Yale

# Harmonic Serialism and Finite-State Optimality Theory

**Yiding Hao**

DEPARTMENT OF LINGUISTICS, YALE UNIVERSITY

FSMNLP · UMEÅ, SWEDEN

September 5, 2017

# Introduction

- Natural language phonology can be modelled as a rational relation between *underlying forms* and *surface forms* (Johnson, 1972), (Kaplan & Kay, 1994).

# Introduction

- Natural language phonology can be modelled as a rational relation between *underlying forms* and *surface forms*.
- However, modern phonology uses the formalism of Optimality Theory (OT), which defines non-rational relations in general (Frank & Satta, 1998), (Karttunen, 1998).

# Introduction

- Natural language phonology can be modelled as a rational relation between *underlying forms* and *surface forms*.
- However, modern phonology uses the formalism of Optimality Theory (OT), which defines non-rational relations in general.
- An interesting question:

*When does OT define rational relations?*

# Motivation

Why do we want a finite-state model of OT?

# Motivation

Why do we want a finite-state model of OT?

- To capture the correct typology of natural language phonologies.

# Motivation

Why do we want a finite-state model of OT?

- To capture the correct typology of natural language phonologies.
- To investigate where the complexity of OT comes from.

# Motivation

Why do we want a finite-state model of OT?

- To capture the correct typology of natural language phonologies.
- To investigate where the complexity of OT comes from.
- To allow for a finite-state implementation of OT grammars.

# Outline

In this talk I will...

# Outline

In this talk I will...

- ... introduce a new finite-state model of OT.

Yale

# Outline

In this talk I will...

- ... introduce a new finite-state model of OT.
  - Strict Locality: Markedness constraints are sets of banned subsequences.

Yale

# Outline

In this talk I will...

- ... introduce a new finite-state model of OT.
  - Strict Locality: Markedness constraints are sets of banned subsequences.
  - Harmonic Serialism (HS): The grammar only change one symbol at a time. The grammar recurses until a fixed-point is reached.

# Outline

In this talk I will...

- ... introduce a new finite-state model of OT.
  - Strict Locality: Markedness constraints are sets of banned subsequences.
  - Harmonic Serialism (HS): The grammar only change one symbol at a time. The grammar recurses until a fixed-point is reached.
- ... prove that this model computes rational relations.

# Outline

In this talk I will...

- ... introduce a new finite-state model of OT.
  - Strict Locality: Markedness constraints are sets of banned subsequences.
  - Harmonic Serialism (HS): The grammar only change one symbol at a time. The grammar recurses until a fixed-point is reached.
- ... prove that this model computes rational relations.
  - First show that one iteration of the grammar is rational.

# Outline

In this talk I will...

- ... introduce a new finite-state model of OT.
  - Strict Locality: Markedness constraints are sets of banned subsequences.
  - Harmonic Serialism (HS): The grammar only change one symbol at a time. The grammar recurses until a fixed-point is reached.
- ... prove that this model computes rational relations.
  - First show that one iteration of the grammar is rational.
  - Then show that recursion preserves rationality.

# Outline

Overview of Optimality Theory

Formalization of Harmonic Serialism

Proof of Finite-Stateness

Conclusions

# Overview of Optimality Theory

# What is Optimality Theory?

- OT is a constraint-based formalism for describing relations between strings.

# What is Optimality Theory?

- OT is a constraint-based formalism for describing relations between strings.
- Constraints are allowed to contradict one another.

Yale

# What is Optimality Theory?

- OT is a constraint-based formalism for describing relations between strings.
- Constraints are allowed to contradict one another.
- Conflicts are mediated by:
    - constraint rankings and

# What is Optimality Theory?

- OT is a constraint-based formalism for describing relations between strings.
- Constraints are allowed to contradict one another.
- Conflicts are mediated by:
  - constraint rankings and
  - gradient violability.

Yale

# An Example of an OT Grammar

In Japanese, a consonant must always be followed by a vowel, except n.

# An Example of an OT Grammar

In Japanese, a consonant must always be followed by a vowel, except n.

In OT, this is described using *markedness constraints*:

# An Example of an OT Grammar

In Japanese, a consonant must always be followed by a vowel, except n.

In OT, this is described using *markedness constraints*:

- CV: Every consonant must be followed by a vowel, except n.

# An Example of an OT Grammar

In Japanese, a consonant must always be followed by a vowel, except n.

In OT, this is described using *markedness constraints*:

- CV: Every consonant must be followed by a vowel, except n.

and *faithfulness constraints*:

# An Example of an OT Grammar

In Japanese, a consonant must always be followed by a vowel, except n.

In OT, this is described using *markedness constraints*:

- CV: Every consonant must be followed by a vowel, except n.

and *faithfulness constraints*:

- MAX: Pronounce every sound in a word.

# An Example of an OT Grammar

In Japanese, a consonant must always be followed by a vowel, except n.

In OT, this is described using *markedness constraints*:

- CV: Every consonant must be followed by a vowel, except n.

and *faithfulness constraints*:

- MAX: Pronounce every sound in a word.
- DEP($x$): Do not add an $x$ to a word.

