A Subregular Bound on the Complexity of Lexical Quantifiers

Thomas Graf

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

Download the slides

Amsterdam Colloquium
December 18–20, 2019

Take-Home Message

- Study Det-quantifiers as formal languages
- Most **quantifiers are remarkably simple**.

Big picture: Cognitive parallelism

Comparable complexity across

- phonology
- morphology
- syntax
- semantics

Empirical insight: A universal of lexical quantifiers

Quantifiers that can be expressed as a single lexical item are **monotonic TSL**. (with a footnote on *most*, *half*)

Outline

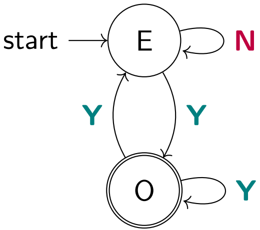
1 Quantifier languages

2 Most quantifier languages are tier-based strictly local

3 Monotonic tier projections

Semantic automata

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



regular	**context-free**
every	most
no	half
some	(at least) a third
(at least) n	
not all	
all but n	
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. $\mathbf{f}(\mathbf{A}, \mathbf{B})$: maps each $\mathbf{a} \in \mathbf{A}$ to **Y** if $\mathbf{a} \in \mathbf{B}$, otherwise **N**
3. $\mathbf{e}(\mathbf{A})$: arbitrary enumeration of **A**
4. $\mathbf{L}(\mathbf{A}, \mathbf{B})$: all $\mathbf{e}(\mathbf{A})$, relabeled by $\mathbf{f}(\mathbf{A}, \mathbf{B})$

Example

1. **Set of students**: {John, Mary, Sue}
   Set of cheaters: {John, Sue, Bill, Peter}

2. **f(A, B)**:
   - John $\mapsto$ **Y**
   - Mary $\mapsto$ **N**
   - Sue $\mapsto$ **Y**

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)**: { **YNY**, **YYN**, **NYY** }

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

- $\mathbf{L}$(**every**) $=$ set of all strings containing no $\mathbf{N}$
- Why?
  - **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)

$\mathbf{L}(\mathbf{Q})$ 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 **N**
- Why?
  - **every**($\mathbf{A}$, $\mathbf{B}$) iff $\mathbf{A} \subseteq \mathbf{B}$
  - If $\mathbf{A} \subseteq \mathbf{B}$, then no binary string contains **N**.
  - If some binary string contains **N**, then $\mathbf{A} \not\subseteq \mathbf{B}$.

A sample of quantifier languages

Quantifier	Constraint
every	
no	
some	
at least **n**	
at most **n**	
exactly **n**	
not all	
all but **n**	
an even number	
most	
half	
at least a third	

A sample of quantifier languages

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

A sample of quantifier languages

Quantifier	**Constraint**
every	$\|\mathbf{N}\| = 0$
no	$\|\mathbf{Y}\| = 0$
some	
at least **n**	
at most **n**	
exactly **n**	
not all	
all but **n**	
an even number	
most	
half	
at least a third	

A sample of quantifier languages

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

A sample of quantifier languages

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

A sample of quantifier languages

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

A sample of quantifier languages

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

A sample of quantifier languages

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

A sample of quantifier languages

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

A sample of quantifier languages

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

A sample of quantifier languages

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

A sample of quantifier languages

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

A sample of quantifier languages

Quantifier	Constraint
every	$\lvert \mathbf{N} \rvert = 0$
no	$\lvert \mathbf{Y} \rvert = 0$
some	$\lvert \mathbf{Y} \rvert \geq 1$
at least **n**	$\lvert \mathbf{Y} \rvert \geq \mathbf{n}$
at most **n**	$\lvert \mathbf{Y} \rvert \leq \mathbf{n}$
exactly **n**	$\lvert \mathbf{Y} \rvert = \mathbf{n}$
not all	$\lvert \mathbf{N} \rvert \geq 1$
all but **n**	$\lvert \mathbf{N} \rvert = \mathbf{n}$
an even number	$\lvert \mathbf{Y} \rvert$ even
most	$\lvert \mathbf{Y} \rvert > \lvert \mathbf{N} \rvert$
half	$\lvert \mathbf{Y} \rvert = \lvert \mathbf{N} \rvert$
at least a third	$3 \times \lvert \mathbf{Y} \rvert \geq \lvert \mathbf{Y} \rvert + \lvert \mathbf{N} \rvert$

