Using Model Theory and Function Types to Formally Relate Symbolic and Dynamic Theories of Speech

DYMOS: Dynamical Models of Speech

Scott Nelson

University of Illinois Urbana-Champaign

July 26, 2025

Download the slides!

Motivation for this Talk

- From the CFP:

  > …there remain a range of unresolved issues and critiques, including the nature of timing control, the relations between symbolics and dynamics, the emergence of phonological structure, and what counts as a necessary condition of a dynamical theory…

- I explore some of these issues in my dissertation, *The Computational Structure of Phonological and Phonetic Knowledge*, and related work using techniques from the theory of computation.

Nelson (2024); Nelson and Heinz (in press)

Why Abstraction?

One of the aims and advantages of abstraction is to make connections between different situations that might previously have seemed very different. It might seem that abstraction takes us further away from "real" situations. This is superficially true, but at the same time abstraction enables us to make connections between situations that are further apart from one another.

Cheng (2022) - *The joy of abstraction: An exploration of math, category theory, and life*

Overview

1. Formally Relating Static Symbols and Dynamic Gestures
2. Describing Grammatical Architecture and Control Structure with Typed Functions

Formally Relating Static Symbols and Dynamic Gestures

Section Outline

1. Model-Theoretic Phonology
2. Articulatory Phonology and gestural representations
3. Defining a bi-directional translation between segments and coupling graphs
4. English past-tense alternation example

Key Idea

I show how to translate between symbolic representations of lexical items and gestural representations of lexical items using first-order logic. These translations can also be used to discuss transformations and therefore provide a way to directly put categorical symbolic alternations into the language of Articulatory Phonology.

Computational *Equivalency*

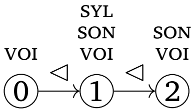
- Mathematical logic and model theory provide a tool for determining the expressivity of different types of phonological knowledge.
  - More powerful logic $\rightarrow$ more expressive pattern/more computational power needed

If one representation scheme can be interpreted in terms of another representation scheme with a certain type of logic, they make the same distinctions within the bounds of the expressivity of that logic.

Rogers et al. (2010, 2013); Strother-Garcia (2019); Oakden (2020)

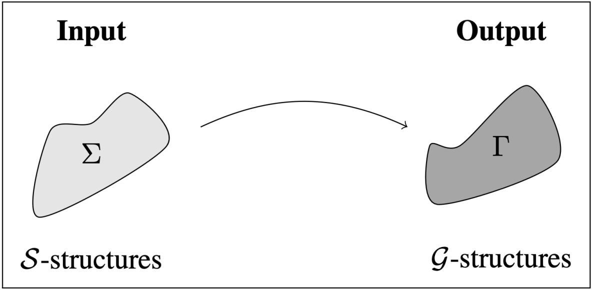
Model Theory - Structure Building

- A *model signature* determines what properties elements in a structure can have and how the elements are related.
  - Basic "feature" signature: $\langle \lhd, \{R_f \mid f \in \mathcal{F}\}\rangle$

- A *structure* is the result of denoting which properties of a signature hold for a given domain $\mathcal{D}$.
  - 

- All structures that satisfy a set of constraints $T = \phi_1, \phi_2, \ldots$ are the *model* of T.
  - $\phi_1 := \exists x[\textsc{Syllabic}(x)]$

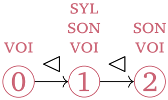
Model Theory - Relating Representations



- We interpret a structure $\mathcal{S}$ in signature Σ as a structure $\mathcal{G}$ in signature Γ.
- This gives us a function from $\mathcal{G}$-structures to $\mathcal{S}$-structures.

Czarnecki (2025)

Model Theory - Relating Representations



$$\phi_{\texttt{voi}}(x) \overset{\text{def}}{=} d(x) \vee n(x) \vee a(x)$$

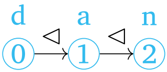
$$\phi_{\texttt{son}}(x) \overset{\text{def}}{=} n(x) \vee a(x)$$

$$\phi_{\texttt{syl}}(x) \overset{\text{def}}{=} a(x)$$

$$\phi_d(x) \overset{\text{def}}{=} \texttt{voi}(x) \wedge \neg\texttt{son}(x) \wedge \neg\texttt{syl}(x)$$

$$\phi_n(x) \overset{\text{def}}{=} \texttt{voi}(x) \wedge \texttt{son}(x) \wedge \neg\texttt{syl}(x)$$

$$\phi_a(x) \overset{\text{def}}{=} \texttt{voi}(x) \wedge \texttt{son}(x) \wedge \texttt{syl}(x)$$



Nelson (2022)

Gestural Representations

- Articulatory Phonology is (primarily) a mono-stratal theory of phonology based around gestural representations.

- Gestures are tasks/goals that are defined via tract variables:
  - Constriction Location: y-dimension; labeled with landmarks like “alveolar”, “velar”, ...
  - Constriction Degree: x-dimension; labeled with degrees like “closed”, “critical”, ...