# An Example of an OT Grammar

In Japanese, a consonant must always be followed by a vowel, except n.

In OT, this is described using *markedness constraints*:

- CV: Every consonant must be followed by a vowel, except n.

and *faithfulness constraints*:

- MAX: Pronounce every sound in a word.
- DEP($x$): Do not add an $x$ to a word.
- ID: Do not change any sounds.

These constraints are *ranked:*

- MAX $\gg$ ID $\gg$ DEP$(\mathrm{a}) \gg \cdots \gg$ DEP$(\mathrm{z}) \gg$ CV $\gg$ DEP$(\mathrm{u})$

Yale

# An Example of an OT Grammar

How is the Swedish name *Frank* pronounced in Japanese?

# An Example of an OT Grammar

How is the Swedish name *Frank* pronounced in Japanese?

| frank | MAX | ID | DEP(a) | ... | DEP(z) | CV | DEP(u) |
|---|---|---|---|---|---|---|---|
| a. frank | | | | | | **! | |
| b. ran | **! | | | | | | |
| c. arana | | **! | | | | | |
| d. faranka | | | **! | | | | |
| ☞ e. furanku | | | | | | | ** |

# An Example of an OT Grammar

How is the Swedish name *Frank* pronounced in Japanese?

| frank | MAX | ID | DEP(a) | ... | DEP(z) | CV | DEP(u) |
|---|---|---|---|---|---|---|---|
| a. frank | | | | | | **! | |
| b. ran | **! | | | | | | |
| c. arana | | **! | | | | | |
| d. faranka | | | **! | | | | |
| ☞ e. furanku | | | | | | | ** |

Add a u after each consonant (other than n) not followed by a vowel.

# OT is Not Finite-State

Constraints (Gerdemann & Hulden, 2012):

Yale

# OT is Not Finite-State

Constraints:

- DEP: No epenthesis (insertion)

# OT is Not Finite-State

Constraints:

- DEP: No epenthesis (insertion)
- ID: No changing one symbol to another

# OT is Not Finite-State

Constraints:

- DEP: No epenthesis (insertion)
- ID: No changing one symbol to another
- AGR: No *ab* or *ba* sequences

# OT is Not Finite-State

Constraints:

- DEP: No epenthesis (insertion)
- ID: No changing one symbol to another
- AGR: No *ab* or *ba* sequences
- MAX: No deletion

# OT is Not Finite-State

| *aaabb* | Dep | Id | Agr | Max |
|---|---|---|---|---|
| a. *aaabb* | | | ∗! | |
| b. *aaacbb* | ∗! | | | |
| c. *aaaaa* | | ∗∗! | | |
| ☞ d. *aaa* | | | | ∗∗ |
| e. *bb* | | | | ∗∗∗! |

# OT is Not Finite-State

| *aaabb* | DEP | ID | AGR | MAX |
|---|---|---|---|---|
| a. *aaabb* | | | *! | |
| b. *aaacbb* | *! | | | |
| c. *aaaaa* | | **! | | |
| ☞ d. *aaa* | | | | ** |
| e. *bb* | | | | ***! |

Delete the *a*s or *b*s, whichever is less common.

# Finite-State Models of OT

How can a finite-state model be accomplished?

# Finite-State Models of OT

How can a finite-state model be accomplished?

- Violation Bound (Frank & Satta, 1998), (Karttunen, 1998)

# Finite-State Models of OT

How can a finite-state model be accomplished?

- Violation Bound
- Matching (Gerdemann & Van Noord, 2000), (Gerdemann & Hulden, 2012)

# Finite-State Models of OT

How can a finite-state model be accomplished?

- Violation Bound
- Matching
- Directional Evaluation (Eisner, 2000), (Eisner, 2002)

# Finite-State Models of OT

How can a finite-state model be accomplished?

- Violation Bound
- Matching
- Directional Evaluation
- OTCA (Riggle, 2004)

# Restrictions on OT

Finite-stateness is achieved "by hook or by crook" (Eisner, 2002) using...

# Restrictions on OT

Finite-stateness is achieved “by hook or by crook” using...

- ...restrictions on markedness constraints

# Restrictions on OT

Finite-stateness is achieved “by hook or by crook” using...

- ...restrictions on markedness constraints
- ...restrictions on how candidates are compared (Eval).

# Restrictions on OT

Finite-stateness is achieved “by hook or by crook” using...

- ...restrictions on markedness constraints
- ...restrictions on how candidates are compared (EVAL).

| *aaabb* | DEP | ID | AGR | MAX |
|---|---|---|---|---|
| a. *aaabb* | | | *! | |
| b. *aaa* | | | | ***! |
| ☞ c. *bb* | | | | ** |

# Restrictions on OT

Finite-stateness is achieved "by hook or by crook" using...

- ...restrictions on markedness constraints
- ...restrictions on how candidates are compared (EVAL).

| *aaabb* | DEP | ID | AGR | MAX |
|---|---|---|---|---|
| a. *aaabb* | | | *! | |
| b. *aaa* | | | | ***! |
| ☞ c. *bb* | | | | ** |

Non-rational power comes from MAX!

# Restrictions on OT

We need restrictions on markedness constraints *and* faithfulness constraints.

