Evaluating Subregular Distinctions in the Complexity of Generalized Quantifiers

Aniello De Santo Thomas Graf John E. Drury

Stony Brook University
aniello.desanto@stonybrook.edu
aniellodesanto.github.io

QUAD: Quantifiers and Determiners
July 17–21, 2017

The Talk in a Nutshell

Generalized Quantifiers and Semantic Complexity

Semantic automata (SA) as a model of quantifiers'verification

- insights into quantifiers' interpretation
- link between formal language theory and model theory

In This Talk: Giving up the SA perspective

- But: formal language theory is richer that automata theory
- Coming back to formal language theory
  $\rightarrow$ subregular hierarchy & quantifier languages

Consequences

- complexity independent of the recognition mechanism
- cross-domain parallels, cognitive predictions and new experimental paradigms!

The Talk in a Nutshell

Generalized Quantifiers and Semantic Complexity

Semantic automata (SA) as a model of quantifiers'verification

- insights into quantifiers' interpretation
- link between formal language theory and model theory

In This Talk: Giving up the SA perspective

- But: formal language theory is richer that automata theory
- Coming back to formal language theory
  $\rightarrow$ subregular hierarchy & quantifier languages

Consequences

- complexity independent of the recognition mechanism
- cross-domain parallels, cognitive predictions and new experimental paradigms!

The Talk in a Nutshell

Generalized Quantifiers and Semantic Complexity

Semantic automata (SA) as a model of quantifiers'verification

- insights into quantifiers' interpretation
- link between formal language theory and model theory

In This Talk: Giving up the SA perspective

- But: formal language theory is richer that automata theory
- Coming back to formal language theory
  $\rightarrow$ subregular hierarchy & quantifier languages

Consequences

- complexity independent of the recognition mechanism
- cross-domain parallels, cognitive predictions and new experimental paradigms!

Outline

1 Semantic Automata

2 Generalized Quantifiers & Subregular Languages

3 Psycholinguistic Predictions

4 Conclusions

Generalized Quantifiers

Generalized quantifier $\mathbf{Q}(\mathbf{A}, \mathbf{B})$:

- two sets **A** and **B** as arguments
- returns truth value $(0, 1)$

Example

(1) Every student cheated.

- $\mathbf{every}(\mathbf{A}, \mathbf{B}) = 1$ iff $\mathbf{A} \subseteq \mathbf{B}$
- **student**: John, Mary, Sue
- **cheat**: John, Mary
- $\mathbf{student} \not\subseteq \mathbf{cheat} \Rightarrow \mathbf{every}(\mathbf{student}, \mathbf{cheat}) = 0$
- "Every student cheated" is false.

Binary Strings

- The language of **A** is the set of all permutations of **A**.

Example

student	John, Mary, Sue
$L($**student**$)$	John Mary Sue, John Sue Mary
	Mary John Sue, Mary Sue John
	Sue John Mary, Sue Mary John

- Now replace every **a** $\in$ **A** by a truth value:
  1 if **a** $\in$ **B**
  0 if **a** $\notin$ **B**
- The result is the **binary string language** of **A** under **B**.

Example

student	John, Mary, Sue
cheat	John, Mary
binary strings	110, 101, 011

Binary Strings

- The language of **A** is the set of all permutations of **A**.

Example

student	John, Mary, Sue
L(**student**)	John Mary Sue, John Sue Mary
	Mary John Sue, Mary Sue John
	Sue John Mary, Sue Mary John

- Now replace every **a** $\in$ **A** by a truth value:
  1 if **a** $\in$ **B**
  0 if **a** $\notin$ **B**
- The result is the **binary string language** of **A** under **B**.

Example

student	John, Mary, Sue
cheat	John, Mary
binary strings	110, 101, 011

Quantifier Languages (van Benthem 1986)

- We can associate each quantifier Q with a language in $\{0,1\}^*$
  $\Rightarrow$ Q accepts only binary strings of specific shape
- This is its **quantifier language**.

Example: *every*

- **every**(**A**, **B**) holds iff **A** $\subseteq$ **B**
- So every element of **A** must be mapped to 1.
- $L(\textbf{every}) = \{1\}^*$

Example: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- Some element of **A** must be mapped to 1.
- $L(\textbf{some}) = \{0,1\}^* 1 \{0,1\}^*$

Quantifier Languages (van Benthem 1986)

- We can associate each quantifier Q with a language in $\{0,1\}^*$ $\Rightarrow$ Q accepts only binary strings of specific shape
- This is its **quantifier language**.

Example: *every*

- **every**(**A**, **B**) holds iff **A** $\subseteq$ **B**
- So every element of **A** must be mapped to 1.
- $L(\textbf{every}) = \{1\}^*$

Example: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- Some element of **A** must be mapped to 1.
- $L(\textbf{some}) = \{0,1\}^* \, 1 \, \{0,1\}^*$

Quantifier Languages (van Benthem 1986)

- We can associate each quantifier Q with a language in $\{0, 1\}^*$
  $\Rightarrow$ Q accepts only binary strings of specific shape
- This is its **quantifier language**.

Example: *every*

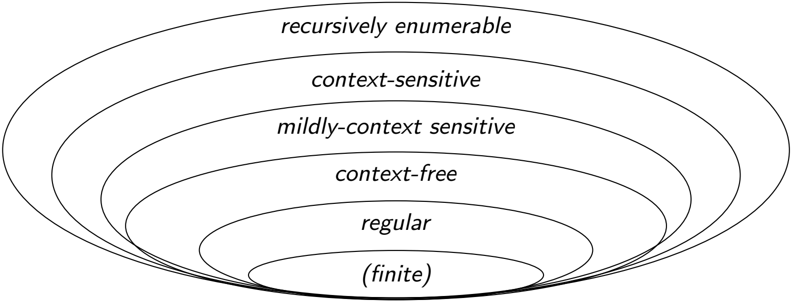
- **every**(**A**, **B**) holds iff **A** $\subseteq$ **B**
- So every element of **A** must be mapped to 1.
- $L($**every**$) = \{1\}^*$

Example: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- Some element of **A** must be mapped to 1.
- $L($**some**$) = \{0, 1\}^* \, 1 \, \{0, 1\}^*$

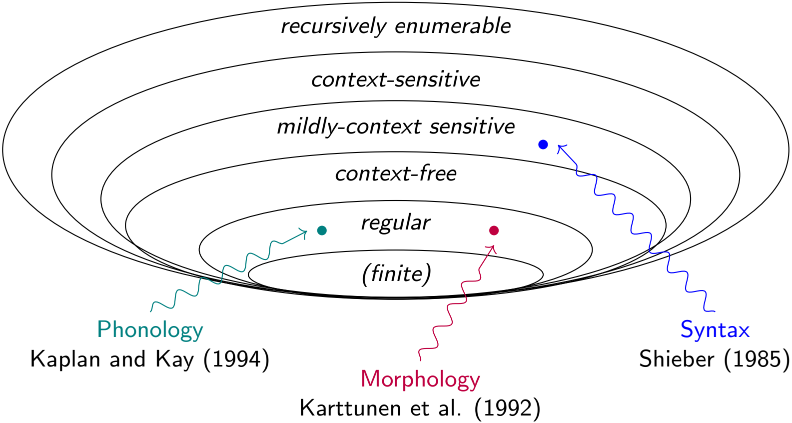
Chomsky Hierarchy of String Languages

Languages (stringsets) can be classified according to the complexity of the grammars that generate them.

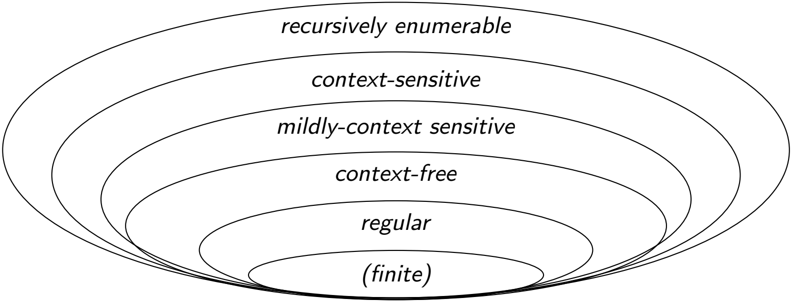


Chomsky Hierarchy of String Languages

Languages (stringsets) can be classified according to the complexity of the grammars that generate them.



