

# Polyadic Quantifiers and Computational Complexity

Jakub Szymanik

# Outline

## Preliminaries

- Quantifiers in finite models
- Computational complexity

Almost all complex quantifiers are simple

Some complex GQs are intractable

- Branching quantifiers
- Strong reciprocity

A complexity perspective on shifts in meaning

# Outline

## Preliminaries

- Quantifiers in finite models
- Computational complexity

Almost all complex quantifiers are simple

Some complex GQs are intractable

- Branching quantifiers
- Strong reciprocity

A complexity perspective on shifts in meaning



# Outline

## Preliminaries

Quantifiers in finite models

Computational complexity

Almost all complex quantifiers are simple

Some complex GQs are intractable

Branching quantifiers

Strong reciprocity

A complexity perspective on shifts in meaning

# Generalized Quantifier Theory

- ▶ Quantifiers occur whenever we speak.

# Generalized Quantifier Theory

- ▶ Quantifiers occur whenever we speak.
- ▶ They influence language expressivity.

# Generalized Quantifier Theory

- ▶ Quantifiers occur whenever we speak.
- ▶ They influence language expressivity.
- ▶ Classical GQT studies definability issues.

# Computational Complexity of Quantifiers

- ▶ What amount of resources is needed for processing?



# Computational Complexity of Quantifiers

- ▶ What amount of resources is needed for processing?
- ▶ Model-checking is a part of comprehension.

# Computational Complexity of Quantifiers

- ▶ What amount of resources is needed for processing?
- ▶ Model-checking is a part of comprehension.
- ▶ Input:  $\mathbb{M}$ ,  $\varphi$ . Output:  $\varphi^{\mathbb{M}}$ .



# Computational Complexity of Quantifiers

- ▶ What amount of resources is needed for processing?
- ▶ Model-checking is a part of comprehension.
- ▶ Input:  $\mathbb{M}$ ,  $\varphi$ . Output:  $\varphi^{\mathbb{M}}$ .
- ▶ W.r.t. to model size.

# Computational Complexity of Quantifiers

- ▶ What amount of resources is needed for processing?
- ▶ Model-checking is a part of comprehension.
- ▶ Input:  $\mathbb{M}$ ,  $\varphi$ . Output:  $\varphi^{\mathbb{M}}$ .
- ▶ W.r.t. to model size.
- ▶ Restriction to finite models.

# Lindström Quantifiers

## Definition

Let  $t = (n_1, \dots, n_k)$  be a  $k$ -tuple of positive integers. A *generalized quantifier* of type  $t$  is a class  $Q$  of models of a vocabulary  $\tau_t = \{R_1, \dots, R_k\}$ , such that  $R_i$  is  $n_i$ -ary for  $1 \leq i \leq k$ , and  $Q$  is closed under isomorphisms.

# Lindström Quantifiers

## Definition

Let  $t = (n_1, \dots, n_k)$  be a  $k$ -tuple of positive integers. A *generalized quantifier* of type  $t$  is a class  $Q$  of models of a vocabulary  $\tau_t = \{R_1, \dots, R_k\}$ , such that  $R_i$  is  $n_i$ -ary for  $1 \leq i \leq k$ , and  $Q$  is closed under isomorphisms.

## Definition

If in the above definition for all  $i$ :  $n_i = 1$ , then we say that a quantifier is *monadic*, otherwise we call it *polyadic*.

# GQs as Classes of Models

$$\forall = \{(M, P) \mid P = M\}.$$

$$\exists = \{(M, P) \mid P \subseteq M \text{ \& } P \neq \emptyset\}.$$

# GQs as Classes of Models

$$\forall = \{(M, P) \mid P = M\}.$$

$$\exists = \{(M, P) \mid P \subseteq M \ \& \ P \neq \emptyset\}.$$

$$\text{even} = \{(M, P) \mid P, S \subseteq M \ \& \ \text{card}(P \cap S) \text{ is even}\}.$$

$$\text{most} = \{(M, P, S) \mid P, S \subseteq M \ \& \ \text{card}(P \cap S) > \text{card}(P - S)\}.$$

$$\text{some} = \{(M, P, S) \mid P, S \subseteq M \ \& \ P \cap S \neq \emptyset\}.$$

