Iconicity at the Level of Formal Complexity

Olivia Griffin (UCSD) and Jia He Sun (Queen’s University)

Introduction

What is Iconicity?

- When the form of a linguistic item represents its meaning in some way, the form and the meaning are in an iconic pairing

Introduction

What is Iconicity?

- When the form of a linguistic item represents its meaning in some way, the form and the meaning are in an iconic pairing

  E.g. Balinese (Arka and Dalrymple, 2017)

cicing	'dog(s) (any number)'
cicing-cicing	'dogs (plural)'

- Multiple copies represents multiple individuals = iconicity

Introduction

Identifying Iconicity

- Identify a property that can be present in both forms and meanings
- Determine if the expression of this property on the side of form can covary with its expression on the side of meaning

Introduction

To what extent is iconicity part of language?

Nothing in language is iconic

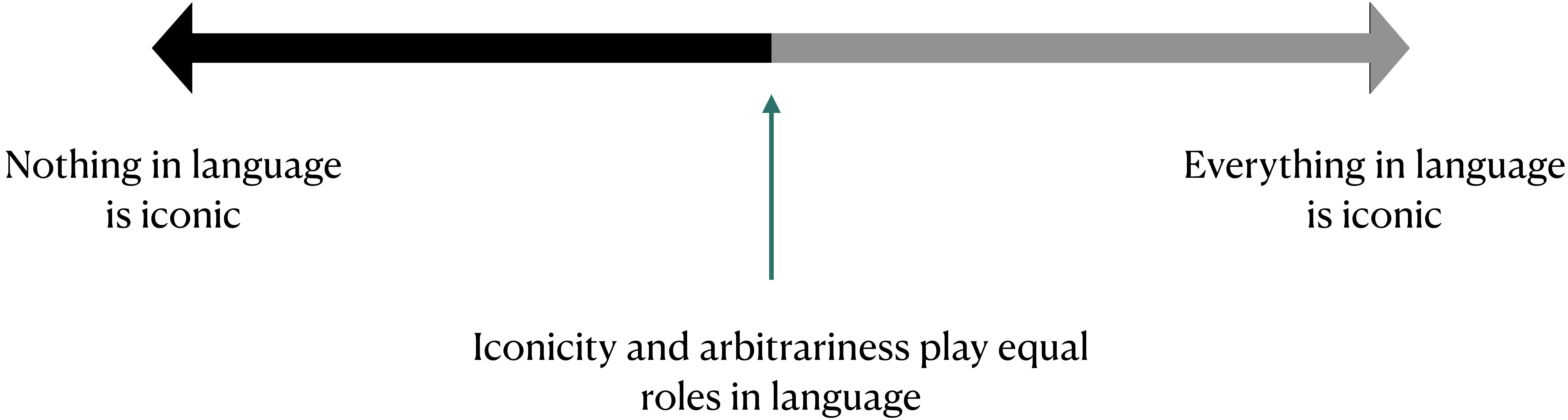
Everything in language is iconic

Introduction

To what extent is iconicity part of language?

Introduction

To what extent is iconicity part of language?



Introduction

To what extent is iconicity part of language?

Introduction

To what extent is iconicity part of language?

Where does human language actually fall?

Nothing in language is iconic

Iconicity and arbitrariness play equal roles in language

Introduction

To what extent is iconicity part of language?

Where does human language actually fall?

Nothing in language is iconic

Iconicity and arbitrariness play equal roles in language

A computational approach can help us to answer this question!

Introduction

To what extent is iconicity part of language?

- Computational approaches allow us to identify certain types of iconic patterns
    - Better picture of the distribution of iconic patterns in language
    - Better understanding of the range of applications of iconicity in language

- Hypothesis: languages can manifest iconicity by marking more complex semantic processes via more complex morphophonological processes
    - Higher complexity on the side of form represents higher complexity on the side of meaning, and lower complexity on the side of meaning represents lower complexity on the side of form

Project Overview

Empirical phenomenon

- Pattern: event-external pluractionality is marked via non-reduplicative affixation, and event-internal pluractionality is marked via reduplication
  - Found in Karuk[1], Yurok[2], and Kaqchikel[3]
- Hypothesis:
  - Complexity difference between event-internal pluractionals and event-external pluractionals
  - Complexity difference between reduplication and non-reduplicative affixation
  - More complex morphophonological process marks more complex semantic process

