Morphologically simplex D-quantifiers are strictly 2-local

Thomas Graf

Stony Brook University
mail@thomasgraf.net
https://thomasgraf.net

SCiL 2024
June 27–29, 2024

Take-Home Message

What?

Morphologically simplex D-quantifiers **every**, **some**, **no**, **most**; but not **all but one**, **not all**, **between three and five**, **an even number**, . . .

are strictly 2-local have truth conditions that can be defined with sets of permitted bigrams

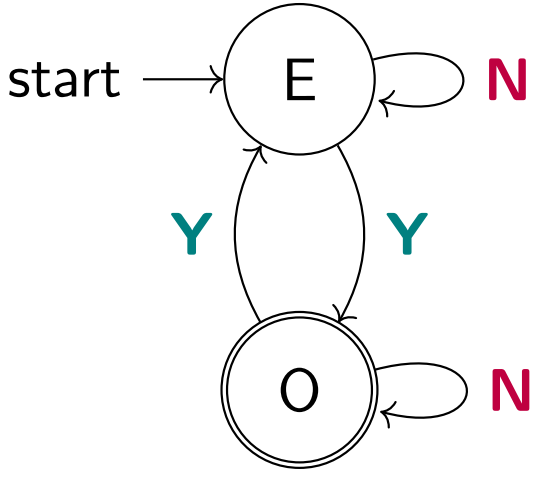
Who gives a hoot?

- explain typology via computational complexity
- linguistic universal on how much meaning fits in a lexical item
- identify abstract properties that are shared across phonology, morphology, syntax, semantics

Outline

Semantic automata approach

- **Quantifier languages**
  Meanings as strings of truth values
  (van Benthem 1986)
- distinguish quantifiers
  based on their **complexity**



regular	**context-free**
every	most
no	(at least) a third
some	
not all	
all but one	
an even number	

Evaluating the truth of quantifiers

(1) a. Every student cheated.
b. No student cheated.
c. Some student cheated.
d. Three students cheated.

students	John	Mary	Sue
cheated	yes	no	yes
string	**Y**	**N**	**Y**

- (1a): **False**, because the string contains a **N**
- (1b): **False**, because the string contains a **Y**
- (1c): **True**, because the string contains a **Y**
- (1d): **False**, because the string does not contain three **Y**s

Evaluating the truth of quantifiers

(1) a. Every student cheated.
b. No student cheated.
c. Some student cheated.
d. Three students cheated.

students	John	Mary	Sue
cheated	yes	no	yes
string	**Y**	**N**	**Y**

- (1a): **False**, because the string contains a **N**
- (1b): **False**, because the string contains a **Y**
- (1c): **True**, because the string contains a **Y**
- (1d): **False**, because the string does not contain three **Y**s

Formalization step 1: Binary string languages

Idea: Convert relation between sets **A** and **B** into set of **Yes/No-strings**

Definition (Binary string language)

1. **A**, **B**: arbitrary sets
2. **f**(**A**, **B**): maps each **a** $\in$ **A** to **Y** if **a** $\in$ **B**, otherwise **N**
3. **e**(**A**): arbitrary enumeration of **A**
4. **L**(**A**, **B**): **all** **e**(**A**), relabeled by **f**(**A**, **B**)

Example

1. **Set of students**: {John, Mary, Sue}
   Set of cheaters: {John, Sue, Bill, Peter}
2. $f(A, B)$:
$$\begin{aligned} \text{John} &\mapsto \mathbf{Y} \\ \text{Mary} &\mapsto \mathbf{N} \\ \text{Sue} &\mapsto \mathbf{Y} \end{aligned}$$
3. $e(A)$:

1)	John	Mary	Sue
2)	John	Sue	Mary
3)	Mary	John	Sue
4)	Mary	Sue	John
5)	Sue	John	Mary
6)	Sue	Mary	John

4. $L(A, B)$: $\left\{ \begin{array}{c} \mathbf{YNY}, \\ \mathbf{YYN}, \\ \mathbf{NYY} \end{array} \right\}$

Formalization step 2: Quantifier language

Idea: Every quantifier is a set of **acceptable Yes/No-strings**

Definition (Quantifier language)

$\mathbf{L}(\mathbf{Q})$ is the **quantifier language** of $\mathbf{Q}$ iff it holds for all $\mathbf{A}$ and $\mathbf{B}$ that $\mathbf{Q}(\mathbf{A}, \mathbf{B})$ is true iff $\mathbf{L}(\mathbf{A}, \mathbf{B}) \subseteq \mathbf{L}(\mathbf{Q})$.

