

# Weak Cardinality Theorems for First-Order Logic

Till Tantau

Fakultät für Elektrotechnik und Informatik  
Technische Universität Berlin

Fundamentals of Computation Theory 2003



# Outline

## 1 History

- Enumerability in Recursion and Automata Theory
- Known Weak Cardinality Theorem
- Why Do Cardinality Theorems Hold Only for Certain Models?

## 2 Unification by First-Order Logic

- Elementary Definitions
- Enumerability for First-Order Logic
- Weak Cardinality Theorems for First-Order Logic

## 3 Applications

- A Separability Result for First-Order Logic

# Outline

## 1 History

- Enumerability in Recursion and Automata Theory
- Known Weak Cardinality Theorem
- Why Do Cardinality Theorems Hold Only for Certain Models?

## 2 Unification by First-Order Logic

- Elementary Definitions
- Enumerability for First-Order Logic
- Weak Cardinality Theorems for First-Order Logic

## 3 Applications

- A Separability Result for First-Order Logic

# Motivation of Enumerability

## Problem

Many functions are not computable or not efficiently computable.

## Example

- #SAT:  
How many satisfying assignments does a formula have?

# Motivation of Enumerability

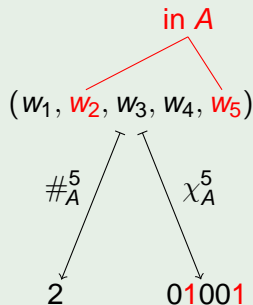
## Problem

Many functions are not computable or not efficiently computable.

## Example

For difficult languages  $A$ :

- Cardinality function  $\#_A^n$ :  
How many input words are in  $A$ ?
- Characteristic function  $\chi_A^n$ :  
Which input words are in  $A$ ?



# Motivation of Enumerability

## Problem

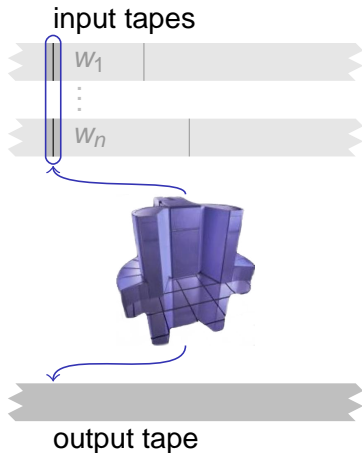
Many functions are not computable or not efficiently computable.

## Solutions

Difficult functions can be

- computed using probabilistic algorithms,
- computed efficiently on average,
- approximated, or
- **enumerated.**

# Enumerators Output Sets of Possible Function Values

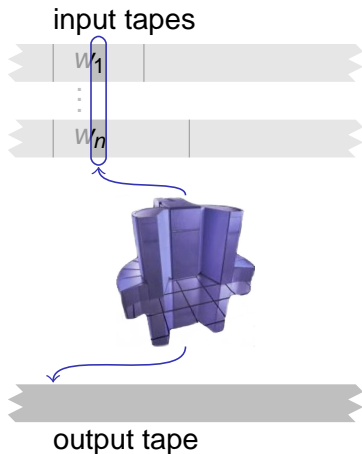


## Definition (1987, 1989, 1994, 2001)

An *m*-enumerator for a function  $f$

- 1 reads  $n$  input words  $w_1, \dots, w_n$ ,
- 2 does a computation,
- 3 outputs at most  $m$  values,
- 4 one of which is  $f(w_1, \dots, w_n)$ .

# Enumerators Output Sets of Possible Function Values



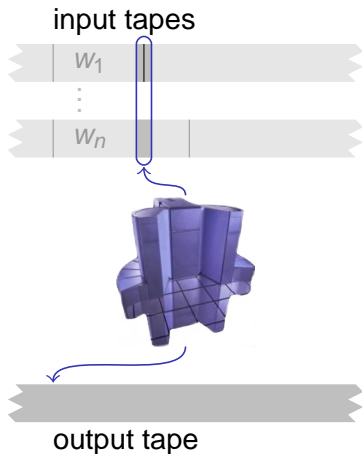
## Definition (1987, 1989, 1994, 2001)

An *m*-enumerator for a function  $f$

- 1 reads  $n$  input words  $w_1, \dots, w_n$ ,
- 2 does a computation,
- 3 outputs at most  $m$  values,
- 4 one of which is  $f(w_1, \dots, w_n)$ .