# GQs as Classes of Models

$$\forall = \{(M, P) \mid P = M\}.$$

$$\exists = \{(M, P) \mid P \subseteq M \text{ \& } P \neq \emptyset\}.$$

$$\text{even} = \{(M, P) \mid P, S \subseteq M \text{ \& } \text{card}(P \cap S) \text{ is even}\}.$$

$$\text{most} = \{(M, P, S) \mid P, S \subseteq M \text{ \& } \text{card}(P \cap S) > \text{card}(P - S)\}.$$

$$\text{some} = \{(M, P, S) \mid P, S \subseteq M \text{ \& } P \cap S \neq \emptyset\}.$$

$$W = \{(M, R) \mid R \subseteq M^2 \text{ \& } R \text{ is a well-order}\}.$$

$$\text{Ram} = \{(M, A, R) \mid A \subseteq M, R \subseteq M^2 \text{ \& } \forall a, b \in A R(a, b)\}.$$

# Monadic GQs - Some Examples

- ▶ Every poet has low self-esteem.
- ▶ Some dean danced nude on the table.
- ▶ At least 3 grad students prepared presentations.
- ▶ An even number of the students saw a ghost.
- ▶ Most of the students think they are smart.
- ▶ Less than half of the students received good marks.

# Monadic GQs

1. They are easy to compute: FA, PDA.
2. Computational model is neuropsychologically plausible.

# Polyadic Quantifiers in Language

- ▶ Use to model multi-quantifier sentences, e.g.:

# Polyadic Quantifiers in Language

- ▶ Use to model multi-quantifier sentences, e.g.:
  1. Most villagers and most townsmen hate each other.
  2. Three PMs referred to each other indirectly

# Polyadic Quantifiers in Language

- ▶ Use to model multi-quantifier sentences, e.g.:
  1. Most villagers and most townsmen hate each other.
  2. Three PMs referred to each other indirectly
- ▶ They have many possible meanings.

# Polyadic Quantifiers in Language

- ▶ Use to model multi-quantifier sentences, e.g.:
  1. Most villagers and most townsmen hate each other.
  2. Three PMs referred to each other indirectly
- ▶ They have many possible meanings.
- ▶ A lot of definability results.

# Polyadic Quantifiers in Language

- ▶ Use to model multi-quantifier sentences, e.g.:
  1. Most villagers and most townsmen hate each other.
  2. Three PMs referred to each other indirectly
- ▶ They have many possible meanings.
- ▶ A lot of definability results.

## Question

1. *What about c.c. of polyadic quantifiers?*

# Polyadic Quantifiers in Language

- ▶ Use to model multi-quantifier sentences, e.g.:
  1. Most villagers and most townsmen hate each other.
  2. Three PMs referred to each other indirectly
- ▶ They have many possible meanings.
- ▶ A lot of definability results.

## Question

1. *What about c.c. of polyadic quantifiers?*
2. *How various readings differ w.r.t. complexity?*

# Quantifiers in Finite Models

- ▶ Finite models can be encoded as strings.
- ▶ GQs as classes of such finite strings are languages.

# Quantifiers in Finite Models

- ▶ Finite models can be encoded as strings.
- ▶ GQs as classes of such finite strings are languages.

## Definition

By the *complexity of a quantifier*  $Q$  we mean the computational complexity of the corresponding class of finite models.

## Question

$M \in Q?$  (equivalently  $M \models Q?$ )

## Definition

Let  $\tau = \{R_1, \dots, R_k\}$  be a relational vocabulary and  $\mathbb{M}$  a  $\tau$ -model of the following form:  $\mathbb{M} = (U, R_1^M, \dots, R_k^M)$ , where  $U = \{1, \dots, n\}$  is the universe of model  $\mathbb{M}$  and  $R_i^M \subseteq U^{n_i}$  is an  $n_i$ -ary relation over  $U$ , for  $1 \leq i \leq k$ . We define a *binary encoding for  $\tau$ -models*. The code for  $\mathbb{M}$  is a word over  $\{0, 1, \#\}$  of length  $O((\text{card}(U))^c)$ , where  $c$  is the maximal arity of the predicates in  $\tau$  (or  $c = 1$  if there are no predicates).

