the end of a word. See work on Substance Free Logical Phonology, such as Hale and Reiss (2008); Reiss (2017) for discussion.

In what follows, we outline the core components of the model in Section 2, accompanied by a demonstration using a basic toy language. Section 3 introduces new parameter combinations that account for patterns traditionally described as opaqueness and transparency. In Section 4, we apply the model to schematic forms based on two Bantu languages, Lamba and Tshiluba, illustrating how it expresses variations in the scope of nasalization. In Section 5, we extend the combinatorial possibilities attributed to opaqueness to derive locality. Finally in section 6, we revisit Lamba to demonstrate how our system accounts for certain long-distance forms. Conclusions and directions for future research appear in 7.

2. SEARCH and CHANGE Basics

The SEARCH and CHANGE framework formulates phonological rules computationally using five basic parameters as inputs to a SEARCH function and a subsequent CHANGE function. Underlying form to surface form mappings result from the composition of these two functions such that the output of the SEARCH defines the domain of input to the CHANGE function. Under this system, input to phonology is a linearly ordered string onto which the functions apply depending on the specifications of the following five parameters for each function:

(1) Parameterized rules have five components specific to the SEARCH function and the CHANGE function:

- SEARCH parameters:
  - INR: Initiator segment. Input to SEARCH function that initiate a SEARCH (corresponds to targets in traditional rules)
  - DIR: direction of SEARCH
  - TRM: terminator of SEARCH (corresponds to trigger/environment in traditional rules)
- CHANGE parameters:
  - CHANGE: INPUT/OUTPUT mapping (all intrasegmental changes)[2]
  - COND: Additional conditions on the specifications of a TRM

In a given string, a SEARCH is initiated by each INR segment and terminates with the first segment meeting the specifications for TRM in a specified direction DIR(Right/Left). If the SEARCH is successful, and additional conditions on the TRM specified in COND are met, SEARCH outputs the INR segment, which serves as input to the CHANGE, which in turn outputs a new symbol, yielding a string to string mapping.[3] Note that not all parameters are mandatory in a rule. The simplest rules contain just a specification for INR, and an input/output CHANGE specification.

To illustrate the logic of SEARCH and CHANGE, consider a simple toy system where rules map between strings built from an inventory of six "segments" : □, ○, △, ■, ● and ▲. Each segment can be characterized by a set consisting of a color value (black or white) and a shape value (square, triangle, circle).[4] An informal version of a SEARCH and CHANGE rule is given in (2):

(2) Rule A: Turn a ■ into a ○ if there is a ▲ anywhere to its right.[5]

Using parameters, this rule is specified in (3):

(3) Rule A parameters:

SEARCH: INR: ■; TRM: ▲; DIR: R
CHANGE: INPUT: INR; OUTPUT: ○

---

[2] In the Logical Phonology literature intrasegmental changes are implemented by combinations of rules based on set substraction and unfication. Here we just use the traditional '→' of rule-based phonology to express feature-cahnging processes.

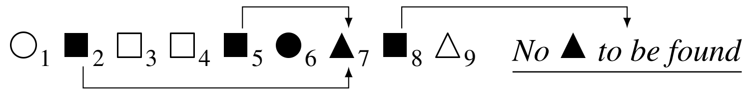
[3] We define input to rules as strings, and rules as string to string mappings, not segment to segment mapping. For more information, refer Bale et al. (2020)

[4] See https://search-and-change.netlify.app/ for a gamified implementation.

[5] Notice how "substance free" our example is: the change to a white circle has no "substantive" connection to the black triangle trigger.

The rule in (3) stipulates that a rightward SEARCH is initiated by each black square and terminates at the first black triangle encountered. If the SEARCH is successful, i.e., a black triangle is found, SEARCH outputs the INR black square which serves as input to the CHANGE function. This, in turn, effects the change specified in the rule, namely turning the INR into a white circle. Notice that Rule A lacks a condition (COND) specification. Any INR that finds a TRM will thus undergo a change. As an example, consider the string in (4).

