Exploring the Utility of Algebraic Lattices in Modelling Syllable Structure

Fae Hicks
University of Edinburgh
fhicks@sms.ed.ac.uk

23rd January 2026

Abstract

I propose two potential solutions to the problems of (i) exhaustive representation in syllabic trees and (ii) referring to nothing. The first using lattice structure to structurally but not explicitly exclude syllables with codas. The second resolves the issue by treating syllabic structure as part of the rule environment. SEARCH parses a rule, if it encounters a syllable boundary, denoted by ., in the rule environment it triggers the SYLLABIFICATION ALGORITHM. Syllabification defines syllable structure in terms of positional order: assigning nuclei and syllable boundaries and defining all other syllable constituents in terms of their order relative to these two points. The output of the SYLLABIFICATION ALGORITHM then replaces the shorthand in the rule and SEARCH applies and if COND are satisfied CHANGE applies. I conclude that the second is better solution.

1 Overview

The Challenge: formulate phonological rules that apply to vowels in open syllables without (i) referring to absence of a coda or (ii) relying on exhaustive tree diagrams

The Competing Solutions:

1. Syllables as lattice structures
2. Syllabification as part of the rule environment

The Winner: 2. Rule environment trigger a SYLLABIFICATION ALGORITHM that inserts syllable boundary markers into a segment string on a language- and rule-specific-basis.

2 The Problem of Referring to Absence

The Problem of Referring to Absence: phonological rules can't operate on something that isn't there.

2.1 Underspecification is Absence

2.1.1 Problem

In Logical Phonology phonological rules can operate on those things that they can refer to (but not those they can't). For any segment underspecified for some feature that underspecified segment is a subset of the fully specified segments making it impossible to directly refer to the underspecified segment without also referring to its not-underspecified counterparts (Dabbous et al, 2025; Gorman & Reiss, 2023; 2025). For example in Turkish as in Fig.1 (Reiss 2022).

Figure 1

$$t = \begin{Bmatrix} -\text{Voice} \\ +\text{Cor} \\ -\text{Cont} \\ -\text{Son} \end{Bmatrix} \qquad d = \begin{Bmatrix} +\text{Voice} \\ +\text{Cor} \\ -\text{Cont} \\ -\text{Son} \end{Bmatrix} \qquad D = \begin{Bmatrix} +\text{Cor} \\ -\text{Cont} \\ -\text{Son} \end{Bmatrix}$$

The natural class defined by /D/ is all the segments which are supersets of the features which are in /D/. /t/ and /d/ are supersets of /D/ and therefore are supersets of all the features in /D/. Therefore, it is impossible to refer to /D/ without referring to /t/ and /d/.

Suppose, as in Turkish, /D/ is affected but /t/ and /d/ are not. We cannot refer to absence, so how can rules target /D/? (see Reiss this morning)

2.1.2 Solution

Logical Phonology solves this issue with unification since unification of { +Voice } with /t/ results in vacuous unification and unification of { +Voice } with /d/ results in unification failure, successful unification is only observed with the underspecified /D/. Thus, **rules can be written that have the effect of referring to absence without having to actually do so** - I aim to find an equivalent for syllable structure.

3 Reference to Open Syllables is Reference to Absence

3.1 Referring to Absence

An open syllable is a syllable without a coda. Therefore, any rule whose environment is "in an open syllable" is referring to absence by saying apply "when there is no coda".

"Everything but the coda" is not a constituent of a syllable structure.

The unification solution does not seem to be applicable to syllable structure. The logic that works at the feature level does not seem to apply in syllable based environments so we need some other way to refer to seeming absence.

3.2 Avoiding Absence with Exhaustivity

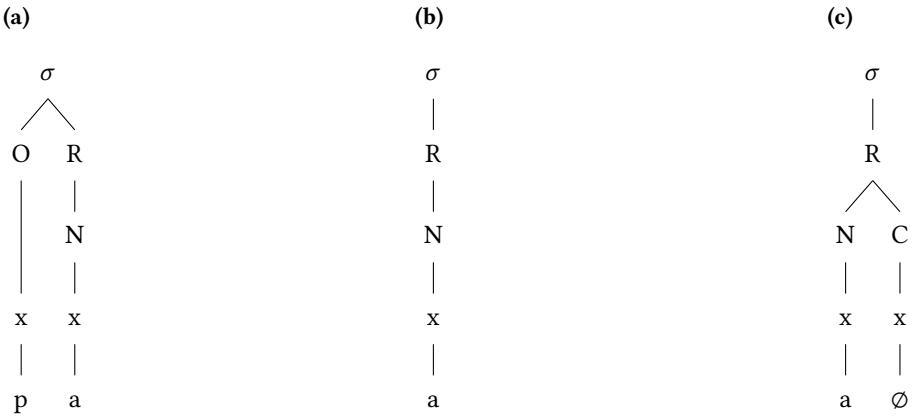
To avoid positing rules that explicitly refer to the absence of a coda tree structure models have been assumed to be exhaustive, meaning that anything not represented in a tree diagram can be assumed to be absent from the syllable.