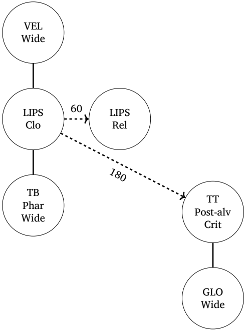
Browman and Goldstein (1986, 1995, *inter alia*)

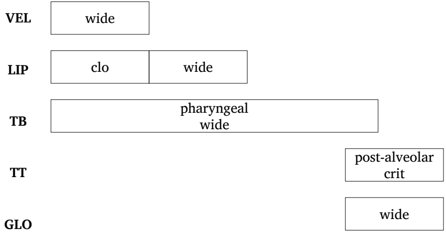
Coupling Graphs as Lexical Representations

"In this model, planning oscillators associated with the set of gestures in a given utterance are coupled in a pairwise, bidirectional manner specified in a coupling graph (or structure) that is part of **the lexical specification of a word**" *(p. 38).*

Nam (2007)

Coupling Graphs as Lexical Representations





- Representations for *mash*
  - Coupling Graph (left)
  - Gestural score (right; automatically determined by left)

Nam (2007)

Coupling Graph Model Signature

Relation	Label
$\diamond$	In-phase
$\lhd_{180}$	Anti-phase
$\lhd_{60}$	Abutting
$\lhd_{30}$	Eccentric

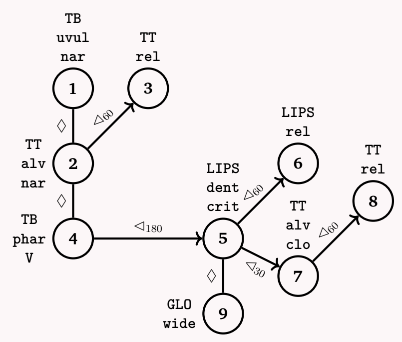
- 4 binary relations based on some phase relations in Articulatory Phonology

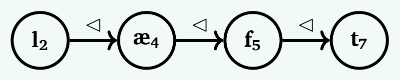
Coupling Graph Model Signature

Relation	Label	Relation	Label
`LIPS`	Labial Articulator	`rel`	Constriction Degree: release
`TT`	Tongue Tip Articulator	`pro`	Constriction Location: protruded
`TB`	Tongue Body Articulator	`dent`	Constriction Location: dental
`VEL`	Velum Articulator	`alv`	Constriction Location: alveolar
`GLO`	Glottis Articulator	`palv`	Constriction Location: postalveolar
`clo`	Constriction Degree: closed	`pal`	Constriction Location: palatal
`crit`	Constriction Degree: critical	`vel`	Constriction Location: velar
`nar`	Constriction Degree: narrow	`uvul`	Constriction Location: uvular
`V`	Constriction Degree: vowel	`phar`	Constriction Location: pharyngeal
`wide`	Constriction Degree: wide		

- Unary labeling relations.

Structures for *laughed*

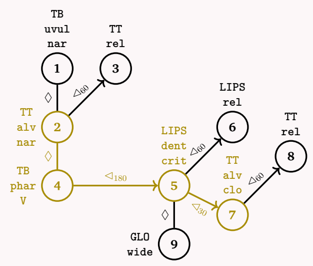




Translating between Structures

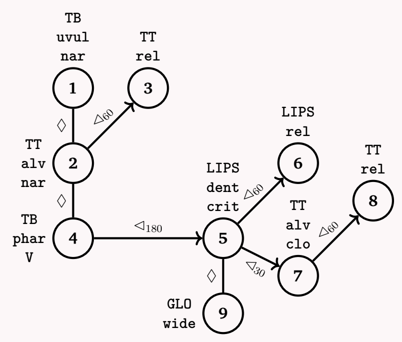
- Using the model-theoretic techniques previously described, I show how to relate these coupling graph representations with segmental strings and vice versa.

The key insight that allows for determining a string from a coupling graph is the notion of a "spine" which is made of what have been called "head gestures" (Gafos, 2002; Smith, 2018).

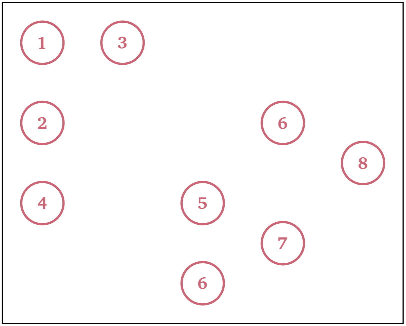


Translating from coupling graph to string

Input



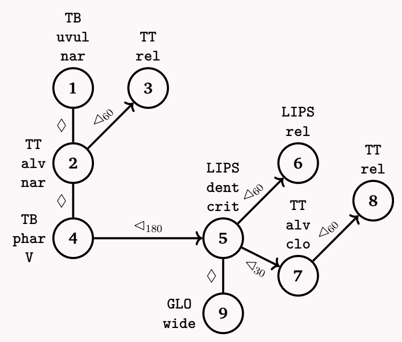
Workspace



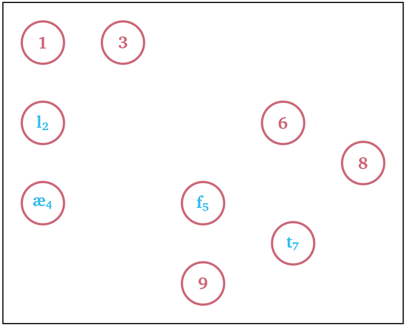
- $C := \{1\}$

Translating from coupling graph to string

Input



Workspace



$$\varphi_{\mathrm{l}}(x) \stackrel{\text{def}}{=} \texttt{TT}(x) \wedge \texttt{alv}(x) \wedge \texttt{nar}(x) \wedge \exists y[x \Diamond y \wedge \texttt{TB}(x) \wedge \texttt{uvul}(x) \wedge \texttt{nar}(x)]$$