(4)



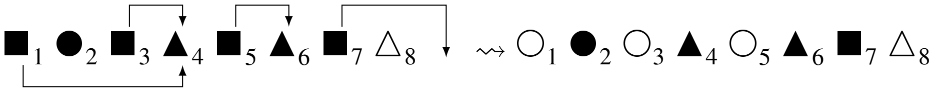
The SEARCH instances initiated by black squares 2 and 5 both find the same TRM, the black triangle in position 7. These SEARCH instances are therefore successful and the INRs each undergo a CHANGE into a white circle. The SEARCH from black square 7, however, fails to find a TRM, and will not proceed to the CHANGE function, so it remains unchanged. Applied to the string in (4), Rule A outputs the string in (5), with changes in positions 2 and 5.

(5) $\bigcirc_1\ \blacksquare_2\ \square_3\ \square_4\ \blacksquare_5\ \bullet_6\ \blacktriangle_7\ \blacksquare_8\ \triangle_9 \rightsquigarrow \bigcirc_1\ \bigcirc_2\ \square_3\ \square_4\ \bigcirc_5\ \bullet_6\ \blacktriangle_7\ \blacksquare_8\ \triangle_9$

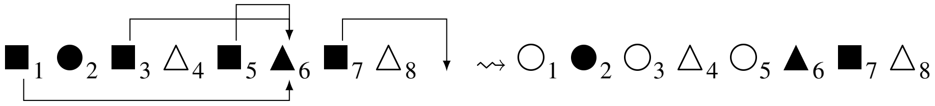
Let's apply Rule A to two minimally distinct strings:

(6) Applying Rule A to minimally distinct strings

a.



b.



In both mappings in (6), every black square that has a black triangle to its right is turned into a white circle: the same shapes are changed in (a) and (b). In both cases, the black squares in position 7 do not change, because there is no black triangle to the right in either (a) or (b). In (b) black squares in positions 1, 3 and 5 all share the TRM black triangle in position 6. Contrast this with (a) where the black squares in positions 1 and 3 find their TRM in position 4. As long as a black square can 'see' a black triangle, the SEARCH is satisfied—locality plays no role. Observe how the white triangle in position 4 in (b) is 'transparent' to the search, as are all the black squares and circles.

It is crucial to note that in our system, this 'regressive' process shows a rightward SEARCH (left-to-right), whereas in most of the literature, especially on harmony processes, regressive processes are conceptualized as involving right-to-left parsing of the string, with feature 'donors' looking for 'recipients'. Considering our assumption of SFSC which supposes that the change is independent from triggering environment, there may not be a donor ‘segment’ for the relevant feature. Following Mailhot and Reiss (2007) and Nevins (2010), our ‘recipient as initiator’ perspective allows each SEARCH instance to be a function, mapping each INR to a single TRM. The standard alternative approach of mapping from ‘donor/trigger’ to all the ‘targets’ in a domain means that trigger-target mapping won’t always be a function, since a given trigger can map to more than one target. In the donor-centric approach, applied to (4), the black triangle in position 7 maps to both the preceding black squares in positions 2 and 5. So, that approach is both conceptually flawed (relying on feature ‘donors’, and thus unable to model ‘substance free rules) and more complex formally (by not guaranteeing unique mapping). To reiterate, the mapping from INRs to their respective TRMs is a function; the mapping from a ‘trigger’ to its ‘targets’ is not necessarily a function. For us the INR-to-TRM mapping re-conceptualizes what is traditionally called a rule environment.

3. Scope of Conditions

Using the parametrized components provided in (1), we can formalize the SFSC assumption by specifying the breakdown of the environment/trigger segment in traditional rules into three distinct components: the Terminator (TRM) that terminates a SEARCH, the COND which specifies additional conditions to be met before a change can take place, and finally, the CHANGE which independently specifies the structural change.

The distinction between TRM and COND, in particular, is fundamental for deriving patterns that have been described in the literature as ‘opaqueness’.[6] Consider the following sets of instructions given to a contractor charged with painting a door on a street lined with single-storied and two-storied houses.