A sample of quantifier languages

Quantifier	Constraint
every	$\|N\| = 0$
no	$\|Y\| = 0$
some	$\|Y\| \geq 1$
at least n	$\|Y\| \geq n$
at most n	$\|Y\| \leq n$
exactly n	$\|Y\| = n$
not all	$\|N\| \geq 1$
all but n	$\|N\| = n$
an even number	$\|Y\|$ even
most	$\|Y\| > \|N\|$
half	$\|Y\| = \|N\|$
at least a third	$3 \times \|Y\| \geq \|Y\| + \|N\|$

Lexical

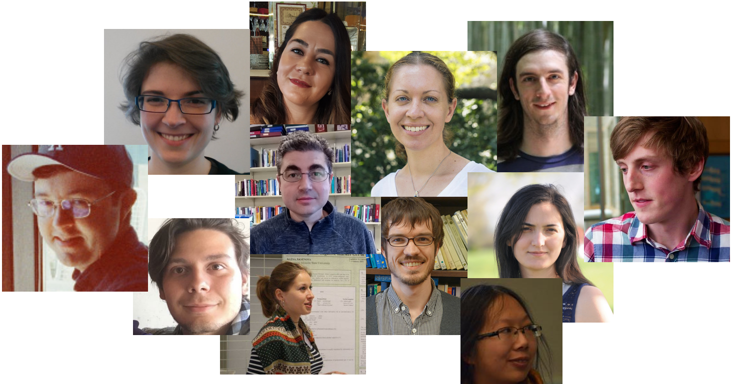
A sample of quantifier languages

Quantifier	Constraint	
every	$\lvert N \rvert = 0$	Lexical
no	$\lvert Y \rvert = 0$	Lexical
some	$\lvert Y \rvert \geq 1$	Lexical
at least n	$\lvert Y \rvert \geq n$	Lexical
at most n	$\lvert Y \rvert \leq n$	Lexical
exactly n	$\lvert Y \rvert = n$	Lexical
not all	$\lvert N \rvert \geq 1$	Regular
all but n	$\lvert N \rvert = n$	Regular
an even number	$\lvert Y \rvert$ even	Regular
most	$\lvert Y \rvert > \lvert N \rvert$	
half	$\lvert Y \rvert = \lvert N \rvert$	
at least a third	$3 \times \lvert Y \rvert \geq \lvert Y \rvert + \lvert N \rvert$	

(Lexical: every through exactly n; Regular: every through an even number.)

Subregular hierarchy

- Regular languages are not the weakest class of languages!
- There is a fine-grained **subregular hierarchy**.
- Many aspects of phonology, morphology, and syntax turn out to be subregular.
  (Heinz 2009, 2010, 2018; Chandlee 2014; Jardine 2016; McMullin 2016; Aksënova et al. 2016; Graf 2018; Shafiei and Graf 2020)



TSL: Tier-based strictly local (Heinz et al. 2011)

1. Fix a tier alphabet T.
2. Project every symbol in T to the tier.
3. Fix a finite list of forbidden substrings that may not occur on the tier.

Linguistic intuition

- inspired by autosegmental phonology (Goldsmith 1976)
- All dependencies are local if ones ignore irrelevant material.

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: ***N**

$ **Y Y Y Y** $ *$ **Y Y N Y** $

no is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: ***Y**

$ **N N N N** $ *$ **N N Y N** $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: ***N**

$ **Y Y Y Y** $
| | | | | |
$ **Y Y Y Y** $

*$ **Y Y N Y** $

no is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: ***Y**

$ **N N N N** $

*$ **N N Y N** $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: ***N**

$ **Y Y Y Y** $
| | | | | |
$ **Y Y Y Y** $

*$ **Y Y N Y** $

no is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: ***Y**

$ **N N N N** $

*$ **N N Y N** $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *N

$ Y Y Y Y $

$ Y Y Y Y $ *$ Y Y N Y $

no is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *Y

$ N N N N $ *$ N N Y N $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: ***N**

