Computational Phonology - Class 2

Jeffrey Heinz (Instructor)
Jon Rawski (TA)

LSA Summer Institute
UC Davis
June 27, 2019

Today

1. Strings
2. String Representations (Word Models)
3. First Order Logic
4. Defining Constraints
5. Defining Transformations

Developing Logical Languages*

Ingredients

1. A *model signature* for the structures of interest
2. A *logical type* (QF, FO, MSO, QFLFP, ...)

Instructions

- Combine and stir well!

(*This is what de Lacy (2011) calls a "Constraint Definition Language")

What are we modeling?

- Strings?
- Trees?
- Syntactic structures?
- Autosegmental structures?
- Prosodic structures?
- ...

What are we modeling?

- Strings?
- Trees?
- Syntactic structures?
- Autosegmental structures?
- Prosodic structures?
- ...

We begin with *strings* because they are *simple*. Once we understand something how things work in the simple cases, we can try to understand how they work in the complex cases.

Part I

What are strings?

Strings and Stringsets

Assume a finite set of symbols. Traditionally, Σ denotes this set. Strings are built inductively with a non-commutative operation called *concatenation*.

1. Base case: λ is a string.
2. Inductive case: If u is a string and $\sigma \in \Sigma$ then $u \cdot \sigma$ is a string.

- The string λ is the *identity*. So for all strings u: $u \cdot \lambda = \lambda \cdot u = u$.

- We refer to all strings of finite length with the notation Σ^*.

A *stringset* is a (possibly infinite) subset of Σ^*.

Part II

Models

Word Models

We use the word 'word' synonymously with 'string.'

- A *model* of a word is a representation of it.

- A relational model contains two kinds of elements.
  1. **A domain.** This is a finite set of elements.
  2. **Some relations** over the domain elements.

- Guiding principles:
  1. Every word has some model.
  2. Different words must have different models.

The successor model

Let $\Sigma = \{a, b, c\}$ and suppose we wish to model strings in Σ^*.
The successor model's signature

$$\mathbb{W}^{\triangleleft} = \langle \mathcal{D}, \triangleleft, \mathtt{a}, \mathtt{b}, \mathtt{c} \rangle$$

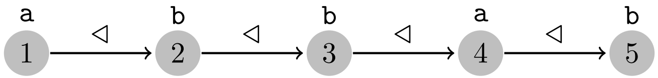
$\mathcal{D}^{\mathbb{W}}$	—	Finite set of elements (positions)
$\triangleleft^{\mathbb{W}}$	—	A binary relation encoding immediate linear precedence on $\mathcal{D}$
$\mathtt{a}, \mathtt{b}, \mathtt{c}$	—	Unary relations (so subsets of $\mathcal{D}$) encoding positions at which $\mathtt{a}$,$\mathtt{b}$,$\mathtt{c}$ occurs

Example: $\mathbb{W}^{\triangleleft}$

Consider the string *abbab*.
The model of *abbab* under the signature $\mathbb{W}^{\triangleleft}$ (denoted $\mathcal{M}^{\triangleleft}_{abbab}$) looks like this.

$$\mathcal{M}^{\triangleleft}_{abbab} = \left\langle \begin{array}{l} \{1,2,3,4,5\}, \\ \{(1,2),(2,3),(3,4),(4,5)\}, \\ \{1,4\}, \\ \{2,3,5\}, \\ \varnothing \end{array} \right\rangle$$

Illustrating the successor model of *abbab*

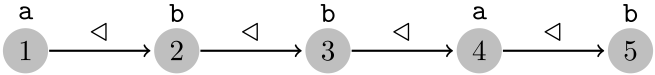


$$\mathcal{M}^{\triangleleft}_{abbab} = \left\langle \begin{array}{l} \{1,2,3,4,5\}, \\ \{(1,2),(2,3),(3,4),(4,5)\}, \\ \{1,4\}, \\ \{2,3,5\} \end{array} \right\rangle$$

In class exercise

1. Give models for these strings.
   1. *abc*
   2. *cacaca*
2. Suppose we removed the unary relations from the signature so the it looks like this: $\mathbb{W}^{\dagger} = \langle \mathcal{D}, \triangleleft \rangle$. Can models with such a signature distinguish all strings in Σ^*?
3. Suppose we removed the successor relation from the signature so it looks like this: $\mathbb{W}^{\ddagger} = \langle \mathcal{D}, \mathtt{a}, \mathtt{b}, \mathtt{c} \rangle$. Can models with such a signature distinguish all strings in Σ^*?
4. Phonological theories often uses features as representational elements, not segments. How could you define a signature for a model that refers to features? What would the model of *can* [kæn] look like?