(7) Distinctions between TRM and COND:

i find a **house** and paint its door **red**. (The TRM is any house, with no COND)

ii find a **two-storey house** and paint its door **red**. (the terminator is a two-storey house, with no COND)

iii find a **house** and if it’s a **two-storey**, paint its door **red**. (The TRM is a house, the condition is that it’s two-storey)

In (i), the TRM is minimally specified as **house**. The contractor will therefore paint the door of the first house they comes across **red**. In (ii) and (iii), by contrast, only the doors of two-storey houses will be painted. However, the conditions under which this occurs differ: in (iii), the TRM is more fully specified with respect to number of storeys as “two-storey”. The contractor will bypass any single-storey house and only stop to paint the first two-storey house they find. In (iii), however, the TRM lacks specification about number of storeys, so the contractor will stop at the first house they find, which can either be one-storey or two-storey. They will, however, only paint the door if the house is two-storey.

---

[6] *Opaqueness* refers to potential interference in non-adjacent INR-TRM relations, which is not to be confused with rule ordering *opacity*.

So while (ii) will always lead to a door being painted, (iii) might not, in the event that the house is single-storey.

We propose that the logic of (iii) is appropriate for the analysis of *opaqueness* in phonological rule environments, whereas (ii) corresponds to *transparency*.

3.1 A Partial Description for TRMs

In the instructions above, the contractor's actions vary in accordance with the amount of information given about the house. Case (ii) provides more specification than cases (i) and (iii) about the kind of house at which the contractor should stop and paint, by specifying the number of storeys. By leaving out the height specification in (i) and (iii), these cases illustrate the notion of a *partial description*. Specifying just "house" is more general, less specific, more 'partial', than specifying 'single-storey house' or 'two-storey house'. This is exactly parallel to the manner in which natural class partial descriptions are used in phonology: one rule might target for change any member of the set of stops, whereas another targets a more richly defined class, with *fewer* members, such as the set of voiced stops.

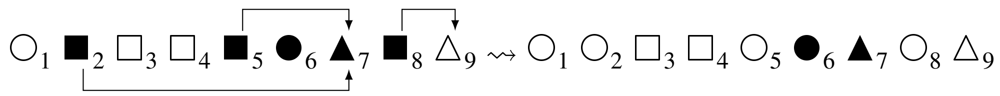
To apply this reasoning to our system, consider rule B in (8) which contains a slight difference from Rule A in (2):

(8) Rule B

- Turn a ■ into a ○ if there is a ▲ anywhere to its right.
- Rule B Parameters

  SEARCH: INR: ■; TRM: ▲ ; DIR: R
  CHANGE: INPUT: INR; OUTPUT: ○

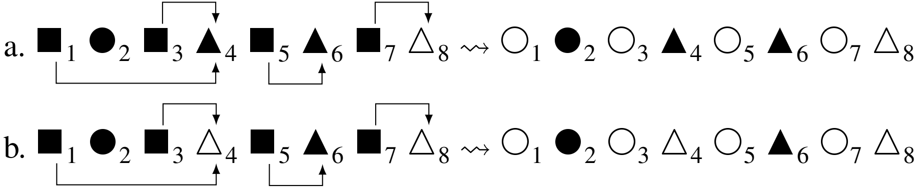
We use the color gray to denote a class of symbols that includes both black and white members. So the gray triangle in (8) is to be understood as referring to the class of all triangles, black and white. (Crucially, a gray triangle is not a basic symbol, but rather a way to denote a set of symbols—this is not a way to refer to a triangle that lacks color, it is not featural underspecification.) In rule B, the specification of the TRM as gray triangle means that either a black or white triangle in a string can terminate SEARCH. This rule provides the mapping in (9):

(9) 

The black squares in positions 2 and 5 'find' the black triangle in position 7 to the right, resulting in those squares becoming white circles. The black square in position 8 finds the white triangle in position 9, so it changes too.

Let's apply Rule B to the same two strings we used in (6):

(10) Mappings from (8)



As shown in (10), every black square that has a triangle to its right (black or white) is turned into a white circle. As in (6), the same shapes are changed in (a) and (b). Here, however, the black squares in position 7 find the white triangles in position 8 and change to circles. Additionally in contrast to (6) where the squares in position 3 found differing Trms due to the specification in rule A that the terminator must be a **black** triangle, the more general specification (the less specific characterization) of Trm in Rule B ensures that any triangle can terminate a Search. These are the parallels to our contractor scenarios thus far: Rule A terminates at a black triangle, which is like (7ii)—find a two-storey house; and Rule B terminates at any triangle, which is like (7i)—find a house.