1. Conathan and Wood (2003)
2. Conathan and Wood (2003)
3. Henderson (2012)

Project Overview

Empirical phenomenon

Event-external pluractionals (EPs)

Event-internal pluractionals (IPs)

Project Overview

Empirical phenomenon

Event-external pluractionals (EPs)

- Set of events of the same type

Event-internal pluractionals (IPs)

- Single complex events that are composed of repetitions of (a subphase of) an event

Project Overview

Empirical phenomenon

Event-external pluractionals (EPs)

- Set of events of the same type

E.g. *Event-external pluractionaliy in Kaqchikel*

X-i-tzuy-e' *'I sat'*
COM-A1s-sit-P.ITV

X-i-tzuy-ulöj *'I sat many times'*
COM-A1s-sit-PLRCT

- Non-reduplicative affixation

Event-internal pluractionals (IPs)

- Single complex events that are composed of repetitions of (a subphase of) an event

Project Overview

Empirical phenomenon

Event-external pluractionals (EPs)

- Set of events of the same type

E.g. *Event-external pluractionaliy in Kaqchikel*

X-i-tzuy-e' *'I sat'*
COM-A1s-sit-P.ITV

X-i-tzuy-ulöj *'I sat many times'*
COM-A1s-sit-PLRCT

- Non-reduplicative affixation

Event-internal pluractionals (IPs)

- Single complex events that are composed of repetitions of (a subphase of) an event

E.g. *Event-internal pluractionaliy in Kaqchikel*

X-i-tzuy-e' *'I sat'*
COM-A1s-sit-P.ITV

X-i-tzuy-utza' *'I made the motion of sitting there repeatedly'*
COM-A1s-sit-PLRCT

SPEAKER COMMENT: your bottom doesn't really hit the chair

- *-Ca'* reduplication

Project Overview

Goal

- Pattern: marking EPs via non-reduplicative affixation, and IPs via reduplication
    - Not obviously iconic on the surface

- Goal: determine whether the pattern is iconically motivated at the level of formal complexity
    - Complexity difference between IPs and EPs
    - Complexity difference between reduplication and non-reduplicative affixation
    - More complex morphophonological process marking more complex semantic process

Project Overview

To what extent is iconicity part of language?

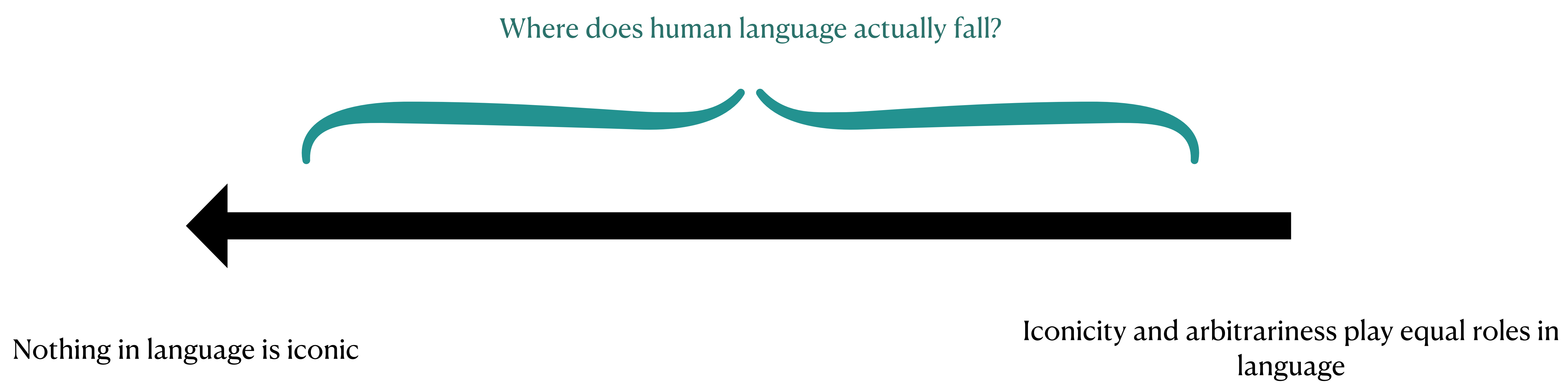
Where does human language actually fall?