# Restrictions on OT

We need restrictions on markedness constraints *and* faithfulness constraints.

- Assume for simplicity that markedness constraints are *strictly local*:

  *"Assign one violation for each occurrence of abc, def, ghi, ..."*

Yale

# Restrictions on OT

We need restrictions on markedness constraints *and* faithfulness constraints.

- Assume for simplicity that markedness constraints are *strictly local*.
- Adopt the *Harmonic Serialism* (HS) version of OT.

# Restrictions on OT

We need restrictions on markedness constraints *and* faithfulness constraints.

- Assume for simplicity that markedness constraints are *strictly local*.
- Adopt the *Harmonic Serialism* (HS) version of OT.
  - The grammar can only insert, delete, or change one symbol.

# Restrictions on OT

We need restrictions on markedness constraints *and* faithfulness constraints.

- Assume for simplicity that markedness constraints are *strictly local*.
- Adopt the *Harmonic Serialism* (HS) version of OT.
  - The grammar can only insert, delete, or change one symbol.
  - If the faithful candidate wins, that candidate is the surface form.

# Restrictions on OT

We need restrictions on markedness constraints *and* faithfulness constraints.

- Assume for simplicity that markedness constraints are *strictly local*.
- Adopt the *Harmonic Serialism* (HS) version of OT.
  - The grammar can only insert, delete, or change one symbol.
  - If the faithful candidate wins, that candidate is the surface form.
  - Otherwise, recursively call the grammar on the winner.

# Restrictions on OT

- Adopt the *Harmonic Serialism* (HS) version of OT.
  - The grammar can only insert, delete, or change one symbol.
  - If the faithful candidate wins, that candidate is the surface form.
  - Otherwise, recursively call the grammar on the winner.

| *aaabb* | DEP | ID | AGR | MAX |
|---|---|---|---|---|
| ☞ a. *aaabb* | | | * | |
| b. *aabb* | | | * | *! |
| c. *aaab* | | | * | *! |

Yale

# Formalization of Harmonic Serialism

# Operations

## Definition

An *operation* is an ordered pair $\langle x, y\rangle$ with $|x|, |y| \leq 1$ and $|xy| > 0$.

# Operations

**Definition**

An *operation* is an ordered pair $\langle x, y \rangle$ with $|x|, |y| \leq 1$ and $|xy| > 0$.

- Insertions: $\langle \lambda, x \rangle$

Yale

# Operations

**Definition**

An *operation* is an ordered pair $\langle x, y\rangle$ with $|x|, |y| \leq 1$ and $|xy| > 0$.

- Insertions: $\langle \lambda, x\rangle$
- Deletions: $\langle x, \lambda\rangle$

# Operations

**Definition**

An *operation* is an ordered pair $\langle x, y \rangle$ with $|x|, |y| \leq 1$ and $|xy| > 0$.

- Insertions: $\langle \lambda, x \rangle$
- Deletions: $\langle x, \lambda \rangle$
- Substitutions: $\langle x, y \rangle$ ($x, y \neq \lambda$)

# Operations

**Definition**

For an operation $\langle x, y \rangle$ and $u, v \in \Sigma^*$, $\langle uxv, uyv \rangle$ is an *application* of $\langle x, y \rangle$. A *change* is an application of an operation.

Yale

# Constraints

## Definition

A *constraint* is a function $c : \Sigma^* \times \Sigma^* \to \mathbb{N}$.

# Constraints

**Definition**

A *constraint* is a function $c : \Sigma^* \times \Sigma^* \to \mathbb{N}$.

- A *markedness constraint* $c$ specifies a set $S_c$ of banned sequences. $c(x, y) = c(y)$ is the number of times a banned sequence occurs in $y$.

Yale

# Constraints

**Definition**

A *constraint* is a function $c : \Sigma^* \times \Sigma^* \to \mathbb{N}$.

- A *markedness constraint* $c$ specifies a set $S_c$ of banned sequences. $c(x, y) = c(y)$ is the number of times a banned sequence occurs in $y$.
- A *faithfulness constraint* $f$ specifies a set $O_f$ of banned operations. $f(x, y) = 1$ if $\langle x, y \rangle$ is the application of a banned operation; $f(x, y) = 0$ otherwise.

Yale

# Constraints

## Example

- AGR: $S_c = \{ab, ba\}$

# Constraints

## Example

- AGR: $S_c = \{ab, ba\}$
- DEP: $O_f = \{\langle \lambda, x \rangle | x \in \Sigma\}$

# Constraints

## Example

- AGR: $S_c = \{ab, ba\}$
- DEP: $O_f = \{\langle\lambda, x\rangle | x \in \Sigma\}$
- ID: $O_f = \{\langle x, y\rangle | x, y \in \Sigma; y \neq x\}$

Yale

# Constraints

## Example

- AGR: $S_c = \{ab, ba\}$
- DEP: $O_f = \{\langle \lambda, x \rangle | x \in \Sigma\}$
- ID: $O_f = \{\langle x, y \rangle | x, y \in \Sigma; y \neq x\}$
- MAX: $O_f = \{\langle x, \lambda \rangle | x \in \Sigma\}$

# Constraint Rankings

## Definition

A *constraint ranking* is a sequence of constraints $\langle c_1, c_2, \ldots, c_n \rangle$.