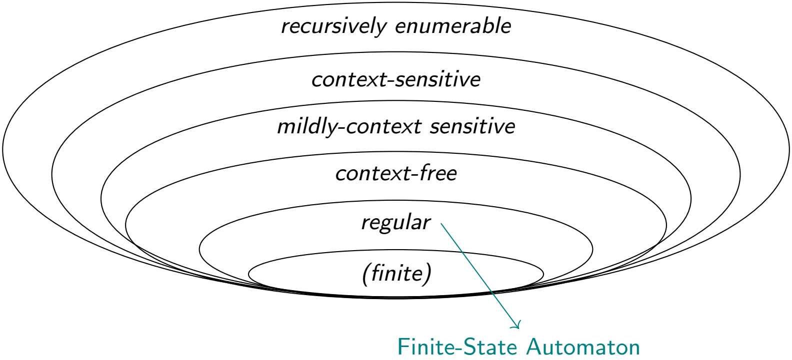
Chomsky Hierarchy and Automata Theory



Semantic Automata (van Benthem 1986, Mostowski 1998)

We can rank quantifiers based on their quantifier languages and the complexity of the machine needed to recognize them.

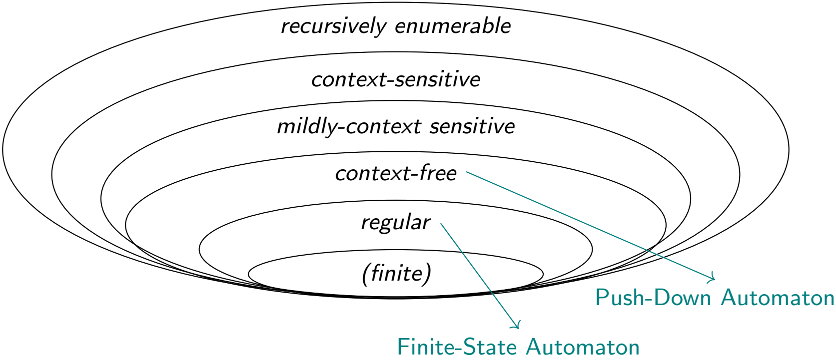
Chomsky Hierarchy and Automata Theory



Semantic Automata (van Benthem 1986, Mostowski 1998)

We can rank quantifiers based on their quantifier languages and the complexity of the machine needed to recognize them.

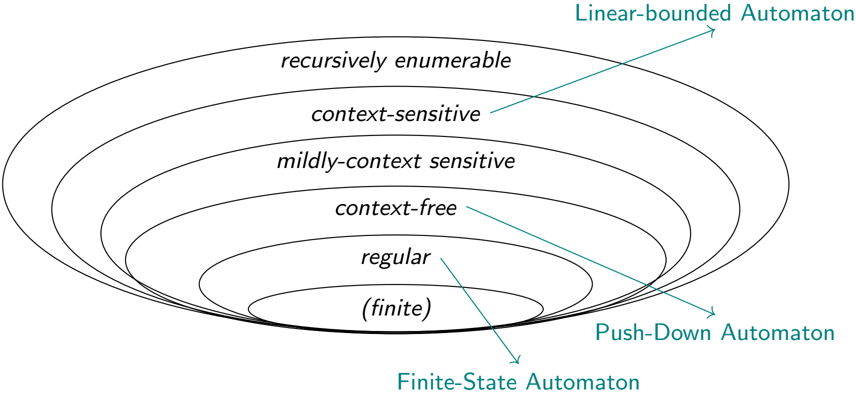
Chomsky Hierarchy and Automata Theory



Semantic Automata (van Benthem 1986, Mostowski 1998)

We can rank quantifiers based on their quantifier languages and the complexity of the machine needed to recognize them.

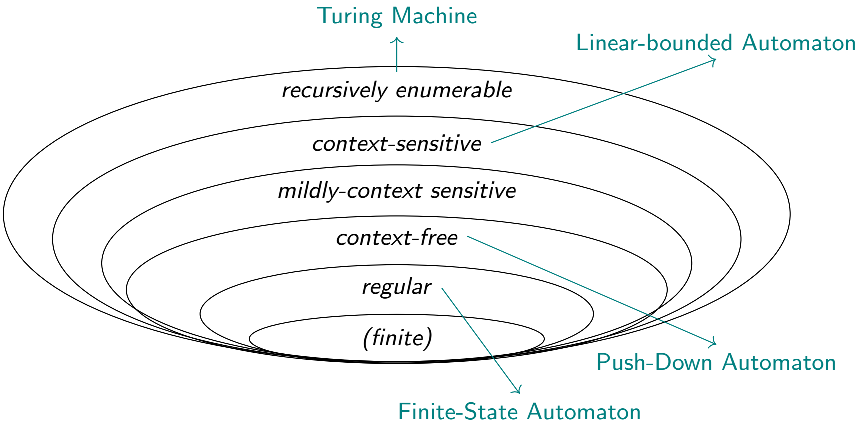
Chomsky Hierarchy and Automata Theory



Semantic Automata (van Benthem 1986, Mostowski 1998)

We can rank quantifiers based on their quantifier languages and the complexity of the machine needed to recognize them.

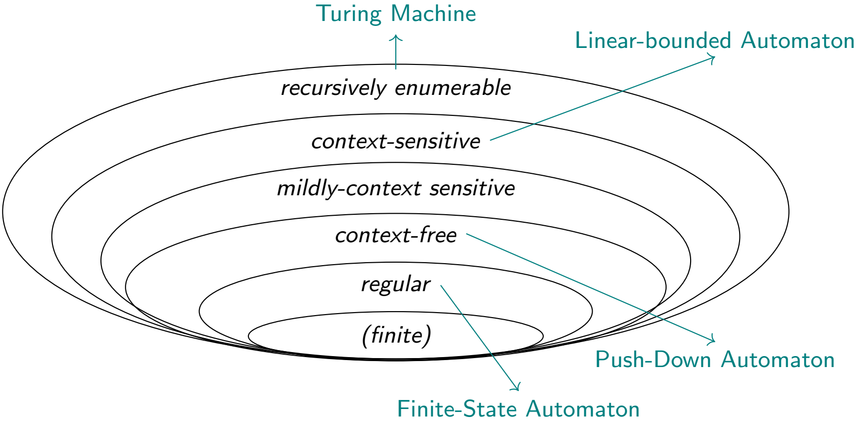
Chomsky Hierarchy and Automata Theory



Semantic Automata (van Benthem 1986, Mostowski 1998)

We can rank quantifiers based on their quantifier languages and the complexity of the machine needed to recognize them.

Chomsky Hierarchy and Automata Theory



Semantic Automata (van Benthem 1986, Mostowski 1998)

We can rank quantifiers based on their quantifier languages and the complexity of the machine needed to recognize them.

Aristotelian Quantifiers are FSA-recognizable

Reminder: *every*

- **every**(**A**, **B**) holds iff **A** $\subseteq$ **B**
- So every element of **A** must be mapped to 1.
- $L($**every**$) = \{1\}^*$

False

student	John, Mary, Sue
cheat	John, Mary
binary strings	110, 101, 011

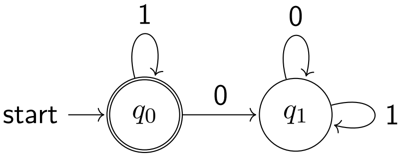
True

student	John, Mary, Sue
cheat	John, Mary,Sue
binary strings	111

Aristotelian Quantifiers are FSA-recognizable

Reminder: *every*

- **every**(**A**, **B**) holds iff **A** $\subseteq$ **B**
- So every element of **A** must be mapped to 1.
- $L($**every**$) = \{1\}^*$



False

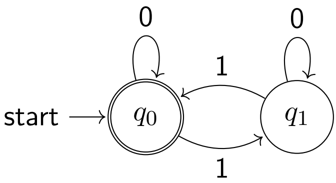
student	John, Mary, Sue
cheat	John, Mary
binary strings	110, 101, 011

True

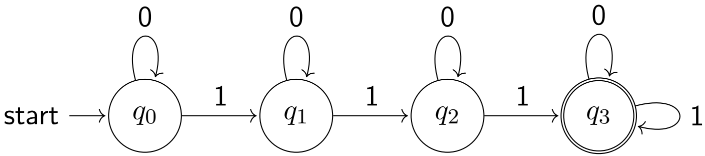
student	John, Mary, Sue
cheat	John, Mary,Sue
binary strings	111

Other FSA-recognizable quantifiers

- Parity quantifiers: **An even number**



- Cardinal quantifiers: **At least 3**



Proportional Quantifiers

- **most**(**A**, **B**) holds iff $|\mathbf{A} \cap \mathbf{B}| > |\mathbf{A} - \mathbf{B}|$
- $L_{most} := \{w \in \{0,1\}^* : |1|_w > |0|_w\}$
- There is no finite automaton recognizing this language.
- We need internal memory.
  $\Rightarrow$ **push-down automata**: two states + a stack

A Hierarchy of Quantifiers' Complexity

FSA **PDA**

$\boxed{\{\textit{All, Some, Even, Odd, At least n, At most n}\}} < \boxed{\{\textit{Less than half, More than half, Most}\}}$

Are these all of equivalent complexity?