3.2 Conditions on CHANGE: another parameter

We now introduce another component into our rule syntax that models (7c), instructing the contractor to stop at any house, but only paint the door if the house is a two-storey. To that end, consider the Rule C in (11) that imposes an extra condition on whether the Change applies.

(11) Rule C

- From a ■ Search for a ▲ to the right and turn the ■ into a ○ if Trm is Black (▲)
- Rule C Parameters

  Search: Inr: ■; Trm: ▲; DIR: R
  Change: INPUT: Inr; OUTPUT: ○; $\boxed{\text{Cond: Trm: B}}$

Observe how this rule differs from both Rule A in (2) and Rule B in (8). The specification of Search here is identical to that in Rule B: Search terminates at the first triangle of any color to the right. In Rule C, however, the change applies only if the terminator is Black. This is not the same as Rule A where the Search terminates only at a black triangle.

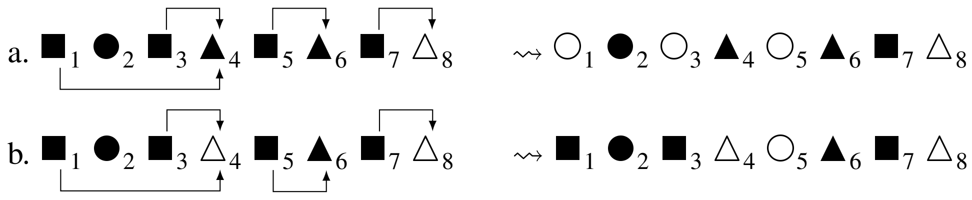
To reiterate, the only difference between Rule A and Rule C is that in the former, the Trm is a black triangle and there is no Cond on Change—the Change happens if Search is satisfied; whereas in the latter, Trm in the rule is *any* triangle (denoted by gray), and there is a condition that Change applies only if the actual Trm in a given instance is black. We have divorced the notions of terminator and trigger.

Since any triangle can terminate the Search initiated by a black square in Rule C, but only a black triangle triggers the Change, it becomes clear that a white triangle behaves as a “blocker”. For an input string ■ △ ▲, the black square will not change because the white triangle ”blocks” access to the black triangle.

3.3 Conditions on CHANGE yield “opaqueness”

Let’s illustrate this blocking effect more thoroughly. We get the following mappings from Rule C:

(12) Mappings from Rule C



For the inputs (12a) and (12b), Rule C makes different changes. In (a), the black squares in positions 1 and 3 are changed to white circles because each of these initiate a SEARCH that is terminated by the triangle in position 4, and that triangle is black. In contrast, in (b), there is no change in positions 1 or 3—they remain as black squares because the relevant instances of SEARCH are terminated by the triangle in position 4, and that triangle is white.

Using terminology familiar to phonologists, we could describe the white triangle in position 4 of (12b) as *opaque*, as it “blocks” the black square from “finding” a black triangle to its right. Conversely, the white triangle in position 4 of (6b), under the mapping of Rule A where TRM is specified as *black* triangle is *transparent*, as it does not terminate the SEARCH initiated by the black square.

With this formalized understanding of opaqueness and transparency, it becomes evident that these phenomena are entirely independent of the “phonetic” substance of the features involved in a rule’s change. In both rules A and C, the change transforms a black square into a white circle. The transparent or opaque behavior of the triangles in (6) and (12) is unrelated to the featural properties of the changes themselves.

We propose that this framework offers a strong model for real phonological patterns: so-called “opaque” segment behavior is simply a reflection of the scope and specificity of conditions on CHANGE as shown in (11). Apparent transparency in (2), on the other hand, emerges from the absence of a COND on CHANGE and from the fact that a white triangle shares certain properties with the black triangle TRM. Transparency, therefore, is not encoded in the grammar but emerges as an epiphenomenon. Similarly, the perceived opaqueness of the white triangle in (11) is also epiphenomenal—it arises from the characterization of the TRM and the additional COND imposed on it.