Yale

# Constraint Rankings

**Definition**

A *constraint ranking* is a sequence of constraints $\langle c_1, c_2, \ldots, c_n \rangle$.

- For any constraint ranking $C$, the number $k_C \geq 0$ is the length of the longest sequence banned by a markedness constraint of $C$.

Yale

# Harmonicity

## Definition

The *cost* of a change $\langle x, y \rangle$ with respect to a constraint ranking $C = \langle c_1, c_2, \ldots, c_n \rangle$ is the vector

$$\mathfrak{c}_C(x, y) = \langle c_1(x, y), c_2(x, y), \ldots, c_n(x, y) \rangle.$$

The *benefit* of $\langle x, y \rangle$ is

$$\mathfrak{b}_C(x, y) = \mathfrak{c}_C(x, y) - \mathfrak{c}_C(x, x).$$

## Harmonicity

### Definition

Let $a = \langle a_1, a_2, \ldots, a_n \rangle, b = \langle b_1, b_2, \ldots, b_n \rangle \in \mathbb{Z}^n$ with $a \neq b$. Let

$$j = \min_{a_i \neq b_i} i.$$

$a$ is *more harmonic* than $b$ iff $a_j < b_j$.

# Harmonicity

## Definition

Let $a = \langle a_1, a_2, \ldots, a_n \rangle, b = \langle b_1, b_2, \ldots, b_n \rangle \in \mathbb{Z}^n$ with $a \neq b$. Let

$$j = \min_{a_i \neq b_i} i.$$

$a$ is *more harmonic* than $b$ iff $a_j < b_j$.

- $a \succ_H b$: "$a$ is more harmonic than $b$"

# Harmonicity

## Definition

Let $a = \langle a_1, a_2, \ldots, a_n \rangle, b = \langle b_1, b_2, \ldots, b_n \rangle \in \mathbb{Z}^n$ with $a \neq b$. Let

$$j = \min_{a_i \neq b_i} i.$$

$a$ is *more harmonic* than $b$ iff $a_j < b_j$.

- $a \succ_H b$: "$a$ is more harmonic than $b$"
- $a \succeq_H b$: "$a$ is at least as harmonic as $b$"

# Harmonicity

## Example

| *ab* | Dep | Id | Agr | Max |
|---|---|---|---|---|
| a. *ab* | | | *! | |
| ☞ b. *a* | | | | * |
| ☞ c. *b* | | | | * |

Yale

# Harmonicity

## Example

| *ab* | DEP | ID | AGR | MAX |
|---|---|---|---|---|
| a. *ab* | | | ∗! | |
| ☞ b. *a* | | | | ∗ |
| ☞ c. *b* | | | | ∗ |

- $\mathfrak{c}_C(ab, a) = \langle 0, 0, 0, 1 \rangle$

Yale

## Example

| *ab* | DEP | ID | AGR | MAX |
|---|---|---|---|---|
| a. *ab* | | | *! | |
| ☞ b. *a* | | | | * |
| ☞ c. *b* | | | | * |

- $\mathfrak{c}_C(ab, a) = \langle 0, 0, 0, 1 \rangle$
- $\mathfrak{b}_C(ab, a) = \mathfrak{c}_C(ab, a) - \mathfrak{c}_C(ab, ab) = \langle 0, 0, 0, 1 \rangle - \langle 0, 0, 1, 0 \rangle = \langle 0, 0, -1, 1 \rangle$

# Harmonicity

## Example

| *ab* | DEP | ID | AGR | MAX |
|---|---|---|---|---|
| a. *ab* | | | *! | |
| ☞ b. *a* | | | | * |
| ☞ c. *b* | | | | * |

- $\mathfrak{c}_C(ab, a) = \langle 0, 0, 0, 1 \rangle$
- $\mathfrak{b}_C(ab, a) = \mathfrak{c}_C(ab, a) - \mathfrak{c}_C(ab, ab) = \langle 0, 0, 0, 1 \rangle - \langle 0, 0, 1, 0 \rangle = \langle 0, 0, -1, 1 \rangle$
- $\mathfrak{b}_C(ab, a) = \langle 0, 0, -1, 1 \rangle \succ_H \langle 0, 0, 0, 0 \rangle = \mathfrak{b}_C(ab, ab)$

Yale

# Harmonic Serialism

## Definition

Let $C$ be a constraint ranking. The relation $\mathscr{H}_C \subseteq \Sigma^* \times \Sigma^*$ is defined by $\langle u, v \rangle \in \mathscr{H}_C$ if and only if

$$\mathfrak{b}_C(u, v) = \max_y \mathfrak{b}_C(u, y),$$

where max is taken with respect to $\succ_H$ over strings $y$ such that $\langle u, y \rangle$ is a change.