A Hierarchy of Quantifiers' Complexity

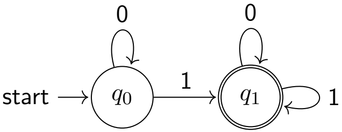
FSA **PDA**

$\boxed{\{\textit{All, Some, Even, Odd, At least n, At most n}\}} < \boxed{\{\textit{Less than half, More than half, Most}\}}$

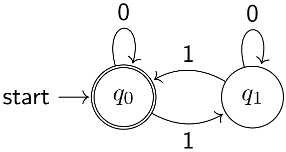
Are these all of equivalent complexity?

Let's Look at the Automata One More Time

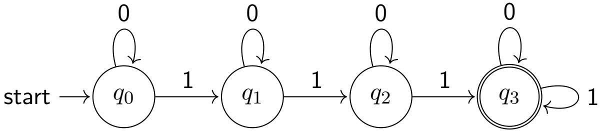
- Aristotelian quantifiers: **Some**



- Parity quantifiers: **An even number**



- Cardinal quantifiers: **At least 3**



A Hierarchy of Quantifiers' Complexity

FSA **PDA**

$\boxed{\{\textit{All, Some}\} < \{\textit{Even, Odd}\} < \{\textit{At least n, At most n}\}} < \boxed{\{\textit{Less than half, More than half, Most}\}}$

Are these all of equivalent complexity? (Szymanik 2016)

- Cyclic vs acyclic automata
- The number of states matters
- But: Complexity = succinctness of automata?

Reminder

It's all grounded in quantifier languages

- FSA recognizable quantifiers → Regular quantifier languages

A Hierarchy of Quantifiers' Complexity

FSA **PDA**

$\boxed{\{\textit{All, Some}\} < \{\textit{Even, Odd}\} < \{\textit{At least n, At most n}\}} < \boxed{\{\textit{Less than half, More than half, Most}\}}$

Are these all of equivalent complexity? (Szymanik 2016)

- Cyclic vs acyclic automata
- The number of states matters
- **But: Complexity = succinctness of automata?**

Reminder

It's all grounded in quantifier languages

- FSA recognizable quantifiers → Regular quantifier languages

A Hierarchy of Quantifiers' Complexity

FSA **PDA**

$\boxed{\{\textit{All, Some}\} < \{\textit{Even, Odd}\} < \{\textit{At least n, At most n}\}} < \boxed{\{\textit{Less than half, More than half, Most}\}}$

Are these all of equivalent complexity? (Szymanik 2016)

- Cyclic vs acyclic automata
- The number of states matters
- **But: Complexity = succinctness of automata?**

Reminder

It's all grounded in quantifier languages

- FSA recognizable quantifiers → Regular quantifier languages

A Hierarchy of Quantifiers' Complexity

FSA **PDA**

$\{$*All, Some*$\} < \{$*Even, Odd*$\} < \{$*At least n, At most n*$\} < \{$*Less than half, More than half, Most*$\}$

Are these all of equivalent complexity? (Szymanik 2016)

- Cyclic vs acyclic automata
- The number of states matters
- **But: Complexity = succinctness of automata?**

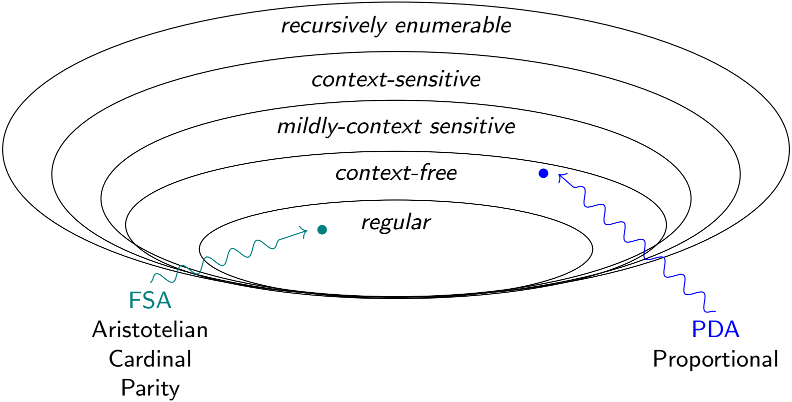
Reminder

It's all grounded in quantifier languages

- FSA recognizable quantifiers → Regular quantifier languages

Chomsky Hierarchy of String Languages (Reprise)

Languages (stringsets) can be classified according to the complexity of the **grammars** that generate them.



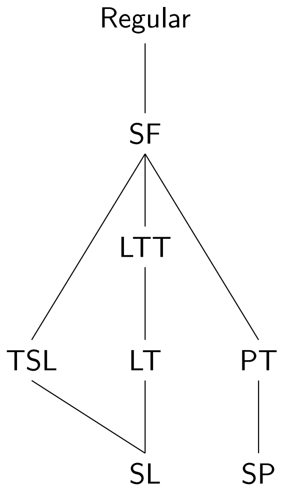
The Subregular Hierarchy

Often Forgotten:

- hierarchy of subregular languages (McNaughton&Papert 1971), (Rogers et al. 2010)

A Richness of Results

- Phonology is subregular (Heinz&Idsardi 2013, Heinz 2015)
- Morphotactics (and Morphology?) is subregular (Aksënova et al. 2016, Chandlee 2016, Aksënova&De Santo 2017)
- Syntax? (Graf&Heinz 2015)



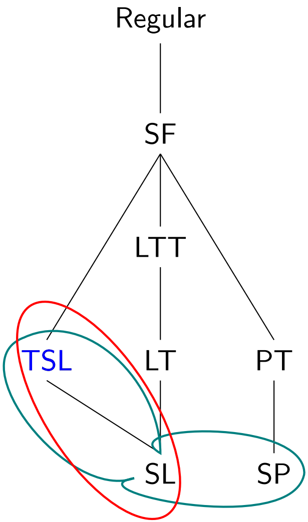
The Subregular Hierarchy

Often Forgotten:

- hierarchy of subregular languages (McNaughton&Papert 1971), (Rogers et al. 2010)

A Richness of Results

- Phonology is subregular (Heinz&Idsardi 2013, Heinz 2015)
- Morphotactics (and Morphology?) is subregular (Aksënova et al. 2016, Chandlee 2016, Aksënova&De Santo 2017)
- Syntax? (Graf&Heinz 2015)



Strictly Local and Tier-based Strictly Local

Strictly Local (SL)

- SL_k grammars are lists of forbidden k-grams;

Tier-based Strictly Local (TSL)

- TSL is a minimal extension of SL, inspired by phonological tiers;
  - define a subset T of the string alphabet;
  - a grammar is a list of strictly k-local constraints over T

Subregular Quantifiers: *Every is SL*

Reminder: *Every*

- **every**(**A**, **B**) holds iff **A** $\subseteq$ **B**
- $L(\textbf{every}) = \{1\}^*$
- Eg. *Every student cheated.*

False

student	John, Mary, Sue
cheat	John, Mary
binary strings	110, 101, 011
grammar	*0

True

student	John, Mary, Sue
cheat	John, Mary, Sue
binary strings	111
grammar	*0

Subregular Quantifiers: *Every is SL*

Reminder: *Every*

- **every**(**A**, **B**) holds iff **A** $\subseteq$ **B**
- $L($**every**$) = \{1\}^*$
- Eg. *Every student cheated.*

