THE LOGICAL STRUCTURE OF PHONOLOGICAL THEORY

EGG 2022

Veno Volenec

Concordia University

Linguistics program

INTRODUCTION

- **Logical Phonology (LP)**
  - A substance-free (purely formal) framework for describing phonological knowledge that uses set theory (Volenec & Reiss 2020)
  - Segments (phonemes, allophones) are sets of features
    - p = {-syl, +cons, -son, -cor, +ant, +lab, -cont, -nas, -lat, -strid, -voi}
  - Natural classes are sets of sets of features (i.e., sets of segments), as defined by general intersection
    - [-son, -cont] = {p, t, k, b, d, g}
  - Rules are set-theoretic operations, such as set subtraction and set unification
    - Subtraction: [+cor, +cont, –son] – {+cor, +ant} / __ [–ant, –back, +cons]
    - Unification [+cont, –son] ⊔ {αcor, –ant} / __ [αcor, –ant, –back, +cons]

INTRODUCTION

- Overview of topics
  - General cognitive and linguistic assumptions that underlie Logical Phonology (LP)
  - Basics of set theory: set, member, union, intersection, superset, subset...
  - Core phonological concepts in LP: redefining representations and computations
  - Using LP in phonological analysis in order to uncover the organization of phonological competence: applying LP to a variety of patterns from different languages
  - Highlighting pros and cons of LP; determining future directions

INTRODUCTION

- Some fundamental assumptions of Logical Phonology
  - Object of description is **phonological competence**, as part of I-language, not speech or overt verbal behavior
  - **I-language** (linguistic competence) = a knowledge-system that can generate an infinity of well-formed sentences; internal, individual, implicit, intensional
  - Adheres to the competence-performance distinction: strict separation of phonology and phonetics
  - **Phonology**: competence, language-specific, mental ('knowing'), discrete, abstract, substance-free
  - **Phonetics**: performance, not language-specific, physical ('doing'), gradual, concrete, substantive

INTRODUCTION

PHONOLOGY

COMPETENCE

underlying phonological representation (sets of features)

↓

computational part of phonology (ordered set-theoretic operations)

↓

surface phonological representation (sets of features)

↓

cognitive phonetics (transduction of features at the phonology-phonetics interface)

↓

phonetic representation (temporally coordinated motor programs)

↓

speech sounds

PHONETICS

PERFORMANCE

BASIC SET THEORY

- **Set** = a collection of distinct members or elements
  - Member, element = any particular entity within a set
  - Curly brackets determine the boundaries of a set
  - {a, b, c} is a set that contains the members a, b and c
- **Membership** = the property of being included in a particular set
  - **Membership symbol**: ∈
  - Not a member: ∉
  - a ∈ {a, b, c}   c ∈ {a, b, c}   d ∉ {a, b, c}
- Naming (labeling) sets is useful if a set contains many members
  - S = {a, b, c}; a ∈ S; d ∉ S

BASIC SET THEORY

- Order does not matter
- $\{a, b, c\} = \{a, c, b\} = \{b, a, c\} = \{b, c, a\} = \{c, a, b\} = \{c, b, a\}$
- Repetition does not matter
- $\{a, b, c\} = \{a, b, c, a\} = \{a, b, b, c\} = \{a, b, b, c, a\} = \{a, b, c, c\} = \{a, c, b, a, c, b\}$
- Sets can also be members of sets
  - $B = \{a, b, c\}$
  - $A = \{e, f, \{a, b, c\}, g\} = \{e, f, B, g\}$
  - A <u>does not</u> have a, b or c as members! These elements are members of a member of A but not members of A itself
  - Valid statements:

    $\{a, b, c\} \in \{e, f, \{a, b, c\}, g\}$

    $a \notin \{e, f, \{a, b, c\}, g\}$

    $e \in \{e, f, \{a, b, c\}, g\}$

BASIC SET THEORY

- It matters if within a set there is another set or a member that is not a set
  - $\{a, \{b\}, c\} \neq \{a, b, c\}$
  - Be careful: $b \notin \{a, \{b\}, c\}$ and $\{b\} \notin \{a, b, c\}$

- **Relations** between sets: subsets and supersets
- **Subset** (inclusion, containment)
  - If A and B are sets, then A is a subset of B if all members of A are also members of B
  - $A \subseteq B$ if and only if for every $x \in A$, $x \in B$
  - $\{a, b, c\} \subseteq \{a, b, c, d, e\}$
  - Every set is a subset of itself (equality): $\{a, b, c\} \subseteq \{a, b, c\}$; $\{1, 2, 3, 4\} \subseteq \{1, 2, 3, 4\}$
  - Notice that the possibility of equality is indicated in the subset symbol $\subseteq$