4. Two versions of nasalization of *l* in (schematic) Bantu

In this section, we apply the SEARCH and CHANGE model to schematic forms based on two Bantu languages, Lamba and Tshiluba, to capture microvariation in the 'spread' of nasality. Odden (1994) compares the unbounded, long-distance nasal assimilation found in Tshiluba (and Kikongo) to the Lamba (and Bemba) type that (basically) applies between adjacent syllable onsets. Consider the following schematic forms representing actual forms from Tshiluba and Lamba:

(13) Nasalization of /l/

Tshiluba pattern	Lamba pattern
a. /pa**m**-i**l**-a/ ⇝ [pa**m**i**n**a]	c. /pa**m**-i**l**-a/ ⇝ [pa**m**i**n**a]
b. /**m**asat-i**l**-a/ ⇝ [**m**asati**n**a]	d. /**m**asat-i**l**-a/ ⇝ [**m**asati**l**a]

In Tshiluba, an *l* changes to *n* when there's a nasal anywhere to its left, as illustrated in (a) and (b). In contrast in Lamba, an *l* changes to *n* only if the *first* consonant to its left is a nasal, as demonstrated in (c), with nasalization of /l/ vs. (d) without nasalization. Here is an informal version of the Tshiluba rule.

(14) Tshiluba "long-distance" rule

> +LATERAL INR initiates SEARCH to the LEFT for a +NASAL consonant TRM. If successful, CHANGE INR to + NASAL

The Tshiluba rule is formalized with parameters in (4).[7]

(15) Tshiluba Rule Parameters:

> SEARCH: INR: +LATERAL; TRM: −SYLLABIC, +NASAL ; DIR: L
>
> CHANGE: INPUT: INR; OUTPUT: INR → + NASAL

In the literature, Tshiluba-type long-distance interactions are treated as problematic or special, requiring additional mechanisms. In contrast, our approach derives the long-distance vs local distinction from the specificity and scope of rule parameters. In traditional terms, the Lamba pattern is considered more basic, given its local nature involving interaction between onsets in adjacent syllables. The Lamba rule can be informally stated as follows:

(16) Lamba "local" Rule

> + LATERAL INR initiates SEARCH to the LEFT for *any* consonant TRM. If the TRM consonant is +NASAL, CHANGE the INR to + NASAL

Here is a parameterized version:

---

[7] In this paper, we assume for ease of exposition that CHANGE can consist of feature-changing rules. This simplification contrasts with our other work in Logical Phonology where featural changes in segments are effected by rules based on feature deletion and insertion (e.g., Reiss 2021).

(17) Lamba Rule parameters (to be revised):

SEARCH: INR: +Lateral; TRM: −SYLLABIC; DIR: L
CHANGE: INPUT: INR; OUTPUT: INR → + NASAL;
COND: TRM is +NASAL

Now, let's consider the (13b,d) input form /masat-il-a/ in both languages. In Tshiluba, the +LATERAL segment /l/ is set as INR and launches a leftward SEARCH for a nasal consonant. Since /m/ meets the TRM specifications, and no additional COND are stipulated in the rule, the SEARCH is successful, so CHANGE applies, resulting in /l/ taking on the + NASAL value and changing to [n]. The input form surfaces as [masati**n**a], with [n], not [l].

In Lamba, this same form surfaces with the lateral unchanged. In our analysis, the /m/ is 'inaccessible' to the INR /l/, just because of the generality of the definition of TRM—any consonant terminates SEARCH. In the two language, the SEARCH mechanism is exactly the same—no special machinery is needed for long-distance SEARCH which just follows from the specificity of the TRM.

Recall the shape examples above in which similar patterns emerged in rule A, which had no COND, yielding 'transparency', versus rule C which had a COND and thus yielded 'opaque' segments. This parallels our Bantu cases: in Tshiluba, the SEARCH proceeds for an unbounded"long-distance," but in Lamba, the rule's SEARCH terminates sooner and just appears to be to be more 'local'.

To leverage our door-painting analogy in (7), Tshiluba corresponds to a contractor looking for a two-storey house and painting the door red, whereas Lamba corresponds to a contractor looking for a house and painting the door red if and only if the house has two storeys.