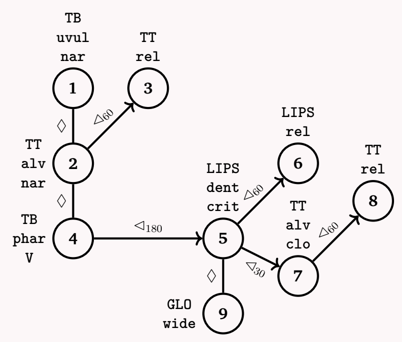
$$\varphi_{\text{æ}}(x) \stackrel{\text{def}}{=} \texttt{TB}(x) \wedge \texttt{phar}(x) \wedge \texttt{V}(x)$$

$$\varphi_{\mathrm{f}}(x) \stackrel{\text{def}}{=} \texttt{LIPS}(x) \wedge \texttt{dent}(x) \wedge \exists y[x \Diamond y \wedge \texttt{GLO}(y)]$$

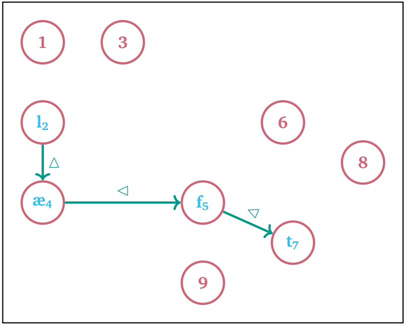
$$\varphi_{\mathrm{t}}(x) \stackrel{\text{def}}{=} \texttt{TT}(x) \wedge \texttt{alv}(x) \wedge \texttt{clo}(x) \wedge \exists y[x \Diamond y \wedge \texttt{GLO}(y)]$$

Translating from coupling graph to string

Input



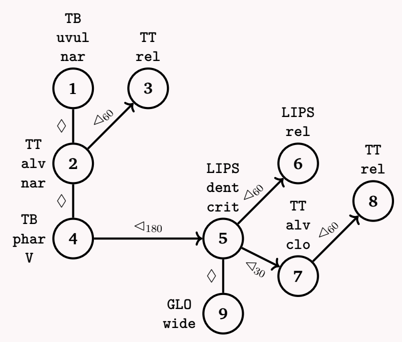
Workspace



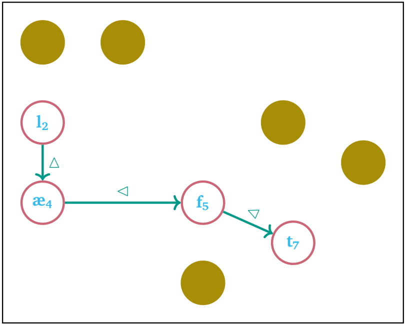
$$\varphi_{\lhd}(x,y) \overset{\text{def}}{=} (x \lhd_{180} y) \vee (x \lhd_{30} y) \vee (x \Diamond y \wedge \mathtt{V}(y) \wedge \neg \exists z[x \lhd_{180} z])$$

Translating from coupling graph to string

Input



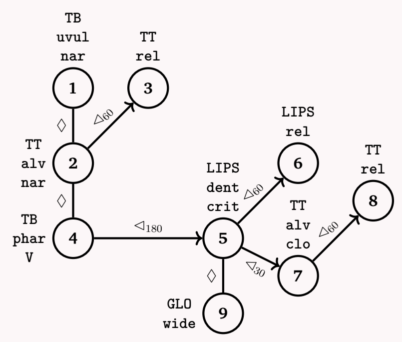
Workspace



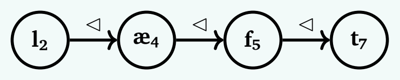
$$\varphi_{license}(x) \overset{\text{def}}{=} \neg\mathtt{rel}(x) \wedge \neg\mathtt{GLO}(x) \wedge \neg\mathtt{VEL}(x) \neg \boxed{\mathtt{SecArc}} (x)$$

Translating from coupling graph to string

Input



Output



Expansion

- Going from coupling graph to string removes information. What happens when we have to expand the representation and add more information by going from a string to a coupling graph?

Expansion

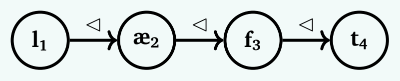
- Here, there is no issue. In fact, the TADA software which implements the Articulatory Phonology system has a program called GEST which turns strings into coupling graphs.

- GEST is written in Perl which is an imperative programing language which lists the steps that must be taken to solve the problem.

The logical transductions are equivalent to logical programs which provide declarative statements on the specs of the program. They tell us what properties must hold of any program solving the problem.