BASIC SET THEORY

- **Proper subset**
  - Like a subset, but prohibits equality
  - If A and B are sets, then A is a proper subset of B if A is a subset of B, but A does not equal B
  - $A \subset B$ if and only if $A \subseteq B$ and $A \neq B$
  - $\{a, b, c\} \subset \{a, b, c, d, e\}$
  - Difference between subset and proper subset relations: $\{a, b, c\} \subseteq \{a, b, c\}$; but $\{a, b, c\} \not\subset \{a, b, c\}$

  - Some examples with sets within sets:

    $\{a, \{b, c, d\}\} \subset \{a, \{b, c, d\}, e, \{f, g\}\}$

    $\{b, c, d\} \not\subset \{a, \{b, c, d\}, e, \{f, g\}\}$

    $\{\{b, c, d\}\} \subset \{a, \{b, c, d\}, e, \{f, g\}\}$

BASIC SET THEORY

- **Superset**
  - If A and B are sets, then A is a superset of B if all members of B are also members of A
  - $A \supseteq B$ if and only if for every $x \in B$, $x \in A$

  Examples: $\{a, b, c\} \supseteq \{a, b\}$

  $\{a, \{b, c, d\}, e, \{f, g\}\} \supseteq \{a, \{b, c, d\}\}$

  $\{a, b\} \supseteq \{a, b\}$
  - The subset vs. superset relation is just a matter of direction, similar to $2 > 1$ vs. $1 < 2$
- **Proper superset**
  - If A and B are sets, then A is a proper superset of B if A is a superset of B but A does not equal B
  - $A \supset B$ if and only if $A \supseteq B$ and $A \neq B$
  - $\{a, b, c\} \supset \{a, b\}$; $\{a, b\} \not\supset \{a, b\}$

BASIC SET THEORY

- **Exercise**
- $V = \{a, b, c, d, \{b, c\}, \{b, e\}\}$
  - Is $\{a\} \in V$?
  - Is $a \in V$?
  - Is $\{b, e\} \in V$?
  - Is $\{b, d\} \in V$?
  - Is $e \in V$?
  - Is $a \subseteq V$?
  - Is $\{b, e\} \subseteq V$?
  - Is $\{a, b, c, d, \{b, c\}, \{b, e\}\} \supset V$?
  - Is $\{\{b, c\}, \{b, e\}, a, b, c, d\} \neq V$?
  - Is $\{a, b, c, d, \{b, c\}, \{b, e\}\} \supseteq V$?

BASIC SET THEORY

- **Exercise**
- $V = \{a, b, c, d, \{b, c\}, \{b, e\}\}$
  - Is $\{a\} \in V$? **No**
  - Is $a \in V$? **Yes**
  - Is $\{b, e\} \in V$? **Yes**
  - Is $\{b, d\} \in V$? **No**
  - Is $e \in V$? **No**
  - Is $a \subseteq V$? **No**
  - Is $\{b, e\} \subseteq V$? **No**
  - Is $\{a, b, c, d, \{b, c\}, \{b, e\}\} \supset V$? **No**
  - Is $\{\{b, c\}, \{b, e\}, a, b, c, d\} \neq V$? **No**
  - Is $\{a, b, c, d, \{b, c\}, \{b, e\}\} \supseteq V$? **Yes**

BASIC SET THEORY

- ‘Subset’ and ‘superset’ are **relations**: statements that can be evaluated as either true or false
- **Operations** are neither true nor false: they are statements that apply to some members, do something to them, and then give back an output also in terms of members
- Intersection, subtraction, union are set-theoretic operations that will be useful for phonology

BASIC SET THEORY

- **Set intersection**
  - If A and B are sets, then $A \cap B$ is the smallest set that contains every element that is both a member of the set A and a member of the set B (basically, just looks for members that are shared between two sets)
  - Examples: $\{a, b, c, d\} \cap \{b, c, e\} = \{b, c\}$

    $\{a, b, c, e\} \cap \{e, f, g\} = \{e\}$

    $\{a, b, c\} \cap \{a, b, c\} = \{a, b, c\}$
  - Intersection can yield an **empty set**
  - $\{a, b\} \cap \{c, d\} = \emptyset$ or $\{ \}$
  - A set that contains no members
  - $\{a, b\} \cap \emptyset = \emptyset$ (universally true for all sets)

BASIC SET THEORY

- **Union**
  - Creates sets by combining all the members of one set with all the members of another set
  - If A and B are sets, then A ∪ B results in the smallest set that contains all the elements of A and all the elements of B
  - Examples: $\{a\} \cup \{b, c\} = \{a, b, c\}$