$ Y Y Y Y $

| | | | | |

$ Y Y Y Y $

*$ Y Y N Y $

no is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: ***Y**

$ N N N N $

*$ N N Y N $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: ***N**

$ **Y** **Y** **Y** **Y** $

| | | | | |

$ **Y** **Y** **Y** **Y** $

*$ **Y** **Y** **N** **Y** $

no is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: ***Y**

$ **N** **N** **N** **N** $

*$ **N** **N** **Y** **N** $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *N

$ Y Y Y Y $

$ Y Y Y Y $ *$ Y Y N Y $

no is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *Y

$ N N N N $ *$ N N Y N $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *N

$ Y Y Y Y $

$ Y Y Y Y $

*$ Y Y N Y $

no is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *Y

$ N N N N $

*$ N N Y N $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *N

$ Y Y Y Y $ → $ Y Y Y Y $

$ Y Y N Y $ → *$ Y Y N Y $

no is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *Y

$ N N N N $

*$ N N Y N $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *N

$ Y Y Y Y $ → $ Y Y Y Y $

$ Y Y N Y $ → *$ Y Y N Y $

no is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *Y

$ N N N N $

*$ N N Y N $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *N

$ Y Y Y Y $
$ Y Y Y Y $

$ Y Y N Y $
*$ Y Y N Y $

no is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *Y

$ N N N N $

*$ N N Y N $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *N

$ Y Y Y Y $ $ Y Y N Y $

$ Y Y Y Y $ *$ Y Y N Y $

no is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *Y

$ N N N N $ *$ N N Y N $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: ***N**

$ **Y Y Y Y** $ → $ **Y Y Y Y** $

$ **Y Y N Y** $ → *$ **Y Y N Y** $

no is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: ***Y**

$ **N N N N** $

*$ **N N Y N** $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet {Y, N}

Forbidden substrings: *N

$ Y Y Y Y $
| | | | | |
$ Y Y Y Y $

$ Y Y N Y $
| | | | | |
*$ Y Y N Y $

no is TSL with tier alphabet {Y, N}

Forbidden substrings: *Y

$ N N N N $

*$ N N Y N $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: ***N**

```
 $ Y Y Y Y $      $ Y Y N Y $
 | | | | | |      | | | | | |
 $ Y Y Y Y $     *$ Y Y N Y $
```

no is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: ***Y**

```
 $ N N N N $
 | | | | | |
 $ N N N N $     *$ N N Y N $
```

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: *N

$ Y Y Y Y $
| | | | | |
$ Y Y Y Y $

$ Y Y N Y $
| | | | | |
*$ Y Y N Y $

no is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: *Y

$ N N N N $
| | | | | |
$ N N N N $

*$ N N Y N $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *N

$ Y Y Y Y $ → $ Y Y Y Y $

$ Y Y N Y $ → *$ Y Y N Y $

no is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *Y

$ N N N N $ → $ N N N N $

*$ N N Y N $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: ***N**

$ Y Y Y Y $ $ Y Y N Y $

| | | | | | | | | | | |

$ Y Y Y Y $ *$ Y Y N Y $

no is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: ***Y**

$ N N N N $

| | | | | |

$ N N N N $ *$ N N Y N $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: *N

$ Y Y Y Y $
| | | | | |
$ Y Y Y Y $

$ Y Y N Y $
| | | | | |
*$ Y Y N Y $

no is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: *Y

$ N N N N $
| | | | | |
$ N N N N $

*$ N N Y N $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: ***N**

$ **Y Y Y Y** $
| | | | | |
$ **Y Y Y Y** $

$ **Y Y N Y** $
| | | | | |
*$ **Y Y N Y** $

no is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: ***Y**

$ **N N N N** $
| | | | | |
$ **N N N N** $

*$ **N N Y N** $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: ***N**

$ Y Y Y Y $ | $ Y Y N Y $
| | | | | | | | | | | | |
$ Y Y Y Y $ | *$ Y Y N Y $

no is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: ***Y**

$ N N N N $
| | | | | |
$ N N N N $ | *$ N N Y N $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *N

```
$ Y Y Y Y $        $ Y Y N Y $
| | | | | |        | | | | | |
$ Y Y Y Y $       *$ Y Y N Y $
```

no is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *Y

```
$ N N N N $        $ N N Y N $
| | | | | |        | | | | | |
$ N N N N $       *$ N N Y N $
```