False

student	John, Mary, Sue
cheat	John, Mary
binary strings	110, 101, 011
grammar	$*0$

True

student	John, Mary, Sue
cheat	John, Mary, Sue
binary strings	111
grammar	$*0$

Subregular Quantifiers: *Every is SL*

Reminder: *Every*

- **every**(**A**, **B**) holds iff **A** $\subseteq$ **B**
- $L($**every**$) = \{1\}^*$
- Eg. *Every student cheated.*

False

student	John, Mary, Sue
cheat	John, Mary
binary strings	110, 101, 011
grammar	^{*0}

$\rtimes$ 1 1 0 $\ltimes$

True

student	John, Mary, Sue
cheat	John, Mary, Sue
binary strings	111
grammar	^{*0}

$\rtimes$ 1 1 1 $\ltimes$

Subregular Quantifiers: *Every is SL*

Reminder: *Every*

- **every**(**A**, **B**) holds iff **A** $\subseteq$ **B**
- $L($**every**$) = \{1\}^*$
- Eg. *Every student cheated.*

False

student	John, Mary, Sue
cheat	John, Mary
binary strings	110, 101, 011
grammar	^{*0}

$\rtimes \; \boxed{1} \; 1 \; 0 \; \ltimes$

True

student	John, Mary, Sue
cheat	John, Mary, Sue
binary strings	111
grammar	^{*0}

$\rtimes \; 1 \; 1 \; 1 \; \ltimes$

Subregular Quantifiers: *Every is SL*

Reminder: *Every*

- **every**(**A**, **B**) holds iff **A** $\subseteq$ **B**
- $L($**every**$) = \{1\}^*$
- Eg. *Every student cheated.*

False

student	John, Mary, Sue
cheat	John, Mary
binary strings	110, 101, 011
grammar	^{*0}

$\rtimes$ 1 $\boxed{1}$ 0 $\ltimes$

True

student	John, Mary, Sue
cheat	John, Mary, Sue
binary strings	111
grammar	^{*0}

$\rtimes$ 1 1 1 $\ltimes$

Subregular Quantifiers: *Every is SL*

Reminder: *Every*

- **every**(**A**, **B**) holds iff **A** $\subseteq$ **B**
- $L($**every**$) = \{1\}^*$
- Eg. *Every student cheated.*

False

student	John, Mary, Sue
cheat	John, Mary
binary strings	110, 101, 011
grammar	^{*0}

$\rtimes \; 1 \; 1 \; \boxed{0} \; \ltimes$

True

student	John, Mary, Sue
cheat	John, Mary, Sue
binary strings	111
grammar	^{*0}

$\rtimes \; 1 \; 1 \; 1 \; \ltimes$

Subregular Quantifiers: *Every is SL*

Reminder: *Every*

- **every**(**A**, **B**) holds iff **A** $\subseteq$ **B**
- $L($**every**$) = \{1\}^*$
- Eg. *Every student cheated.*

False

student	John, Mary, Sue
cheat	John, Mary
binary strings	110, 101, 011
grammar	^{*0}

F

$\rtimes$ 1 1 $\boxed{0}$ $\ltimes$

True

student	John, Mary, Sue
cheat	John, Mary, Sue
binary strings	111
grammar	^{*0}

$\rtimes$ 1 1 1 $\ltimes$

Subregular Quantifiers: *Every is SL*

Reminder: *Every*

- **every**(**A**, **B**) holds iff **A** $\subseteq$ **B**
- $L($**every**$) = \{1\}^*$
- Eg. *Every student cheated.*

False

student	John, Mary, Sue
cheat	John, Mary
binary strings	110, 101, 011
grammar	^{*0}

F

$\rtimes \quad 1 \quad 1 \quad \boxed{0} \quad \ltimes$

True

student	John, Mary, Sue
cheat	John, Mary, Sue
binary strings	111
grammar	^{*0}

$\rtimes \quad \boxed{1} \quad 1 \quad 1 \quad \ltimes$

Subregular Quantifiers: *Every is SL*

Reminder: *Every*

- **every**(**A**, **B**) holds iff **A** $\subseteq$ **B**
- $L(\textbf{every}) = \{1\}^*$
- Eg. *Every student cheated.*

False

student	John, Mary, Sue
cheat	John, Mary
binary strings	110, 101, 011
grammar	^{*0}

F

$\rtimes$ 1 1 0 $\ltimes$

True

student	John, Mary, Sue
cheat	John, Mary, Sue
binary strings	111
grammar	^{*0}

$\rtimes$ 1 $\boxed{1}$ 1 $\ltimes$

Subregular Quantifiers: *Every is SL*

Reminder: *Every*

- **every**(**A**, **B**) holds iff **A** $\subseteq$ **B**
- $L($**every**$) = \{1\}^*$
- Eg. *Every student cheated.*

False

student	John, Mary, Sue
cheat	John, Mary
binary strings	110, 101, 011
grammar	^{*0}

F

$\rtimes$ 1 1 0 $\ltimes$

True

student	John, Mary, Sue
cheat	John, Mary, Sue
binary strings	111
grammar	^{*0}

$\rtimes$ 1 1 $\boxed{1}$ $\ltimes$

Subregular Quantifiers: *Every is SL*

Reminder: *Every*

- **every**(**A**, **B**) holds iff **A** $\subseteq$ **B**
- $L($**every**$) = \{1\}^*$
- Eg. *Every student cheated.*

False

student	John, Mary, Sue
cheat	John, Mary
binary strings	110, 101, 011
grammar	^{*0}

F

$\rtimes\ 1\ 1\ \boxed{0}\ \ltimes$

True

student	John, Mary, Sue
cheat	John, Mary, Sue
binary strings	111
grammar	^{*0}

T

$\rtimes\ 1\ 1\ 1\ \ltimes$

Subregular Quantifiers: *Some* is SL?

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L(\textbf{some}) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$*0$

True

student	John, Mary, Sue
cheat	John
binary strings	100,010,001
grammar	$*0$

Subregular Quantifiers: *Some* is SL?

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L(\textbf{some}) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	^{*0}

True

student	John, Mary, Sue
cheat	John
binary strings	100,010,001
grammar	^{*0}

Subregular Quantifiers: *Some* is SL?

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L($**some**$) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	^{*0}

$\rtimes \;\; 0 \;\; 0 \;\; 0 \;\; \ltimes$

True

student	John, Mary, Sue
cheat	John
binary strings	100,010,001
grammar	^{*0}

$\rtimes \;\; 0 \;\; 0 \;\; 1 \;\; \ltimes$

Subregular Quantifiers: *Some* is SL?

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L($**some**$) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	^{*0}

$\rtimes$ $\boxed{0}$ 0 0 $\ltimes$

True

student	John, Mary, Sue
cheat	John
binary strings	100,010,001
grammar	^{*0}

$\rtimes$ 0 0 1 $\ltimes$

Subregular Quantifiers: *Some* is SL?

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- L(**some**) $= \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	^{*0}

$\rtimes$ 0 $\boxed{0}$ 0 $\ltimes$

True

student	John, Mary, Sue
cheat	John
binary strings	100,010,001
grammar	^{*0}

$\rtimes$ 0 0 1 $\ltimes$

Subregular Quantifiers: *Some* is SL?

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L(\textbf{some}) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	^{*0}

$\rtimes$ 0 0 $\boxed{0}$ $\ltimes$

True

student	John, Mary, Sue
cheat	John
binary strings	100,010,001
grammar	^{*0}

$\rtimes$ 0 0 1 $\ltimes$

Subregular Quantifiers: *Some* is SL?

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L($**some**$) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	^{*0}

F

$\rtimes$ $\boxed{0}\boxed{0}\boxed{0}$ $\ltimes$

True

student	John, Mary, Sue
cheat	John
binary strings	100,010,001
grammar	^{*0}

$\rtimes$ 0 0 1 $\ltimes$

Subregular Quantifiers: *Some* is SL?

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L($**some**$) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	^{*0}

F

$\rtimes$ 0 0 0 $\ltimes$

True

student	John, Mary, Sue
cheat	John
binary strings	100,010,001
grammar	^{*0}

$\rtimes$ 0 0 1 $\ltimes$

Subregular Quantifiers: *Some* is SL?