    $\{a, b, c, d\} \cup \{b, c, e\} = \{a, b, c, d, e\}$ (repetition doesn't matter)

    $\{a, b, c\} \cup \{a, b\} = \{a, b, c\}$
  - Union with an empty set
  - $\{a, b\} \cup \emptyset = \{a, b\}$
  - Universally, nothing changes: $X \cup \emptyset = X$

BASIC SET THEORY

- **Subtraction**
- If A and B are sets, then A − B results in the set that contains all and only the elements of A that are not elements of B
- Examples: $\{a, b\} - \{b, c\} = \{a\}$

  $\{a, b, c, d\} - \{d, a\} = \{b, c\}$

  $\{a, b\} - \{f, g\} = \{a, b\}$

  $\{a, b\} - \emptyset = \{a, b\}$ (universally: $X - \emptyset = X$)
- The ordering of sets matters: $\{a, b\} - \{b, c\} = \{a\}$

  $\{b, c\} - \{a, b\} = \{c\}$
- In union and intersection the ordering of sets does not matter; these are **commutative** operations
- $A \cap B = B \cap A$; $A \cup B = B \cup A$
- Subtraction is not commutative: $A - B \neq B - A$

BASIC SET THEORY

- **Exercise**
- $A = \{a, b, c, d, e\}$; $B = \{a, b, c\}$; $C = \{a, c, d\}$; $D = \{a, c\}$
  - $B \cup D =$
  - $C \cap B =$
  - $D - A =$
  - $A - D =$
  - $C \cap \emptyset =$
  - $D \cup \emptyset =$

BASIC SET THEORY

- **Exercise**
- A = {a, b, c, d, e}; B = {a, b, c}; C = {a, c, d}; D = {a, c}
  - $B \cup D =$ **{a, b, c}**
  - $C \cap B =$ **{a, c}**
  - $D - A =$ **{ } or Ø**
  - $A - D =$ **{b, d, e}**
  - $C \cap \emptyset =$ **{ } or Ø**
  - $D \cup \emptyset =$ **{a, c}**

LOGICAL PHONOLOGY

- Logical Phonology (LP) uses set theory to redefine various aspects of phonological representations and computations
- Phonological **segments** are consistent and potentially 'incomplete' sets of valued (binary) features
  - s = {+CONS, +COR, +ANT, +CONT, –SON, –VOICED … } (ellipsis means we assume all other features are present)
  - ʒ = {+CONS, +COR, –ANT, +CONT, –SON, +VOICED … }
  - ɪ = {+HIGH, –LOW, –BACK, –ROUND, –ATR …} (we assume vowels and consonants contain the same features)
  - u = {+HIGH, –LOW, +BACK, +ROUND, +ATR …}
- **Consistency**
  - For a given feature F, a segment cannot contain both –F and +F
  - The set {–VOICED, +VOICED} is inconsistent
  - Inconsistent segments are phonologically impossible; any operation that would yield them is **undefined** (it gives an unchanged output)

LOGICAL PHONOLOGY

- **Underspecification**
  - If a segment lacks one or more valued feature(s), then such a segment is underspecified
  - Fully specified: p = {–VOI, –SON, –COR, +ANT, +LAB, –NAS, –CONT}

    b = {+VOI, –SON, –COR, +ANT, +LAB, –NAS, –CONT}
  - Underspecified: B = {–SON, –COR, +ANT, +LAB, –NAS, –CONT} (crucially, no VOI feature!)
  - An absence of a feature is not the same as a third value (e.g., 0VOI): **an absence of a feature cannot be targeted by a rule**, while a third value could be targeted
  - E.g., B cannot be targeted without p and b also being targeted (B is a proper subset of p and b, so any natural class that contains both p and b will also contain B)
  - This seems to be in line with empirical data: underspecified segments never seem to make a natural class on their own, to the exclusion of their fully specified counterparts
  - Underspecification simplifies phonology (it's automatically generated by subtraction) and makes it more powerful (there are more possible segments)