Example

- $\mathbf{L}(\textbf{every})$ = set of all strings containing no $\mathbf{N}$
- Why?
  - $\textbf{every}(\mathbf{A}, \mathbf{B})$ iff $\mathbf{A} \subseteq \mathbf{B}$
  - If $\mathbf{A} \subseteq \mathbf{B}$, then no binary string contains $\mathbf{N}$.
  - If some binary string contains $\mathbf{N}$, then $\mathbf{A} \not\subseteq \mathbf{B}$.

Formalization step 2: Quantifier language

Idea: Every quantifier is a set of **acceptable Yes/No-strings**

Definition (Quantifier language)

$\mathsf{L}(\mathsf{Q})$ is the **quantifier language** of Q iff it holds for all A and B that $\mathsf{Q}(\mathsf{A}, \mathsf{B})$ is true iff $\mathsf{L}(\mathsf{A}, \mathsf{B}) \subseteq \mathsf{L}(\mathsf{Q})$.

Example

- $\mathsf{L}(\textbf{every})$ = set of all strings containing no N
- Why?
  - $\textbf{every}(\mathsf{A}, \mathsf{B})$ iff $\mathsf{A} \subseteq \mathsf{B}$
  - If $\mathsf{A} \subseteq \mathsf{B}$, then no binary string contains N.
  - If some binary string contains N, then $\mathsf{A} \not\subseteq \mathsf{B}$.

A sample of quantifier languages (Graf 2019)

Quantifier	Constraint
every	
no	
some	
not all	
all but one	
an even number	
most	
at least a third	

A sample of quantifier languages (Graf 2019)

Quantifier	Constraint
every	$\lvert\mathbf{N}\rvert = 0$
no	
some	
not all	
all but one	
an even number	
most	
at least a third	

A sample of quantifier languages (Graf 2019)

Quantifier	Constraint
every	$\lvert\mathbf{N}\rvert = 0$
no	$\lvert\mathbf{Y}\rvert = 0$
some	
not all	
all but one	
an even number	
most	
at least a third	

A sample of quantifier languages (Graf 2019)

Quantifier	Constraint
every	$\lvert\mathbf{N}\rvert = 0$
no	$\lvert\mathbf{Y}\rvert = 0$
some	$\lvert\mathbf{Y}\rvert \geq 1$
not all	
all but one	
an even number	
most	
at least a third	

A sample of quantifier languages (Graf 2019)

Quantifier	Constraint
every	$\lvert\mathbf{N}\rvert = 0$
no	$\lvert\mathbf{Y}\rvert = 0$
some	$\lvert\mathbf{Y}\rvert \geq 1$
not all	$\lvert\mathbf{N}\rvert \geq 1$
all but one	
an even number	
most	
at least a third	

A sample of quantifier languages (Graf 2019)

Quantifier	Constraint
every	$\lvert\mathbf{N}\rvert = 0$
no	$\lvert\mathbf{Y}\rvert = 0$
some	$\lvert\mathbf{Y}\rvert \geq 1$
not all	$\lvert\mathbf{N}\rvert \geq 1$
all but one	$\lvert\mathbf{N}\rvert = 1$
an even number	
most	
at least a third	

A sample of quantifier languages (Graf 2019)

Quantifier	Constraint
every	$\lvert\mathbf{N}\rvert = 0$
no	$\lvert\mathbf{Y}\rvert = 0$
some	$\lvert\mathbf{Y}\rvert \geq 1$
not all	$\lvert\mathbf{N}\rvert \geq 1$
all but one	$\lvert\mathbf{N}\rvert = 1$
an even number	$\lvert\mathbf{Y}\rvert$ even
most	
at least a third	

A sample of quantifier languages (Graf 2019)

Quantifier	Constraint
every	$\lvert\mathbf{N}\rvert = 0$
no	$\lvert\mathbf{Y}\rvert = 0$
some	$\lvert\mathbf{Y}\rvert \geq 1$
not all	$\lvert\mathbf{N}\rvert \geq 1$
all but one	$\lvert\mathbf{N}\rvert = 1$
an even number	$\lvert\mathbf{Y}\rvert$ even
most	$\lvert\mathbf{Y}\rvert > \lvert\mathbf{N}\rvert$
at least a third	