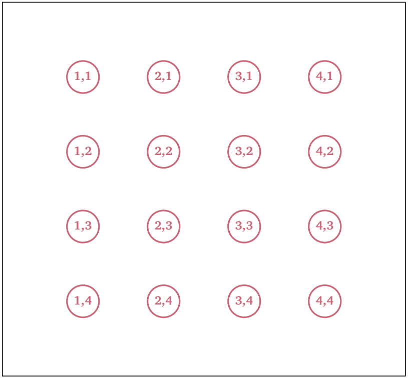
Translating from string to coupling graph

Input



- $C := \{1, 2, 3, 4\}$

Workspace



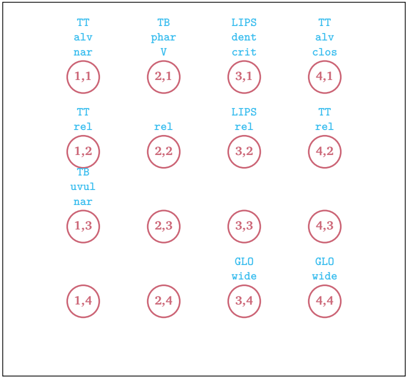
Translating from string to coupling graph

Input

$$\mathrm{l}_1 \xrightarrow{\lhd} \text{æ}_2 \xrightarrow{\lhd} \mathrm{f}_3 \xrightarrow{\lhd} \mathrm{t}_4$$

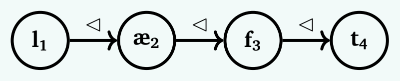
$$\varphi^1_{\texttt{LIPS}}(x) := \mathrm{f}(x)$$
$$\varphi^2_{\texttt{LIPS}}(x) := \varphi^1_{\texttt{LIPS}}(x)$$
$$\varphi^1_{\texttt{TT}}(x) := \mathrm{t}(x) \lor \mathrm{l}(x)$$
$$\varphi^2_{\texttt{TT}}(x) := \varphi^1_{\texttt{TT}}(x)$$
$$\varphi^1_{\texttt{TB}}(x) := \text{æ}(x)$$
$$\varphi^3_{\texttt{TB}}(x) := \mathrm{l}(x)$$
$$\varphi^4_{\texttt{GLO}}(x) := \mathrm{t}(x) \lor \mathrm{f}(x)$$
$$\varphi^1_{\texttt{dent}} := \mathrm{f}(x)$$
$$\varphi^1_{\texttt{alv}} := \mathrm{t}(x)$$
$$\varphi^1_{\texttt{phar}} := \text{æ}(x)$$
$$\varphi^3_{\texttt{uvul}} := \mathrm{l}(x)$$
$$\varphi^1_{\texttt{clo}} := \mathrm{t}(x)$$
$$\varphi^1_{\texttt{crit}} := \mathrm{f}(x)$$
$$\varphi^1_{\texttt{V}} := \text{æ}(x)$$
$$\varphi^1_{\texttt{nar}} := \mathrm{l}(x)$$
$$\varphi^3_{\texttt{nar}} := \mathrm{l}(x)$$
$$\varphi^4_{\texttt{wide}} := \mathrm{t}(x) \lor \mathrm{f}(x)$$

Workspace



Translating from string to coupling graph

Input



$$\varphi^{1,1}_{\diamond}(x,y) := x \lhd y \wedge \text{æ}(y) \wedge \neg\text{æ}(x)$$

$$\varphi^{1,3}_{\diamond}(x,y) := (x = y) \wedge \mathrm{l}(x)$$

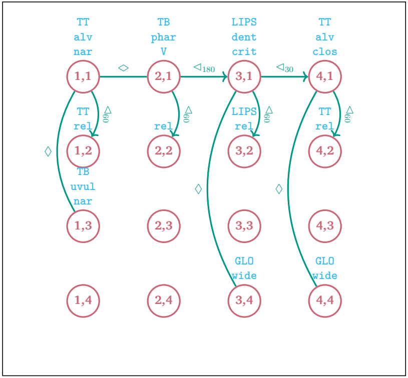
$$\varphi^{1,4}_{\diamond}(x,y) := (x = y) \wedge \mathrm{t}(x) \vee \mathrm{f}(x)$$

$$\varphi^{1,1}_{\lhd 180}(x,y) := x \lhd y \wedge \text{æ}(x) \wedge \neg\text{æ}(y)$$

$$\varphi^{1,2}_{\lhd 60}(x,y) := (x = y)$$

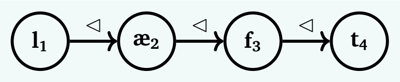
$$\varphi^{1,1}_{\lhd 30}(x,y) := \neg\text{æ}(x) \wedge \neg\text{æ}(y) \wedge \exists z[z \lhd x \wedge \text{æ}(z)]$$