The code has the following form:

$$\tilde{n}\#\tilde{R}_1^M\#\dots\#\tilde{R}_n^M, \text{ where:}$$

- ▶  $\tilde{n}$  is the part coding the universe of the model and consists of  $n$  1s.
- ▶  $\tilde{R}_i^M$  — the code for the  $n_i$ -ary relation  $R_i^M$  — is an  $n^{n_i}$ -bit string whose  $j$ -th bit is 1 iff the  $j$ -th tuple in  $U^{n_i}$  (ordered lexicographically) is in  $R_i^M$ .
- ▶  $\#$  is a separating symbol.

# Coding Example

Consider vocabulary  $\sigma = \{P, R\}$ , where  $P$  is a unary predicate and  $R$  a binary relation. Take the  $\sigma$ -model  $\mathbb{M} = (M, P^M, R^M)$ , where the universe  $M = \{1, 2, 3\}$ , the unary relation  $P^M \subseteq M$  is equal to  $\{2\}$  and the binary relation  $R^M \subseteq M^2$  consists of the pairs  $(2, 2)$  and  $(3, 2)$ .

- ▶  $\tilde{n}$  consists of three 1s as there are three elements in  $M$ .
- ▶  $\tilde{P}^M$  is the string of length three with 1s in places corresponding to the elements from  $M$  belonging to  $P^M$ . Hence  $\tilde{P}^M = 010$  as  $P^M = \{2\}$ .
- ▶  $\tilde{R}^M$  is obtained by writing down all  $3^2 = 9$  binary strings of elements from  $M$  in lexicographical order and substituting 1 in places corresponding to the pairs belonging to  $R^M$  and 0 in all other places. As a result  $\tilde{R}^M = 000010010$ .

Adding all together the code for  $\mathbb{M}$  is  $111\#010\#000010010$ .

# Outline

## Preliminaries

Quantifiers in finite models

Computational complexity

Almost all complex quantifiers are simple

Some complex GQs are intractable

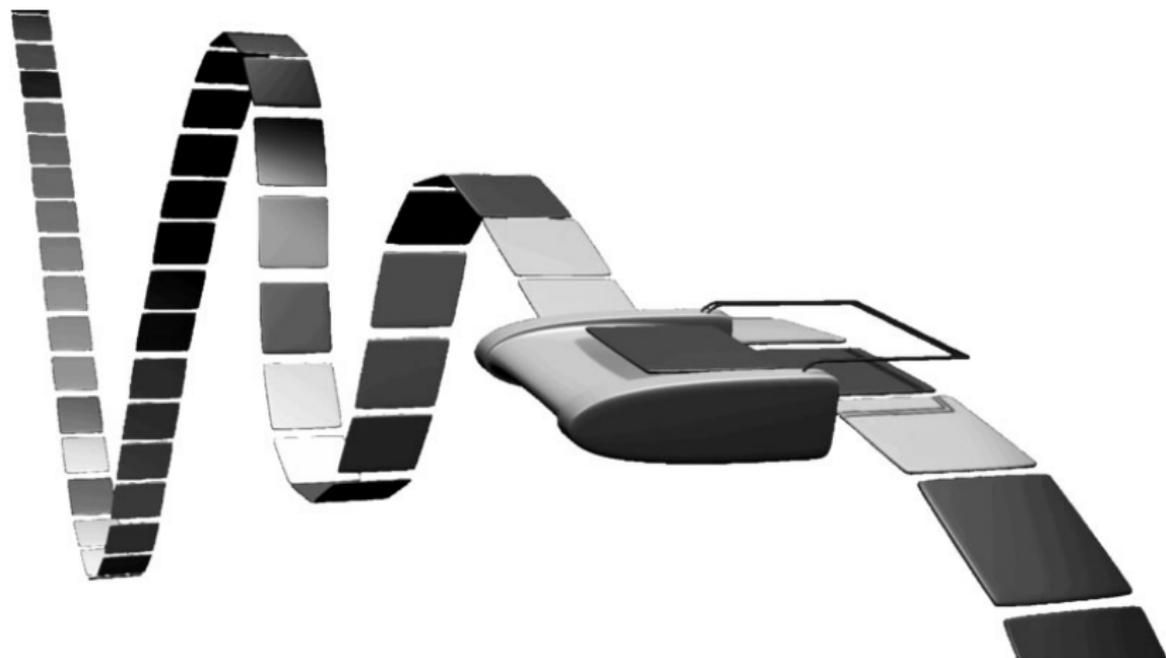
Branching quantifiers

Strong reciprocity

A complexity perspective on shifts in meaning



# Model of Computation



# Computational Complexity Theory

## Question

*What amount of resources TM needs to solve a task?*



# Computational Complexity Theory

## Question

*What amount of resources TM needs to solve a task?*

## Theorem (determinism vs. non-determinism)

*If there is a non-deterministic Turing machine  $N$  recognizing a language  $L$ , then there exists a deterministic Turing machine  $M$  for language  $L$ .*

## Question

*The simulation takes  $O(c^{f(n)})$ . Can we do it **significantly** faster?*

# Time Complexity

Let  $f : \omega \longrightarrow \omega$ .