A sample of quantifier languages (Graf 2019)

Quantifier	Constraint
every	$\lvert\mathbf{N}\rvert = 0$
no	$\lvert\mathbf{Y}\rvert = 0$
some	$\lvert\mathbf{Y}\rvert \geq 1$
not all	$\lvert\mathbf{N}\rvert \geq 1$
all but one	$\lvert\mathbf{N}\rvert = 1$
an even number	$\lvert\mathbf{Y}\rvert$ even
most	$\lvert\mathbf{Y}\rvert > \lvert\mathbf{N}\rvert$
at least a third	$2\lvert\mathbf{Y}\rvert \geq \lvert\mathbf{N}\rvert$

A sample of quantifier languages (Graf 2019)

Quantifier	Constraint	Simplex?
every	$\lvert\mathbf{N}\rvert = 0$	Yes
no	$\lvert\mathbf{Y}\rvert = 0$	Yes
some	$\lvert\mathbf{Y}\rvert \geq 1$	Yes
not all	$\lvert\mathbf{N}\rvert \geq 1$	No
all but one	$\lvert\mathbf{N}\rvert = 1$	No
an even number	$\lvert\mathbf{Y}\rvert$ even	No
most	$\lvert\mathbf{Y}\rvert > \lvert\mathbf{N}\rvert$	Yes
at least a third	$2\lvert\mathbf{Y}\rvert \geq \lvert\mathbf{N}\rvert$	No

A sample of quantifier languages (Graf 2019)

Quantifier	Constraint	Simplex?	Comp.
every	$\lvert\mathbf{N}\rvert = 0$	Yes	REG
no	$\lvert\mathbf{Y}\rvert = 0$	Yes	REG
some	$\lvert\mathbf{Y}\rvert \geq 1$	Yes	REG
not all	$\lvert\mathbf{N}\rvert \geq 1$	No	REG
all but one	$\lvert\mathbf{N}\rvert = 1$	No	REG
an even number	$\lvert\mathbf{Y}\rvert$ even	No	REG
most	$\lvert\mathbf{Y}\rvert > \lvert\mathbf{N}\rvert$	Yes	CFL
at least a third	$2\lvert\mathbf{Y}\rvert \geq \lvert\mathbf{N}\rvert$	No	CFL

A sample of quantifier languages (Graf 2019)

Quantifier	Constraint	Simplex?	Comp.
every	$\lvert\mathbf{N}\rvert = 0$	Yes	REG
no	$\lvert\mathbf{Y}\rvert = 0$	Yes	REG
some	$\lvert\mathbf{Y}\rvert \geq 1$	Yes	REG
not all	$\lvert\mathbf{N}\rvert \geq 1$	No	REG
all but one	$\lvert\mathbf{N}\rvert = 1$	No	REG
an even number	$\lvert\mathbf{Y}\rvert$ even	No	REG
most	$\lvert\mathbf{Y}\rvert > \lvert\mathbf{N}\rvert$	Yes	CFL
at least a third	$2\lvert\mathbf{Y}\rvert \geq \lvert\mathbf{N}\rvert$	No	CFL

A sample of quantifier languages (Graf 2019)

Quantifier	Constraint	Simplex?	Comp.	Refined comp.
every	$\lvert\mathbf{N}\rvert = 0$	Yes	REG	SL-1
no	$\lvert\mathbf{Y}\rvert = 0$	Yes	REG	SL-1
some	$\lvert\mathbf{Y}\rvert \geq 1$	Yes	REG	TSL-2
not all	$\lvert\mathbf{N}\rvert \geq 1$	No	REG	TSL-2
all but one	$\lvert\mathbf{N}\rvert = 1$	No	REG	TSL-2
an even number	$\lvert\mathbf{Y}\rvert$ even	No	REG	FO_{mod}
most	$\lvert\mathbf{Y}\rvert > \lvert\mathbf{N}\rvert$	Yes	CFL	1-Counter
at least a third	$2\lvert\mathbf{Y}\rvert \geq \lvert\mathbf{N}\rvert$	No	CFL	1-Counter