# Enumerators Output Sets of Possible Function Values

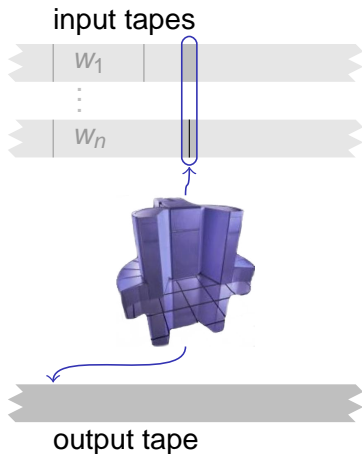


## Definition (1987, 1989, 1994, 2001)

An *m*-enumerator for a function  $f$

- 1 reads  $n$  input words  $w_1, \dots, w_n$ ,
- 2 does a computation,
- 3 outputs at most  $m$  values,
- 4 one of which is  $f(w_1, \dots, w_n)$ .

# Enumerators Output Sets of Possible Function Values

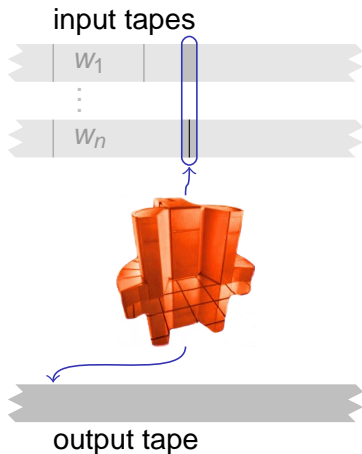


## Definition (1987, 1989, 1994, 2001)

An *m*-enumerator for a function  $f$

- 1 reads  $n$  input words  $w_1, \dots, w_n$ ,
- 2 does a computation,
- 3 outputs at most  $m$  values,
- 4 one of which is  $f(w_1, \dots, w_n)$ .

# Enumerators Output Sets of Possible Function Values

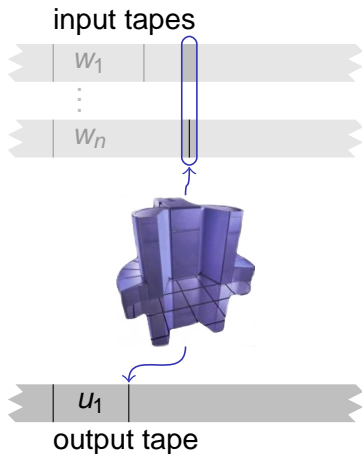


## Definition (1987, 1989, 1994, 2001)

An *m*-enumerator for a function  $f$

- 1 reads  $n$  input words  $w_1, \dots, w_n$ ,
- 2 does a computation,
- 3 outputs at most  $m$  values,
- 4 one of which is  $f(w_1, \dots, w_n)$ .

# Enumerators Output Sets of Possible Function Values

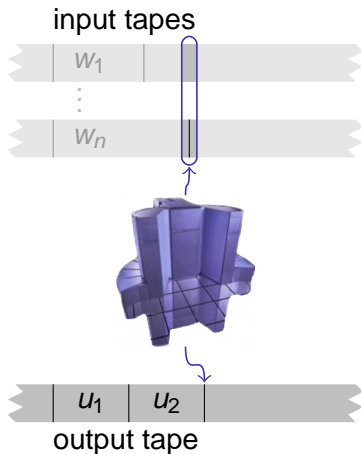


## Definition (1987, 1989, 1994, 2001)

An ***m*-enumerator** for a function  $f$

- 1 reads  $n$  input words  $w_1, \dots, w_n$ ,
- 2 does a computation,
- 3 **outputs at most  $m$  values,**
- 4 one of which is  $f(w_1, \dots, w_n)$ .