# Time Complexity

Let  $f : \omega \longrightarrow \omega$ .

## Definition

$\text{TIME}(f)$  is the class of languages (problems) which can be recognized by a deterministic Turing machine in time bounded by  $f$  with respect to the length of the input.

# Time Complexity

Let  $f : \omega \longrightarrow \omega$ .

## Definition

$\text{TIME}(f)$  is the class of languages (problems) which can be recognized by a deterministic Turing machine in time bounded by  $f$  with respect to the length of the input.

## Definition

$\text{NTIME}(f)$ , is the class of languages  $L$  for which there exists a non-deterministic Turing machine  $M$  such that for every  $x \in L$  all branches in the computation tree of  $M$  on  $x$  are bounded by  $f(n)$  and moreover  $M$  decides  $L$ .

# Complexity Classes P and NP

## Definition

- ▶  $\text{PTIME} = \bigcup_{k \in \omega} \text{TIME}(n^k)$
- ▶  $\text{NPTIME} = \bigcup_{k \in \omega} \text{NTIME}(n^k)$



# Complexity Classes P and NP

## Definition

- ▶  $\text{PTIME} = \bigcup_{k \in \omega} \text{TIME}(n^k)$
- ▶  $\text{NPTIME} = \bigcup_{k \in \omega} \text{NTIME}(n^k)$

## Question (Millenium Problem)

$P=NP?$

# (In)tractability

## Definition

We say that a function  $f : A \rightarrow A$  is a *polynomial time computable function* iff there exists a deterministic Turing machine computing  $f(w)$  for every  $w \in A$  in polynomial time.

# (In)tractability

## Definition

We say that a function  $f : A \rightarrow A$  is a *polynomial time computable function* iff there exists a deterministic Turing machine computing  $f(w)$  for every  $w \in A$  in polynomial time.

## Definition

A problem  $L$  is polynomial reducible to a problem  $L'$  if there is a polynomial time computable function such that

$$w \in L \iff f(w) \in L'.$$

# (In)tractability

## Definition

We say that a function  $f : A \rightarrow A$  is a *polynomial time computable function* iff there exists a deterministic Turing machine computing  $f(w)$  for every  $w \in A$  in polynomial time.

## Definition

A problem  $L$  is polynomial reducible to a problem  $L'$  if there is a polynomial time computable function such that

$$w \in L \iff f(w) \in L'.$$

## Definition

A language  $L$  is NP-complete if  $L \in NP$  and every language in  $NP$  is reducible to  $L$ .

## NP Problems

P Problems

NP-complete  
Problems



# Outline

## Preliminaries

- Quantifiers in finite models
- Computational complexity

## Almost all complex quantifiers are simple

## Some complex GQs are intractable

- Branching quantifiers
- Strong reciprocity

## A complexity perspective on shifts in meaning



# Boolean Combinations of Determininers

1. At least 5 or at most 10 departments can win EU grants.
2. Between 100 and 200 students run in the marathon.
3. Not all students passed.
4. All students did not pass.

# Boolean Combinations of Quantifiers

## Definition

Let  $Q, Q'$  be generalized quantifiers, both of type  $(n_1, \dots, n_k)$ .  
We define:

$$(Q \wedge Q')_M[R_1, \dots, R_k] \iff Q_M[R_1, \dots, R_k] \text{ and } Q'_M[R_1, \dots, R_k]$$

$$(Q \vee Q')_M[R_1, \dots, R_k] \iff Q_M[R_1, \dots, R_k] \text{ or } Q'_M[R_1, \dots, R_k]$$

$$(\neg Q)_M[R_1, \dots, R_k] \iff \text{not } Q_M[R_1, \dots, R_k]$$

$$(Q\neg)_M[R_1, \dots, R_k] \iff Q_M[R_1, \dots, R_{k-1}, M - R_k]$$



# Boolean Operations are Tractable

## Theorem

*Let  $Q$  and  $Q'$  be generalized quantifiers computable in PTIME. Then the quantifiers: (1)  $\neg Q$ ; (2)  $Q\neg$ ; (3)  $Q \wedge Q'$ ; (3)  $Q \vee Q'$  are PTIME computable.*