Workspace



Translating from string to coupling graph

Input



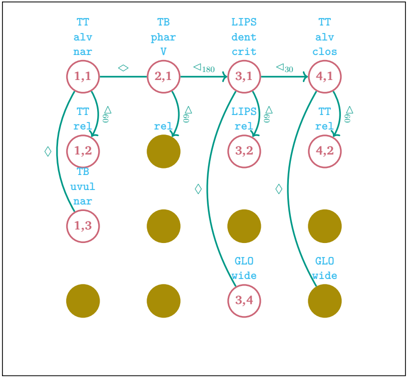
$$\varphi^{1}_{license}(x) := \texttt{True}$$

$$\varphi^{2}_{license}(x) := \mathrm{f}(x) \vee \mathrm{t}(x) \vee \mathrm{l}(x)$$

$$\varphi^{3}_{license}(x) := \mathrm{l}(x)$$

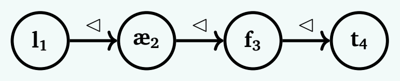
$$\varphi^{4}_{license}(x) := \mathrm{t}(x) \vee \mathrm{f}(x) \wedge \neg\exists y[y \lhd x \wedge \mathrm{f}(y) \vee \mathrm{t}(y)]$$

Workspace

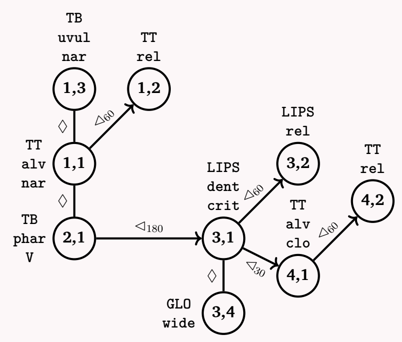


Translating from string to coupling graph

Input



Output



What now?

- “At present, there is no real consensus on how to deal with categorical alternations in AP” (Hall, 2010).

We can use the equations from this section to translate between constraints and transformations. These provide a way to embed these types of analyses directly into the language of Articulatory Phonology and vice versa.

- I’ll use the example of the English past-tense alternation (specifically determining when epenthesis occurs).

Model Theory - Phonological Transformations

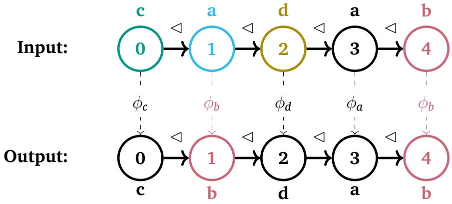
$$a \rightarrow b/c\_d$$

$$\phi_a(x) \stackrel{\text{def}}{=} a(x) \wedge \neg\exists y, z[y \lhd x \lhd z \wedge c(y) \wedge d(z)]$$

$$\phi_b(x) \stackrel{\text{def}}{=} b(x) \vee (a(x) \wedge \exists y, z[y \lhd x \lhd z \wedge c(y) \wedge d(z)])$$

$$\phi_c(x) \stackrel{\text{def}}{=} c(x)$$

$$\phi_d(x) \stackrel{\text{def}}{=} d(x)$$



Theoretical Accounts of the Past Tense Alternations

- Rules
  - $\emptyset \rightarrow \text{ɨ}/ \begin{bmatrix} -\texttt{son} \\ -\texttt{cont} \\ +\texttt{cor} \end{bmatrix} \_ \begin{bmatrix} -\texttt{son} \\ -\texttt{cont} \\ +\texttt{cor} \end{bmatrix}$
  - $[-\texttt{son}] \rightarrow [-\texttt{voi}]/[-\texttt{voi}]\_\#$

Theoretical Accounts of the Past Tense Alternations

- Optimality Theory
  - NoGem, Agree(Voice) $\gg$ Dep(V) $\gg$ Ident(Voice)

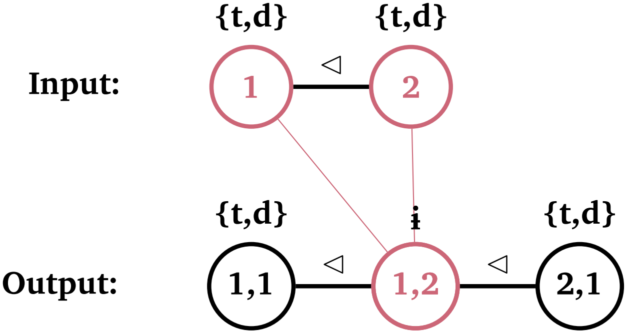
Baković (2005)

- Coupling Graphs
  - Couple the TT closure gesture of the suffix coupling graph anti-phase to a stem-final vowel constriction gesture.
  - Couple the TT closure gesture of the suffix eccentric phase to a stem-final non-coronal obstruent constriction gesture.
  - Couple the TT closure gesture of the suffix eccentric phase to a stem-final coronal obstruent release gesture.

Goldstein (2011)

Rules and English Past Tense Epenthesis

$$\varphi^2_{\text{ɨ}}(x) \overset{\text{def}}{=} (\mathrm{t}(x) \vee \mathrm{d}(x)) \wedge \exists y[x \lhd y \wedge \mathrm{t}(y) \vee \mathrm{d}(y)]$$



- “if there is a sequence of two coronals, then on the output insert a [ɨ] between those coronals.”

OT and English Past Tense Epenthesis

- The constraint ranking essentially gets us an `IF...THEN...ELSE` statement.

If there is a sequence of two coronal plosives on the input (violating NoGem), then changing the voicing valuation will not make a difference in harmony.The only way to salvage a candidate is by epenthesizing due to the ranking.