Nothing in language is iconic

Iconicity and arbitrariness play equal roles in language

Project Overview

To what extent is iconicity part of language?



- For IPs, a repetition of events is represented by repetition of part of the form of the verb
  - Iconic!

- Hypothesis: iconicity motivates the form-meaning pairing for **both** IPs and EPs
  - Looking at iconicity at a deeper level (formal complexity) will reveal that iconicity plays a larger role in shaping the system
  - Test via a computational approach

Computational Approach

Iconicity at the level of formal complexity

- Calculate the relative complexity of reduplication and non-reduplicative affixation
- Calculate the relative complexity of IPs and EPs
- The pattern is iconic if the more complex marking strategy is applied the more complex semantics

Computational Approach

Modeling the form of verbs

- Determining the relative complexity of morphophonological processes begins with a model of the morphophonological forms of words

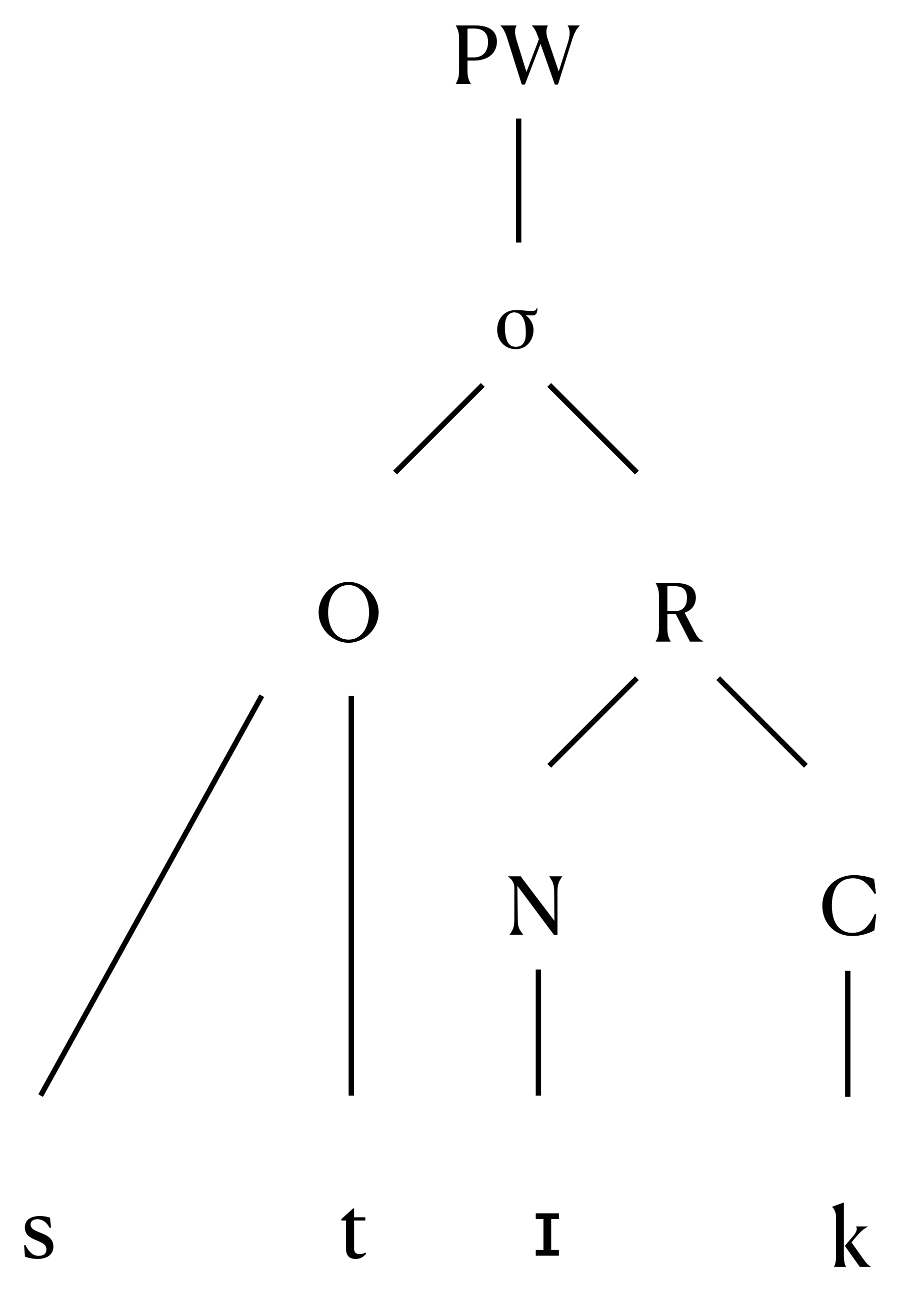
Computational Approach

Modeling the form of verbs

- Determining the relative complexity of morphophonological processes begins with a model of the morphophonological forms of words
- Forms modeled as sets