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: ***N**

$ Y Y Y Y $
$ Y Y Y Y $

$ Y Y N Y $
*$ Y Y N Y $

no is TSL with tier alphabet {**Y**, **N**}

Forbidden substrings: ***Y**

$ N N N N $
$ N N N N $

$ N N Y N $
*$ N N Y N $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *N

$ Y Y Y Y $ | | | | | | $ Y Y Y Y $

$ Y Y N Y $ | | | | | | *$ Y Y N Y $

no is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *Y

$ N N N N $ | | | | | | $ N N N N $

$ N N Y N $ | | | | | | *$ N N Y N $

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *N

	$	Y	Y	Y	Y	$		$	Y	Y	N	Y	$
	|	|	|	|	|	|		|	|	|	|	|	|
	$	Y	Y	Y	Y	$	*	$	Y	Y	N	Y	$

no is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: *Y

	$	N	N	N	N	$		$	N	N	Y	N	$
	|	|	|	|	|	|		|	|	|	|	|	|
	$	N	N	N	N	$	*	$	N	N	Y	N	$

TSL quantifier languages for *every* and *no*

every is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: ***N**

$ Y Y Y Y $ → $ Y Y Y Y $

$ Y Y N Y $ → *$ Y Y N Y $

no is TSL with tier alphabet $\{\mathbf{Y}, \mathbf{N}\}$

Forbidden substrings: ***Y**

$ N N N N $ → $ N N N N $

$ N N Y N $ → *$ N N Y N $

Quantifier languages with a non-trivial tier

some is TSL with tier alphabet {Y}

Forbidden substrings: *$$

$ N N Y N $ *$ N N N N $

all but 3 is TSL with tier alphabet {N}

Forbidden substrings: *$$, *N, *NN, *NNNN

$ N N Y N $ *$ N Y Y N $ *$ N N N N $

Quantifier languages with a non-trivial tier

some is TSL with tier alphabet {**Y**}

Forbidden substrings: *$$

$ **Y** $

$ **N N Y N** $ *$ **N N N N** $

all but 3 is TSL with tier alphabet {N}

Forbidden substrings: *$$, *$**N**$, *$**NN**$, ***NNNN**

$ **N N Y N** $ *$ **N Y Y N** $ *$ **N N N N** $

Quantifier languages with a non-trivial tier

some is TSL with tier alphabet {**Y**}

Forbidden substrings: *$$

$ **Y** $

$ **N N Y N** $ *$ **N N N N** $

all but 3 is TSL with tier alphabet {N}

Forbidden substrings: *$$, *$**N**$, *$**NN**$, ***NNNN**

$ **N N Y N** $ *$ **N Y Y N** $ *$ **N N N N** $

Quantifier languages with a non-trivial tier

some is TSL with tier alphabet {**Y**}

Forbidden substrings: *$$

$ Y $

$ **N N Y N** $ *$ **N N N N** $

all but 3 is TSL with tier alphabet {N}

Forbidden substrings: *$$, *$**N**$, *$**NN**$, ***NNNN**

$ **N N Y N** $ *$ **N Y Y N** $ *$ **N N N N** $

Quantifier languages with a non-trivial tier

some is TSL with tier alphabet {**Y**}

Forbidden substrings: *$$

$ Y $	$ $

$ **N N Y N** $ *$ **N N N N** $

all but 3 is TSL with tier alphabet {N}