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L(\textbf{some}) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	^{*0}

F

$\rtimes\ 0\ 0\ 0\ \ltimes$

True

student	John, Mary, Sue
cheat	John
binary strings	100,010,001
grammar	^{*0}

F

$\rtimes\ 0\ 0\ 1\ \ltimes$

Subregular Quantifiers: *Some* is SL?

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** ∩ **B** $\neq \emptyset$
- $L(\textbf{some}) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	^{*00}

F

$\rtimes$ 0 0 0 $\ltimes$

True

student	John, Mary, Sue
cheat	John
binary strings	100,010,001
grammar	^{*00}

F

$\rtimes$ 0 0 1 $\ltimes$

Subregular Quantifiers: *Some* is SL?

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L($**some**$) = \{0, 1\}^* \, 1 \, \{0, 1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	^{*000}

F

$\rtimes$ $\boxed{0\ 0\ 0}$ $\ltimes$

True

student	John, Mary, Sue
cheat	John
binary strings	100,010,001
grammar	^{*000}

T

$\rtimes$ $\boxed{0\ 0\ 1}$ $\ltimes$

Subregular Quantifiers: *Some* is SL?

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L(\textbf{some}) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	??

$\rtimes \; \boxed{0 \; 0^n \; 0} \; \ltimes$

True

student	John, Mary, Sue
cheat	John
binary strings	100,010,001
grammar	??

$\rtimes \; \boxed{0 \; 0^n \; 1} \; \ltimes$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L(\textbf{some}) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

$\rtimes$ 0 0 $\ltimes$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

$\rtimes$ 1 $\ltimes$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L(\textbf{some}) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{^*\rtimes\ltimes\}$

$\rtimes$ 0 0 $\ltimes$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{^*\rtimes\ltimes\}$

$\rtimes$ 1 $\ltimes$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L($**some**$) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{^* \rtimes \ltimes\}$

$\rtimes$ 0 0 $\ltimes$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{^* \rtimes \ltimes\}$

$\rtimes$ 1 $\ltimes$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L($**some**$) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{^* \rtimes \ltimes\}$

$\rtimes \; 0 \; 0 \; 0 \; \ltimes$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{^* \rtimes \ltimes\}$

$\rtimes \; 0 \; 1 \; 0 \; \ltimes$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L($**some**$) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

$\rtimes$ $\ltimes$

$\rtimes \; 0 \; 0 \; 0 \; \ltimes$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

$\rtimes \; 0 \; 1 \; 0 \; \ltimes$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L($**some**$) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{^* \rtimes \ltimes\}$

$\rtimes \quad \ltimes$

$\rtimes \; \boxed{0} \; 0 \; 0 \; \ltimes$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{^* \rtimes \ltimes\}$

$\rtimes \; 0 \; 1 \; 0 \; \ltimes$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- L(**some**) $= \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{^*\rtimes\ltimes\}$

$\rtimes$ $\ltimes$

$\rtimes$ 0 $\boxed{0}$ 0 $\ltimes$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{^*\rtimes\ltimes\}$

$\rtimes$ 0 1 0 $\ltimes$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L($**some**$) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{^* \rtimes \ltimes\}$

$\rtimes \quad \ltimes$

$\rtimes \; 0 \; 0 \; \boxed{0} \; \ltimes$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{^* \rtimes \ltimes\}$

$\rtimes \; 0 \; 1 \; 0 \; \ltimes$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L($**some**$) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

F $\boxed{\rtimes \quad \ltimes}$

$\rtimes \; 0 \; 0 \; 0 \; \ltimes$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

$\rtimes \; 0 \; 1 \; 0 \; \ltimes$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff $\mathbf{A} \cap \mathbf{B} \neq \emptyset$
- $L(\textbf{some}) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

$$F\ \boxed{\rtimes \quad \ltimes}$$

$$\rtimes\ 0\ 0\ 0\ \ltimes$$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

$$\rtimes \quad \ltimes$$

$$\rtimes\ 0\ 1\ 0\ \ltimes$$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L($**some**$) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

F $\boxed{\rtimes \quad \ltimes}$

$\rtimes \; 0 \; 0 \; 0 \; \ltimes$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

$\rtimes \quad \ltimes$

$\rtimes \; \boxed{0} \; 1 \; 0 \; \ltimes$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff $\mathbf{A} \cap \mathbf{B} \neq \emptyset$
- $L(\textbf{some}) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

F $\boxed{\rtimes \quad \ltimes}$

$\rtimes \; 0 \; 0 \; 0 \; \ltimes$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

$\rtimes \; 1 \; \ltimes$

$\rtimes \; 0 \; \boxed{1} \; 0 \; \ltimes$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L($**some**$) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

$$F \quad \boxed{\rtimes \qquad \ltimes}$$

$$\rtimes \; 0 \; 0 \; 0 \; \ltimes$$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

$$\rtimes \quad 1 \quad \ltimes$$

$$\rtimes \; 0 \; 1 \; \boxed{0} \; \ltimes$$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff $\mathbf{A} \cap \mathbf{B} \neq \emptyset$
- $L(\textbf{some}) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{^* \rtimes \ltimes\}$

F $\rtimes \quad \ltimes$

$\rtimes \; 0 \; 0 \; 0 \; \ltimes$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{^* \rtimes \ltimes\}$

$\rtimes \quad 1 \quad \ltimes$

$\rtimes \; 0 \; 1 \; 0 \; \ltimes$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L($**some**$) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{^* \rtimes \ltimes\}$

$$F \quad \boxed{\rtimes \qquad \ltimes}$$

$$\rtimes \quad 0 \quad 0 \quad 0 \quad \ltimes$$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{^* \rtimes \ltimes\}$

$$\rtimes \qquad 1 \qquad \ltimes$$

$$\rtimes \quad 0 \quad 1 \quad 0 \quad \ltimes$$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff $\mathbf{A} \cap \mathbf{B} \neq \emptyset$
- $L(\textbf{some}) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

F $\boxed{\rtimes \quad \ltimes}$

$\rtimes \; 0 \; 0 \; 0 \; \ltimes$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

T $\boxed{\rtimes \quad 1} \boxed{1 \quad \ltimes}$

$\rtimes \; 0 \; 1 \; 0 \; \ltimes$

Subregular Quantifiers: *Some* is TSL

Reminder: *some*

- **some**(**A**, **B**) holds iff **A** $\cap$ **B** $\neq \emptyset$
- $L(\textbf{some}) = \{0,1\}^* \, 1 \, \{0,1\}^*$
- Eg. *Some student cheated.*

False

student	John, Mary, Sue
cheat	
binary strings	000
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

F $\boxed{\rtimes \qquad \ltimes}$

$\rtimes \;\; 0 \;\; 0^n \;\; 0 \;\; \ltimes$

True

student	John, Mary, Sue
cheat	John,
binary strings	100, 010, 001
grammar	$T = \{1\}$
	$S = \{{}^*\rtimes\ltimes\}$

T $\boxed{\rtimes \quad 1}$ $\boxed{1 \quad \ltimes}$

$\rtimes \;\; 0^n \;\; 1 \;\; 0^n \;\; \ltimes$

Parity Quantifiers?

An even number

- **An even number**(**A**, **B**) holds iff |**A** ∩ **B**| $\geq 2n$, with $n > 0$
- $L($**even**$) = \{w \in 0, 1^* s.t. |1|_w \geq 2n$, with $n > 0\}$

Is $L($**even**$)$ a TSL language?

F 1 1 1 0 0 T 1 1 1 1 0 F 1 1 1 1 1

Parity Quantifiers?

An even number

- **An even number**(**A**, **B**) holds iff $|$**A** $\cap$ **B**$| \geq 2n$, with $n > 0$
- $L($**even**$) = \{w \in 0, 1^* s.t. |1|_w \geq 2n$, with $n > 0\}$

Is $L($**even**$)$ a TSL language?

F 1 1 1 0 0 T 1 1 1 1 0 F 1 1 1 1 1

Parity Quantifiers?

An even number

- **An even number**(**A**, **B**) holds iff |**A** ∩ **B**| $\geq 2n$, with $n > 0$
- $L($**even**$) = \{w \in 0, 1^* s.t. |1|_w \geq 2n$, with $n > 0\}$

Is $L($**even**$)$ a TSL language?