Computational Approach

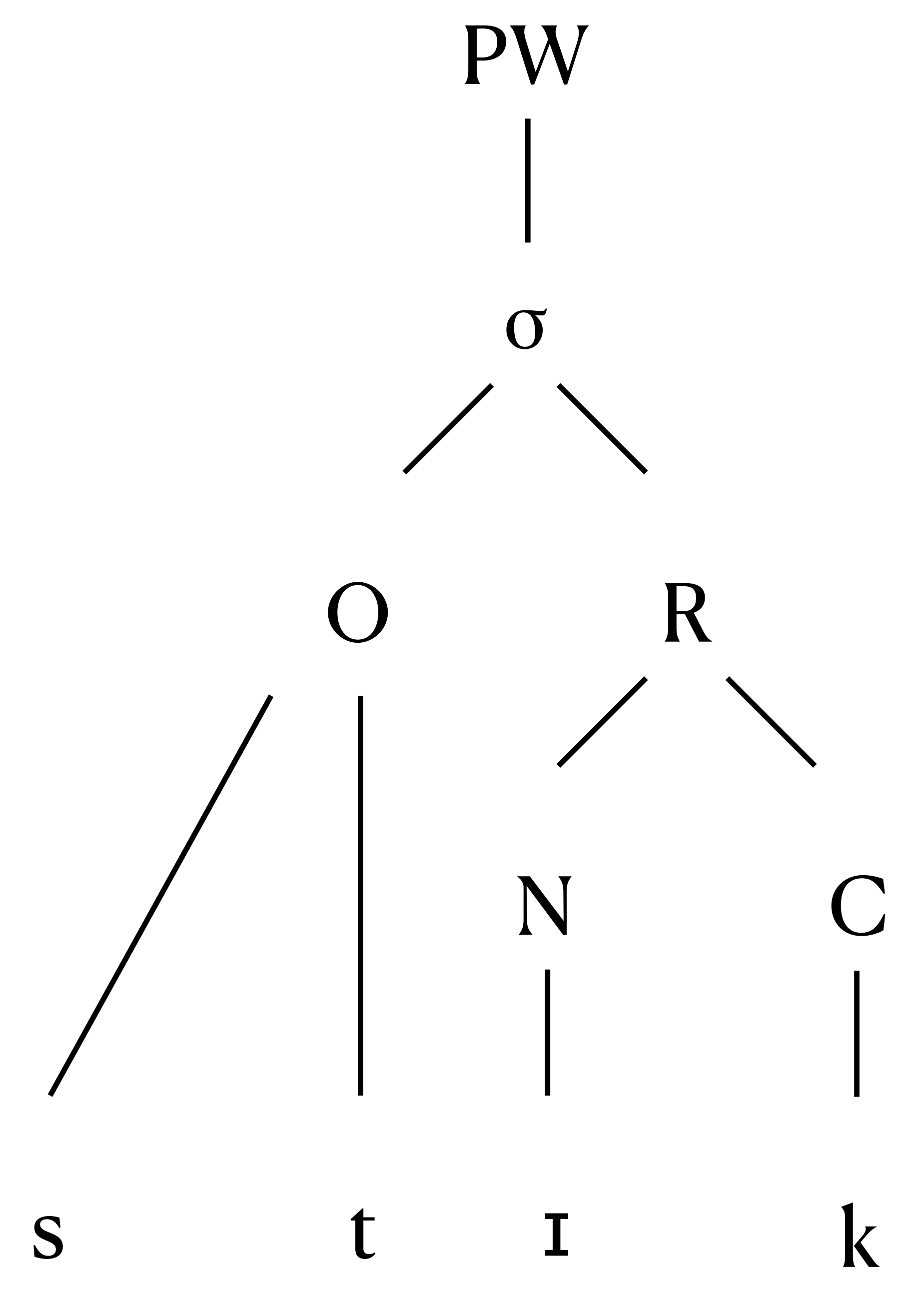
Modeling the form of verbs



- Begin with a familiar phonological representation

Computational Approach

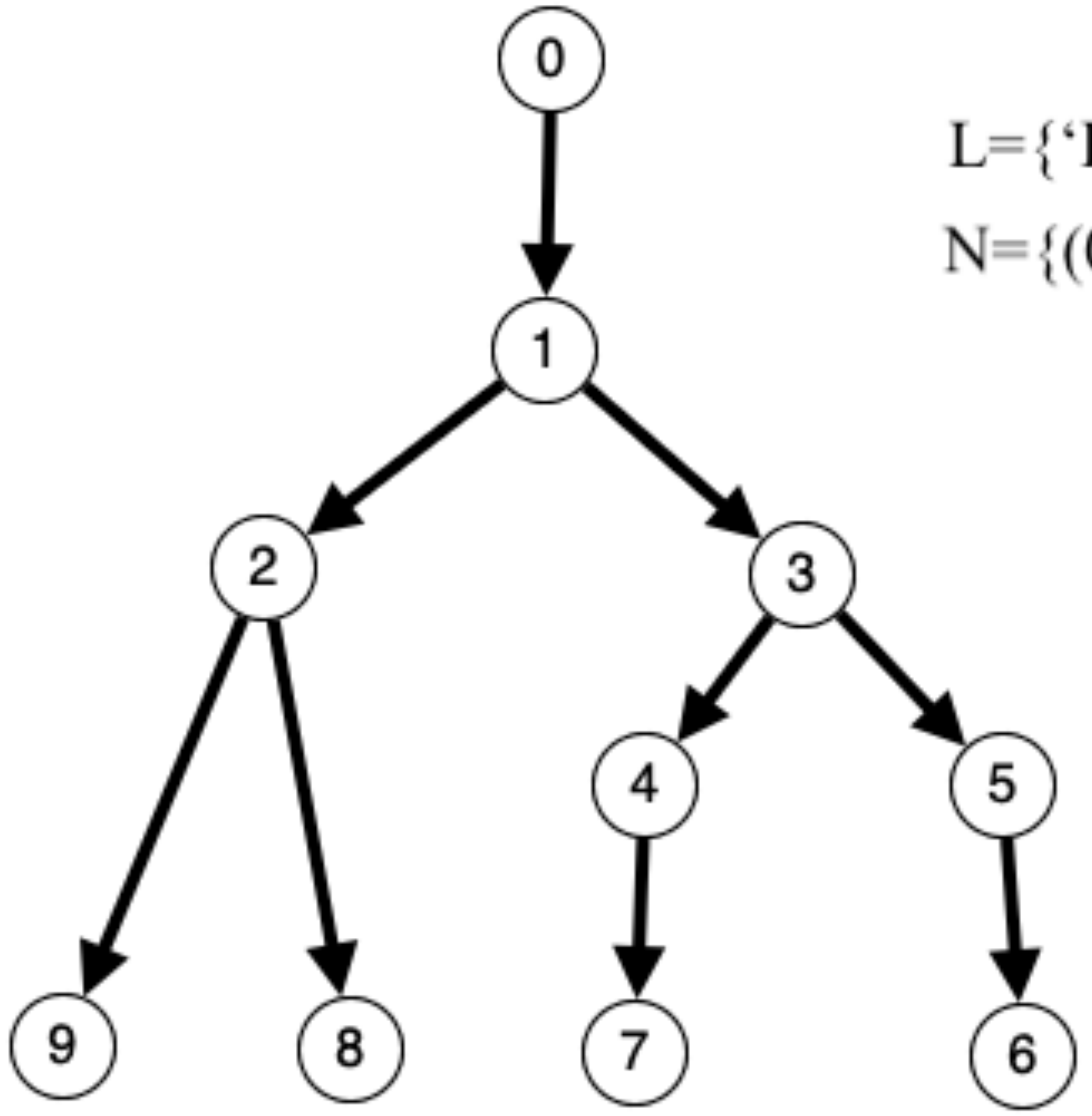
Modeling the form of verbs



- Begin with a familiar phonological representation
- Translate into a formal tree:
  - Replace phonological symbols with numbered vertices
  - Replace lines with directed edges
  - Define a set of labels for the vertices
  - Define a function that assigns labels to vertices