# Enumerators Output Sets of Possible Function Values

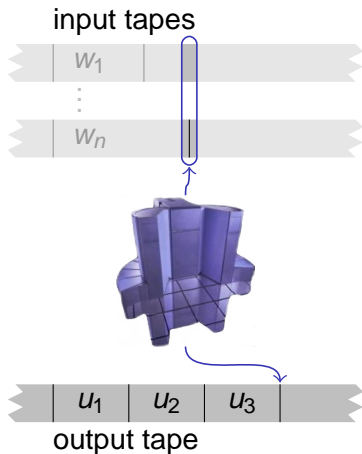


## Definition (1987, 1989, 1994, 2001)

An ***m*-enumerator** for a function  $f$

- 1 reads  $n$  input words  $w_1, \dots, w_n$ ,
- 2 does a computation,
- 3 **outputs at most  $m$  values,**
- 4 one of which is  $f(w_1, \dots, w_n)$ .

# Enumerators Output Sets of Possible Function Values

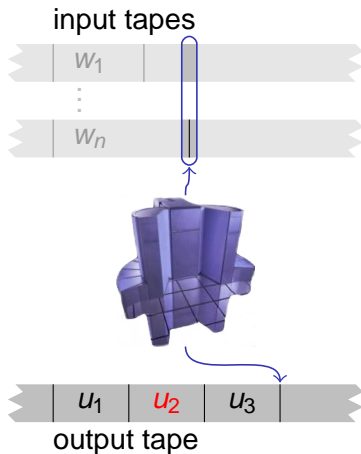


## Definition (1987, 1989, 1994, 2001)

An *m*-enumerator for a function  $f$

- 1 reads  $n$  input words  $w_1, \dots, w_n$ ,
- 2 does a computation,
- 3 outputs at most  $m$  values,
- 4 one of which is  $f(w_1, \dots, w_n)$ .

# Enumerators Output Sets of Possible Function Values



## Definition (1987, 1989, 1994, 2001)

An ***m*-enumerator** for a function  $f$

- 1 reads  $n$  input words  $w_1, \dots, w_n$ ,
- 2 does a computation,
- 3 outputs at most  $m$  values,
- 4 **one of which is  $f(w_1, \dots, w_n)$ .**

# How Well Can the Cardinality Function Be Enumerated?

## Observation

For fixed  $n$ , the cardinality function  $\#_A^n$

- can be **1**-enumerated by Turing machines only for **recursive**  $A$ , but
- can be  **$(n + 1)$** -enumerated for **every** language  $A$ .

## Question

What about 2-, 3-, 4-, ...,  $n$ -enumerability?



# How Well Can the Cardinality Function Be Enumerated?

## Observation

For fixed  $n$ , the cardinality function  $\#_A^n$

- can be 1-enumerated by Turing machines only for **recursive**  $A$ , but
- can be  $(n + 1)$ -enumerated for **every** language  $A$ .

## Question

What about 2-, 3-, 4-, ...,  $n$ -enumerability?

# How Well Can the Cardinality Function Be Enumerated by Turing Machines?

## Cardinality Theorem (Kummer, 1992)

*If  $\#_A^n$  is  $n$ -enumerable by a Turing machine, then  $A$  is recursive.*

## Weak Cardinality Theorems ( )

- *If  $\#_A^1$  is 1-enumerable by a Turing machine, then  $A$  is recursive.*
- *If  $\#_A^2$  is 2-enumerable by a Turing machine, then  $A$  is recursive.*

# How Well Can the Cardinality Function Be Enumerated by Turing Machines?

## Cardinality Theorem (Kummer, 1992)

*If  $\#_A^n$  is  $n$ -enumerable by a Turing machine, then  $A$  is recursive.*

## Weak Cardinality Theorems (1987, 1989, 1992)

- 1 *If  $\chi_A^n$  is  $n$ -enumerable by a Turing machine, then  $A$  is recursive.*
- 2 *If  $\#_A^2$  is 2-enumerable by a Turing machine, then  $A$  is recursive.*
- 3 *If  $\#_A^n$  is  $n$ -enumerable by a Turing machine that never enumerates both 0 and  $n$ , then  $A$  is recursive.*

# How Well Can the Cardinality Function Be Enumerated by Turing Machines?

## Cardinality Theorem (Kummer, 1992)

*If  $\#_A^n$  is  $n$ -enumerable by a Turing machine, then  $A$  is recursive.*