Yale

# Harmonic Serialism

## Definition

Let $C$ be a constraint ranking. Let $\hat{\mathscr{H}}_C$ be the transitive closure of $\mathscr{H}_C$. The relation $\mathscr{H}_C^*$ is defined by

$$\mathscr{H}_C^* = \{\langle x, y\rangle \in \hat{\mathscr{H}}_C | \langle y, y\rangle \in \mathscr{H}_C\}.$$

# Harmonic Serialism

## Definition

Let $C$ be a constraint ranking. Let $\hat{\mathscr{H}}_C$ be the transitive closure of $\mathscr{H}_C$. The relation $\mathscr{H}_C^*$ is defined by

$$\mathscr{H}_C^* = \{\langle x, y\rangle \in \hat{\mathscr{H}}_C | \langle y, y\rangle \in \mathscr{H}_C\}.$$

- $y$ is a surface form for underlying form $x$ if and only if $\langle x, y\rangle \in \mathscr{H}_C^*$.

Yale

# Proof of Finite-Stateness

# Proof of Finite-Stateness

The proof will be done in two steps.

# Proof of Finite-Stateness

The proof will be done in two steps.

- Show that $\mathscr{H}_C$ is rational.

Yale

# Proof of Finite-Stateness

The proof will be done in two steps.

- Show that $\mathscr{H}_C$ is rational.
- Show that if $\mathscr{H}_C$ is rational, then so is $\mathscr{H}_C^*$.

Yale

# $\mathscr{H}_C$ is Rational

We will use strict locality to show that $\mathscr{H}_C$ is rational.

# $\mathscr{H}_C$ is Rational

We will use strict locality to show that $\mathscr{H}_C$ is rational.

- If $\langle lxr, lyr \rangle$ is a change, then $\mathfrak{b}_C(lxr, lyr)$ only depends on a context of bounded size around $x$ and $y$.

Yale

# $\mathscr{H}_C$ is Rational

We will use strict locality to show that $\mathscr{H}_C$ is rational.

- If $\langle lxr, lyr \rangle$ is a change, then $\mathfrak{b}_C(lxr, lyr)$ only depends on a context of bounded size around $x$ and $y$.

## Definition

A *rule* is an ordered quadruple $\langle x, y, c, d \rangle$, where $\langle x, y \rangle$ is an operation and $c, d \in (\Sigma \cup \{\rtimes, \ltimes\})^*$. We denote $\langle x, y, c, d \rangle$ by $x \rightarrow y \,/\, c \,\_\, d$.

Yale

# $\mathscr{H}_C$ is Rational

We will use strict locality to show that $\mathscr{H}_C$ is rational.

- If $\langle lxr, lyr \rangle$ is a change, then $\mathfrak{b}_C(lxr, lyr)$ only depends on a context of bounded size around $x$ and $y$.

## Definition

A *rule* is an ordered quadruple $\langle x, y, c, d \rangle$, where $\langle x, y \rangle$ is an operation and $c, d \in (\Sigma \cup \{ \rtimes, \ltimes \})^*$. We denote $\langle x, y, c, d \rangle$ by $x \to y \,/\, c \,\_\, d$.

## Definition

An *application of a rule* $x \to y \,/\, c \,\_\, d$ is a pair $\langle a, b \rangle$ such for some $u, v \in (\Sigma \cup \{ \rtimes, \ltimes \})^*$, $\rtimes a \ltimes = ucxdv$ and $\rtimes b \ltimes = ucydv$.

Yale

# $\mathscr{H}_C$ is Rational

*Proposition*

*Let C be a constraint ranking, and suppose* $\langle a_1, b_1 \rangle$ *and* $\langle a_2, b_2 \rangle$ *are applications of a rule* $x \to y \,/\, c\_d$. *If* $|c|, |d| \geq k_C - 1$, *then* $\mathfrak{b}_C(a_1, b_1) = \mathfrak{b}_C(a_2, b_2)$.

Yale

# $\mathscr{H}_C$ is Rational

### *Proposition*

*Let C be a constraint ranking, and suppose $\langle a_1, b_1 \rangle$ and $\langle a_2, b_2 \rangle$ are applications of a rule $x \to y$ / $c$ _ $d$. If $|c|, |d| \geq k_C - 1$, then $\mathfrak{b}_C(a_1, b_1) = \mathfrak{b}_C(a_2, b_2)$.*

- There are only finitely many rules $x \to y$ / $c$ _ $d$ with $|c| = |d| = k_C - 1$.

# $\mathscr{H}_C$ is Rational

**Proposition**

*Let $C$ be a constraint ranking, and suppose $\langle a_1, b_1\rangle$ and $\langle a_2, b_2\rangle$ are applications of a rule $x \to y \,/\, c\,\_\,d$. If $|c|, |d| \geq k_C - 1$, then $\mathfrak{b}_C(a_1, b_1) = \mathfrak{b}_C(a_2, b_2)$.*

- There are only finitely many rules $x \to y \,/\, c\,\_\,d$ with $|c| = |d| = k_C - 1$.
- This means that the set of possible changes $\langle x, y\rangle$ can be reduced to finitely many cases!