A sample of quantifier languages (Graf 2019)

Quantifier	Constraint	Simplex?	Comp.	Refined comp.
every	$\lvert\mathbf{N}\rvert = 0$	Yes	REG	SL-1
no	$\lvert\mathbf{Y}\rvert = 0$	Yes	REG	SL-1
some	$\lvert\mathbf{Y}\rvert \geq 1$	Yes	REG	TSL-2
not all	$\lvert\mathbf{N}\rvert \geq 1$	No	REG	TSL-2
all but one	$\lvert\mathbf{N}\rvert = 1$	No	REG	TSL-2
an even number	$\lvert\mathbf{Y}\rvert$ even	No	REG	FO_{mod}
most	$\lvert\mathbf{Y}\rvert > \lvert\mathbf{N}\rvert$	Yes	CFL	1-Counter
at least a third	$2\lvert\mathbf{Y}\rvert \geq \lvert\mathbf{N}\rvert$	No	CFL	1-Counter

Quantifier verification: A thought experiment

1 Setup

- The Assistant Dean of the Office of Deranged Tasks has built a long line of black and white marbles that spans all the rooms of said Office.
- Mary has to determine whether **most** of the marbles are black. But it's been a long day, she doesn't want to count. What can Mary do?

2 Strategy

- Collect all marbles and try to rearrange them as follows.
- **Rule 1**: The first marble must be black.
- **Rule 2**: White marbles must not be adjacent.
- This is possible iff most of the marbles are black.

Quantifier verification: A thought experiment

1. **Setup**
   - The Assistant Dean of the Office of Deranged Tasks has built a long line of black and white marbles that spans all the rooms of said Office.
   - Mary has to determine whether **most** of the marbles are black. But it's been a long day, she doesn't want to count. What can Mary do?
2. **Strategy**
   - Collect all marbles and try to rearrange them as follows.
   - **Rule 1**: The first marble must be black.
   - **Rule 2**: White marbles must not be adjacent.
   - This is possible iff most of the marbles are black.

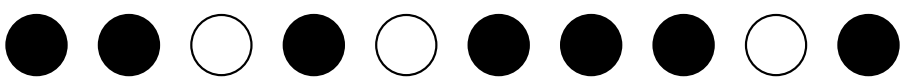
Quantifier verification: A thought experiment

1 Setup

- The Assistant Dean of the Office of Deranged Tasks has built a long line of black and white marbles that spans all the rooms of said Office.
- Mary has to determine whether **most** of the marbles are black. But it's been a long day, she doesn't want to count. What can Mary do?

2 Strategy

- Collect all marbles and try to rearrange them as follows.
- **Rule 1**: The first marble must be black.
- **Rule 2**: White marbles must not be adjacent.
- This is possible iff most of the marbles are black.



Verification patterns

- Quantifier languages give a monolithic view of meaning.
- Let's take a decompositonal hint from syntax instead:

Parser $+$ **Grammar** $\Rightarrow$ **acceptability judgment**

Verification patterns

- Quantifier languages give a monolithic view of meaning.
- Let's take a decompositonal hint from syntax instead:

Parser	+	**Grammar**	⇒	**acceptability judgment**
Verifier	+	**Pattern**	⇒	**truth-value judgment**

Verification patterns

- Quantifier languages give a monolithic view of meaning.
- Let's take a decompositonal hint from syntax instead:

Parser	+	**Grammar**	⇒	**acceptability judgment**
Verifier	+	**Pattern**	⇒	**truth-value judgment**

Definition

A set of binary strings is a **verification pattern** **V**(**Q**) for **Q** iff the permutation closure of **V**(**Q**) is the quantifier language of **Q**.

Intuition: V(Q) can omit strings that have easier equivalents

Definition

Given a class **C** of string languages, **Q** is **C-verifiable** iff **C** contains a verification pattern for **Q**.

Formal account of **most**

L(most) set of all strings with $|\mathbf{Y}| > |\mathbf{N}|$

V(most) set of all strings that are iterations of **Y**(**N**)

String	in L(Q)?	in V(Q)?
YYNYN	✓	✓
NNYNN	✗	✗
YYYNN	✓	✗

- V(most) can be described by a finite set of permitted bigrams:

$**Y**	strings can only start with **Y**
YY	**Y** can be iterated
YN	**N** can follow **Y**
(**Y**$	strings can only end with **Y**)

- Hence **most** is **SL-2 verifiable**,
  even though its quantifier language isn't even regular.