# Iteration

1. Most logicians criticized some papers.
2. It(most, some)[Logicians, Papers, Criticized].

## Definition

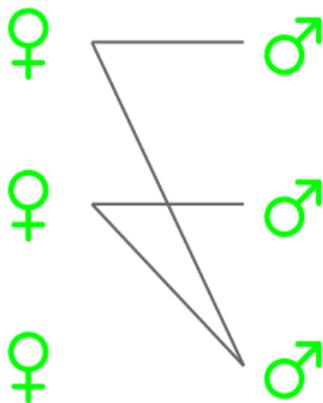
Let  $Q$  and  $Q'$  be generalized quantifiers of type  $(1, 1)$ . Let  $A, B$  be subsets of the universe and  $R$  a binary relation over the universe. Suppressing the universe, we will define the *iteration* operator as follows:

$$\text{It}(Q, Q')[A, B, R] \iff Q[A, \{a \mid Q'[B, R_{(a)}]\}],$$

where  $R_{(a)} = \{b \mid R(a, b)\}$ .

# Illustration

- ▶ Most girls and most boys hate each other.



# Iteration is Easy

## Theorem

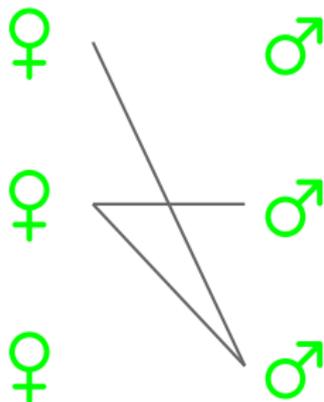
*Let  $Q$  and  $Q'$  be generalized quantifiers computable in PTIME. Then the quantifier  $It(Q, Q')$  is also PTIME computable.*





# Illustration

- ▶ Most girls and most boys hate each other.



# Cumulation is Easy

## Theorem

*Let  $Q$  and  $Q'$  be generalized quantifiers computable in PTIME.  
Then the quantifier  $\text{Cum}(Q, Q')$  is PTIME computable.*

# Resumption

- ▶ Most twins never separate.

## Definition

Let  $Q$  be any monadic quantifier with  $n$  arguments,  $U$  a universe, and  $R_1, \dots, R_n \subseteq U^k$  for  $k \geq 1$ . We define the *resumption* operator as follows:

$$\text{Res}^k(Q)_U[R_1, \dots, R_n] \iff (Q)_{U^k}[R_1, \dots, R_n].$$

# Resumption is Easy

## Theorem

*Let  $Q$  and  $Q'$  be generalized quantifiers computable in PTIME.  
Then the quantifier  $\text{Res}(Q, Q')$  is PTIME computable.*



# Basic Operations are Tractable

## Theorem

*Let  $Q$  and  $Q'$  be generalized quantifiers computable in PTIME. Then the quantifiers: (1)  $\neg Q$ ; (2)  $Q\neg$ ; (3)  $Q \wedge Q'$ ; (4)  $\text{It}(Q, Q')$ ; (5)  $\text{Cum}(Q, Q')$ ; (6)  $\text{Res}(Q)$  are PTIME computable.*

# Take Home Message

Everyday simple determiners in NL are in PTIME.

# Take Home Message

Everyday simple determiners in NL are in PTIME.  
PTIME quantifiers are closed under the common polyadic lifts.



# Outline

## Preliminaries

- Quantifiers in finite models
- Computational complexity

Almost all complex quantifiers are simple

## Some complex GQs are intractable

- Branching quantifiers
- Strong reciprocity

A complexity perspective on shifts in meaning

# Possibly Branching Sentences

1. Most villagers and most townsmen hate each other.
2. One third of villagers and half of townsmen hate each other.
3. 5 villagers and 7 townsmen hate each other.