This raises its own problems as it necessitates explicitness throughout the structure. So, a rule applying in open syllables regardless of the status of the onset requires at least two autosegmental representations, one to represent rule application in syllables without onsets and a second to show rule application in syllables with onsets[1]. Furthermore, situation in 2a & 2b does not reflect natural language. I am unaware of any language where the

action of a phonological function on a vowel is conditioned by the presence/absence of an onset i.e., that would apply in 2b but not 2a or vice versa.

[1] N.B. The number of rules necessary further increases when you consider the possibility of complex onsets.

Figure 2: Autosegmental structures representing open syllables



3.3 Removing Redundancy with Absence

To remove the redundancy of writing a different rule for each different syllable type we could assume that autosegmental representations are not exhaustive and explicitly mark an empty coda position as in 2c. But, this brings us back to square one, referring to absence.

3.4 Summary: The Problem of Open Syllables

Format phonological rules that apply to segments in open syllables without:

(i) Referring to absence

(ii) Relying on exhaustivity

4 Solution 1: Syllables as Lattices

Positing the syllable as a lattice allows non-target syllable types to be structurally excluded without making assumptions about other elements of the structure.

A lattice a special type of partially ordered set which, when built from ordered tuples, can be used to model syllable structure as in 3 (Davey & Priestly, 2002; Partee, ter Meulen & Wall, 1990).

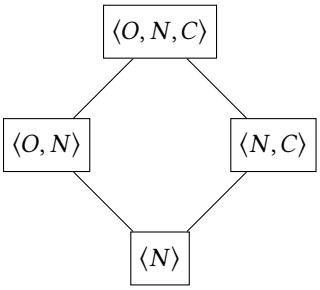


Figure 3: Closed lattice of licit syllable structures

The diagonal line (EDGE) between the boxes (VERTICES) denotes a COVER RELATION, e.g. $\langle N \rangle$ is covered by $\langle O, N \rangle$. Referring to the cover relation allows reference to either member of it to the exclusion of any other member of the lattice. So, nuclei in the cover relation ($\langle N \rangle$,$\langle O, N \rangle$) can be referred to without knowledge of the status of any onsets and without reference to absence of a coda.

So, using lattice cover relations solves problems (i) and (ii).

(i) Closed syllables can be structurally excluded without explicitly referring to their absence

(ii) An element of any member of a cover relation can be referred to by the same function (= we don't need to state whether there is an onset)

However, the lattice structure allows for the modelling of patterns that are not observed in natural languages such as rules that refer exclusively to syllables that consist only of nuclei.

5 Solution 2: Syllables in the Environment

Here I return to the old idea that syllabification is something that the phonology *does* (Kahn, 1976; Dell & Elmedlaoui 1985; Gussenhoven, 1986; Giergerich, 1992). I propose that the presence of syllable structure in the conditioning environment of a rule triggers a syllabification algorithm. Thus, a syllabification algorithm is a part of a particular rule in a particular language.

Key assumptions

1. Syllable structure is part of the phonological computation not stored in the lexicon
2. Reference to syllable structure in the rule environment triggers syllabification
3. Syllabification rules are **not** separate rules of phonology, they are part of the conditioning environment of other rules

This solves (ii) as syllabification is only triggered when necessary for rule computation and only those syllabic structures specified in the environment can affect the rule so no assumptions are made about unspecified bits of structure.

The nature of the SYLLABIFICATION ALGORITHM (SA) laid out in 5.3 will resolve (i).