Computational Approach

Modeling the form of verbs



L={'Pw', 'σ', 'O', 'R', 'N', 'C', 's', 't', 'I', 'k'}

N={(0, 'Pw'), (1, 'σ'), (2, 'O'), (3, 'R'), (4, 'N'), (5, 'C'), (9, 's'), (8, 't'), (7, 'I'), (6, 'k')}

- Begin with a familiar phonological representation
- Translate into a formal tree:
  - Replace phonological symbols with numbered vertices
  - Replace lines with directed edges
  - Define a set of labels for the vertices
  - Define a function that assigns labels to vertices

Computational Approach

Modeling the form of verbs

- Translate the tree into set-theoretic representation:

Computational Approach

Modeling the form of verbs

- Translate the tree into set-theoretic representation:

$$U = \{V \sqcup L\}$$

$$V = \{0,1,2,3,4,5,6,7,8,9\}$$

$$L = \{'Pw','\sigma','O','R','N','C','s','t','\text{ɪ}','k',\}$$

$$D = \{(0,1),(1,2),(1,3),(2,9),(2,8),(3,4),(3,5),(4,7),(5,6)\}$$

$$P = \{(2,3),(4,5),(9,8)\}$$

$$N = \{(0,'Pw'),(1,'o'),(2,'O'),(3,'R'),(4,'N'),(5,'C'),(9,'s'),(8,'t'),(8,'t'),(7,'i'),(6,'k')\}$$

Computational Approach

Modeling the form of verbs

- The full model of a verb's morphophonological form contains:
  - set-theoretic representation of the verb's phonological tree
  - set-theoretic representation of the verb's morphological tree
  - set that assigns objects from the phonological tree to objects from the morphological tree

Computational Approach

Modeling the form of verbs

$$W = \{P,\ M,\ C_{MP}\}$$

Phonological

$$P = \{U_{phon},\ V_{phon},\ L_{phon},\ D_{phon},\ B_{phon},\ N_{phon}\}$$
$$U_{phon} = \{V_{phon} \cup L_{phon}\}$$
$$V_{phon} = \{1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10,\}$$
$$L_{phon} = \{'Pw',\ 'σ',\ 'O',\ 'R',\ 'N',\ 'C',\ 's',\ 't',\ 'ɪ',\ 'k'\}$$
$$D_{phon} = \{(1,2),\ (2,\ 3),\ (2,\ 4),\ (3,\ 7),\ (3,\ 8),\ (4,\ 5),\ (4,\ 6),\ (5,\ 9),\ (6,\ 10)\}$$
$$Pr_{phon} = \{(3,\ 4),\ (5,\ 6),\ (7,\ 8)\}$$
$$N_{phon} = \{(1,\ 'Pw'),\ (2,\ 'σ'),\ (3,\ 'O'),\ (4,\ 'R'),\ (5,'N'),\ (6,\ 'C'),\ (7,\ 's'),\ (8,\ 't'),\ (9,\ 'ɪ'),\ (10,'k')\}$$

Morphological

$$M = \{U_{morph},\ V_{morph},\ L_{morph},\ D_{morph},\ B_{morph},\ N_{morph}\}$$
$$U_{morph} = \{V_{morph} \cup L_{morph}\}$$
$$V_{morph} = \{100,\ 101\}$$
$$L_{morph} = \{'Mw',\ 'Mst'\}$$
$$D_{morph} = \{(100,\ 101)\}$$
$$Pr_{morph} = \{ø\}$$
$$N_{morph} = \{(100,\ 'W'),\ (101,\ 'Mst')\}$$

Correspondences →

$$C_{MP} = \{(1,\ 100),\ (7,\ 101),\ (8,\ 101),\ (9,\ 101),\ (10,\ 101)\}$$

Computational Approach

Modeling morphophonological processes

- Model verbs as sets
- Define a series of algorithms that create the set representing an affixed verb from the set representing an unaffixed verb