# Branching Reading

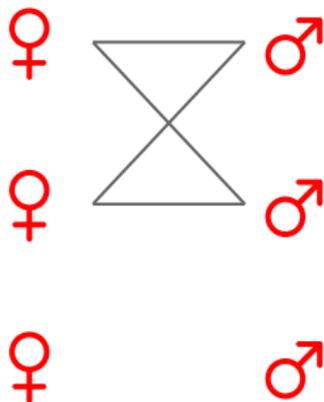
- ▶ Most girls and most boys hate each other.

$$\begin{array}{l} \text{most } x : G(x) \\ \text{most } y : B(y) \end{array} H(x, y).$$

$$\exists A \exists A' [\text{most}(G, A) \wedge \text{most}(B, A') \wedge \forall x \in A \forall y \in A' H(x, y)].$$

# Illustration

- ▶ Most girls and most boys hate each other.



# Definition

## Definition

Let  $Q$  and  $Q'$  be both  $\text{MON}\uparrow$  quantifiers of type  $(1, 1)$ . Define the *branching* of quantifier symbols  $Q$  and  $Q'$  as the type  $(1, 1, 2)$  quantifier symbol  $\text{Br}(Q, Q')$ .

A structure  $\mathbb{M} = (M, A, B, R) \in \text{Br}(Q, Q')$  if the following holds:

$$\exists X \subseteq A \exists Y \subseteq B [(X, A) \in Q \wedge (Y, B) \in Q' \wedge X \times Y \subseteq R].$$

# Branching Readings are Intractable

## Theorem

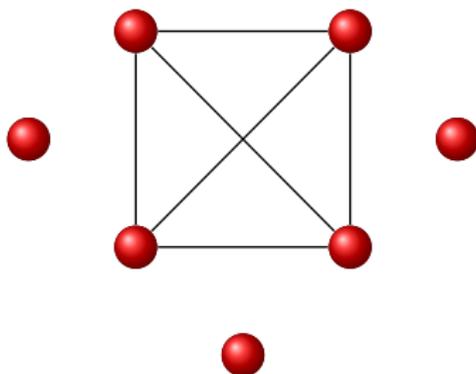
*Proportional branching sentences are NP-complete.*

# Potentially Strong Reciprocal Sentences

1. Andi, Jarmo and Jakub laughed at **one another**.
2. 15 men are hitting **one another**.
3. Most of the PMs refer to **each other**.

# Strong Reading

- ▶ Most of the PMs refer to each other.



# Strong Reciprocal Lift

## Definition

Let  $Q$  be a right monotone increasing quantifier of type  $(1, 1)$ .

We define:

$$\text{Ram}_S(Q)[A, R] \iff \exists X \subseteq A [Q(A, X) \\ \wedge \forall x, y \in X (x \neq y \implies R(x, y))].$$

# Strong Reciprocity is Intractable

## Theorem

*Model-checking for strong reciprocal sentences with proportional quantifiers is NP-complete.*





# Intermediate Reciprocal Lift

## Definition

$$\text{Ram}_1(Q)[A, R] \iff \exists X \subseteq A[Q(A, X) \\ \wedge \forall x, y \in X (x \neq y \implies \exists \text{ sequence } z_1, \dots, z_\ell \in X \text{ such that} \\ (z_1 = x \wedge R(z_1, z_2) \wedge \dots \wedge R(z_{\ell-1}, z_\ell) \wedge z_\ell = y))].$$

# Weak Reading

- ▶ Some pirates were staring at each other in surprise.



# Weak Reciprocal Lift

## Definition

$$\text{Ram}_W(Q)[A, R] \iff \exists X \subseteq A[Q(A, X) \\ \wedge \forall x \in X \exists y \in X (x \neq y \wedge R(x, y))].$$



# Complexity Dichotomy

As opposed to the strong case:

# Complexity Dichotomy

As opposed to the strong case:

## Theorem

*If  $Q$  is PTIME, then also  $\text{Ram}_I(Q)$  and  $\text{Ram}_W(Q)$  are in PTIME.*

# Outline

## Preliminaries

- Quantifiers in finite models
- Computational complexity

Almost all complex quantifiers are simple

Some complex GQs are intractable

- Branching quantifiers
- Strong reciprocity

A complexity perspective on shifts in meaning



# Strong Meaning Hypothesis

## Hypothesis

*Reading associated with the reciprocal in a given sentence is the strongest available reading which is consistent with relevant information supplied by the context.*

# Strong Meaning Hypothesis

## Hypothesis

*Reading associated with the reciprocal in a given sentence is the strongest available reading which is consistent with relevant information supplied by the context.*

## Example

1. The children followed each other into the church.
2. The children followed each other around the Maypole.



Dalrymple et al., Reciprocal Expressions and the Concept of Reciprocity. Linguistics and Philosophy, 1998.