5.1 Rule Elements

Bale & Reiss (2018) posit the boundary markers # for the beginning of strings of segments and % for the end[2]. In order to refer to syllable boundaries I propose the addition of the symbol "**.**"

BOUNDARY SYMBOLS: %, #, **.**

Additionally, in order to calculate syllable structure according to positional ordering the SA must assign nuclei, N. I am assuming that N is an object with the same ontological status as **.**

STRING STRUCTURE MARKERS: %, #, **.**, N

5.2 Rule Structure

Imagine a rule that backs low vowels in open syllables as in (3)

(3) [+LOW] ⊔ {+BACK } / $_{N\_\_\_N}$ **.**

Parsing (3)

1. SEARCH reads this rule
2. **.** tells SEARCH that the environment of this rule requires syllabification

   $_{N\_\_\_N}$ **.** tells CHANGE that COND is *open syllables.*[3]
3. The input string is syllabified by the SYLLABIFICATION ALGORITHM.
4. The syllabified string is reparsed by SEARCH
   (a) If the environmental conditions, TRM and COND, are satisfied CHANGE performs the function
   (b) If the environmental conditions, TRM and COND are not satisfied CHANGE does not perform the function

[2]The ontological status of these is up for debate but following B&R I will assume that they form part of our mental representation.
[3]This representation is meant as a shorthand for syllable position, see A for more.

5.3 Syllabification

We are jumping in at Step 3. for discussion of rule parsing with SEARCH and CHANGE see Dabbous et al (2025).

Syllabification proceeds in stages. Assume we are syllabifying the English word *browsing* represented as the string #braʊzɪŋ%. In (4) *browsing* is represented as an ordered string.

(4) $\{\langle \#_1,b_2\rangle, \langle b_2,r_3\rangle, \langle r_3,a_4\rangle\ , \langle a_4,ʊ_5\rangle, \langle ʊ_5,z_6\rangle, \langle z_6,ɪ_7\rangle, \langle ɪ_7,ŋ_8\rangle, \langle ŋ_8,\%_9\rangle\}$

N.B. The subscript numbers in (4) represent the POSITIONAL (ORDER) VALUE of their respective segments

5.3.1 Identify Nuclei

The SA parses the string of segments assigning nuclei according to sonority then returns the positional values of each N.

Each N has a lower bound, B (the positional value of the nuclear element with the lowest positional value) and an upper bound, E (the positional value of a nuclear element with the highest positional value). Which can be represented as in the general template (5).

(5) $N_{B:E}$

The specific nuclear ranges for the string in (4), *browsing*, are shown in (6).

(6) $N_{4:5}$, $N_{7:7}$

5.3.2 Assign Syllable Boundaries

Once nuclei are assigned the SA assigns syllable boundaries according the maximal onset principle and the specific sonority rules of the language.

a **Mark existing boundaries:** % and # subsume **.** so their positional order is automatically recorded as syllable boundaries

b **Insert syllable boundaries:** other syllable boundaries must be inserted by SA. This counts back from position B assigning onsets first according to language specific sonority constraints. So, for English a syllable boundary is inserted after a maximum of three positional slots from B.

c **Recalculate order:** inserting string internal syllable boundaries will have changed the positional ordering, so order must be updated to reflect this, (4) becomes (7).

(4) $\{\langle \#_1,b_2\rangle, \langle b_2,r_3\rangle, \langle r_3,a_4\rangle\ , \langle a_4,ʊ_5\rangle, \langle ʊ_5,z_6\rangle, \langle z_6,ɪ_7\rangle, \langle ɪ_7,ŋ_8\rangle, \langle ŋ_8,\%_9\rangle\}$

(7) $\{\langle \#_1,b_2\rangle, \langle b_2,r_3\rangle, \langle r_3,a_4\rangle\ , \langle a_4,ʊ_5\rangle, \langle ʊ_5,._6\rangle, \langle ._6,z_7\rangle, \langle z_7,ɪ_8\rangle, \langle ɪ_8,ŋ_9\rangle, \langle ŋ_9,\%_{10}\rangle\}$

5.3.3 Reassign Nuclear Ranges

The insertion of syllable boundaries changes the order of segments in the string so the nuclear ranges calculated in 5.3.1 may no longer be accurate, so they must be recalculated.

Since the order of segments changed be (4) and (7) the nuclear ranges must also be updated such that (6) is recalculated as (8).

(6) $N_{4:5}$, $N_{7:7}$

(8) $N_{4:5}$, $N_{8:8}$

5.3.4 Determine Syllable Constituents by Positional Ordering

Now we know the positional order of the syllable boundaries and the nuclei other syllabic positions can be derived. General statements can be made using the notation in (9).[4]

(9) B: the lowest positional order value in a given nucleus

E: the highest positional order value in a given nucleus

A: the closest syllable boundary that has a lower positional order value than B

F: the closest syllable boundary that has a higher positional order value than *e*

Open Syllables can be defined as:

(10) A syllable where the ordered position of the syllable boundary is exactly equal to the ordered position of the upper bound of the nucleus +1.