Quantifier languages and verification patterns

Quantifier	L(Q)	V(Q)
every		
no		
some		
not all		
all but one		
most		
at least a third		
an even number		

Quantifier languages and verification patterns

Quantifier	L(Q)	V(Q)
every	$\mathbf{Y}^*$	$\mathbf{Y}^*$
no		
some		
not all		
all but one		
most		
at least a third		
an even number		

Quantifier languages and verification patterns

Quantifier	L(Q)	V(Q)
every	Y*	Y*
no	N*	N*
some		
not all		
all but one		
most		
at least a third		
an even number		

Quantifier languages and verification patterns

Quantifier	L(Q)	V(Q)
every	$\mathbf{Y}^*$	$\mathbf{Y}^*$
no	$\mathbf{N}^*$	$\mathbf{N}^*$
some	$(\mathbf{N}^*\mathbf{Y}^+\mathbf{N}^*)^+$	$\mathbf{Y}^+\mathbf{N}^*$
not all		
all but one		
most		
at least a third		
an even number		

Quantifier languages and verification patterns

Quantifier	L(Q)	V(Q)
every	$\mathbf{Y}^*$	$\mathbf{Y}^*$
no	$\mathbf{N}^*$	$\mathbf{N}^*$
some	$(\mathbf{N}^*\mathbf{Y}^+\mathbf{N}^*)^+$	$\mathbf{Y}^+\mathbf{N}^*$
not all	$(\mathbf{Y}^*\mathbf{N}^+\mathbf{Y}^*)^+$	$\mathbf{N}^+\mathbf{Y}^*$
all but one		
most		
at least a third		
an even number		

Quantifier languages and verification patterns

Quantifier	L(Q)	V(Q)
every	$\mathbf{Y}^*$	$\mathbf{Y}^*$
no	$\mathbf{N}^*$	$\mathbf{N}^*$
some	$(\mathbf{N}^*\mathbf{Y}^+\mathbf{N}^*)^+$	$\mathbf{Y}^+\mathbf{N}^*$
not all	$(\mathbf{Y}^*\mathbf{N}^+\mathbf{Y}^*)^+$	$\mathbf{N}^+\mathbf{Y}^*$
all but one	$\mathbf{Y}^*\mathbf{N}\mathbf{Y}^*$	$\mathbf{N}\mathbf{Y}^*$
most		
at least a third		
an even number		

Quantifier languages and verification patterns

Quantifier	L(Q)	V(Q)
every	$\mathbf{Y}^*$	$\mathbf{Y}^*$
no	$\mathbf{N}^*$	$\mathbf{N}^*$
some	$(\mathbf{N}^*\mathbf{Y}^+\mathbf{N}^*)^+$	$\mathbf{Y}^+\mathbf{N}^*$
not all	$(\mathbf{Y}^*\mathbf{N}^+\mathbf{Y}^*)^+$	$\mathbf{N}^+\mathbf{Y}^*$
all but one	$\mathbf{Y}^*\mathbf{N}\mathbf{Y}^*$	$\mathbf{N}\mathbf{Y}^*$
most	$\lvert\mathbf{Y}\rvert > \lvert\mathbf{N}\rvert$	$(\mathbf{Y}^+\mathbf{N})^*$
at least a third		
an even number		

Quantifier languages and verification patterns

Quantifier	L(Q)	V(Q)
every	$\mathbf{Y}^*$	$\mathbf{Y}^*$
no	$\mathbf{N}^*$	$\mathbf{N}^*$
some	$(\mathbf{N}^*\mathbf{Y}^+\mathbf{N}^*)^+$	$\mathbf{Y}^+\mathbf{N}^*$
not all	$(\mathbf{Y}^*\mathbf{N}^+\mathbf{Y}^*)^+$	$\mathbf{N}^+\mathbf{Y}^*$
all but one	$\mathbf{Y}^*\mathbf{N}\mathbf{Y}^*$	$\mathbf{N}\mathbf{Y}^*$
most	$\lvert\mathbf{Y}\rvert > \lvert\mathbf{N}\rvert$	$(\mathbf{Y}^+\mathbf{N})^*$
at least a third	$2\lvert\mathbf{Y}\rvert \geq \lvert\mathbf{N}\rvert$	$(\mathbf{Y}\mathbf{Y}^+\mathbf{N})^*$
an even number		