5. What is Adjacency?

Now that we have outlined the basics of our system and demonstrated how various outcomes result from the detail and scope of certain parameter specifications, we are in a position to define a taxonomy of phonological patterns that correlate with specific combinations. We have focused on notions like locality, transparency, and opaqueness but a wide variety of phenomena appear to be reducible to the effects of specificity and scope of rule parameters (e.g. icy targets[8]) and can be similarly captured using the same basic principles.

5.1 Opaqueness is Lack of TRM Specificity

The difference between Tshiluba and Lamba sketched above suggests a simple argument by *reductio ad absurdum*. In Tshiluba, the TRM of SEARCH is fairly precisely specified: the TRM is a nasal consonant. In Lamba, the TRM of SEARCH is less precisely specified: the TRM is a consonant. In general, then, a Tshiluba SEARCH instance will traverse

---

[8] Icy targets, also known as semi-opaque segments, are vowels that 'harmonize' but block the harmony from spreading (Jurgec 2011). See Dabbous et al. (2021) for a treatment using SEARCH and CHANGE.

more segments in a string than a Lamba SEARCH instance. If we make the conditions on SEARCH maximally general, by using the vaguest partial description possible, the result is that SEARCH will terminate even sooner.

Opaqueness, then, arises from a restriction to the scope of a SEARCH within a given string. We have demonstrated in previous sections how specifications on TRMs can be manipulated to effect this restriction via SEARCH terminating on a segment which does not necessarily provide the proper ‘environment’ due to some information being ascribed to the COND parameter instead. This division of labor, we have argued, is a side-effect of separating the environment/trigger into two distinct components: (TRM and COND).

To recap, transparent segments are the TRMs that are maximally specified for a given change, but lacking a COND, resulting in a maximal scope for SEARCH that encompasses a string unobstructed until it finds a proper TRM. Opaque segments, on the other hand, result when a TRM lacks specification, which is instead found in the COND. This causes a SEARCH to terminate ‘pre-maturely’ at a TRM, narrowing the ‘distance’ between INR and TRM.

Absolute adjacency, then, is a maximal restriction of the domain of application of a rule. In other words, absolute adjacency is the kind of opaqueness that arises from minimal feature specifications in a TRM such that a TRM is any segment.

5.2 Adjacency is a Special Case of Long-Distance: least specific TRM

Let us refer back to our shape grammar for an exposition. Assume that S is the minimally specified ‘segment’ that can refer to any feature, color or shape.

To express a rule for the following pattern: “Turn a ■ into a ○ if there is a ▲ to its immediate right”, we must specify that the “trigger” segment (in this case, the black triangle) is located *immediately* to the right of the “target” black square. This contrasts with (2), where the trigger could appear anywhere to the right of the target. In our system, adjacency requires more specifications than long-distance environments. However, our model need not be more complex to handle this—adjacency can be captured using the tools already available. Consider the following formulation:

(18) Rule D

- From a ■ SEARCH for a SHAPE to the right and turn the ■ into a ○ if TRM is ▲
- Rule D Parameters

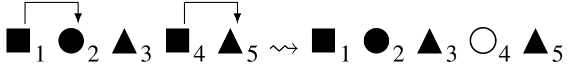
  SEARCH: INR: ■, TRM: S, DIR: R

  CHANGE: INPUT: INR, OUTPUT: ○, COND: TRM: ▲

In (18) the TRM is specified as the maximally general category S (SHAPE). As a result, the SEARCH will terminate at the first symbol to the right of INR, no matter what it is. The CHANGE only applies if the COND is met—in this case, if the first segment to the right is a black triangle. In other words, only a black triangle to the immediate right of the INR black square will trigger the change of INR to a white circle.

As (18) demonstrates, no other machinery needs to be stipulated to capture local patterns. Adjacency is derived via the same mechanisms underpinning 'opaque' segments– namely, a minimal specification on TRM accompanied by a COND. Consider the effects of (18) engendered in the following mapping:

(19) Mapping from Rule D in (18)



The black squares in positions 1 and 4 satisfy the INR specifications and will each launch a SEARCH. Because the TRM is defined as 'any shape/color', the SEARCH will stop at the shape immediately following each INR—the black circle in position 2 and the black triangle in position 5, respectively. However, the COND stipulating that a CHANGE is only triggered when the TRM is a black triangle ensures that the black circle TRM in position 2 'blocks' the CHANGE from applying, resulting in an unchanged INR. The TRM in position 5, however, meets the criteria set by the CON, which triggers the INR in position 4 to become a white circle.