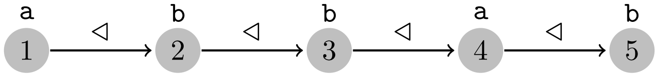
Talking Truth: Models ($\models$)



True or False?

1. $1 \lhd 2$
2. $1 \lhd 3$
3. $\lhd (1, 2)$
4. $\lhd (1, 3)$
5. $\mathtt{a}(1)$
6. $\mathtt{a}(2)$
7. $\mathtt{b}(3)$
8. $\mathtt{b}(4)$

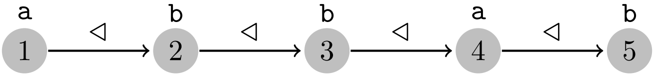
Talking Truth: Models ($\models$)



True or False?

1. $1 \lhd 2$ `True`
2. $1 \lhd 3$
3. $\lhd(1, 2)$
4. $\lhd(1, 3)$
5. $\mathtt{a}(1)$
6. $\mathtt{a}(2)$
7. $\mathtt{b}(3)$
8. $\mathtt{b}(4)$

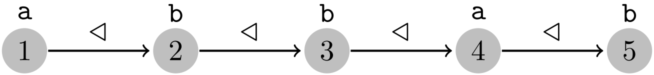
TALKING TRUTH: MODELS ($\models$)



True or False?

1. $1 \lhd 2$ `True`
2. $1 \lhd 3$ `False`
3. $\lhd\ (1,2)$
4. $\lhd\ (1,3)$
5. $\mathtt{a}(1)$
6. $\mathtt{a}(2)$
7. $\mathtt{b}(3)$
8. $\mathtt{b}(4)$

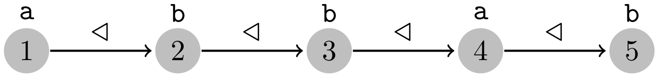
Talking Truth: Models (⊨)



True or False?

1. $1 \lhd 2$ `True`
2. $1 \lhd 3$ `False`
3. $\lhd\ (1,2)$ `True`
4. $\lhd\ (1,3)$
5. `a`(1)
6. `a`(2)
7. `b`(3)
8. `b`(4)

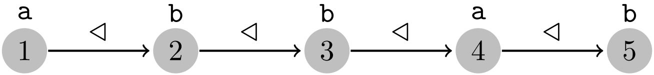
Talking Truth: Models (⊨)



True or False?

1. $1 \lhd 2$ `True`
2. $1 \lhd 3$ `False`
3. $\lhd (1,2)$ `True`
4. $\lhd (1,3)$ `False`
5. $\mathtt{a}(1)$
6. $\mathtt{a}(2)$
7. $\mathtt{b}(3)$
8. $\mathtt{b}(4)$

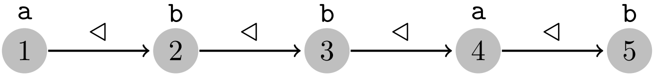
Talking Truth: Models ($\models$)



True or False?

1. $1 \lhd 2$ `True`
2. $1 \lhd 3$ `False`
3. $\lhd (1, 2)$ `True`
4. $\lhd (1, 3)$ `False`
5. $\texttt{a}(1)$ `True`
6. $\texttt{a}(2)$
7. $\texttt{b}(3)$
8. $\texttt{b}(4)$

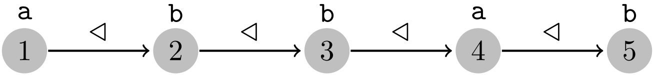
Talking Truth: Models ($\models$)



True or False?

1. $1 \lhd 2$ `True`
2. $1 \lhd 3$ `False`
3. $\lhd\ (1,2)$ `True`
4. $\lhd\ (1,3)$ `False`
5. $\mathtt{a}(1)$ `True`
6. $\mathtt{a}(2)$ `False`
7. $\mathtt{b}(3)$
8. $\mathtt{b}(4)$

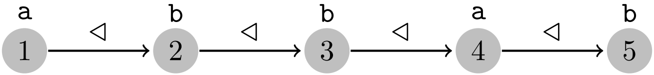
Talking Truth: Models (⊨)



True or False?