- $\varphi^2_{\text{ɨ}}(x) := \texttt{IF cor}(x) \wedge \texttt{cor}(\texttt{s}(\texttt{x}))\ \texttt{THEN true ELSE false}$

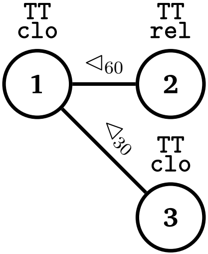
- In the FO language used previously, this is equivalent to the rule formula.

Bhaskar et al. (2020); Chandlee and Jardine (2021); Bhaskar et al. (2023)

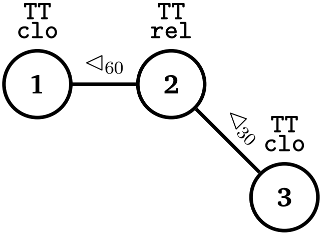
Dynamic Grammar and English Past Tense Epenthesis

$$\varphi_{\lhd_{30}}(x,y) \stackrel{\text{def}}{=} \mathtt{TT}(y) \wedge \mathtt{clo}(y) \wedge \exists z[z \lhd_{30} y \wedge \mathtt{TT}(z) \wedge \mathtt{clo}(z) \wedge z \lhd_{60} x]$$

Input:



Output:



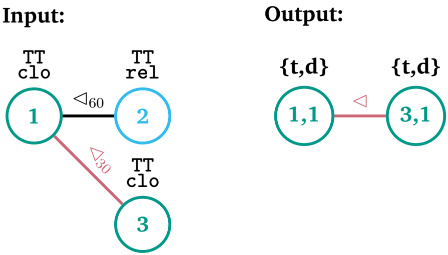
- If we abstract just a little bit, the extension of the Dynamic Grammar includes pairs where, if a given substructure exists in an input graph, then the output graph changes the coupling structure such that epenthesis occurs.

Dynamic Grammar and English Past Tense Epenthesis

$$\varphi_{\triangleleft}(x,y) \overset{\text{def}}{=} (x \triangleleft_{180} y) \vee (x \triangleleft_{30} y) \vee (x \diamond y \wedge \mathtt{V}(y) \wedge \neg\exists z[x \triangleleft_{180} z])$$

$$\varphi_{license}(x) \overset{\text{def}}{=} \neg\mathtt{rel}(x) \wedge \neg\mathtt{GLO}(x) \wedge \neg\mathtt{VEL}(x) \neg \boxed{\mathtt{SecArc}}\ (x)$$

$$\varphi_{\mathrm{t,d}}(x) \overset{\text{def}}{=} \mathtt{TT}(x) \wedge \mathtt{clo}(x)$$



- The substructure that drives epenthesis in the Dynamic Grammar is equivalent to the substructure driving epenthesis in the Rule Grammar.

Local Summary

- Model theory provides an abstract way to directly translate, and therefore make connections, between symbolic strings and coupling graphs. FO logic is relatively restricted (and we can probably go lower in expressive power...)

- It appears that any theory of phonology needs a way to change basic structure when new morphemes are added. If we think about symbolic phonology as operating over equivalence classes of coupling graph structure then the gap between theories narrows.

- Still plenty of technical stuff to work out:
  - Variation, long-distance phenomena, coupling strength, stiffness, ...

Describing Grammatical Architecture and Control Structure with Typed Functions

Section Outline

1. Thinking in Functions
2. Moving beyond the “Modular Feed-Forward” View
3. Rethinking Gestural Hiding + Reduction

Key Idea

I show how to use function types to rethink about the way that certain types of information interact within a system. This view reveals that qualitative structure rather than quantitative implementations can explain a good chunk of phenomena. From this, it follows that the symbolic vs. dynamic distinction can be subtle and may rely at times on auxiliary assumptions.

Computational *Structure*

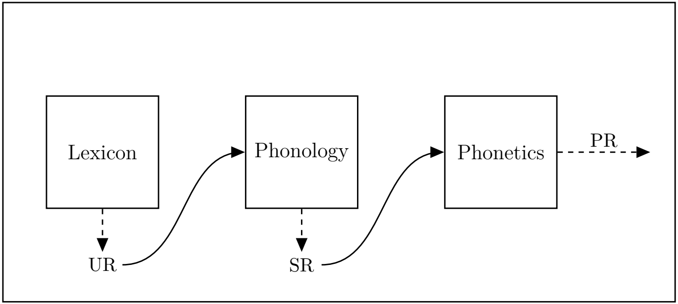
- Typed functions, a generalization of the λ-calculus (a universal model of computation) can bring together two definitions of computation.

- $A :: B \rightarrow C$ describes a function A that maps B type things to C type things.

- An information processing device (A) takes some information (B) and processes it into something new (C).

Sipser (2013); Marr (1982); Pierce (2002); Church (1932, 1933)

Modular Feed-Forward Model



- The common view of the production process in generative linguistics is often referred to as the modular feed-forward model.