Let's apply this reasoning to the following nasalization rule which targets only laterals (right-)adjacent to a nasal. The minimally specified natural class description '[ ]' defines the natural class of *all* segments (Reiss 2021), parallel to use of the symbol S in Rule D above.

(20) Strictly Adjacent Nasalization Rule

- l → n / m,n __
- Rule Parameters

SEARCH: INR: +Lateral; TRM: [ ]; DIR: L
CHANGE: INPUT: INR; OUTPUT: INR → + NASAL;
COND: TRM is +NASAL

Here, much like Tshiluba and Lamba, a SEARCH is launched by each lateral segment /l/. The maximally general partial description for TRM, denoting any segment, ensures that the SEARCH range is restricted to the immediately preceding segment—any segment terminates the SEARCH. If a particular TRM segment is a nasal, the CHANGE condition is met, resulting in an /l/ surfacing as [n].

6. Lamba sometimes has long-distance nasalization

It turns out that underlying forms containing a sequence of laterals can be found in languages like Lamba, something like /pam-il-il-a/. Since we have characterized Lamba as having local nasal spread, it is surprising that such a form surfaces with *both* laterals nasalized, as in [paminina].

All we need to do to ensure this outcome is to narrow down the partial description of the TRM natural class. Instead of terminating SEARCH at any consonant, we can make laterals transparent by adding the specification that TRM must be −LATERAL, as in (21).

(21) Lamba Rule parameters (final version):

SEARCH: INR: +Lateral; TRM: −SYLLABIC, −LATERAL; DIR: L
CHANGE: INPUT: INR; OUTPUT: INR → + NASAL;
COND: TRM is +NASAL

This rule allows SEARCH to proceed through a lateral, which means that both underlying /l/'s will find the same TRM. This is important, because it means that we do not need iterative rule application—we do not need the rule to first affect the leftmost /l/ and then the rightmost one. The SEARCH and CHANGE model simplifies rule application by doing away with apparent iterative application. This example suggests that locality and adjacency are not properties of languages or rules, but are epiphenomena arising from the interaction of the scope and specificity of rule parameters.

7. Discussion

Locality, alongside opaqueness and transparency, have played a pivotal role in phonological research across decades and theoretical paradigms. In this paper, we have aimed to demonstrate that these concepts are, in fact, epi-phenomenal and purely descriptive, lying outside the domain of core phonological competence.

We introduced a phonological model that shifts focus from traditional notions of locality to a framework emphasizing unbounded relationships. Unlike approaches that privilege immediate adjacency as a distinct case of precedence[9], our model treats adjacency as emergent phenomena, unified under mechanisms that also account for segment ”opaqueness.”

Building on prior work, such as McMullin (2016) where local consonant agreement is linked to the treatment of consonants as blockers, our model takes a further step by decoupling phonological processes from a reliance on copying mechanisms. Central to this framework is a redefinition of key concepts: targets and triggers are reimagined as INR and TRM, maintaining the principle of Substance-Freeness of Structural Changes (SFSC) while refining their traditional roles.

---

[9] For discussions on precedence and immediate precedence, see Raimy (2000); Shen (2016) and Papillon (2020).

The core principles of our model can be summarized as follows:

(22) Key aspects of SEARCH and CHANGE model

a. SEARCH-Based Rules: Rule are best conceived as SEARCH procedures. If the SEARCH terminates successfully and the COND is satisfied, a rule's CHANGE applies. Otherwise, the output form is identical to the input.

b. Descriptive Phenomena as Derivables: Opaqueness and transparency are not intrinsic properties but instead arise naturally from the parameters of our rules.

c. Deprioritization of Taxonomic Labels: Constructs like "vowel harmony" or "place assimilation" are viewed as taxonomic artifacts rather than fundamental components of phonological systems.