## Weak Cardinality Theorems (1987, 1989, 1992)

- 1 *If  $\chi_A^n$  is  $n$ -enumerable by a Turing machine, then  $A$  is recursive.*
- 2 *If  $\#_A^2$  is 2-enumerable by a Turing machine, then  $A$  is recursive.*
- 3 *If  $\#_A^n$  is  $n$ -enumerable by a Turing machine that never enumerates both 0 and  $n$ , then  $A$  is recursive.*

# How Well Can the Cardinality Function Be Enumerated by Turing Machines?

## Cardinality Theorem (Kummer, 1992)

*If  $\#_A^n$  is  $n$ -enumerable by a Turing machine, then  $A$  is recursive.*

## Weak Cardinality Theorems (1987, 1989, 1992)

- 1 *If  $\chi_A^n$  is  $n$ -enumerable by a Turing machine, then  $A$  is recursive.*
- 2 *If  $\#_A^2$  is 2-enumerable by a Turing machine, then  $A$  is recursive.*
- 3 *If  $\#_A^n$  is  $n$ -enumerable by a Turing machine that never enumerates both 0 and  $n$ , then  $A$  is recursive.*

# How Well Can the Cardinality Function Be Enumerated by Finite Automata?

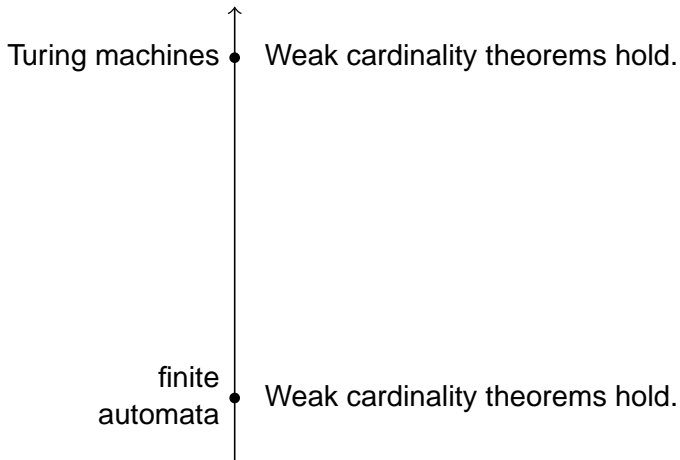
## Conjecture

If  $\#_A^n$  is  $n$ -enumerable by a **finite automaton**, then  $A$  is **regular**.

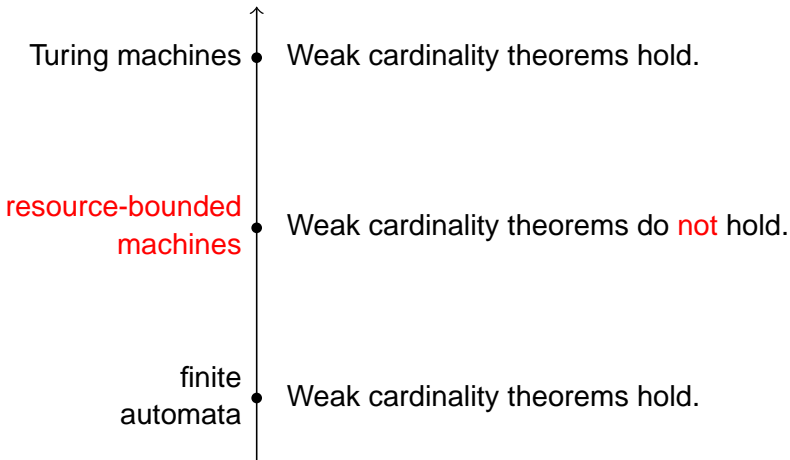
## Weak Cardinality Theorems (2001, 2002)

- 1 If  $\chi_A^n$  is  $n$ -enumerable by a **finite automaton**, then  $A$  is **regular**.
- 2 If  $\#_A^2$  is 2-enumerable by a **finite automaton**, then  $A$  is **regular**.
- 3 If  $\#_A^n$  is  $n$ -enumerable by a **finite automaton** that never enumerates both 0 and  $n$ , then  $A$  is **regular**.

# Cardinality Theorems Do Not Hold for All Models



# Cardinality Theorems Do Not Hold for All Models





# Why?

## First Explanation

The weak cardinality theorems hold both for recursion and automata theory **by coincidence**.