LOGICAL PHONOLOGY

- **Natural class**
  - A set of segments (a set of sets of features); the segments of a natural class all behave alike with respect to some phonological process
  - A set of all and only the segments that contain a given set of valued features
  - Every natural class has a name (label), which is stated in **square brackets**
  - Example: **[+NAS]** = {m, n, ŋ} (supposing those are the only nasals in the phoneme inventory)
  - The name/label are <u>all</u> the features that are shared between the members of a class; it is determined by **generalized intersection**:

$$\bigcap \left\{ \begin{Bmatrix} \text{m} \\ +\textsc{Son} \\ -\textsc{Cont} \\ +\textsc{Voi} \\ +\textsc{Ant} \\ +\textsc{Lab} \\ -\textsc{Cor} \\ +\textsc{Nas} \\ \vdots \end{Bmatrix}, \begin{Bmatrix} \text{n} \\ +\textsc{Son} \\ -\textsc{Cont} \\ +\textsc{Voi} \\ +\textsc{Ant} \\ -\textsc{Lab} \\ +\textsc{Cor} \\ +\textsc{Nas} \\ \vdots \end{Bmatrix}, \begin{Bmatrix} \text{ŋ} \\ +\textsc{Son} \\ -\textsc{Cont} \\ +\textsc{Voi} \\ +\textsc{Ant} \\ -\textsc{Lab} \\ -\textsc{Cor} \\ +\textsc{Nas} \\ \vdots \end{Bmatrix} \right\} = \begin{Bmatrix} +\textsc{Son} \\ -\textsc{Cont} \\ +\textsc{Voi} \\ +\textsc{Nas} \\ \vdots \end{Bmatrix}$$

LOGICAL PHONOLOGY

- **Natural class**
  - Technically, a natural class is defined by the largest set of features that is shared by all of the members of that class; but in practice (merely for the sake of brevity), we only state a minimally sufficient set of shared features
  - Actual natural class of nasals: {m, n, ŋ} = [+VOI, +SON, +NAS, –CONT, –STRI ...]
  - Simplified natural class of nasals: {m, n, ŋ} = [+NAS]
  - Intensional vs. extensional definition of a set
  - Don't confuse a natural class (set of segments) with its label/name/definition (all of the shared features among member-sets)!
    - Every segment of a NC is a subset of that NC, but every segment of NC is a superset of the label of that NC
  - The fewer the members in a natural class, the more restricted its definition

LOGICAL PHONOLOGY

- **Natural class**
  - Why are natural classes important in phonology? LP makes a strong claim: **phonological rules can only be stated with natural classes as their targets and triggering environments**
  - Universal format of a rule: **[ ] → { } / [ ] __ [ ]**
    - Structural positions (syllable boundary, coda, etc.) may also be in the environment
  - A phonological description / generalization that is not stated in these terms is not a (single) phonological rule
  - Is the statement 's becomes ʃ after r, u, k, i' a phonological rule?
  - If {r, u, k, i} cannot be extracted from the inventory by generalized intersection of features to the exclusion of all other segments, then the above statement is <u>not</u> a phonological rule

LOGICAL PHONOLOGY

- **Exercise**
- Given the vowel inventory below, determine the members and natural classes in (1) to (5)

i	ɨ	u
e		o
	a	ɑ

(1) [+ROUND] =

(2) [+BACK, –ROUND] =

(3) [–HIGH] =

(4) Define the natural class that contains only the members i and e.

(5) Define the natural class that contains only the member o.

(6) Define the natural class that contains only the members e and u.

LOGICAL PHONOLOGY

- **Exercise**
- Given the vowel inventory below, determine the members and natural classes in (1) to (5)

i	ɨ	u
e		o
	a	ɑ

(1) [+ROUND] = **{u, o}**

(2) [+BACK, –ROUND] = **{ɨ, a, ɑ}**

(3) [–HIGH] = **{e, o, a, ɑ}**

(4) Define the natural class that contains only the members i and e. **[–BACK] = {i, e}**

(5) Define the natural class that contains only the member o. **[+ROUND, –HIGH] = {o}**

(6) Define the natural class that contains only the members e and u. **{e, u} cannot be defined as a natural class**

LOGICAL PHONOLOGY

- Interlude
- Using logical reasoning – instead of functionalist reasoning or just intuition – in determining underlying representations (URs) on the basis of surface data
  - Note that the following types of reasoning are analytical methods and not components of phonological competence
  - In fancy terms: these are part of our epistemological toolkit, not claims about the ontology of phonology
  - Comparison and subtraction, Non-alternation Assumption, Surfacing Underlying Form Assumption, Modus Tollendo Ponens, and Reductio ad Absurdum

LOGICAL PHONOLOGY

SRs	Meaning
[naču]	‘a dog’
[nači]	‘the dog’
[padu]	‘a foot’
[padi]	‘the foot’
[naku]	‘a hat’
[nači]	’the hat’

- Determining URs from SRs: **comparison** and **subtraction**
- Comparing [naču] and [nači]; comparing [padu] and [padi]
  - [nač] means ‘dog’; [pad] means ‘foot’
- Subtracting [nač] from [naču] and [nači]; subtracting [pad] from [padu] and [padi]
  - [u] means ‘a’; [i] means ‘the’