By emphasizing general computational principles, our framework addresses key issues in phonology while offering a new perspective on locality, opaqueness and transparency. The model's reliance on Search procedures enables flexibility in handling intervening material without the need to explicitly define permissible configurations between targets and triggers. Additionally, our approach clarifies the independence of the changes enacted by phonological rules and the environments they define. While many questions remain open, we believe this work advances the discussion toward uncovering general representational and computational principles of phonological systems and away from taxonomic artifacts like "vowel harmony," "opaqueness" and "assimilation".

References

Bale, Alan, and Charles Reiss. 2018. *Phonology: A formal introduction*. Cambridge, MA: MIT Press.

Bale, Alan, Charles Reiss, and David Ta-Chun Shen. 2020. Sets, rules and natural classes: { } *vs.* [ ]. *Loquens* 6(2): e065.

Burness, Phillip Alexander, Kevin James McMullin, and Jane Chandlee. 2021. Long-distance phonological processes as tier-based strictly local functions. *Glossa: a journal of general linguistics* 6. 99.

Chandlee, Jane, and Jeffrey Heinz. 2018. Strict locality and phonological maps. *Linguistic Inquiry* 49(1): 23–60.

Chomsky, Noam, and Morris Halle. 1968. *The sound pattern of English*. New York: Harper & Row.

Clements, G.N. 1980. *Vowel harmony in nonlinear generative phonology*. Indiana Univ. Linguistics Club.

Dabbous, Rim, Marjorie Leduc, Fatemeh Mousavi, Charles Reiss, and David Ta-Chun Shen. 2021. Satisfying long-distance relationships (without tiers): A strictly anti-local approach to phonology. *Submitted*. Lingbuzz/006329.

Goldsmith, John. 1979. On the thematic nature of *See*. *Linguistic Inquiry* 10(2): 347–352.

Hale, Mark, and Charles Reiss. 2008. *The phonological enterprise*. Oxford: Oxford University Press.

Heinz, Jeffrey. 2010. Learning long-distance phonotactics. *Linguistic Inquiry* 41(4): 623–661.

Heinz, Jeffrey. 2018. The computational nature of phonological generalizations. In *Phonological typology*, ed. Larry Hyman and Frans Plank, Phonetics and Phonology, 126–195. Berlin: De Gruyter Mouton.

Heinz, Jeffrey, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints for phonology. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, ed. Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, 58–64. Portland, OR: Association for Computational Linguistics. https://aclanthology.org/P11-2011/.

Jurgec, Peter. 2011. Feature spreading 2.0: A unified theory of assimilation. Doctoral dissertation, Universitetet i Tromsø.

Mailhot, Fred, and Charles Reiss. 2004. The form and interpretation of phonological rules. Ms., Concordia University.

Mailhot, Frederic, and Charles Reiss. 2007. Computing long-distance dependencies in vowel harmony. *Biolinguistics* 1(1): 28–48.

McMullin, Kevin James. 2016. *Tier-based locality in long-distance phonotactics: learnability and typology*. Doctoral dissertation, University of British Columbia.

Nevins, Andrew. 2010. *Locality in vowel harmony*. Cambridge, MA: MIT Press.

Odden, David. 1994. Adjacency parameters in phonology. *Language* 70(2): 289–330.

Papillon, Maxime. 2020. Precedence and the lack thereof: Precedence-relation-oriented phonology. Doctoral dissertation, University of Maryland.

Raimy, Eric. 2000. *The phonology and morphology of reduplication*. Berlin: Mouton de Gruyter.

Reiss, Charles. 2017. Substance Free Phonology. In *Handbook of Phonological Theory*, ed. S.J. Hannahs and Anna Bosch, 425–452. New York: Routledge.

Reiss, Charles. 2021. Towards a complete Logical Phonology model of intrasegmental changes. *Glossa* 6(1): 107.

Samuels, Bridget D. 2011. *Phonological architecture: a biolinguistic perspective*. Oxford: Oxford University Press.

Shen, David Ta-Chun. 2016. *Precedence and Search: Primitive concepts in morpho-phonology*. Doctoral dissertation, National Taiwan Normal University.