Pierrehumbert (2002); Bermúdez-Otero (2007); Kenstowicz (2010)

- Phonology is a function that maps *UR*'s to *SR*'s ($P :: UR \rightarrow SR$)
- Phonetics is a function that maps *SR*'s to *PR*'s ($A_{\text{MFF}} :: SR \rightarrow PR$)

Roark and Sproat (2007), Heinz (2018)

Modular Feed-Forward Model Redux

- Since phonology has the type $UR \rightarrow SR$ and phonetics has the type $SR \rightarrow PR$, the Modular Feed-Forward Model is a composed function $MFF :: UR \rightarrow PR$.
  - $A_{\text{MFF}}(P(UR)) = PR$

$$\begin{tikzcd} UR \arrow[r, "P"] \arrow[rr, bend left, "A_{\text{MFF}}"] & SR \arrow[r, "A"] & PR \end{tikzcd}$$

Moving beyond the MFF Model

- Some problems for the MFF Model:
  - Incomplete Neutralization
  - Variation in Homophone Duration
  - Word Specific Effects
  - …
- Solution from Dynamics (Gafos and Benus, 2006):
  - ...it is both necessary and promising to do away with the metaphor of precedence between the qualitative phonology and the quantitative phonetics, without losing sight of the essential distinction between the two...

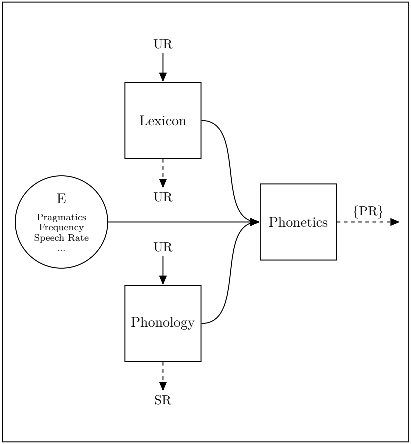
Port and Leary (2005); Gahl (2008); Pierrehumbert (2002)

An Abstract Solution

What we continue to stress in this paper is that language production involves the interaction of lexical, phonological, and extra-grammatical factors which the modular feedforward model fails to capture. Since this idea is able to be expressed using different types of mathematical formalisms, we believe that this idea is not a property of the specific mathematical implementation, but rather a property of the high-level architecture (a "model" in our terms).

Nelson and Heinz (in press) - *The Blueprint Model of Production*

The Blueprint Model of Production



Nelson and Heinz (in press)

The Blueprint Model of Production

- The inputs to the phonetics function are the lexicon and phonology functions, as well as extra-grammatical information that influences production.
  - The lexicon is a function with type $L :: UR \rightarrow UR$.
  - Phonology is *still* a function with type $P :: UR \rightarrow SR$.
  - E is a cover type for extra-grammatical information.

- When a function takes a function as an input it is called a **higher-order function**.

- The phonetics function in the Blueprint Model of Production has type $A_{\mathrm{BP}} :: L \rightarrow P \rightarrow E \rightarrow \{PR\}$.

Simulating Empirical Phenomena

- Simulations show that the Blueprint Model of Production can produce systematic gradient phonetic effects with discrete phonological knowledge.
  - Final Devoicing in German
  - Tonal Near Merger in Cantonese
  - Epenthesis in Lebanese Arabic
  - Homophone duration variation in English

Nelson and Heinz (in press)

Reconsidering Gestural Phenomena

- Gestural Hiding
  - *perfect memory, must be, ground pressure*
  - Magnitude of alveolar gesture similar between "deleted" and present forms.
  - Weak evidence for output being a blend of lexical + phonological forms (with deletion rule)

- Gestural Reduction
  - *late calls*
  - Magnitude of alveolar gesture reduced between "deleted" and present forms.
  - Stronger evidence for output being a blend of lexical + phonological forms (with deletion rule)

Browman and Goldstein (1990); Beckman (1996); Nolan (1992)

Local Summary

- The assumed structure of a system influences how data are interpreted.
  - New structure = possible new interpretation of old data
- The Blueprint Model of Production shows that we can tweak one parameter (structure of interface) and keep another unchanged (discrete phonological knowledge) to explain thought-to-be-problematic phenomena.
  - What are core assumptions of a symbolic/generative theory?
- Again, abstraction allows us to make connections in ways that may not seem obvious.
  - Philosophical vs. empirical justification for one type of theory over the other?

Conclusion

Final Thoughts

- Abstract analyses help identify which aspects of theories are necessary properties to account for speech and language data.
- These can be thought of as complementary approach to the continuing work that shows how dynamic models of speech sufficiently capture various empirical phenomena.
- What counts as a necessary condition [in favor] of a dynamical theory?
  - Any condition which can only be explained directly by dynamics and not through an auxiliary assumption.

THANK YOU

"Mathematical models with their equations and proofs and computational models with their programs and simulations provide different and important windows of insight into the phenomena at hand. In the first, one constructs idealized and simplified models but one can now reason about the behavior of such models and therefore be very sure of one's conclusions. In the second, one constructs more realistic models but because of the complexity, one will need to resort to heuristic arguments and simulations. In summary, for mathematical models the assumptions are more questionable but the conclusions are more reliable – for computational models, the assumptions are more believable but the conclusions more suspect." - Niyogi (2006, pp. 38–39)

Bibliography I

Baković, E. (2005). Antigemination, assimilation and the determination of identity. Phonology, 22(3):279–315.

Beckman, M. E. (1996). When is a syllable not a syllable. Phonological structure and language processing: Cross-linguistic studies, pages 95–123.