Decisions about URs

1. /nač/ is the UR for the morpheme meaning ‘dog’
2. /pad/ is the UR for the morpheme meaning ‘foot’
3. /-u/ is the UR for the suffix meaning ‘the’
4. /-i/ is the UR for the suffix meaning ‘a’

- Problem: What is the UR for the word meaning ‘hat’?
- Subtraction of /-i/ and /-u/ gives two options: /nak/ and /nač/
- We should not choose /nak/ for ‘hat’ just because it’s different from ‘dog’ (i.e., in order to avoid homophony)
  - That’s faulty functionalist reasoning

LOGICAL PHONOLOGY

- Some useful tools in deciding about URs (heuristics, not universal principles; they can fail)
- **Non-alternation Assumption** (NA; called the Alternation Condition in older generative literature)
  - If there is one surface form of a morpheme in all environments, then the underlying representation of that morpheme is identical to that surface form
- **Surfacing Underlying Form Assumption** (SUFA)
  - If there is more than one surface form, then the underlying representation of the morpheme is identical to one of those surface forms
- **Modus Tollendo Ponens** (MTP; 'mode that affirms by denying'):
  - For any two propositions p and q, if (p or q) is true and p is false, then it can be concluded that q is true
- **Reductio ad Absurdum** (RA; 'reduction to absurdity'):
  - For any proposition p, if assuming that p is true leads to a contradiction, then it can be concluded that p is false

LOGICAL PHONOLOGY

- NA is implicit in our decisions about /nač/, /pad/, /-i/, and /-u/
- SUFA, MTP and RA can help in deciding about the UR for 'hat'
- SUFA reasoning
  - The UR of the morpheme meaning 'hat' is /nač/ or /nak/
- RA reasoning
  - Assumption p: the UR for 'hat' is /nač/
  - If that were the case, we'd need a /nač/ to [nak] mapping before /-u/ (č → k / __ u)
  - But if /č/ mapped to [k] before /-u/, 'a dog' would surface as [naku], but it does not
  - Assumption p leads to a contradiction; therefore, p is false

LOGICAL PHONOLOGY

- MTP reasoning
  - Premise 1: The UR of the word meaning 'hat' is /nač/ or /nak/ (via SUFA)
  - Premise 2: /nač/ is not the UR for 'hat' (via RA)
  - Conclusion via MTP: Therefore, /nak/ is the UR for 'hat'
- 'k → č / __ i' maps /nak-i/ to [nači]
- We did not appeal to functionalist notions such as homophony avoidance, typology (/k/ is more common than /č/), or phonetics (/k/ is easier to articulate than /č/)
  - If we manage to construct an MTP argument, then its conclusion is **necessarily true**
- SUFA and MTP work well with **neutralization**, but not so well with underspecification (/t/, /d/, /D/)

LOGICAL PHONOLOGY

- In traditional generative phonology, a **phonological rule** is a somewhat loose statement given in this format: A → B / C __ D
- What exactly are the target (A), the change (B), and the environment (C __ D)? What exactly does '→' mean?
- Logical Phonology sharpens the notion of a phonological rule, proposing the following claims about rules
  - 1. The **target** and the **environment** are stated as natural classes
    - The environment can also include reference to a structural position, such as word boundary or coda
  - 2. The **structural change** is stated as a set of features
  - 3. The **arrow** '→' is either set **subtraction** or set **unification**
- The **universal format of rules** in LP – the innate blueprint for constructing particular language-specific rules:

$$[\,] - \{\,\} \,/\, [\,] \_\_ [\,] \qquad \text{or} \qquad [\,] \sqcup \{\,\} \,/\, [\,] \_\_ [\,]$$

LOGICAL PHONOLOGY

- Targets, changes, and environments in LP
- Slovenian data

UR	SR	Meaning
/grob/	[grop]	N. ‘grave’
	cf. G. [groba]	
/grad/	[grat]	N. ‘town’
	cf. G. [grada]	
/vrag/	[vrak]	N. ‘devil’
	cf. G. [vraga]	
/krov/	[krof]	N. ‘roof’
	cf. G. [krova]	
/mraz/	[mras]	N. ‘frost’
	cf. G. [mraza]	

Rules that account for the Slovenian patterns:

b → p / __ #
d → t / __ #
g → k / __ #
v → f / __ #
z → s / __ #

Descriptively adequate, but misses important generalizations...