Yale

# $\mathscr{H}_C$ is Rational

**Proposition**

*Let $C$ be a constraint ranking, and suppose $\langle a_1, b_1 \rangle$ and $\langle a_2, b_2 \rangle$ are applications of a rule $x \to y \,/\, c\_d$. If $|c|, |d| \geq k_C - 1$, then $\mathfrak{b}_C(a_1, b_1) = \mathfrak{b}_C(a_2, b_2)$.*

- There are only finitely many rules $x \to y \,/\, c\_d$ with $|c| = |d| = k_C - 1$.
- This means that the set of possible changes $\langle x, y \rangle$ can be reduced to finitely many cases!
- Let us write $\mathfrak{b}_C(x \to y \,/\, c\_d) = \mathfrak{b}_C(cxd, cyd)$.

Yale

# $\mathscr{H}_C$ is Rational

We can now predict the behavior of $\mathscr{H}_C$ by inspecting inputs.

# $\mathscr{H}_C$ is Rational

We can now predict the behavior of $\mathscr{H}_C$ by inspecting inputs.

- For $|c| = |d| = k_C - 1$ and $|x| \leq 1$, let

$$R_{cxd} = \left\{ r = x \to y \,/\, c\_d \,\middle|\, \mathfrak{b}_C(r) = \max_{y'} \mathfrak{b}_C(x \to y' \,/\, c\_d) \right\}$$

  where max is taken wrt $\prec_H$.

# $\mathscr{H}_C$ is Rational

We can now predict the behavior of $\mathscr{H}_C$ by inspecting inputs.

- For $|c| = |d| = k_C - 1$ and $|x| \leq 1$, let

$$R_{cxd} = \left\{ r = x \to y \,/\, c\,\_\,d \,\middle|\, \mathfrak{b}_C(r) = \max_{y'} \mathfrak{b}_C(x \to y' \,/\, c\,\_\,d) \right\}.$$

- All rules of $R_{cxd}$ have the same benefit. Call this value $\mathfrak{b}_C(cxd)$.

# $\mathscr{H}_C$ is Rational

We can now predict the behavior of $\mathscr{H}_C$ by inspecting inputs.

- For $|c| = |d| = k_C - 1$ and $|x| \leq 1$, let

$$R_{cxd} = \left\{ r = x \to y \,/\, c\,\_\,d \,\middle|\, \mathfrak{b}_C(r) = \max_{y'} \mathfrak{b}_C(x \to y' \,/\, c\,\_\,d) \right\}.$$

- All rules of $R_{cxd}$ have the same benefit. Call this value $\mathfrak{b}_C(cxd)$.
- $\mathscr{H}_C$ applies a rule of $R_{cxd}$ to an input $a = lcxdr$ iff $a$ does not contain any substring $c'x'd'$ with $\mathfrak{b}_C(c'x'd') \succ_H \mathfrak{b}_C(cxd)$.

Yale

# $\mathscr{H}_C$ is Rational

We can now predict the behavior of $\mathscr{H}_C$ by inspecting inputs.

- For $|c| = |d| = k_C - 1$ and $|x| \leq 1$, let

$$R_{cxd} = \left\{ r = x \to y \,/\, c\,\_\,d \,\middle|\, \mathfrak{b}_C(r) = \max_{y'} \mathfrak{b}_C(x \to y' \,/\, c\,\_\,d) \right\}.$$

- All rules of $R_{cxd}$ have the same benefit. Call this value $\mathfrak{b}_C(cxd)$.
- $\mathscr{H}_C$ applies a rule of $R_{cxd}$ to an input $a = lcxdr$ iff $a$ does not contain any substring $c'x'd'$ with $\mathfrak{b}_C(c'x'd') \succ_H \mathfrak{b}_C(cxd)$.
- For every $r \in R_{cxd}$, let

$$F_r = \{c'x'd' | \mathfrak{b}_C(c'x'd') \succ_H \mathfrak{b}_C(cxd)\}.$$

Yale

# $\mathscr{H}_C$ is Rational

*Theorem*

*Let $C = \langle c_1, c_2, \ldots, c_n \rangle$ be a constraint ranking. Then, $\mathscr{H}_C$ is rational.*

# $\mathscr{H}_C$ is Rational

**Theorem**

*Let $C = \langle c_1, c_2, \ldots, c_n \rangle$ be a constraint ranking. Then, $\mathscr{H}_C$ is rational.*

**Proof.**

Let

$$R = \bigcup_{cxd} R_{cxd}.$$

□

Yale

# $\mathscr{H}_C$ is Rational

*Theorem*

*Let $C = \langle c_1, c_2, \ldots, c_n \rangle$ be a constraint ranking. Then, $\mathscr{H}_C$ is rational.*

Proof.

For each $r = x \rightarrow y \, / \, c \,\_\, d \in R$, we need to construct a transducer $T_r$ that applies $r$ iff...

- ...it is the most beneficial rule applicable to its input.

□

Yale

# $\mathscr{H}_C$ is Rational

**Theorem**

*Let $C = \langle c_1, c_2, \ldots, c_n \rangle$ be a constraint ranking. Then, $\mathscr{H}_C$ is rational.*

**Proof.**

For each $r = x \rightarrow y \,/\, c \,\_\, d \in R$, we need to construct a transducer $T_r$ that applies $r$ iff...

- ...it is the most beneficial rule applicable to its input.
- $\iff$ the input does not have any substrings in $F_r$.