F 1 1 1 0 0   T 1 1 1 1 0   F 1 1 1 1 1

Parity Quantifiers?

An even number

- **An even number**(**A**, **B**) holds iff |**A** ∩ **B**| $\geq 2n$, with $n > 0$
- $L($**even**$) = \{w \in 0, 1^* s.t. |1|_w \geq 2n$, with $n > 0\}$

Is $L($**even**$)$ a TSL language?

1 1 1	1 1 1 1	1 1 1 1 1
F 1 1 1 0 0	T 1 1 1 1 0	F 1 1 1 1 1

Parity Quantifiers?

An even number

- **An even number**(**A**, **B**) holds iff $|\mathbf{A} \cap \mathbf{B}| \geq 2n$, with $n > 0$
- $L(\textbf{even}) = \{w \in 0, 1^* s.t. |1|_w \geq 2n$, with $n > 0\}$

Is $L(\textbf{even})$ a TSL language?

F $\boxed{1\ 1\ 1}$ — F $\boxed{1\ 1\ 1}$ 1 — F $\boxed{1\ 1\ 1}$ 1 1

F 1 1 1 0 0 — T 1 1 1 1 0 — F 1 1 1 1 1

Parity Quantifiers?

An even number

- **An even number**(**A**, **B**) holds iff |**A** ∩ **B**| $\geq 2n$, with $n > 0$
- $L($**even**$) = \{w \in 0, 1^* s.t. |1|_w \geq 2n$, with $n > 0\}$

Is $L($**even**$)$ a TSL language?

F 1 1 1 T 1 1 1 1 T 1 1 1 1 1

F 1 1 1 0 0 T 1 1 1 1 0 F 1 1 1 1 1

Parity Quantifiers?

An even number

- **An even number**(**A**, **B**) holds iff $|$**A** $\cap$ **B**$| \geq 2n$, with $n > 0$
- $L($**even**$) = \{w \in 0, 1^* s.t. |1|_w \geq 2n$, with $n > 0\}$

Is $L($**even**$)$ a TSL language?

F 1 1 1	T 1 1 1 1	F 1 1 1 1 1
F 1 1 1 0 0	T 1 1 1 1 0	F 1 1 1 1 1

Parity Quantifiers?

An even number

- **An even number**(**A**, **B**) holds iff $|\mathbf{A} \cap \mathbf{B}| \geq 2n$, with $n > 0$
- $L(\textbf{even}) = \{w \in 0, 1^* s.t. |1|_w \geq 2n$, with $n > 0\}$

Is $L(\textbf{even})$ a TSL language?

F	T	F
$\boxed{1\ 1\ 1}$	$\boxed{1\ 1\ 1\ 1}$	$\boxed{1\ 1\ 1\ 1\ 1}$
F 1 1 1 0 0	T 1 1 1 1 0	F 1 1 1 1 1

Since n is arbitrary, there is **no general TSL grammar** that can generate $L(\textbf{even})$.

Characterization of Quantifier Languages: Summary

Language	Constraint	Complexity	Subregular Grammar
every	$\|0\|_w = 0$	SL-1	**S** $:= \{\neg 0\}$
no	$\|1\|_w = 0$	SL-1	**S** $:= \{\neg 1\}$
some	$\|1\|_w \geq 1$	TSL-2	**T** $:= \{1\}$, **S** $:= \{\neg \rtimes \ltimes\}$
not all	$\|0\|_w \geq 1$	TSL-2	**T** $:= \{0\}$, **S** $:= \{\neg \rtimes \ltimes\}$
(at least) n	$\|1\|_w \geq n$	TSL-$(n+1)$	**T** $:= \{1\}$, **S** $:= \left\{\neg \rtimes 1^k \ltimes\right\}_{k \leq n}$
(at most) n	$\|1\|_w \leq n$	TSL-$(n+1)$	**T** $:= \{1\}$, **S** $:= \left\{\neg 1^{k+1}\right\}$
all but n	$\|0\|_w = n$	TSL-$(n+1)$	**T** $:= \{0\}$, **S** $:= \left\{\neg 0^{n+1}, \neg \rtimes 0^k \ltimes\right\}_{k \leq n}$
even number	$\|1\|_w = 2n,\ n \geq 0$	regular	**impossible**
most	$\|1\|_w \geq \|0\|_w$	context-free	**impossible**

A Complexity Hierarchy (Revisited)

- Semantic Automata predictions

FSA	PDA
$\{$*All, Some*$\} < \{$*Even, Odd*$\} < \{$*At least n, At most n*$\}$	$< \{$*Less than half, More than half, Most*$\}$

- Subregular characterization predictions

SL	TSL	REG	CF
$\{$*All*$\}$	$< \{$*Some, At least n, At most n*$\}$	$< \{$*Even, Odd*$\}$	$< \{$*Less than half, More than half, Most*$\}$

Automata vs Quantifier Languages

- cardinal $<$ parity;
- complexity independent of the specific recognition machine
- **what's the cognitive reality of these predictions?**

A Complexity Hierarchy (Revisited)

- Semantic Automata predictions

FSA				PDA
{*All*, *Some*}	< {*Even*, *Odd*}	< {*At least n*, *At most n*}	<	{*Less than half*, *More than half*, *Most*}

- Subregular characterization predictions

SL		TSL		REG		CF
{*All*}	<	{*Some*, *At least n*, *At most n*}	<	{*Even*, *Odd*}	<	{*Less than half*, *More than half*, *Most*}

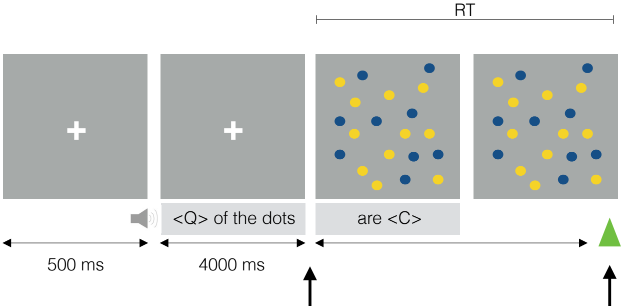
Automata vs Quantifier Languages

- cardinal < parity;
- complexity independent of the specific recognition machine
- **what's the cognitive reality of these predictions?**

Formal Complexity and Cognition

- FO-quantifiers vs higher order quantifiers
  - neuroimaging (McMillan et al. 2005, Clark & Grossman 2007)
  - patient literature (McMillan et al. 2009, Troiani et al 2009,)
- Psycholinguistic evidence for semantic automata
  - many behavioral findings (Szymanik & Zajenkowsky 2009, 2010, Steinert-Threlkeld & Icard 2013, i.a.)
  - for a survey: Szymanik (2016)
- Subregular hierarchy and cognition
  - general discussion(Rogers et al. 2013)
  - animal vs human cognition (Pulum & Rogers 2006, Rogers & Pullum 2011)
  - learnability and acquisition (Lai 2015, Avcu 2017)

Testing the Subregular Predictions



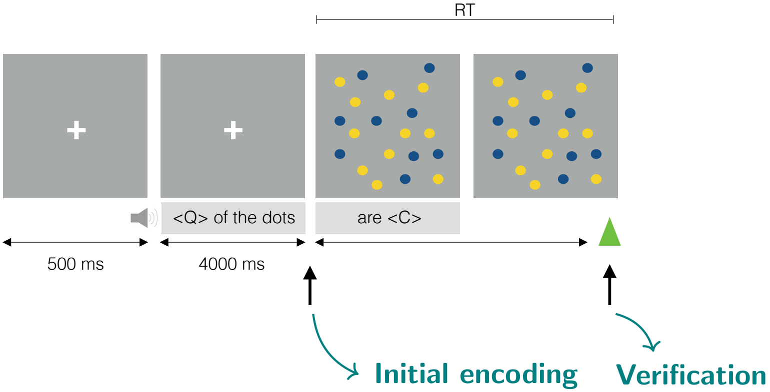
Initial encoding **Verification**

- McMillan et al. (2005)

Disentangling encoding from verification

- ERP → upcoming (De Santo et al, SNL2017), MEG, ...
- Pupil size ← **ongoing...**