Bermúdez-Otero, R. (2007). Diachronic phonology. In de Lacy, P., editor, The Cambridge handbook of phonology, pages 497–517. Cambridge.

Bhaskar, S., Chandlee, J., and Jardine, A. (2023). Rational functions via recursive schemes. arXiv preprint arXiv:2302.03074.

Bhaskar, S., Chandlee, J., Jardine, A., and Oakden, C. (2020). Boolean monadic recursive schemes as a logical characterization of the subsequential functions. In Language and Automata Theory and Applications: 14th International Conference, pages 157–169. Springer.

Browman, C. P. and Goldstein, L. (1990). Tiers in articulatory phonology, with some implications for casual speech. Papers in laboratory phonology I: Between the grammar and physics of speech, pages 341–376.

Browman, C. P. and Goldstein, L. M. (1986). Towards an articulatory phonology. Phonology, 3:219–252.

Browman, C. P. and Goldstein, L. M. (1995). Dynamics and articulatory phonology. In Mind as Motion, chapter 7, pages 175–193. MIT Press.

Chandlee, J. and Jardine, A. (2021). Computational universals in linguistic theory: Using recursive programs for phonological analysis. Language, 97(3):485–519.

Cheng, E. (2022). The joy of abstraction: An exploration of math, category theory, and life. Cambridge University Press.

Church, A. (1932). A set of postulates for the foundation of logic. Annals of mathematics, pages 346–366.

Church, A. (1933). A set of postulates for the foundation of logic. Annals of mathematics, pages 839–864.

Czarnecki, V. (2025). The logic of linearization: Interpretations of trees via strings. Proceedings of the Society for Computation in Linguistics, 8(1).

Gafos, A. I. (2002). A grammar of gestural coordination. Natural language & linguistic theory, 20(2):269–337.

Gafos, A. I. and Benus, S. (2006). Dynamics of phonological cognition. Cognitive science, 30(5):905–943.

Gahl, S. (2008). Time and thyme are not homophones: The effect of lemma frequency on word durations in spontaneous speech. Language, 84(3):474–496.

Bibliography II

Goldstein, L. (2011). Back to the past tense in English. In Gutiérrez-Bravo, R., Mikkelsen, L., and Potsdam, E., editors, Representing language: Essays in honor of Judith Aissen, pages 69–88.

Hall, N. (2010). Articulatory phonology. Language and Linguistics Compass, 4(9):818–830.

Heinz, J. (2018). The computational nature of phonological generalizations. In Hyman, L. and Plank, F., editors, Phonological Typology, Phonetics and Phonology, chapter 5, pages 126–195. De Gruyter Mouton.

Kenstowicz, M. (2010). The cambridge handbook of phonology. Language, 86(1):216–219.

Marr, D. (1982). Vision: A computational investigation into the human representation and processing of visual information. CUMINCAD.

Nam, H. (2007). A gestural coupling model of syllable structure. PhD thesis, Yale University.

Nelson, S. (2022). A model theoretic perspective on phonological feature systems. Proceedings of the Society for Computation in Linguistics, 5(1).

Nelson, S. (2024). The Computational Structure of Phonological and Phonetic Knowledge. PhD thesis, Stony Brook University.

Nelson, S. and Heinz, J. (in press). The blueprint model of production. Phonology.

Nolan, F. (1992). The descriptive role of segments: Evidence from assimilation. In Docherty, G. J. and Ladd, D. R., editors, Papers in Laboratory Phonology 2: Gesture, Segment, Prosody. Cambridge University Press.

Oakden, C. (2020). Notational equivalence in tonal geometry. Phonology, 37(2):257–296.

Pierce, B. C. (2002). Types and programming languages. MIT press.

Pierrehumbert, J. (2002). Word-specific phonetics. Laboratory phonology, 7.

Port, R. F. and Leary, A. P. (2005). Against formal phonology. Language, 81(4):927–964.

Roark, B. and Sproat, R. (2007). Computational approaches to morphology and syntax, volume 4. Oxford University Press.

Rogers, J., Heinz, J., Bailey, G., Edlefsen, M., Visscher, M., Wellcome, D., and Wibel, S. (2010). On languages piecewise testable in the strict sense. In The Mathematics of Language: 10th and 11th Biennial Conference, MOL 10, Los Angeles, CA, USA, July 28-30, 2007, and MOL 11, Bielefeld, Germany, August 20-21, 2009, Revised Selected Papers, pages 255–265. Springer.

Bibliography III

Rogers, J., Heinz, J., Fero, M., Hurst, J., Lambert, D., and Wibel, S. (2013). Cognitive and sub-regular complexity. In Formal Grammar: 17th and 18th International Conferences, FG 2012, Opole, Poland, August 2012, Revised Selected Papers, FG 2013, Düsseldorf, Germany, August 2013. Proceedings, pages 90–108. Springer.

Sipser, M. (2013). Introduction to the Theory of Computation. Cengage Learning, third edition edition.

Smith, C. M. (2018). Harmony in gestural phonology. PhD thesis, University of Southern California.

Strother-Garcia, K. (2019). Using model theory in phonology: a novel characterization of syllable structure and syllabification. PhD thesis, University of Delaware.