Forbidden substrings: *$$, *$**N**$, *$**NN**$, ***NNNN**

$ **N N Y N** $ *$ **N Y Y N** $ *$ **N N N N** $

Quantifier languages with a non-trivial tier

some is TSL with tier alphabet {**Y**}

Forbidden substrings: *$$

$ Y $

$ **N N Y N** $

$ $

*$ **N N N N** $

all but 3 is TSL with tier alphabet {N}

Forbidden substrings: *$$, *$**N**$, *$**NN**$, ***NNNN**

$ **N N Y N** $

*$ **N Y Y N** $

*$ **N N N N** $

Quantifier languages with a non-trivial tier

some is TSL with tier alphabet {**Y**}

Forbidden substrings: *$$

$ Y $ $ $

$ N N Y N $ *$ N N N N $

all but 3 is TSL with tier alphabet {N}

Forbidden substrings: *$$, *N, *NN, *NNNN

$ N N Y N $ *$ N Y Y N $ *$ N N N N $

Quantifier languages with a non-trivial tier

some is TSL with tier alphabet {**Y**}

Forbidden substrings: *$$

all but 3 is TSL with tier alphabet {N}

Forbidden substrings: *$$, *$**N**$, *$**NN**$, ***NNNN**

Quantifier languages with a non-trivial tier

some is TSL with tier alphabet {**Y**}

Forbidden substrings: *$$

$ Y $ $ $

$ **N N Y N** $ *$ **N N N N** $

all but 3 is TSL with tier alphabet {N}

Forbidden substrings: *$$, *$**N**$, *$**NN**$, ***NNNN**

$ **N N N** $

$ **N N Y N** $ *$ **N Y Y N** $ *$ **N N N N** $

Quantifier languages with a non-trivial tier

some is TSL with tier alphabet {**Y**}

Forbidden substrings: *$$

$ Y $

$ N N Y N $

$ $

*$ N N N N $

all but 3 is TSL with tier alphabet {N}

Forbidden substrings: *$$, *$**N**$, *$**NN**$, ***NNNN**

$ N N N $

$ N N Y N $

*$ N Y Y N $

*$ N N N N $

Quantifier languages with a non-trivial tier

some is TSL with tier alphabet {**Y**}

Forbidden substrings: *$$

$ Y $
$ N N Y N $

$ $
*$ N N N N $

all but 3 is TSL with tier alphabet {N}

Forbidden substrings: *$$, *$**N**$, *$**NN**$, ***NNNN**

$ N N N $
$ N N Y N $

$ N N $
*$ N Y Y N $

*$ N N N N $

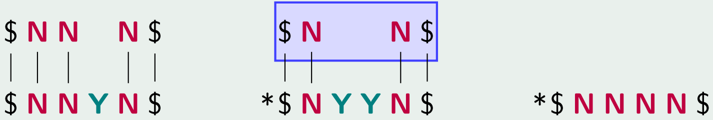
Quantifier languages with a non-trivial tier

some is TSL with tier alphabet {**Y**}

Forbidden substrings: *$$

all but 3 is TSL with tier alphabet {N}

Forbidden substrings: *$$, *$**N**$, *$**NN**$, ***NNNN**



Quantifier languages with a non-trivial tier

some is TSL with tier alphabet {**Y**}

Forbidden substrings: *$$

$ Y $ — $ N N Y N $

$ $ — *$ N N N N $

all but 3 is TSL with tier alphabet {N}

Forbidden substrings: *$$, *$**N**$, *$**NN**$, ***NNNN**

$ N N N $ — $ N N Y N $

$ N N $ — *$ N Y Y N $

$ N N N N $ — *$ N N N N $

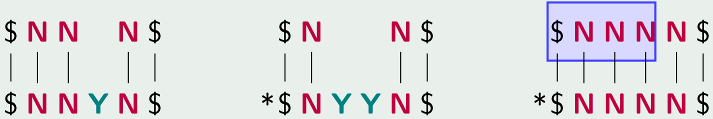
Quantifier languages with a non-trivial tier