1. $1 \lhd 2$ `True`
2. $1 \lhd 3$ `False`
3. $\lhd\ (1,2)$ `True`
4. $\lhd\ (1,3)$ `False`
5. $\texttt{a}(1)$ `True`
6. $\texttt{a}(2)$ `False`
7. $\texttt{b}(3)$ `True`
8. $\texttt{b}(4)$

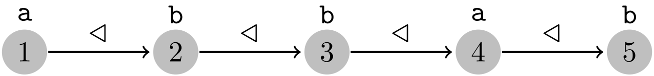
Talking Truth: Models ($\models$)



True or False?

1. $1 \lhd 2$ `True`
2. $1 \lhd 3$ `False`
3. $\lhd(1,2)$ `True`
4. $\lhd(1,3)$ `False`
5. $\mathtt{a}(1)$ `True`
6. $\mathtt{a}(2)$ `False`
7. $\mathtt{b}(3)$ `True`
8. $\mathtt{b}(4)$ `False`

Talking Truth: Models (⊨)



Whenever $R(\vec{x})$ is true in $\mathcal{M}_w$ we write

$$\mathcal{M}_w \vDash R(\vec{x})$$

"The model of w satisfies $R(\vec{x})$" or "$\mathcal{M}_w$ models $R(\vec{x})$"

Part III

Defining Constraints with FO Logic

Formulas of First Order Logic

We let $x, y, z, x_1, x_2, \ldots$ be variables. They range over elements of the domain.

- Base Cases
  1. (x=y) (equality)
  2. R(x) (for each unary relation R∊ $\mathbb{W}$)
  3. R(x,y) (for each binary relation R∊ $\mathbb{W}$)

  (binary relations are often written in infix notation as xRy)

- Inductive Cases. If φ, ψ are formulas of FO logic so are:
  1. $\neg\varphi, (\varphi \wedge \psi), (\varphi \vee \psi)$,
     $(\varphi \Rightarrow \psi), (\varphi \Leftrightarrow \psi)$ (Boolean connectives)
  2. $(\exists x)[\varphi]$ (existential quantification)
  3. $(\forall x)[\varphi]$ (universal quantification)

Defining new predicates

Defining new predicates is like writing little programs or scripts that can be used again and again. We write them from the basic aforementioned pieces.

Examples

$$
\begin{aligned}
x \neq y &\stackrel{\text{def}}{=} \neg(x = y) \\
\mathsf{first}(x) &\stackrel{\text{def}}{=} \neg(\exists y)[y \lhd x] \\
\mathsf{last}(x) &\stackrel{\text{def}}{=} \neg(\exists y)[x \lhd y] \\
\mathsf{C}_x\mathsf{C}(x) &\stackrel{\text{def}}{=} \exists(y)[y \lhd x \wedge \mathtt{cons}(x) \wedge \mathtt{cons}(y)]
\end{aligned}
$$

Interpreting sentence of FO logic (the extensions!)

- Only sentences where every variable is *bound* can be interpreted to define stringsets. Variables that are not bound are called *free*.
- The details of interpretations are in the handout. There tends to be a lot of bookkeeping to write out the definitions, but it is easy to explain with examples.
- In a nutshell: A word w *models* a sentence of FO logic φ if the sentence is true of $\mathcal{M}_w$. We write $\mathcal{M}_w \vDash \varphi$.

$$[\![\varphi]\!] = \{w \in \Sigma^* \mid \mathcal{M}_w \vDash \varphi\}$$

Examples

1. $\varphi = (\forall x)[\neg \mathsf{C}_x \mathsf{C}(x)]$

   Words like *paba* and *ana* satisfy φ but words like *pikka* and *pint* do not.

Examples

1. $\varphi = (\forall x)[\neg \mathsf{C}_x\mathsf{C}(x)]$

   Words like *paba* and *ana* satisfy φ but words like *pikka* and *pint* do not.

2. $\varphi = (\forall x)[\mathsf{first}(x) \Rightarrow \mathsf{cons}(x)]$

   Words like *paba* and *tana* satisfy φ but words like *asa* and *ilio* do not.

Examples

1. $\varphi = (\forall x)[\neg \mathsf{C}_x\mathsf{C}(x)]$

   Words like *paba* and *ana* satisfy φ but words like *pikka* and *pint* do not.