Testing the Subregular Predictions

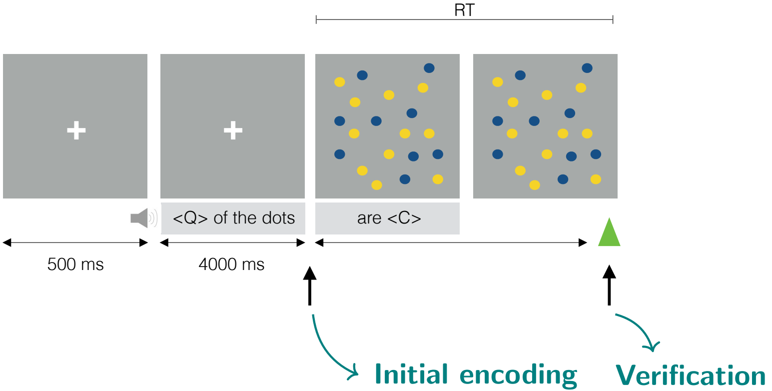


- McMillan et al. (2005)

Disentangling **encoding** from **verification**

- ERP → upcoming (De Santo et al, SNL2017), MEG, ...
- Pupil size ← **ongoing...**

Testing the Subregular Predictions



- McMillan et al. (2005)

Disentangling encoding from verification

- ERP → upcoming (De Santo et al, SNL2017), MEG, ...
- Pupil size ← **ongoing...**

Conclusions

Tracing Back our Steps

- SA as a plausible model of quantifier complexity
- Refined by looking at weaker classes in the Chomsky hierarchy
  $\Rightarrow$ **subregular characterization of generalized quantifiers**

Outcomes & Future Work

- Computational complexity and cognition
  - precise, testable predictions about cognitive resources
  - strong, cross-domain linking hypothesis
- Support for cross-domain subregular generalizations
  - typological predictions (Graf 2017)
  - insights on learnability/acquisition
- New theoretical questions
  - e.g. permutation closure & subregular languages?

Conclusions

Tracing Back our Steps

- SA as a plausible model of quantifier complexity
- Refined by looking at weaker classes in the Chomsky hierarchy
  $\Rightarrow$ **subregular characterization of generalized quantifiers**

Outcomes & Future Work

- Computational complexity and cognition
  - precise, testable predictions about cognitive resources
  - strong, cross-domain linking hypothesis
- Support for cross-domain subregular generalizations
  - typological predictions (Graf 2017)
  - insights on learnability/acquisition
- New theoretical questions
  - e.g. permutation closure & subregular languages?

An algorithm is likely to be understood more readily by understanding **the nature of the problem** *being solved than by examining the mechanism (and the hardware) in which it is embodied.*

(Marr 1983, p.27)

Selected References I

1 **Aksënova, Alëna, Thomas Graf, and Sedigheh Moradi**. 2016. Morphotactics as tier-based strictly local dependencies. In Proceedings of SIGMorPhon 2016.

2 **Aksënova, Alëna and Aniello De Santo**. 2017. Strict Locality in morphological derivations. (to appear)In Proceedings of CLS53 2017.

3 **Avcu Enes**. 2017. Experimental investigation of the subregular hierarchy. In Proceedings of PLC41, 2017.

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

5 **Chandlee, Jane**. 2016. Computational locality in morphological maps. Ms., Haverford College.

6 **Graf, Thomas**. 2017. The subregular complexity of monomorphemic quantifiers. Ms., Stony Brook University.

7 **Graf, Thomas and Heinz, Jeffrey**. 2016. *Tier-based strictly locality in phonology and syntax*. Ms., Stony Brook University and University of Delaware.

8 **Heinz, Jeffrey**. 2015. *The Computational Nature of Phonological Generalizations*. Ms., University of Delaware.

9 **Heinz, Jeffrey**, Chetan Rawal, and Herbert G. Tanner. 2011.Tier-based strictly local constraints in phonology. In *Proceedings of ACL 49th*, 58–64.

10 **Kaplan, Ronald M., and Martin Kay**. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20:331?378.

11 **Karttunen, Lauri, Ronald M. Kaplan, and Annie Zaenen**. 1992. Two-level morphology with composition. In *COLING'92,* 141-148.

12 **Just, M. A., P. A. Carpenter, and A. Miyake**. 2003. Neuroindices of cognitive workload: Neuroimaging, pupillometric and event-related potential.

13 **Lai, Regine**. 2015. Learnable vs. unlearnable harmony patterns. Linguistic Inquiry 46:425?451.

Selected References II

14 **Marr, David** (1983). Vision: A Computational Investigation into the Human Representation and Processing Visual Information. San Francisco: W. H. Freeman.

15 **McMillan Corey T. , Robin Clark, Peachie Moore, Christian Devita, and Murray Grossman**. 2005. Neural basis for generalized quantifier comprehension. Neuropsychologia, 43(12):1729?1737, Jan2005.

16 **McNaughton, Robert, and Seymour Papert**. 1971. Counter-free automata. Cambridge, MA: MIT Press.

17 **Mostowski, M.** 1998. Computational semantics for monadic quantifiers. Journal of Applied Non-Classical Logics, 8, 107–121.

18 **Pullum, Geoffrey K., and James Rogers**. 2006. Animal pattern-learning experiments: Some mathematical background. Ms., Radcliffe Institute for Advanced Study, Harvard University.

19 **Rogers, J., J. Heinz, M. Fero, J. Hurst, D. Lambert, and S. Wibel** (2013). Cognitive and Subregular Complexity, Chapter Formal Grammar, pp. 90?108. Springer.

20 **Rogers, James and Geoffrey Pullum**. 2007. Aural Pattern Recognition Experiments and the Subregular Hierarchy. In Proc. of *Mathematics of Language 10*, 1 –16.

21 **Shieber, Stuart M.** 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*. 8:333–345.

22 **Steinert-Threlkeld Shane and Thomas F. Icard III**. 2013. Iterating Semantic Automata. Linguistics and Philosophy, 2013.

23 **Szymanik Jakub and Marcin Zajenkowski**. 2010. Comprehension of simple quantifiers: empirical evaluation of a computational model. Cognitive Science, 34(3):521–532, April 2010.

24 **Szymanik Jakub and Marcin Zajenkowski**. 2011. Contribution of working memory in parity and proportional judgments. Belgian Journal of Linguistics, 25(1):176–194, January 2011.

25 **Szymanik, Jacub**. 2016. Quantifiers and Cognition: Logical and Computational Perspectives. Springer International Publishing.

26 **Troiani, V., Peelle, J., Clark, R., and Grossman, M.** 2009. Is it logical to count on quantifiers?

Dissociable neural networks underlying numerical and logical quantifiers. Neuropsychologia, 47, 104–111.

Appendix

Logical Definability of Subregular Classes

	Regular	Monadic Second-Order Logic
	$\cup$	
Locally Threshold Testable $\subset$	Star Free	First-Order Logic
$\cup$	$\cup$	
Locally Testable	Piecewise Testable	Propositional Logic
$\cup$	$\cup$	
Strictly Local $\subset$ TSL	Strictly Piecewise	Conjunction of Negative Literals
$S/\triangleleft$	$< /\triangleleft^{+}$	

Strictly Local: Example

Word-Final Devoicing is SL

- SL grammars are lists of forbidden n-grams;
- Word-Final Devoicing: voiced segments at the end of a word are forbidden.

Word-Final Devoicing: German Example

- Grammar $S := \{{}^*z\ltimes, {}^*v\ltimes, {}^*d\ltimes\}$
- $\{\rtimes, \ltimes\} \rightarrow$ left and right word edge

$$^* \rtimes \; r \; a \; d \; \ltimes$$

$$\rtimes \; r \; a \; t \; \ltimes$$

Strictly Local: Example

Word-Final Devoicing is SL

- SL grammars are lists of forbidden n-grams;
- Word-Final Devoicing: voiced segments at the end of a word are forbidden.

Word-Final Devoicing: German Example

- Grammar $S := \{{}^{*}z\ltimes, {}^{*}v\ltimes, {}^{*}d\ltimes\}$
- $\{\rtimes, \ltimes\}$ → left and right word edge

$$^{*}\ \rtimes\ \mathrm{r}\ \mathrm{a}\ \mathrm{d}\ \ltimes$$

$$\rtimes\ \mathrm{r}\ \mathrm{a}\ \mathrm{t}\ \ltimes$$

Strictly Local: Example

Word-Final Devoicing is SL

- SL grammars are lists of forbidden n-grams;
- Word-Final Devoicing: voiced segments at the end of a word are forbidden.