Quantifier languages and verification patterns

Quantifier	L(Q)	V(Q)
every	$\mathbf{Y}^*$	$\mathbf{Y}^*$
no	$\mathbf{N}^*$	$\mathbf{N}^*$
some	$(\mathbf{N}^*\mathbf{Y}^+\mathbf{N}^*)^+$	$\mathbf{Y}^+\mathbf{N}^*$
not all	$(\mathbf{Y}^*\mathbf{N}^+\mathbf{Y}^*)^+$	$\mathbf{N}^+\mathbf{Y}^*$
all but one	$\mathbf{Y}^*\mathbf{N}\mathbf{Y}^*$	$\mathbf{N}\mathbf{Y}^*$
most	$\lvert\mathbf{Y}\rvert > \lvert\mathbf{N}\rvert$	$(\mathbf{Y}^+\mathbf{N})^*$
at least a third	$2\lvert\mathbf{Y}\rvert \geq \lvert\mathbf{N}\rvert$	$(\mathbf{Y}\mathbf{Y}^+\mathbf{N})^*$
an even number	$\lvert\mathbf{Y}\rvert$ even	$\lvert\mathbf{Y}\rvert$ even

Quantifier languages and verification patterns

Quantifier	L(Q)	V(Q)	Complexity drop
every	$\mathbf{Y}^*$	$\mathbf{Y}^*$	SL-1⇒SL-1
no	$\mathbf{N}^*$	$\mathbf{N}^*$	SL-1⇒ SL-1
some	$(\mathbf{N}^*\mathbf{Y}^+\mathbf{N}^*)^+$	$\mathbf{Y}^+\mathbf{N}^*$	TSL-2⇒SL-2
not all	$(\mathbf{Y}^*\mathbf{N}^+\mathbf{Y}^*)^+$	$\mathbf{N}^+\mathbf{Y}^*$	TSL-2⇒SL-2
all but one	$\mathbf{Y}^*\mathbf{N}\mathbf{Y}^*$	$\mathbf{N}\mathbf{Y}^*$	TSL-2⇒SL-2
most	$\lvert\mathbf{Y}\rvert > \lvert\mathbf{N}\rvert$	$(\mathbf{Y}^+\mathbf{N})^*$	1-counter⇒**SL-2**
at least a third	$2\lvert\mathbf{Y}\rvert \geq \lvert\mathbf{N}\rvert$	$(\mathbf{Y}\mathbf{Y}^+\mathbf{N})^*$	1-counter⇒SL-3
an even number	$\lvert\mathbf{Y}\rvert$ even	$\lvert\mathbf{Y}\rvert$ even	FO_{mod}⇒FO_{mod}

Quantifier languages and verification patterns

Quantifier	L(Q)	V(Q)	Complexity drop
every	$\mathbf{Y}^*$	$\mathbf{Y}^*$	SL-1⇒SL-1
no	$\mathbf{N}^*$	$\mathbf{N}^*$	SL-1⇒ SL-1
some	$(\mathbf{N}^*\mathbf{Y}^+\mathbf{N}^*)^+$	$\mathbf{Y}^+\mathbf{N}^*$	TSL-2⇒SL-2
not all	$(\mathbf{Y}^*\mathbf{N}^+\mathbf{Y}^*)^+$	$\mathbf{N}^+\mathbf{Y}^*$	TSL-2⇒SL-2
all but one	$\mathbf{Y}^*\mathbf{N}\mathbf{Y}^*$	$\mathbf{N}\mathbf{Y}^*$	TSL-2⇒SL-2
most	$\lvert\mathbf{Y}\rvert > \lvert\mathbf{N}\rvert$	$(\mathbf{Y}^+\mathbf{N})^*$	1-counter⇒**SL-2**
at least a third	$2\lvert\mathbf{Y}\rvert \geq \lvert\mathbf{N}\rvert$	$(\mathbf{Y}\mathbf{Y}^+\mathbf{N})^*$	1-counter⇒SL-3
an even number	$\lvert\mathbf{Y}\rvert$ even	$\lvert\mathbf{Y}\rvert$ even	FO_{mod}⇒FO_{mod}

Typological generalization

All attested simplex D-quantifiers are **SL-2 verifiable**.