(11) $x_q|\text{B}\leq q <\text{F}\&(\text{F}-\text{E}) = 1$

Whether either of the two syllables in *browsing* is open can be determined by substituting the positional orders of its segments into the logical form in (11).

Syllable 1: braʊ

A. $x_q|\text{B}\leq q <\text{F}$

$x_q|4 \leq q < 6 = \text{TRUE}$

B. $\text{F} - \text{E} = 1$

$6 - 5 = 1 = \text{TRUE}$

C. σ_1 is open = TRUE

Syllable 2: zɪŋ

A. $x_q|\text{B}\leq q <\text{F}$

$x_q|8 \leq q < 10 = \text{TRUE}$

B. $\text{F} - \text{E} = 1$

$10 - 8 = 1 = \text{FALSE}$

C. σ_2 is open = FALSE

This notation type can also be used to refer other syllable types see appendix B for details.

5.4 Parsing the Syllabified String

Now that the string has been syllabified it can be reparsed by SEARCH to determine whether the environmental conditions are met.

(3) [+LOW] ⊔ {+BACK } / $_{N\_\_\_N}$.

SEARCH will only stop if it encounters TRM and CHANGE will only apply the function if COND is met (Dabbous et al, 2025):

- TRM: [+LOW]
- COND: $x_q|\text{B}\leq q <\text{F}\&(\text{F}-\text{E}) = 1$ (= open syllables)

[4] N.B. I assume that this is part of the phonologists' meta-language not the grammar itself

For [a] in σ_1

- Trm: [+Low] = true
- Cond: $x_q|4 \leq q < 6$ & $6 - 5 = 1$ = true
- Trm and Cond are satisfied so Change applies and unification occurs

For [ʊ] in σ_1

- Trm: [+Low] = false
- Cond: $x_q|4 \leq q < 6$ & $6 - 5 = 1$ = true
- Cond is satisfied but Trm is not, so Change does not apply applies

For [ɪ] in σ_2

- Trm: [+Low] = false
- Cond: $x_q|8 \leq q < 10$ & $10 - 8 = 1$ = false
- Trm and Cond are not satisfied so Change does not apply

Thus, the product of (3) applying to [braʊzɪŋ] is /braʊ.zɪŋ/

5.5 Summary

Compare the situations below in 1. syllable structure is relevant to the application of a rule, therefore it features in the rule environment. However, in 2. syllable structure is irrelevant so is not mentioned.

1. Search parses a rule and encounters "**.**"
   (a) syllabification algorithm applies:
      i. Nuclei identified
      ii. Syllable boundaries assigned
      iii. Positional ordering recalculated
      iv. Syllable constituents are derived from ordering relations
      v. Syllabification is complete and sa outputs an ordered string with syllable positions defined by ordering relations between syllable boundaries and nuclear elements
   (b) Search parses the rule
      i. If Trm and Cond are satisfied, change performs the function
      ii. If Trm and Cond are **not** satisfied met the rule fails to apply
2. Search parses a rule and does not encounter "**.**"
   (a) Nothing triggers the application of sa so no syllabification occurs
      i. If Trm and Cond are satisfied, Change performs the function
      ii. If Trm and Cond are **not** satisfied met the rule fails to apply

6 What This Solves

(i) **Syllabification on positional order resolves the problem of referring to absence**

By defining syllable constituents and syllable types on this basis we are defining them on positively, on what is included not on what is absent.

e.g., an open syllable is defined by its own properties, the positional value of the upper bound of its nucleus compared to a syllable boundary - the absence of a coda is implied but is not defining.

(ii) **Treating syllabic structure as part of the rule environment solves the problem of exhaustivity**

Since (1) is implicit in any phonological rule if we assume that syllabic structure is just the same any other part of the rule environment we can assume (2).

(1) Only those properties listed in the environment of a phonological rule condition the application of that rule

(2) Only syllabic structures listed in ENV can condition the application of the rule

Assuming (1) and (2) means that all and only the specified bits of syllable structure can affect the application of the phonological function, i.e., we do not need to know what is going on in the onset to write a rule that affects open syllables.

7 References

Bale, Alan., & Reiss, C. (2018). Phonology: A Formal Introduction. The MIT Press.

Davey, B. A., & Priestley, H. A. (2002). Introduction to Lattices and Order (2nd ed.). Cambridge University Press.