some is TSL with tier alphabet {**Y**}

Forbidden substrings: *$$

all but 3 is TSL with tier alphabet {N}

Forbidden substrings: *$$, *$**N**$, *$**NN**$, ***NNNN**



Quantifier languages with a non-trivial tier

some is TSL with tier alphabet {**Y**}

Forbidden substrings: *$$

$ Y $
$ N N Y N $

$ $
*$ N N N N $

all but 3 is TSL with tier alphabet {N}

Forbidden substrings: *$$, *$**N**$, *$**NN**$, ***NNNN**

$ N N N $
$ N N Y N $

$ N N $
*$ N Y Y N $

$ N N N N $
*$ N N N N $

TSL descriptions for quantifier languages

Quantifier	Constraint	Tier	Forbidden
every	$\|N\| = 0$	Y, N	*N
no	$\|Y\| = 0$	Y, N	*Y
some	$\|Y\| \geq 1$	Y	*$$
at least n	$\|Y\| \geq n$	Y	$*\$Y^{m}\$$ ($m < n$)
at most n	$\|Y\| \leq n$	Y	$*Y^{n+1}$
exactly n	$\|Y\| = n$	Y	at least + at most
not all	$\|N\| \geq 1$	N	*$$
all but n	$\|N\| = n$	N	$*\$N^{m}\$$, $*N^{n+1}$
an even number	regular	—	—
most	context-free	—	—
half	context-free	—	—
a third	context-free	—	—

- **Insight:** common quantifiers are even simpler than we realized
- **Open issue:** still unclear why only some quantifiers can be expressed as a single lexical item (Paperno 2011)

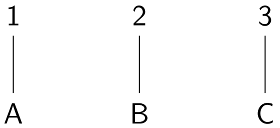
Monotonicity

Definition (Monotonicity)

- Let A and B be two sets with orders $\leq_{\mathsf{A}}$ and $\leq_{\mathsf{B}}$, respectively.
- A function f from A to B is **monotonic**ally increasing iff

$$x \leq_{\mathsf{A}} y \Rightarrow \mathsf{f}(x) \leq_{\mathsf{B}} \mathsf{f}(y)$$

- Monotonicity is similar to **No Crossing Branches** constraint.



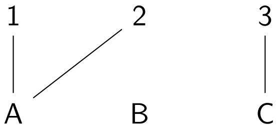
Monotonicity

Definition (Monotonicity)

- Let A and B be two sets with orders $\leq_A$ and $\leq_B$, respectively.
- A function f from A to B is **monotonic**ally increasing iff

$$x \leq_A y \Rightarrow f(x) \leq_B f(y)$$

- Monotonicity is similar to **No Crossing Branches** constraint.



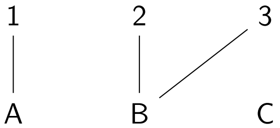
Monotonicity

Definition (Monotonicity)

- Let A and B be two sets with orders $\leq_A$ and $\leq_B$, respectively.
- A function f from A to B is **monotonic**ally increasing iff

$$x \leq_A y \Rightarrow f(x) \leq_B f(y)$$

- Monotonicity is similar to **No Crossing Branches** constraint.



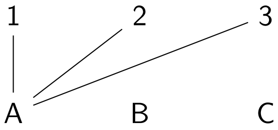
Monotonicity

Definition (Monotonicity)

- Let **A** and **B** be two sets with orders $\leq_{\mathbf{A}}$ and $\leq_{\mathbf{B}}$, respectively.
- A function f from **A** to **B** is **monotonic**ally increasing iff

$$x \leq_{\mathbf{A}} y \Rightarrow \mathsf{f}(x) \leq_{\mathbf{B}} \mathsf{f}(y)$$

- Monotonicity is similar to **No Crossing Branches** constraint.



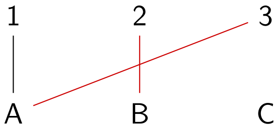
Monotonicity

Definition (Monotonicity)

- Let A and B be two sets with orders $\leq_A$ and $\leq_B$, respectively.
- A function f from A to B is **monotonic**ally increasing iff

$$x \leq_A y \Rightarrow f(x) \leq_B f(y)$$

- Monotonicity is similar to **No Crossing Branches** constraint.