2. $\varphi = (\forall x)[\mathsf{first}(x) \Rightarrow \mathtt{cons}(x)]$

   Words like *paba* and *tana* satisfy φ but words like *asa* and *ilio* do not.

3. $\varphi = (\exists x)[\mathtt{nasal}(x)]$

   Words like *pana* and *mule* satisfy φ but words like *asa* and *lazi* do not.

Examples

1. $\varphi = (\forall x)[\neg \mathsf{C}_x \mathsf{C}(x)]$

   Words like *paba* and *ana* satisfy φ but words like *pikka* and *pint* do not.

2. $\varphi = (\forall x)[\mathsf{first}(x) \Rightarrow \mathtt{cons}(x)]$

   Words like *paba* and *tana* satisfy φ but words like *asa* and *ilio* do not.

3. $\varphi = (\exists x)[\mathtt{nasal}(x)]$

   Words like *pana* and *mule* satisfy φ but words like *asa* and *lazi* do not.

4. $\varphi = (\exists x, y, z)[\mathtt{nasal}(x) \wedge \mathtt{nasal}(y) \wedge \mathtt{nasal}(z) \wedge x \neq y \wedge x \neq z \wedge y \neq z]$

   Words like *panaman* and *mulenumina* satisfy φ but words like *asa* and *munile* do not.

In class exercise

What generalization (=markedness constraint) is this?

1 $(\forall x, y)\Big[\big(x \lhd y \wedge \texttt{nasal}(x) \wedge \texttt{cons}(y)\big) \Rightarrow \texttt{voice}(y)\Big]$

Interim Summary

We have defined our first Constraint Definition Language!

Ingredients

1. A *model signature*: $\mathcal{M}^{\triangleleft}$
2. A *logical type*: First Order Logic

This logical language is known as First Order with Successor FO($\triangleleft$).

Part IV

Analysis

Outstanding Questions

1. What constraints can we write (and not write) with FO($\triangleleft$)?

Outstanding Questions

1. What constraints can we write (and not write) with FO($\triangleleft$)?
2. This defines constraints as functions $f : \Sigma^* \rightarrow \{\texttt{True}, \texttt{False}\}$? How do we count violations? Assign probabilities?

Outstanding Questions

1. What constraints can we write (and not write) with FO($\triangleleft$)?
2. This defines constraints as functions $f : \Sigma^* \to \{\texttt{True}, \texttt{False}\}$? How do we count violations? Assign probabilities?
3. This defines constraints. How do we define transformations?

Outstanding Questions

1. What constraints can we write (and not write) with FO($\triangleleft$)?
2. This defines constraints as functions $f : \Sigma^* \rightarrow \{\texttt{True}, \texttt{False}\}$? How do we count violations? Assign probabilities?
3. This defines constraints. How do we define transformations?
4. What happens when we change the signature? What other signatures are there?

Outstanding Questions

1. What constraints can we write (and not write) with FO($\triangleleft$)?
2. This defines constraints as functions $f : \Sigma^* \rightarrow \{\texttt{True}, \texttt{False}\}$? How do we count violations? Assign probabilities?
3. This defines constraints. How do we define transformations?
4. What happens when we change the signature? What other signatures are there?
5. What happens when we change the logic? What other logics are there?

What constraints can we write (and not write) with FO($\triangleleft$)?

Theorem

A constraint is FO-definable with successor if and only if there are two natural numbers k and t such that for any two strings w and v, if w and v contain the same substrings x of length k *the same number of times counting only up to* t, then either both w and v violate the constraint or neither does.

In other words, FO($\triangleleft$) *cannot distinguish* two strings which have the same number of substrings x of length k (counting up to some threshold t).

(Thomas 1982, "Classifying regular events in symbolic logic")

Some Phonology

Kikongo
/ku-kinis-i**l**-a/ becomes [kukinisi**n**a] ‘to make dance for’

A Common Analysis
This alternation is motivated by the following constraint:

- *N..L: Laterals cannot follow nasals at *any* distance.