## Second Explanation

The weak cardinality theorems hold both for recursion and automata theory, **because they are instantiations of single, unifying theorems**.

# Why?

## First Explanation

The weak cardinality theorems hold both for recursion and automata theory **by coincidence**.

## Second Explanation

The weak cardinality theorems hold both for recursion and automata theory, **because they are instantiations of single, unifying theorems**.

The second explanation is correct.

The theorems can (almost) be unified using first-order logic.

# Outline

- 1 History
  - Enumerability in Recursion and Automata Theory
  - Known Weak Cardinality Theorem
  - Why Do Cardinality Theorems Hold Only for Certain Models?
- 2 **Unification by First-Order Logic**
  - Elementary Definitions
  - Enumerability for First-Order Logic
  - Weak Cardinality Theorems for First-Order Logic
- 3 Applications
  - A Separability Result for First-Order Logic

# What Are Elementary Definitions?

## Definition

A relation  $R$  is **elementarily definable in a logical structure  $\mathcal{S}$**  if

- 1 there exists a first-order formula  $\phi$ ,
- 2 that is true exactly for the elements of  $R$ .

## Example

The set of even numbers is elementarily definable in  $(\mathbb{N}, +)$  via the formula  $\phi(x) \equiv \exists z . z + z = x$ .

## Example

The set of powers of 2 is not elementarily definable in  $(\mathbb{N}, +)$ .

# Characterisation of Classes by Elementary Definitions

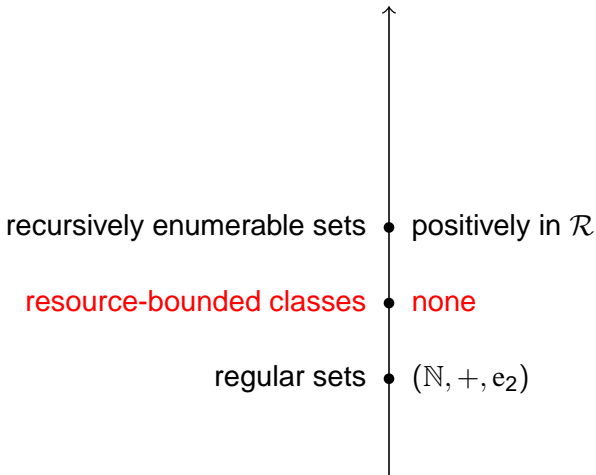
## Theorem (Büchi, 1960)

*There exists a logical structure  $(\mathbb{N}, +, e_2)$  such that a set  $A \subseteq \mathbb{N}$  is **regular** iff it is **elementarily definable** in  $(\mathbb{N}, +, e_2)$ .*

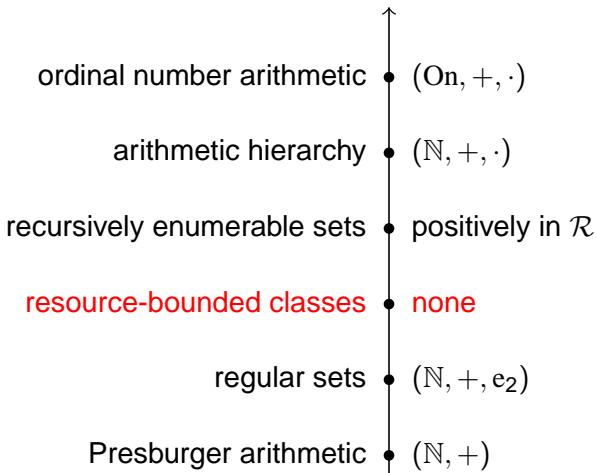
## Theorem

*There exists a logical structure  $\mathcal{R}$  such that a set  $A \subseteq \mathbb{N}$  is **recursively enumerable** iff it is **positively elementarily definable** in  $\mathcal{R}$ .*

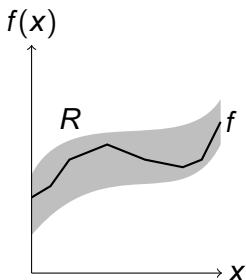
# Characterisation of Classes by Elementary Definitions



# Characterisation of Classes by Elementary Definitions



# Elementary Enumerability is a Generalisation of Elementary Definability



## Definition

A function  $f$  is

**elementarily  $m$ -enumerable** in a structure  $\mathcal{S}$  if

- 1 its graph is contained in an **elementarily definable** relation  $R$ ,
- 2 which is  **$m$ -bounded**, i.e., for each  $x$  there are at most  $m$  different  $y$  with  $(x, y) \in R$ .