LOGICAL PHONOLOGY

- The target is a set of segments {b, d, g, v, z}, and each segment is a set of features
- Generalized intersection to determine the natural class label: ∩{b, d, g, v, z} = {+CONS, –SON, +VOI, ...}
- The smallest set of shared features that unambiguously define the correct natural class is {–SON, +VOI}
- Target: [–SON, +VOI] = {b, d, g, v, z}
  - We will dispense with VOI when we introduce unification, simplifying the analysis even further

- Rule that captures the Slovenian patterns (first approximation):

[–SON, +VOI] → {–VOI} / __ #

- We will return to complete the analysis of Slovenian once we take care of the arrow '→'

LOGICAL PHONOLOGY

- LP argues that the arrow '→' needs to be replaced with more clearly defined set-theoretic operations
- Conceptual argument: In the interpretation 'A becomes B', the notion of 'becoming' is unclear
  - How does it work: 'remove a feature, then add a feature', 'add a feature, then remove a feature', or 'remove and add features simultaneously'?
- Empirical argument: The traditional arrow '→' has at least three different interpretations in the literature
  - Feature **deletion**: Debuccalization of /k/ to [K] (or [ʔ]) in Arbore (Ethiopia) (Hayward 1984; McCarthy 2008)
  - Feature **addition**: Turkish underspecified /D/ filled-in with either +VOI (onset) or –VOI (coda) (Inkelas & Orgun 1995; Bale et al. 2014)
  - Feature **replacement**: Slovenian word-final devoicing

LOGICAL PHONOLOGY

- LP proposes that the processes that the arrow '→' captures should be described with **set subtraction** and **set unification**
- Back to Slovenian: [–SON, +VOI] → {–VOI} / __ #
- Set subtraction: [–SON, +VOI] – {+VOI} / __ #
  - Changes all voiced obstruents into segments underspecified for voicing
  - B = {+CONS, –SON, +LAB, –COR… (no VOI feature)}
  - D = {+CONS, –SON, –LAB, +COR… (no VOI feature)}
  - Once +VOI is removed, we need an operation that will insert –VOI

LOGICAL PHONOLOGY

- Set union doesn't work
  - We can't target only underspecified segments, to the exclusion of their fully specified counterparts, and union with fully specified obstruents yields impossible segments
  - Underspecified segments are subsets of their fully specified versions, therefore every natural class that includes them will also include their fully specified version
  - Example $\quad$ b = {–SON, +LAB, +VOI}

    B = {–SON, +LAB} $\quad$ B ⊂ b
  - b ∩ B = {–SON, +LAB}
  - Since b is a more specific version of B, b will be in every natural class that contains B; there is no way to isolate B by intersection
  - If fully specified obstruents are included, union will yield impossible segments:
    b ∪ {–VOI} = {–SON, +LAB, **+VOI, –VOI**}

LOGICAL PHONOLOGY

- **Set unification**
  - If A and B are sets, then A ⊔ B = A ∪ B if A ∪ B is **consistent**. Otherwise, A ⊔ B is undefined
    - An **undefined** operation yields an unchanged output
    - Examples {–SON, +LAB, +VOI} ∪ {–VOI} = {–SON, +LAB, +VOI, –VOI}

      {–SON, +LAB, +VOI} ⊔ {–VOI} = {–SON, +LAB, +VOI}

      {–SON, +LAB} ⊔ {–VOI} = {–SON, +LAB, –VOI}

      {–SON, +LAB, –VOI} ⊔ {–VOI} = {–SON, +LAB, –VOI}
- Back to Slovenian: after +VOI is removed by subtraction, unification inserts –VOI
- [–SON] ⊔ {–VOI} / __ #

LOGICAL PHONOLOGY

- Set unification in Slovenian: [–SON] ⊔ {–VOI} / __ #
  - Notice that VOI cannot be in the target because we have to target the natural class that also includes underspecified segments, which have no VOI
  - Now, the target is more general: it includes <u>all</u> obstruents, both fully specified and underspecified ones
- Full derivation table – summary of Slovenian word-final devoicing in Logical Phonology

Underlying representation	/grob/ 'grave'	/grad/ 'city'	/vrag/ 'devil'	/krov/ 'roof'	/mraz/ 'frost'	/mrak/ 'dark'
Set subtraction [–SON, +VOI] – {+VOI} / __ #	groB	graD	vraG	kroV	mraZ	–
Set unification [–SON] ⊔ {–VOI} / __ #	grop	grat	vrak	krof	mras	–
Surface representation	[grop]	[grat]	[vrak]	[krof]	[mras]	[mrak]

LOGICAL PHONOLOGY

- Assimilation in Logical Phonology makes use of **variables**
  - If a feature is specified in the environment with one value (+ or –), then the same feature in the structural change will have the same value
- Croatian data (assimilation of nasals)