□

Yale

# $\mathscr{H}_C$ is Rational

**Theorem**

*Let $C = \langle c_1, c_2, \ldots, c_n \rangle$ be a constraint ranking. Then, $\mathscr{H}_C$ is rational.*

**Proof.**

For each $r = x \rightarrow y \, / \, c \, \_ \, d \in R$, we need to construct a transducer $T_r$ that applies $r$ iff...

- ...it is the most beneficial rule applicable to its input.
- $\iff$ the input does not have any substrings in $F_r$.
- $\iff$ the input is in the set $S_r$ of strings without substrings in $F_r$.

□

Yale

# $\mathscr{H}_C$ is Rational

*Theorem*

*Let $C = \langle c_1, c_2, \ldots, c_n \rangle$ be a constraint ranking. Then, $\mathscr{H}_C$ is rational.*

Proof.

For each $r = x \rightarrow y \,/\, c \_ d \in R$, we need to construct a transducer $T_r$ that applies $r$ iff...

- ... the input is in the set $S_r$ of strings without substrings in $F_r$.

Notice that $S_r$ is $(2k_C - 1)$-strictly local!

□

Yale

## $\mathscr{H}_C$ is Rational

**Theorem**

*Let $C = \langle c_1, c_2, \ldots, c_n \rangle$ be a constraint ranking. Then, $\mathscr{H}_C$ is rational.*

**Proof.**

Define $T_r$ to apply $r$ to any input not in $S_r$:

$$[T_r] = \{\langle ucxdv, ucydv \rangle | ucxdv \in S_r, y \in \Sigma \cup \{\lambda\}\}.$$

□

Yale

# $\mathscr{H}_C^*$ is Rational

It remains to show that $\mathscr{H}_C^*$ is rational.

Yale

# $\mathscr{H}_C^*$ is Rational

It remains to show that $\mathscr{H}_C^*$ is rational.

- Recall that $\mathscr{H}_C^* = \{\langle x, y\rangle \in \hat{\mathscr{H}}_C | \langle y, y\rangle \in \mathscr{H}_C\}$.

# $\mathscr{H}_C^*$ is Rational

It remains to show that $\mathscr{H}_C^*$ is rational.

- Recall that $\mathscr{H}_C^* = \{\langle x, y\rangle \in \hat{\mathscr{H}_C} | \langle y, y\rangle \in \mathscr{H}_C\}$.
- The challenge is to show that the transitive closure $\hat{\mathscr{H}_C}$ is rational.

Yale

# $\mathscr{H}_C^*$ is Rational

It remains to show that $\mathscr{H}_C^*$ is rational.

- Recall that $\mathscr{H}_C^* = \{\langle x, y\rangle \in \hat{\mathscr{H}}_C | \langle y, y\rangle \in \mathscr{H}_C\}$.
- The challenge is to show that the transitive closure $\hat{\mathscr{H}}_C$ is rational.
- This is possible due to a result from *regular model checking*.

Yale

# $\mathscr{H}_C^*$ is Rational

## Definition

A same-length relation $R$ has *bounded local depth* if there exists $m_R$ such that for each $i \in \mathbb{N}$, applying $R$ repeatedly can change the $i$th symbol of its input at most $m_R$-many times.

# $\mathscr{H}_C^*$ is Rational

**Definition**

A same-length relation $R$ has *bounded local depth* if there exists $m_R$ such that for each $i \in \mathbb{N}$, applying $R$ repeatedly can change the $i$th symbol of its input at most $m_R$-many times.

*Theorem (Abdulla et al. (2002), Abdulla et al. (2003))*

*If a same-length rational relation has bounded local depth, then its transitive closure is rational.*

Yale

# $\mathscr{H}_C^*$ is Rational

*Theorem (Abdulla et al. (2002), Abdulla et al. (2003))*

*If a same-length rational relation has bounded local depth, then its transitive closure is rational.*

Strategy:

- Pad $\mathscr{H}_C$ to turn it into a same-length relation.

# $\mathscr{H}_C^*$ is Rational

**Theorem (Abdulla et al. (2002), Abdulla et al. (2003))**

*If a same-length rational relation has bounded local depth, then its transitive closure is rational.*

Strategy:

- Pad $\mathscr{H}_C$ to turn it into a same-length relation.
- Show that this padded relation has bounded local depth.

# $\mathscr{H}_C^*$ is Rational

**Theorem (Abdulla et al. (2002), Abdulla et al. (2003))**

*If a same-length rational relation has bounded local depth, then its transitive closure is rational.*

Strategy:

- Pad $\mathscr{H}_C$ to turn it into a same-length relation.
- Show that this padded relation has bounded local depth.
- Use the transitive closure to obtain $\mathscr{H}_C^*$.

# $\mathscr{H}_C^*$ is Rational

*Theorem (Abdulla et al. (2002), Abdulla et al. (2003))*

*If a same-length rational relation has bounded local depth, then its transitive closure is rational.*

Proof.

Introduce two new symbols: `i` and `d`.

□

Yale

# $\mathscr{H}_C^*$ is Rational

**Theorem (Abdulla et al. (2002), Abdulla et al. (2003))**

*If a same-length rational relation has bounded local depth, then its transitive closure is rational.*

**Proof.**

Introduce two new symbols: `i` and `d`.