# The Original Notions of Enumerability are Instantiations

## Theorem

A function is  $m$ -enumerable by a *finite automaton* iff it is elementarily  $m$ -enumerable in  $(\mathbb{N}, +, e_2)$ .

## Theorem

A function is  $m$ -enumerable by a *Turing machine* iff it is positively elementarily  $m$ -enumerable in  $\mathcal{R}$ .

# The First Weak Cardinality Theorem

## Theorem

Let  $S$  be a logical structure with universe  $U$  and let  $A \subseteq U$ . If

- 1  $S$  is well-orderable and
- 2  $\chi_A^n$  is elementarily  $n$ -enumerable in  $S$ ,

then  $A$  is elementarily definable in  $S$ .

# The First Weak Cardinality Theorem

## Theorem

Let  $S$  be a logical structure with universe  $U$  and let  $A \subseteq U$ . If

- 1  $S$  is well-orderable and
- 2  $\chi_A^n$  is elementarily  $n$ -enumerable in  $S$ ,

then  $A$  is elementarily definable in  $S$ .

## Corollary

If  $\chi_A^n$  is  $n$ -enumerable by a finite automaton, then  $A$  is regular.

# The First Weak Cardinality Theorem

## Theorem

Let  $S$  be a logical structure with universe  $U$  and let  $A \subseteq U$ . If

- 1  $S$  is well-orderable and
- 2  $\chi_A^n$  is elementarily  $n$ -enumerable in  $S$ ,

then  $A$  is elementarily definable in  $S$ .

## Corollary (with more effort)

If  $\chi_A^n$  is  $n$ -enumerable by a Turing machine, then  $A$  is recursive.

# The Second Weak Cardinality Theorem

## Theorem

Let  $S$  be a logical structure with universe  $U$  and let  $A \subseteq U$ . If

- 1  $S$  is well-orderable,
- 2 every finite relation on  $U$  is elementarily definable in  $S$ , and
- 3  $\#_A^2$  is elementarily 2-enumerable in  $S$ ,

then  $A$  is elementarily definable in  $S$ .

# The Third Weak Cardinality Theorem

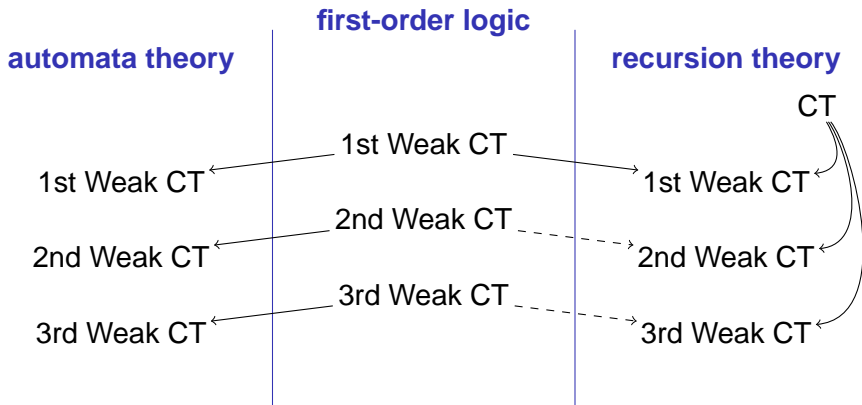
## Theorem

Let  $S$  be a logical structure with universe  $U$  and let  $A \subseteq U$ . If