Dabbous, R., Gorman, K., & Reiss, C. (2025). Tutorial on Substance-Free Logical Phonology. `lingbuzz/008746`

Dabbous, R., Leduc, M., Reiss, C., & Shen, D. T.-C. (2025). Locality is epiphenomenal: adjacency is opaqueness. `lingbuzz/008785`

Dell, F., & Elmedlaoui, M. (1985). Syllabic Consonants and Syllabification in Imdlawn Tashlhiyt Berber. *Journal of African Languages and Linguistics*, 105–130.

Giegerich, H. J. (1992). English Phonology: An Introduction. Cambridge: Cambridge University Press.

Gorman, K. & Reiss, C. (2023) Maximal feature specification is feasible; minimal feature specification is not

Gorman, K., & Reiss, C. (2025). Natural class reasoning in segment deletion rules. `lingbuzz/009501`

Gussenhoven, C. (1986) English Plosive Allophones and Ambisyllabicity. *Gramma*, 119 - 141

Kahn, D. (1976) Syllable-Based Generalisations in English Phonology. Unpublished PhD Thesis. MIT Partee, B. B. H., ter Meulen, A. G., & Wall, R. (1990). Mathematical Methods in Linguistics. Springer Netherlands.

Reiss, C., (2021). Priority union and feature logic in phonology. *Linguistic Inquiry*.

A Shorthand for Syllable Structure in the Environment

Now that we have defined the positional ranges corresponding to the traditional syllable structure labels: onset, nucleus, coda, and rhyme we can devise a shorthand to use in rule notation.

- ONSET = $x_q|_{\text{A}<\, q\, <\text{B}} = {}_{O}\_\_\_{}_{O}$
- NUCLEUS = ${}_{N}\_\_\_{}_{N}$
- CODA = $x_q|_{\text{E}<\, q\, <\text{F}} = {}_{C}\_\_\_{}_{C}$
- RHYME = $x_q|_{\text{B}\leq\, q\, <\text{F}} = {}_{R}\_\_\_{}_{R}$

B Positional Order Definitions of Syllable Structures

Syllable Constituents

Where x is some segment and q is the positional order of that segment we can determine the syllabic role of x according to the following order relations

- ONSET:

  $x_q|_{\text{A}<\, q\, <\text{B}}$

There is some element, x, whose positional order, q, is less than the lower bound of the nucleus, B, and greater to than the positional value of the closest syllable boundary lower than B, A.

- CODA:

  $x_q|_{\text{E}} < q <_{\text{F}}$

  There is some element, x, whose positional order, q, is greater than the upper bound of the nucleus, E, and less than the positional value of the closest syllable boundary greater than E, F.

- RHYME:

  $x_q|_{\text{B}} \leq q <_{\text{F}}$

  There is some element, x, whose positional order, q, is greater than or equal to the lower bound of the nucleus, B, and less than the positional value of the closest syllable boundary greater than B, F

Segment Type in Constituent Type

- (VOWEL IN) NUCLEUS:

  $x_q|_{\text{B}} \leq q \leq_{\text{E}}$

  There is some element, x, whose positional order, q, is greater than or equal to the lower bound of the nucleus, B, and less than or equal to the upper bound of the nucleus, E

- (VOWEL IN) OPEN SYLLABLE:

  $x_q|_{\text{B}} \leq q <_{\text{F}} \wedge (\text{F} - \text{E}) = 1$

  There is some element, x, whose positional order, q, is greater than or equal to the lower bound of the nucleus, B, and less than the less than the positional value of the closest syllable boundary greater than E (the upper bound of the nucleus), F where the positional order of F is exactly one higher than the positional order of the upper bound of the nucleus i.e., F = E + 1.

- CONSONANT IN A CLOSED SYLLABLE

  $x_q|_{\text{E}} < q <_{\text{F}}$

  There is some element, x, whose positional order, q, is greater than the upper bound of the nucleus, E, and less than the positional value of the closest syllable boundary greater than E, F.

- VOWEL IN A CLOSED SYLLABLE

  $x_q|_{\text{B}} \leq q \leq_{\text{E}} \wedge \text{E} \leq \text{F} - (\text{F} - \text{E})$

  There is some element, x, whose positional order, q, is greater than or equal to the lower bound of the nucleus, B, and less than or equal to the upper bound of the nucleus,E. And the value of E is less than the value of the positional order of F (where F is the closest syllable boundary greater than E) minus the value of the positional order of F.