# P-Cognition Thesis

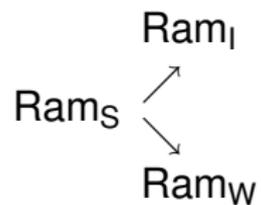
## Hypothesis

*Human cognitive (linguistic) capacities are constrained by polynomial time computability.*

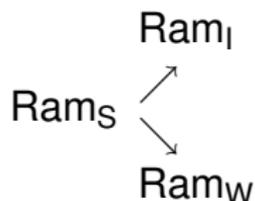


Frixione, Tractable Competence. Minds and Machines, 2001.

# Shifts Forced by Intractability



# Shifts Forced by Intractability



## Example

What is the default meaning of:

Most members of parliament refer to each other indirectly?

# Summary

- ▶ Standard multi-quantifier constructions are tractable.

# Summary

- ▶ Standard multi-quantifier constructions are tractable.
- ▶ Computational dichotomy between different readings.

# Summary

- ▶ Standard multi-quantifier constructions are tractable.
- ▶ Computational dichotomy between different readings.
- ▶ Revising the Strong Meaning Hypothesis with complexity.



Szymanik, Computational Complexity of Polyadic Lifts of Generalized Quantifiers in Natural Language, Linguistics and Philosophy, 2010.

# Experimental Question

## Question

*Is there any empirical support for those claims?*

# Experimental Question

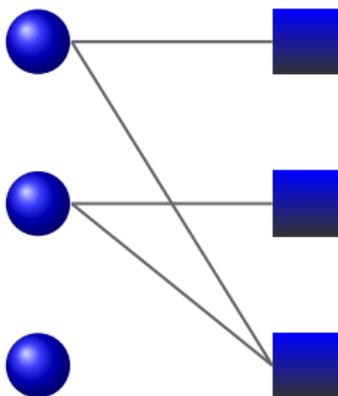
## Question

*Is there any empirical support for those claims?*

Well ...

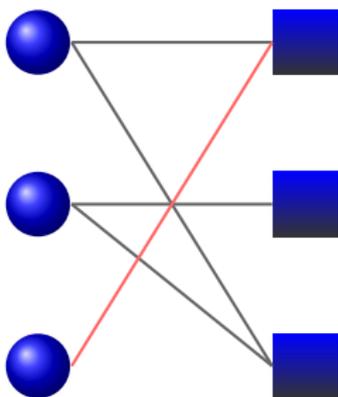
# Two-way Quantification

$$\text{It}(Q_1, Q_2) \wedge \text{It}(Q_2, Q_1)$$



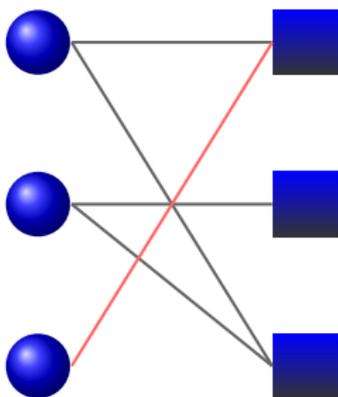
# Two-way Quantification

$$\text{It}(Q_1, Q_2) \wedge \text{It}(Q_2, Q_1)$$



# Two-way Quantification

$$\text{It}(Q_1, Q_2) \wedge \text{It}(Q_2, Q_1)$$



Subjects are happy to accept such interpretation.



Gierasimczuk and Szymanik, Branching Quantification vs. Two-way Quantification, Journal of Semantics, 2009

# Default Meanings Differ

Draw:

1. All/Most of the dots are connected to each other.

# Default Meanings Differ

Draw:

1. All/Most of the dots are connected to each other.
  - ▶ Against SMH:
    - ▶ Ambiguous between strong and intermediate.

# Default Meanings Differ

Draw:

1. All/Most of the dots are connected to each other.
  - ▶ Against SMH:
    - ▶ Ambiguous between strong and intermediate.
  - ▶ In line with complexity: fewer strong pictures for 'most'.



Bott et al., Interpreting Tractable versus Intractable Reciprocal Sentences, Proceedings of the International Conference on Computational Semantics, 2011.