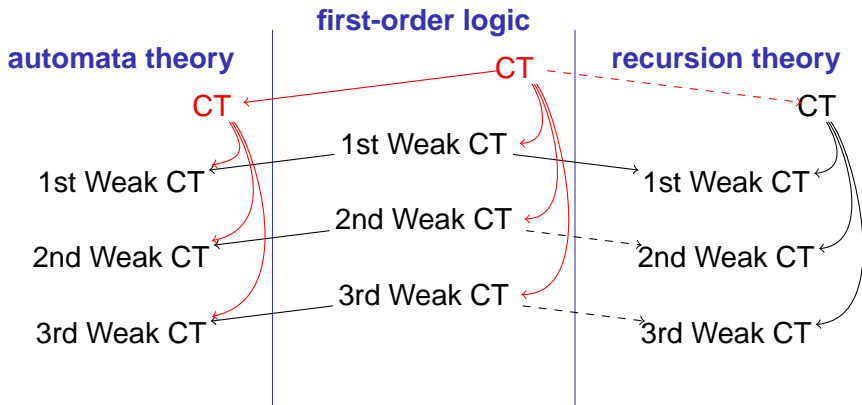
- 1  $S$  is well-orderable,
- 2 every finite relation on  $U$  is elementarily definable in  $S$ , and
- 3  $\#_A^n$  is elementarily  $n$ -enumerable in  $S$  via a relation that never 'enumerates' both 0 and  $n$ ,

then  $A$  is elementarily definable in  $S$ .

# Relationships Between Cardinality Theorems (CT)



# Relationships Between Cardinality Theorems (CT)

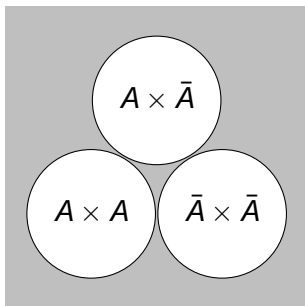




# Outline

- 1 History
  - Enumerability in Recursion and Automata Theory
  - Known Weak Cardinality Theorem
  - Why Do Cardinality Theorems Hold Only for Certain Models?
- 2 Unification by First-Order Logic
  - Elementary Definitions
  - Enumerability for First-Order Logic
  - Weak Cardinality Theorems for First-Order Logic
- 3 **Applications**
  - A Separability Result for First-Order Logic

## A Separability Result for First-Order Logic



### Theorem

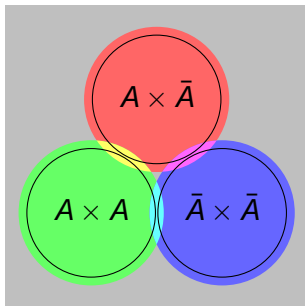
*Let  $S$  be a well-orderable logical structure in which all finite relations are elementarily definable.*

*If there exist elementarily definable supersets of  $A \times A$ ,  $A \times \bar{A}$ , and  $\bar{A} \times \bar{A}$  whose intersection is empty, then  $A$  is elementarily definable in  $S$ .*

### Note

The theorem is no longer true if we add  $\bar{A} \times A$  to the list.

## A Separability Result for First-Order Logic



## Theorem

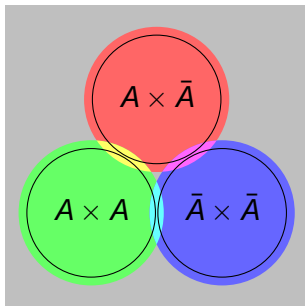
*Let  $S$  be a well-orderable logical structure in which all finite relations are elementarily definable.*

*If there exist elementarily definable supersets of  $A \times A$ ,  $A \times \bar{A}$ , and  $\bar{A} \times \bar{A}$  whose intersection is empty, then  $A$  is elementarily definable in  $S$ .*

## Note

The theorem is no longer true if we add  $\bar{A} \times A$  to the list.

## A Separability Result for First-Order Logic

**Theorem**

*Let  $S$  be a well-orderable logical structure in which all finite relations are elementarily definable.*

*If there exist elementarily definable supersets of  $A \times A$ ,  $A \times \bar{A}$ , and  $\bar{A} \times \bar{A}$  whose intersection is empty, then  $A$  is elementarily definable in  $S$ .*

**Note**

The theorem is no longer true if we add  $\bar{A} \times A$  to the list.

# Summary

## Summary

- The weak cardinality theorems for first-order logic **unify** the weak cardinality theorems of automata and recursion theory.
- The logical approach yields weak cardinality theorems for **other computational models**.
- Cardinality theorems are **separability theorems** in disguise.

## Open Problems

- Does a cardinality theorem for first-order logic hold?
- What about non-well-orderable structures like  $(\mathbb{R}, +, \cdot)$ ?