/n/-assimilation

/jedan-put/	→	[jeda**m**put]	‘once’
/t͡ʃin-b-en-ik/	→	[t͡ʃi**m**benik]	‘factor’
/on prat-i/	→	[o**m**prati]	‘he follows’
/inʋalid/	→	[i**ɱ**ʋalid]	‘invalid’
/on ʋid-i/	→	[o**ɱ**ʋidi]	‘he sees’
/kon-form-iz-am/	→	[ko**ɱ**formizam]	‘conformity’
/bank-a/	→	[ba**ŋ**ka]	‘bank’
/kongres/	→	[ko**ŋ**gres]	‘congress’
/inxibir-a-ti/	→	[i**ŋ**xibirati]	‘to inhibit’

/m/-assimilation

/tramʋaj/	→	[traɱʋaj]	‘tram’
/amfor-a/	→	[aɱfora]	‘amphora’
/pitam ʋas/	→	[pitaɱʋas] ‘	I am asking you’
(/iznim-k-a/	→	[iznimka]	‘exception’; *[izniŋka])
(/kamp/	→	[kamp]	‘camp’)

absence of /ɲ/-assimilation

/koɲ bi/ → [koɲbi] ‘horse would’; *[kombi]

/ʋoɲ te mut͡ʃ-i/ → [ʋoɲte mut͡ʃi] ‘smell bothers you’; *[ʋonte mut͡ʃi]

saɲk-e → [saɲke] ‘sled’; *[saŋke]

LOGICAL PHONOLOGY

- Croatian data (nasal place assimilation)
  - Before velars, /n/ also assimilates in continuancy
  - In [ŋk] and [ŋg] clusters, [ŋ] is non-continuant (Figure 1)
  - In [ŋx] clusters, [ŋ] is continuant (Figure 2)

	0	0	0	0	0	0	
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
100	16	0	0	0	0	0	100
100	100	100	50	16	50	100	100

Figure 1. [ŋ] in [baŋka]

	0	0	0	0	0	0	
0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	100
100	16	0	0	0	0	66	100
100	100	0	0	0	0	100	100
100	100	16	0	0	16	100	100
100	100	83	0	0	83	100	100
100	100	0	0	0	33	100	100

Figure 2. [ŋ] in [iŋxibirati]

LOGICAL PHONOLOGY

- Croatian place and continuancy assimilation of nasals
  - Assimilation of /m/ and /n/ to bilabial and labiodental consonants

(S1) [+NAS, +ANT] – {+COR, –LAB, –CONT} /__ [+LAB]

(U1) [+NAS, +ANT] ⊔ {–COR, +LAB, αCONT} / __ [+LAB, αCONT]

  - Assimilation of /n/ to velar consonants

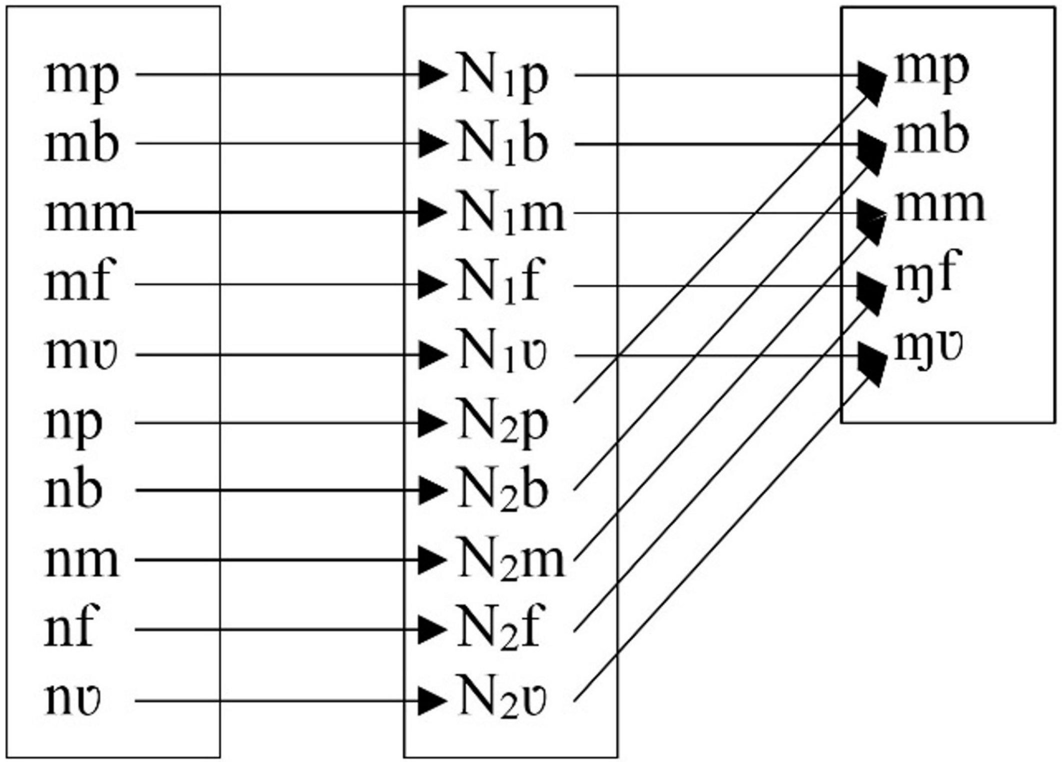
(S2) [+NAS, +ANT, +COR] – {+ANT, +COR, –BACK, –CONT} / __ [–SON, +BACK]