Monotonicity in language

- **Monotonicity in phonology**
  - No Crossing Branches constraint
  - Natural classes are convex
- **Monotonicity in morphology**
  - adjectival gradation
  - person pronoun paradigms
  - tense
  - resolved gender
- **Monotonicity in syntax**
  - Subcategorization $<$ A-Move $<$ A$'$-Move
  - Adjunct Island Constraint & Coordinate Structure Constraint
  - Williams cycle
  - Ban against improper case
  - Expletive negation
- **Monotonicity in semantics**
  - Everywhere...

Montonicity in tier projection

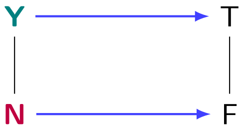
- Suppose, then, that monotonicity is a desirable trait.
- How does monotonicity relate to tier projection?

Project:

- Monotonicity **forbids projecting only N**.

Montonicity in tier projection

- Suppose, then, that monotonicity is a desirable trait.
- How does monotonicity relate to tier projection?

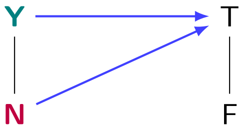


Project: Y

- Monotonicity **forbids projecting only N**.

Montonicity in tier projection

- Suppose, then, that monotonicity is a desirable trait.
- How does monotonicity relate to tier projection?

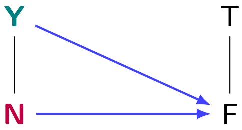


Project: Y and N

- Monotonicity **forbids projecting only N**.

Montonicity in tier projection

- Suppose, then, that monotonicity is a desirable trait.
- How does monotonicity relate to tier projection?

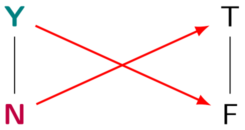


Project: nothing

- Monotonicity **forbids projecting only N**.

Montonicity in tier projection

- Suppose, then, that monotonicity is a desirable trait.
- How does monotonicity relate to tier projection?

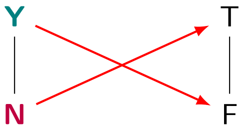


Project: forbidden

- Monotonicity **forbids projecting only N**.

Montonicity in tier projection

- Suppose, then, that monotonicity is a desirable trait.
- How does monotonicity relate to tier projection?



Project: forbidden

- Monotonicity **forbids projecting only N**.

Typology of quantifiers

Quantifier	TSL?	Tier	Mono. (Paperno 2011)
every	yes	Y, N	yes
no	yes	Y, N	yes
some	yes	Y	yes
at least n	yes	Y	yes
at most n	yes	Y	yes
exactly n	yes	Y	yes
not all	yes	N	no
all but one	yes	N	no
an even number	no		no
most	no		—
half	no		—
at least a third	no		—

Typology of quantifiers

Quantifier	TSL?	Tier	Mono.
every	yes	Y, N	yes
no	yes	Y, N	yes
some	yes	Y	yes
at least n	yes	Y	yes
at most n	yes	Y	yes
exactly n	yes	Y	yes
not all	yes	N	no
all but one	yes	N	no
an even number	no		no
most	no		—
half	no		—
at least a third	no		—

(Paperno 2011)

Context-free

Typology of quantifiers

Quantifier	TSL?	Tier	Mono.
every	yes	Y, N	yes
no	yes	Y, N	yes
some	yes	Y	yes
at least n	yes	Y	yes
at most n	yes	Y	yes
exactly n	yes	Y	yes
not all	yes	N	no
all but one	yes	N	no
an even number	no		no
most	no		—
half	no		—
at least a third	no		—

(Paperno 2011)

Regular (every through an even number)

Context-free (all rows)

Typology of quantifiers

Quantifier	TSL?	Tier	Mono.
every	yes	Y, N	yes
no	yes	Y, N	yes
some	yes	Y	yes
at least n	yes	Y	yes
at most n	yes	Y	yes
exactly n	yes	Y	yes
not all	yes	N	no
all but one	yes	N	no
an even number	no		no
most	no		—
half	no		—
at least a third	no		—

(Paperno 2011)

TSL

Regular

Context-free

Typology of quantifiers

(Paperno 2011)

Quantifier	TSL?	Tier	Mono.	
every	yes	Y, N	yes	Lexical
no	yes	Y, N	yes	Lexical
some	yes	Y	yes	Lexical
at least n	yes	Y	yes	Lexical
at most n	yes	Y	yes	Lexical
exactly n	yes	Y	yes	Lexical
not all	yes	N	no	TSL
all but one	yes	N	no	TSL
an even number	no		no	Regular
most	no		—	Context-free
half	no		—	Context-free
at least a third	no		—	Context-free

A semantic universal

Monotic TSL restriction

If a regular quantifier can be expressed by a single lexical item, then its quantifier language must be monotonic TSL.