Computational Approach

Modeling morphophonological processes

- Process for the reduplication series of algorithms:
  - Identify the base for reduplication within the unreduplicated verb
  - Determine which items in the model of the base should be present in the model of the reduplicant
  - Create the model of the reduplicant using identified items from the base and other items pre-specified in the algorithm (e.g. fixed segments)
  - Concatenate the reduplicant with the base

Computational Approach

Modeling morphophonological processes

- Process for the series of algorithms:
  - Identify the base for reduplication within the unreduplicated verb
  - Determine which items in the model of the base should be present in the model of the reduplicant
  - Create the model of the reduplicant using identified items from the base and other items pre-specified in the algorithm (e.g. fixed segments)
  - Concatenate the reduplicant with the base

Computational Approach

Modeling morphophonological processes

- In this model, the base for reduplication is always a morphological constituent of the unreduplicated verb
- The single vertex corresponding to that morphological constituent is given by applying the function **base(**x**)** to the set V-morph
- The output of **base(**V-morph**)** is defined in the input

Computational Approach

Modeling morphophonological processes

Algorithm for defining the morphological content of the base for reduplication (input: W)

A. put the vertex output by **base(**V-morph**)** into the set V'-morph
B. check each pair of vertices (x, y) in the set D-morph and identify the ones whose first element is the same as the output of **base(**V-morph**)**
C. for every pair of vertices identified in step (b), put the second vertex in the pair into the set V'-morph
   LOOP until no new additions to V'-morph:
D. check each pair of vertices (x, y) in the set D-morph and identify the ones whose first element is in the set V'-morph
E. for every pair of vertices identified in step (d), put the second vertex in the pair into the set V'-morph
   END LOOP
F. check each pair of vertices (x, y) in the set Pr-morph and identify the ones where both x and y are in the set V'-morph
G. put every pair identified in step f into the set Pr'-morph
H. check each pair of vertices (x, y) in the set N-morph and identify the ones whose first element is in the set V'-morph
I. put every pair identified in step (h) into the set N'-morph
J. put every label in the set L-morph into the set L'-morph
K. put every element that is in the set V'-morph or the set L'-morph into the set U'-morph
L. output M'

Computational Approach

Modeling morphophonological processes

A. put the vertex output by **base(**V-morph**)** into the set V’-morph
B. check each pair of vertices (x, y) in the set D-morph and identify the ones whose first element is the same as the output of **base(**V-morph**)**
C. for every pair of vertices identified in step (b), put the second vertex in the pair into the set V’-morph

- Copy the vertex corresponding to the top of the base’s morphological tree, along with all of the vertices that it directly dominates

Computational Approach

Modeling morphophonological processes

LOOP until no new additions to V'-morph:

D. check each pair of vertices (x, y) in the set D-morph and identify the ones whose first element is in the set V'-morph

E. for every pair of vertices identified in step (d), put the second vertex in the pair into the set V'-morph

END LOOP

- Copy everything else dominated by the top of the base's morphological tree
  - Loop = higher complexity for reduplication
- Result: the entire morphological structure is identified and copied into a set called 'base'

Computational Approach

Modeling morphophonological processes

F. check each pair of vertices (x, y) in the set Pr-morph and identify the ones where both x and y are in the set V'-morph
G. put every pair identified in step f into the set Pr'-morp
H. check each pair of vertices (x, y) in the set N-morph and identify the ones whose first element is in the set V'-morph
I. put every pair identified in step (h) into the set N'-morph

- Copy all of the relevant precendence relations, and the part of the naming function that is relevant to the base

Computational Approach

Modeling morphophonological processes

J. put every label in the set L-morph into the set L’-morph
K. put every element that is in the set V’-morph or the set L’-morph into the set U’-morph
L. output M’

- Copy over all of the labels and create the universe set for the base
- Output the completed set that represents the base

Computational Approach

Calculating complexity

Computational complexity:

- Expressed in terms of input size (e.g. n, n^2, n^3...)
- Can be conceptualized as estimate of the number of steps to complete a task (based on the size of the input)
    - This estimate is based on complexity as well as the size of the input
- Overall complexity = number of steps asymptotically
    - Algorithms are distinct in terms of complexity if the term of highest order is different (n vs. n^2 vs. n^3...)