(U2) [+NAS] ⊔ {–ANT, –COR, +BACK, αCONT} / __ [–SON, +BACK, αCONT]

- The use of **α variable** is required to account for assimilation

LOGICAL PHONOLOGY

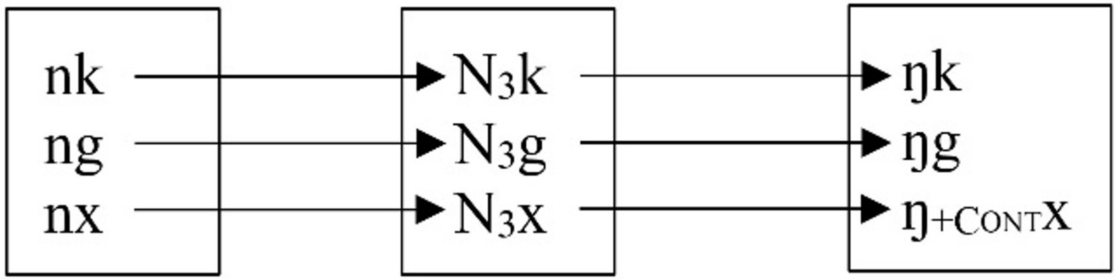
- Croatian place and continuancy assimilation of nasals
  - Effect of S1 and U1



	m	ɱ	n	N_1	N_2	N_3
CONS	+	+	+	+	+	+
SON	+	+	+	+	+	+
COR	–	–	+	–		
ANT	+	+	+	+	+	
LAB	+	+	–	+		–
NAS	+	+	+	+	+	+
CONT	–	+	–			
BACK	–	–	–	–	–	

LOGICAL PHONOLOGY

- Croatian place and continuancy assimilation of nasals
  - Effect of S2 and U2



	m	ɱ	n	N_1	N_2	N_3
Cons	+	+	+	+	+	+
Son	+	+	+	+	+	+
Cor	–	–	+	–		
Ant	+	+	+	+	+	
Lab	+	+	–	+		–
Nas	+	+	+	+	+	+
Cont	–	+	–			
Back	–	–	–	–	–	

LOGICAL PHONOLOGY

- Croatian place and continuancy assimilation of nasals
  - Derivation table

URs	banka	inxibirati	on bi	inʋalid	tramʋaj	iznimka	kamp	saɲke	koɲ bi
(S1)	–	–	oN$_2$bi	iN$_2$ʋalid	traN$_1$ʋaj	–	kaN$_1$p	–	–
(S2)	baN$_3$ka	iN$_3$xibirati	–	–	–	–	–	–	–
(U1)	–	–	ombi	iɱʋalid	traɱʋaj	–	kamp	–	–
(U2)	baŋka	iŋ$_{[+\text{Cont}]}$xibirati	–	–	–	–	–	–	–
SRs	baŋka	iŋ$_{[+\text{Cont}]}$xibirati	ombi	iɱʋalid	traɱʋaj	iznimka	kamp	saɲke	koɲbi
Gloss	‘bank’	‘to inhibit’	‘he would’	‘invalid’	‘tram’	‘exception’	‘camp’	‘sled’	‘horse would’

LOGICAL PHONOLOGY

- Summary of the main principles of **Logical Phonology**
  - The object of study is phonological competence, which is part of I-language.
  - Phonology is free of phonetic substance.
  - There is no language-specific phonetics. All sound patterns of language that are encoded in the mind are phonology.
  - Segments are consistent and potentially underspecified sets of valued binary features.
  - Natural classes are sets of segments, defined by generalized intersection.
  - Rules are set-theoretic operations: set subtraction and set unification. Their targets and environments are stated as natural classes.
  - The features and the general format of rules are innate – they are part of Universal Grammar.