- Pad the input with `i`s.

□

Yale

# $\mathscr{H}_C^*$ is Rational

**Theorem (Abdulla et al. (2002), Abdulla et al. (2003))**

*If a same-length rational relation has bounded local depth, then its transitive closure is rational.*

**Proof.**

- Pad the input with `i`s.
- Insertion: $\texttt{i} \rightarrow x \,/\, c \,\_\, d$

□

Yale

## $\mathscr{H}_C^*$ is Rational

**Theorem (Abdulla et al. (2002), Abdulla et al. (2003))**

*If a same-length rational relation has bounded local depth, then its transitive closure is rational.*

**Proof.**

- Pad the input with `i`s.
- Insertion: $\mathtt{i} \rightarrow x \,/\, c \,\_\, d$
- Deletion: $x \rightarrow \mathtt{d} \,/\, c \,\_\, d$

□

Yale

## $\mathscr{H}_C^*$ is Rational

**Theorem (Abdulla et al. (2002), Abdulla et al. (2003))**

*If a same-length rational relation has bounded local depth, then its transitive closure is rational.*

**Proof.**

- Pad the input with `i`s.
- Insertion: $\texttt{i} \rightarrow x \,/\, c \,\_\, d$
- Deletion: $x \rightarrow \texttt{d} \,/\, c \,\_\, d$
- Substitution: $\texttt{i}x \rightarrow y\texttt{d} \,/\, c \,\_\, d$

□

Yale

# $\mathscr{H}_C^*$ is Rational

*Theorem (Abdulla et al. (2002), Abdulla et al. (2003))*

*If a same-length rational relation has bounded local depth, then its transitive closure is rational.*

Proof.

- Pad the input with $\mathtt{i}$s.
- Insertion: $\mathtt{i} \to x \,/\, c \,\_\, d$
- Deletion: $x \to \mathtt{d} \,/\, c \,\_\, d$
- Substitution: $\mathtt{i}x \to y\mathtt{d} \,/\, c \,\_\, d$

Each symbol can only be changed at most twice! □

# Conclusions

# Conclusions

Harmonic Serialism with strictly local constraints is finite-state.

# Conclusions

Harmonic Serialism with strictly local constraints is finite-state.

- HS is actually "simpler" than OT even though it seems to be more "complex" (McCarthy, 2000), (Moreton, 1999)

# Conclusions

Harmonic Serialism with strictly local constraints is finite-state.

- HS is actually "simpler" than OT even though it seems to be more "complex."
- This is the first finite-state model inspired by phonological literature.

# Future Work

- See if the result extends to more classes of markedness constraints (e.g. tier-based strictly local, weakly deterministic).

# Future Work

- See if the result extends to more classes of markedness constraints (e.g. tier-based strictly local, weakly deterministic).
- Investigate varying assumptions about HS.

# Future Work

- See if the result extends to more classes of markedness constraints (e.g. tier-based strictly local, weakly deterministic).
- Investigate varying assumptions about HS.
- Implement this model.

# Future Work

- See if the result extends to more classes of markedness constraints (e.g. tier-based strictly local, weakly deterministic).
- Investigate varying assumptions about HS.
- Implement this model.
- Analyze the complexity of the model.

Thank you!

# References

Abdulla, Parosh Aziz, Bengt Jonsson, Marcus Nilsson & Julien d'Orso. 2002. Regular model checking made simple and effcient. In *International conference on concurrency theory*, 116–131. Springer.

Abdulla, Parosh Aziz, Bengt Jonsson, Marcus Nilsson & Julien d'Orso. 2003. Algorithmic improvements in regular model checking. In *International conference on computer aided verification*, 236–248. Springer.

Eisner, Jason. 2000. Directional constraint evaluation in optimality theory. *Proceedings of COLING* .

Eisner, Jason. 2002. Comprehension and compilation in optimality theory. In *Proceedings of the 40th annual meeting on association for computational linguistics*, 56–63. Association for Computational Linguistics.

Frank, Robert & Giorgio Satta. 1998. Optimality theory and the generative complexity of constraint violability. *Computational Linguistics* 24(2). 307–315.

Gerdemann, Dale & Mans Hulden. 2012. Practical finite state optimality theory. In *Fsmnlp*, 10–19.

Gerdemann, Dale & Gertjan Van Noord. 2000. Approximation and exactness in finite state optimality theory. *arXiv preprint cs/0006038* .

Johnson, C. Douglas. 1972. *Formal aspects of phonological desription*. De Gruyter Mouton.

Kaplan, Ronald M & Martin Kay. 1994. Regular models of phonological rule systems. *Computational linguistics* 20(3). 331–378.

Karttunen, Lauri. 1998. The proper treatment of optimality in computational phonology: plenary talk. In *Proceedings of the international workshop on finite state methods in natural language processing*, 1–12. Association for Computational Linguistics.

McCarthy, John J. 2000. Harmonic serialism and parallelism .

Moreton, Elliott. 1999. Non-computable functions in optimality theory .

Riggle, Jason Alan. 2004. *Generation, recognition, and learning in finite state optimality theory*: Citeseer dissertation.