(See the paper for how to tighten the typology even more.)

Infinity

- SL easily generalized to infinite strings
  (in fact, SL can't require strings to be finite)

Making sense of infinite **most**

- Many people believe that the following statement is true:
  "Most natural numbers are not prime."
- Under the proportionality definition of **most** as $|\mathbf{Y}| > |\mathbf{N}|$
  this is false because $|\mathbf{Y}| = |\mathbf{N}| = \aleph_0$.
- But the statement is **true under the SL-2 definition**:
  we can arrange the natural numbers such that
  there are never two primes in a row.
  (in fact, this requires no rearrangement at all after 3)

Pragmatics

- It is odd to say that "every married bachelor laughed" because there are no married bachelors.
- Similarly, "some bachelors are men" is odd because all bachelors are men.
- We can capture this by requiring that specific (all?) n-grams must occur in the string.
- **Formally:** semantics is SL-2, pragmatics is LT-2

Example

- **Unigrams for every**: **Y**
  if **Y** must occur, then the domain cannot be empty
- **Bigrams for some**: $**Y**, **YY**, **YN**, **NN**
  if **YN** must occur, then **some** cannot mean **every**

Typological frequency

- SL-verifiability can be tightened so that **every**, **some**, **no**, and **most** form a natural class to the exclusion of **all but one** and **not all**.
- **Observation:** within that subclass, there are multiple ways to define **every** and **some**, but just one each for **no** and **most**.
- This **matches typological frequency**: **every** and **some** are common, **no** and **most** are rare.

Parallels across language modules

- SL also shows up in phonology, morphology, and syntax.
- Verification patterns have an analogue in syntax:
  - Over strings, syntax is mildly context-sensitive.
  - But if one factors out displacement, it is context-free.
  - cf. Parikh's theorem (Parikh 1966)

Example

- MIX contains all strings over a, b, c that contain the same number of all three.
- MIX is a 2-MCFL (Salvati 2011).
- But it is the permutation closure of $(abc)^+$, which is SL-2.
- In our parlance: MIX is SL-2 verifiable.

Psychosemantics

- Experimental work on the processing of **most** seems to argue against the verification pattern I posit (Lidz et al. 2011).
- But processing observations can only argue against specific theories of semantic processing.
- Interpretation of processing behavior requires
  1. a **grammar** (= specification of the verification pattern)
  2. a **parser** (= specification of how speakers build string representations of semantic domains)
  3. a **linking theory** that maps the parser's behavior to processing predictions (e.g. via memory usage)
- I haven't said anything about 2 and 3 here.
- But at least this approach allows us to tackle those points in a rigorous manner.

Wrapping up

Linguistic puzzle

- How much meaning can be packed into simplex forms?
- For D-quantifiers, **SL-2** seems to be the limit.

Programmatic goals

- Study semantics like phonology, morphology, syntax
  formal language theory, subregular complexity
- Inventory of alternative representations of meaning
  quantifier languages, verification patterns

The future

pragmatics, typological frequency, semantic processing, learnability, verification patterns beyond D-quantifiers, ...

Thanks

The work reported in this paper was supported by
the National Science Foundation under Grant No. BCS-1845344.

References

van Benthem, Johan. 1986. Semantic automata. In *Essays in logical semantics*, 151–176. Dordrecht: Springer.

Graf, Thomas. 2019. A subregular bound on the complexity of lexical quantifiers. In *Proceedings of the 22nd Amsterdam Colloquium*, ed. Julian J. Schlöder, Dean McHugh, and Floris Roelofsen, 455–464.

Lidz, Jeff, Paul Pietroski, Tim Hunter, and Justin Halberda. 2011. Interface transparency and psychosemantics of *most*. *Natural Language Semantics* 19:227–256.

Parikh, Rohit. 1966. On context-free languages. *Journal of the Association for Computing Machinery* 13:570–581.

Salvati, Sylvain. 2011. MIX is a 2-MCFL and the word problem in $\mathbb{Z}^2$ is captured by the IO and the OI hierarchies. Technical report, INRIA Bordeaux, France.