(Odden 2004)

*N..L cannot be expressed with FO($\triangleleft$)

Proof

Pick any k and t. Compare $w = a^k n a^k \ell a^k$ with $v = a^k \ell a^k n a^k$.

count	$w = \rtimes a^k n a^k \ell a^k \ltimes$	Notes
1	$\rtimes a^{k-1}$	
3	a^k	
1	$a^i n a^j$	(for each $0 \le i, j \le k-1$, $i + j = k - 1$)
1	$a^i \ell a^j$	(for each $0 \le i, j \le k-1$, $i + j = k - 1$)
1	$a^{k-1} \ltimes$	

count	$v = \rtimes a^k \ell a^k n a^k \ltimes$	Notes
1	$\rtimes a^{k-1}$	
3	a^k	
1	$a^i n a^j$	(for each $0 \le i, j \le k-1$, $i + j = k - 1$)
1	$a^i \ell a^j$	(for each $0 \le i, j \le k-1$, $i + j = k - 1$)
1	$a^{k-1} \ltimes$	

How can we express the constraint *N..L?

Two options

1. Increase the power of the logic
2. Change the representation

Part V

The Precedence Model

The precedence model

Let $\Sigma = \{a, b, c\}$ and suppose we wish to model strings in Σ^*.

The precedence model's signature

$$\mathbb{W}^{\triangleleft} = \langle \mathcal{D}, <, \mathtt{a}, \mathtt{b}, \mathtt{c} \rangle$$

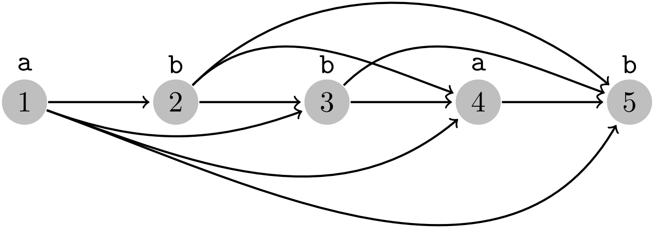
$\mathcal{D}^{\mathbb{W}}$	—	Finite set of elements (positions)
$<^{\mathbb{W}}$	—	A binary relation encoding **general linear precedence** on $\mathcal{D}$
$\mathtt{a}, \mathtt{b}, \mathtt{c}$	—	Unary relations (so subsets of $\mathcal{D}$) encoding positions at which $\mathtt{a}$,$\mathtt{b}$,$\mathtt{c}$ occurs

Example: $\mathbb{W}^<$

Consider the string *abbab*.
The model of *abbab* under the signature $\mathbb{W}^<$ (denoted $\mathcal{M}^<_{abbab}$) looks like this.

$$
\mathcal{M}^<_{abbab} = \left\langle \begin{array}{l}
\{1,2,3,4,5\}, \\
\{(1,2),(1,3),(1,4),(1,5),(2,3), \\
\quad (2,4),(2,5),(3,4),(3,5),(4,5)\} \\
\{1,4\}, \\
\{2,3,5\}, \\
\varnothing
\end{array} \right\rangle
$$

Illustrating the precedence model of *abbab*



$$\mathcal{M}^{<}_{abbab} = \left\langle \begin{array}{l} \{1,2,3,4,5\}, \\ \{(1,2),(1,3),(1,4),(1,5),(2,3), \\ \quad (2,4),(2,5),(3,4),(3,5),(4,5)\} \\ \{1,4\}, \\ \{2,3,5\}, \\ \varnothing \end{array} \right\rangle$$

The constraint *N..L

- *N..L: Laterals cannot follow nasals at *any* distance.

$$\ast N..L \stackrel{\text{def}}{=} \forall x, y\big[\texttt{nasal}(x) \land \texttt{lateral}(y) \Rightarrow \neg(x < y)\big]$$

Revisiting Outstanding Questions

1. What constraints can we write (and not write) with FO(<)?
2. This defines constraints as functions $f : \Sigma^* \rightarrow \{\texttt{True}, \texttt{False}\}$? How do we count violations? Assign probabilities?
3. This defines constraints. How do we define transformations?
4. What happens when we change the signature? What other signatures are there?
5. What happens when we change the logic? What other logics are there?

Revisiting Outstanding Questions

1. What constraints can we write (and not write) with FO(<)?
2. This defines constraints as functions $f : \Sigma^* \to \{\texttt{True}, \texttt{False}\}$? How do we count violations? Assign probabilities?
3. This defines constraints. How do we define transformations?
4. What happens when we change the signature?

   What other signatures are there?
5. What happens when we change the logic? What other logics are there?

Part VI

Summary

Summary

1. We learned the successor model for words.
2. We learned how to express constraints in First Order Logic with this model.
3. We encountered some limitations in the expressivity of this class.
4. There are many paths forward from FO($\triangleleft$).
5. We briefly encountered the precedence model for words.