Computational Approach

Calculating computational complexity

Non-reduplicative affixation

- Input: model of the verb (form), pre-specified information (e.g. content of the affix, affixation site)
- Highest complexity of operations: $O(n)$

Reduplication

- Input: model of the verb (form), pre-specified information (e.g. fixed segments, base constituent)
- Highest complexity of operations: $O(n^3)$
  - Contributed by the loop

Computational Approach

Meaning

- Same strategy:
  - Model the semantics of verbs as sets
    - [[verb]] = $\{\{e_1, x_{11}, x_{12}, \ldots\}, \{e_2, x_{21}, x_{22}, \ldots\}, \ldots, \{e_m, x_{m1}, x_{m2}, \ldots\}\}$ where, for all $i \in \{1, \ldots, m\}$, $e_i \in$ **property_v**(D_ϵ), and for all $j \in \{1, \ldots, m\}$ and $k \in \mathbb{N}$, $x_{jk} \in$ **agent**(D_e) or **theme**(D_e). In other words, this is the set of all sets that meet the requirements for the verb.
  - Define a series of algorithms that take you from the set representing non-pluractional semantics to the set representing pluractional semantics
  - Calculate the complexity of the IP series and the EP series

Computational Approach

Calculating computational complexity

Event-external pluractionals (EPs)

- Input: model of the verb (meaning), set of events that satisfy the base verb (e.g. set of all sitting events)
- Highest complexity of operations: no more than $O(n)$
  - Determined by the significantly smaller input

Event-internal pluractionals (IPs)

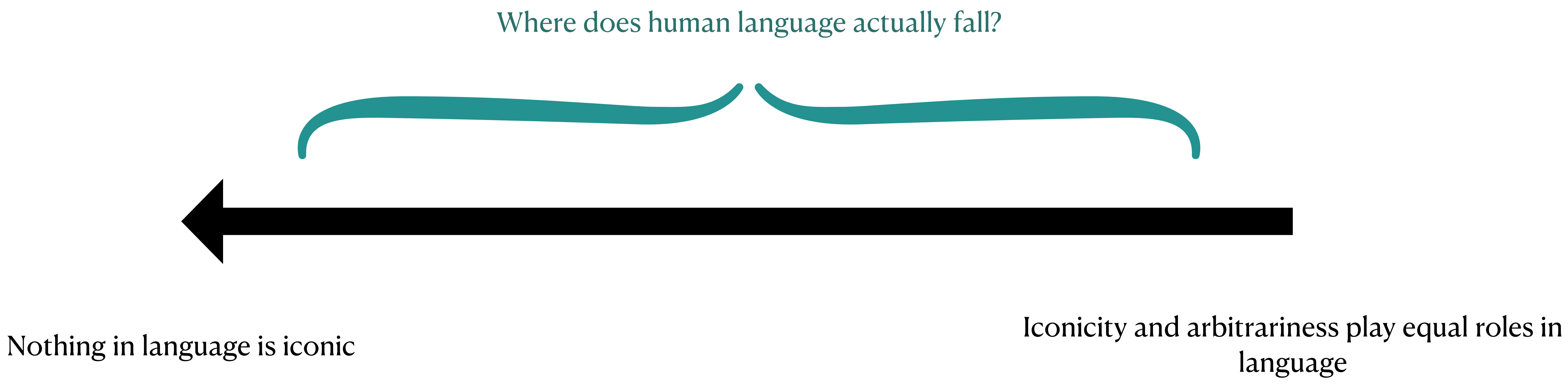
- Input: model of the verb (meaning), set of all events (including events that satisfy the base verb and events that do not satisfy the base verb)
- Highest complexity of operations: $O(n^2)$

Computational Approach

Results

- Reduplication (more complex) marks IPs (more complex)
- Non-reduplicative affixation (less complex) marks EPs (less complex)
- The pattern is iconic at the level of formal complexity

Conclusion



Two contributions:

1. Iconicity plays a larger role in the Karuk, Yurok, and Kaqchikel pattern than might be supposed by surface observation

2. Efforts to answer this question should involve a computational element to account for types of iconicity that cannot be observed on the surface

Thank you.