Word-Final Devoicing: German Example

- Grammar $S := \{{}^{*}z\ltimes,\ {}^{*}v\ltimes,\ {}^{*}d\ltimes\}$
- $\{\rtimes, \ltimes\} \rightarrow$ left and right word edge

$$^{*}\ \rtimes\ r\ a\ \boxed{d\ \ltimes}$$

$$\rtimes\ r\ a\ t\ \ltimes$$

Strictly Local: Example

Word-Final Devoicing is SL

- SL grammars are lists of forbidden n-grams;
- Word-Final Devoicing: voiced segments at the end of a word are forbidden.

Word-Final Devoicing: German Example

- Grammar $S := \{{}^*z\ltimes,\ {}^*v\ltimes,\ {}^*d\ltimes\}$
- $\{\rtimes, \ltimes\} \rightarrow$ left and right word edge

$$^{*}\quad \rtimes \quad r \quad a \quad \boxed{d \quad \ltimes}$$

$$^{ok}\quad \rtimes \quad r \quad a \quad \boxed{t \quad \ltimes}$$

Tier-based Strictly Local (Heinz et al. 2011)

- TSL is a minimal expansion of SL
- Inspired by phonological tiers

TSL Grammars

- a projection function $E : \Sigma \to T \cup \lambda$, with $T \subseteq \Sigma$
- strictly local constraints over T

TSL Example: Sibilant harmony in Aari

- If multiple sibilants {s, z, ʒ, ʃ} occur in the same word, they must all be voiceless {s, ʃ} or voiced {z, ʒ} .

Grammar

- T := {s, z, ʒ, ʃ}
- S := {*ʒs, *sʒ, *sʃ, *ʃs}

* ʒ aː e r s e

^{ok} ʒ aː e r ʃ e

TSL Example: Sibilant harmony in Aari

- If multiple sibilants {s, z, ʒ, ʃ} occur in the same word, they must all be voiceless {s, ʃ} or voiced {z, ʒ} .

Grammar

- T := {s, z, ʒ, ʃ}
- S := {*ʒs, *sʒ, *sʃ, *ʃs}

ʒ

T: sibilant harmony

* ʒ a: e r s e

ok ʒ a: e r ʃ e

TSL Example: Sibilant harmony in Aari

- If multiple sibilants {s, z, ʒ, ʃ} occur in the same word, they must all be voiceless {s, ʃ} or voiced {z, ʒ} .

Grammar

- T := {s, z, ʒ, ʃ}
- S := {*ʒs, *sʒ, *sʃ, *ʃs}

ʒ

T: sibilant harmony

* ʒ a: e r s e

ok ʒ a: e r ʃ e

TSL Example: Sibilant harmony in AARI

- If multiple sibilants {s, z, ʒ, ʃ} occur in the same word, they must all be voiceless {s, ʃ} or voiced {z, ʒ} .

Grammar

- T := {s, z, ʒ, ʃ}
- S := {*ʒs, *sʒ, *sʃ, *ʃs}

ʒ

T: sibilant harmony

* ʒ a: e r s e

ok ʒ a: e r ʃ e

TSL Example: Sibilant harmony in AARI

- If multiple sibilants {s, z, ʒ, ʃ} occur in the same word, they must all be voiceless {s, ʃ} or voiced {z, ʒ} .

Grammar

- T := {s, z, ʒ, ʃ}
- S := {*ʒs, *sʒ, *sʃ, *ʃs}

ʒ

T: sibilant harmony

* ʒ a: e r s e

ok ʒ a: e r ʃ e

TSL Example: Sibilant harmony in Aari

- If multiple sibilants {s, z, ʒ, ʃ} occur in the same word, they must all be voiceless {s, ʃ} or voiced {z, ʒ} .

Grammar

- T := {s, z, ʒ, ʃ}
- S := {*ʒs, *sʒ, *sʃ, *ʃs}

ʒ s

T: sibilant harmony

* ʒ aː e r s e

ok ʒ aː e r ʃ e

TSL Example: Sibilant harmony in Aari

- If multiple sibilants {s, z, ʒ, ʃ} occur in the same word, they must all be voiceless {s, ʃ} or voiced {z, ʒ} .

Grammar

- T := {s, z, ʒ, ʃ}
- S := {*ʒs, *sʒ, *sʃ, *ʃs}

ʒ s

T: sibilant harmony

* ʒ aː e r s e

ok ʒ aː e r ʃ e

TSL Example: Sibilant harmony in AARI

- If multiple sibilants {s, z, ʒ, ʃ} occur in the same word, they must all be voiceless {s, ʃ} or voiced {z, ʒ} .

Grammar

- T := {s, z, ʒ, ʃ}
- S := {*ʒs, *sʒ, *sʃ, *ʃs}

*ʒ s

T: sibilant harmony

* ʒ aː e r s e

ok ʒ aː e r ʃ e

TSL Example: Sibilant harmony in Aari

- If multiple sibilants {s, z, ʓ, ʃ} occur in the same word, they must all be voiceless {s, ʃ} or voiced {z, ʓ} .

Grammar

- T := {s, z, ʓ, ʃ}
- S := {*ʓs, *sʓ, *sʃ, *ʃs}

* ʓ s

T: sibilant harmony

* ʓ a: e r s e

T: sibilant harmony

ok ʓ a: e r ʃ e

TSL Example: Sibilant harmony in Aari

- If multiple sibilants {s, z, ʒ, ʃ} occur in the same word, they must all be voiceless {s, ʃ} or voiced {z, ʒ} .

Grammar

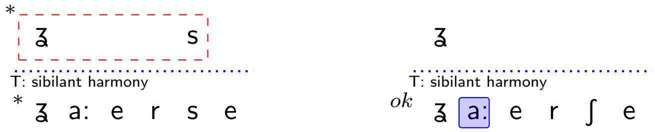
- T := {s, z, ʒ, ʃ}
- S := {*ʒs, *sʒ, *sʃ, *ʃs}

TSL Example: Sibilant harmony in Aari

- If multiple sibilants {s, z, ʒ, ʃ} occur in the same word, they must all be voiceless {s, ʃ} or voiced {z, ʒ} .

Grammar

- T := {s, z, ʒ, ʃ}
- S := {*ʒs, *sʒ, *sʃ, *ʃs}



TSL Example: Sibilant harmony in Aari

- If multiple sibilants {s, z, ʒ, ʃ} occur in the same word, they must all be voiceless {s, ʃ} or voiced {z, ʒ} .

Grammar

- T := {s, z, ʒ, ʃ}
- S := {*ʒs, *sʒ, *sʃ, *ʃs}

TSL Example: Sibilant harmony in Aari

- If multiple sibilants {s, z, ʒ, ʃ} occur in the same word, they must all be voiceless {s, ʃ} or voiced {z, ʒ} .

Grammar

- T := {s, z, ʒ, ʃ}
- S := {*ʒs, *sʒ, *sʃ, *ʃs}

* ʒ s

T: sibilant harmony

* ʒ aː e r s e

ʒ

T: sibilant harmony

ok ʒ aː e r ʃ e

TSL Example: Sibilant harmony in Aari

- If multiple sibilants {s, z, ʒ, ʃ} occur in the same word, they must all be voiceless {s, ʃ} or voiced {z, ʒ} .

Grammar

- T := {s, z, ʒ, ʃ}
- S := {*ʒs, *sʒ, *sʃ, *ʃs}

* ʒ s

T: sibilant harmony

* ʒ a: e r s e

ʒ ʃ

T: sibilant harmony

ok ʒ a: e r ʃ e

TSL Example: Sibilant harmony in Aari

- If multiple sibilants {s, z, ʒ, ʃ} occur in the same word, they must all be voiceless {s, ʃ} or voiced {z, ʒ} .

Grammar

- T := {s, z, ʒ, ʃ}
- S := {*ʒs, *sʒ, *sʃ, *ʃs}

TSL Example: Sibilant harmony in Aari

- If multiple sibilants {s, z, ʒ, ʃ} occur in the same word, they must all be voiceless {s, ʃ} or voiced {z, ʒ} .

Grammar

- T := {s, z, ʒ, ʃ}
- S := {*ʒs, *sʒ, *sʃ, *ʃs}