What about **most**/**half**?

1. not regular, hence not subject to the universal
2. might be multi-lexical underlyingly (Hackl 2009)
3. monotonic TSL if we can **impose specific orders**

Example: **most**

1. Tier: **Y**, **N**
2. Start with **Y**
3. End with **Y**
4. Don't have **NN**

- **YNYNY**
- **YYYNY**
- ***YNNNY**
- ***YNYN**

A semantic universal

Monotic TSL restriction

If a regular quantifier can be expressed by a single lexical item, then its quantifier language must be monotonic TSL.

What about **most**/**half**?

1. not regular, hence not subject to the universal
2. might be multi-lexical underlyingly (Hackl 2009)
3. monotonic TSL if we can **impose specific orders**

Example: **most**

1. Tier: **Y**, **N**
2. Start with **Y**
3. End with **Y**
4. Don't have **NN**

- **YNYNY**
- **YYYNY**
- ***YNNNY**
- ***YNYN**

Conclusion

- Common quantifiers are even simpler than we thought (TSL).
- **Cognitive parallelism**
  TSL also plays a major role in phonology, morphology, syntax
- TSL brings a new kind of monotonicity to quantifiers.
- Lexical quantifiers are starting to look like a natural class.
- Of course, plenty of work remains to be done.
  adverbials, modals, typology, ...

Thanks

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

References I

Aksënova, Alëna, Thomas Graf, and Sedigheh Moradi. 2016. Morphotactics as tier-based strictly local dependencies. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, 121–130. URL https://www.aclweb.org/anthology/W/W16/W16-2019.pdf.

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

Chandlee, Jane. 2014. *Strictly local phonological processes*. Doctoral Dissertation, University of Delaware. URL http://udspace.udel.edu/handle/19716/13374.

Goldsmith, John. 1976. *Autosegmental phonology*. Doctoral Dissertation, MIT.

Graf, Thomas. 2018. Why movement comes for free once you have adjunction. In *Proceedings of CLS 53*, ed. Daniel Edmiston, Marina Ermolaeva, Emre Hakgüder, Jackie Lai, Kathryn Montemurro, Brandon Rhodes, Amara Sankhagowit, and Miachel Tabatowski, 117–136.

Hackl, Martin. 2009. On the grammar and processing of proportional quantifiers: Most versus more than half. *Natural Language Semantics* 17:63–98.

Heinz, Jeffrey. 2009. On the role of locality in learning stress patterns. *Phonology* 26:303–351. URL https://doi.org/10.1017/S0952675709990145.

Heinz, Jeffrey. 2010. Learning long-distance phonotactics. *Linguistic Inquiry* 41:623–661. URL http://dx.doi.org/10.1162/LING_a_00015.

References II

Heinz, Jeffrey. 2018. The computational nature of phonological generalizations. In *Phonological typology*, ed. Larry Hyman and Frank Plank, Phonetics and Phonology, chapter 5, 126–195. Mouton De Gruyter.

Heinz, Jeffrey, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints in phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 58–64. URL `http://www.aclweb.org/anthology/P11-2011`.

Jardine, Adam. 2016. Computationally, tone is different. *Phonology* 33:247–283. URL `https://doi.org/10.1017/S0952675716000129`.

McMullin, Kevin. 2016. *Tier-based locality in long-distance phonotactics: Learnability and typology*. Doctoral Dissertation, University of British Columbia.

Paperno, Denis. 2011. Learnable classes of natural language quantifiers: Two perspectives. URL `http://paperno.bol.ucla.edu/q_learning.pdf`, ms., UCLA.

Shafiei, Nazila, and Thomas Graf. 2020. The subregular complexity of syntactic islands. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2020*. To appear.