Generic reducibilities

Alexander Rybalov

Sobolev Institute of Mathematics, Omsk

September, 2016
Definition

A \subseteq I \text{ reduces to } B \subseteq J, \text{ if there is a computable function } f : I \rightarrow J \text{ such that } x \in A \iff f(x) \in B \text{ for all } x \in I.
Properties of the classical reducibility

**Denotation**

\( A \subseteq I \) reduces to \( B \subseteq J \) we denote \( A \leq B \).
### Denotation

$A \subseteq I$ reduces to $B \subseteq J$ we denote $A \leq B$.

### Properties

For all sets $A, B, C \subseteq I$
Denotation

\( A \subseteq I \) reduces to \( B \subseteq J \) we denote \( A \leq B \).

Properties

For all sets \( A, B, C \subseteq I \)

1. \( A \leq A \) – reflexivity,
### Denotation

$A \subseteq I$ reduces to $B \subseteq J$ we denote $A \leq B$.

### Properties

For all sets $A, B, C \subseteq I$

1. $A \leq A$ – reflexivity,
2. $A \leq B$ and $B \leq C$ implies $A \leq C$ – transitivity,
Properties of the classical reducibility

Denotation

A ⊆ I reduces to B ⊆ J we denote A ≤ B.

Properties

For all sets A, B, C ⊆ I

1. A ≤ A – reflexivity,
2. A ≤ B and B ≤ C implies A ≤ C – transitivity,
3. if A ≤ B and B is decidable, then A is decidable – preserving of decidability.
Set $A \subseteq I$ generically reduces to $B \subseteq J$ ($A \leq_{\text{Gen}} B$) if there is a computable function $f : I \times \mathbb{N} \to J \cup \{?\}$ such that

1. $\forall x \in I$ if $f(x, 0) \neq ?$, then $\forall n \in \mathbb{N} f(x, n) \neq ?$.
2. $\rho(\{x \in I : f(x, 0) \neq ?\}) = 1$.
3. $\forall x \in I$ if $f(x, 0) \neq ?$, then $\rho(\{f(x, n) : n \in \mathbb{N}\}) > 0$.
4. $\forall x \in I$ if $f(x, 0) \neq ?$, then $x \in A \Rightarrow \forall n \in \mathbb{N} f(x, n) \in B$.
   $x \notin A \Rightarrow \forall n \in \mathbb{N} f(x, n) \notin B$.
**Denotation**

Set $A \subseteq I$ generically reduces to $B \subseteq J$ ($A \leq_{Gen} B$) if there is a computable function $f : I \times \mathbb{N} \rightarrow J \cup \{?\}$ such that

1. \( \forall x \in I \text{ if } f(x, 0) \neq ?, \text{ then } \forall n \in \mathbb{N} \ f(x, n) \neq ? \).
Generic reducibility

Denotation
Set $A \subseteq I$ generically reduces to $B \subseteq J$ ($A \leq_{Gen} B$) if there is a computable function $f : I \times \mathbb{N} \to J \cup \{?\}$ such that

1. $\forall x \in I$ if $f(x, 0) \neq ?$, then $\forall n \in \mathbb{N}$ $f(x, n) \neq ?$.
2. $\rho\left(\{x \in I : f(x, 0) \neq ?\}\right) = 1$. 
Generic reducibility

Denotation

Set $A \subseteq I$ generically reduces to $B \subseteq J$ ($A \leq_{\text{Gen}} B$) if there is a computable function $f : I \times \mathbb{N} \to J \cup \{?\}$ such that

1. $\forall x \in I$ if $f(x, 0) \neq?$, then $\forall n \in \mathbb{N}$, $f(x, n) \neq?$. 
2. $\rho\left(\{x \in I : f(x, 0) \neq?\}\right) = 1$. 
3. $\forall x \in I$ if $f(x, 0) \neq?$, then $\rho\left(\{f(x, n) : n \in \mathbb{N}\}\right) > 0$. 

Alexander Rybalov

Generic reducibilities
### Denotation

Set $A \subseteq I$ generically reduces to $B \subseteq J$ ($A \leq_{\text{Gen}} B$) if there is a computable function $f : I \times \mathbb{N} \rightarrow J \cup \{?\}$ such that

1. $\forall x \in I$ if $f(x, 0) \neq ?$, then $\forall n \in \mathbb{N} \ f(x, n) \neq ?$.
2. $\rho\left(\{x \in I : f(x, 0) \neq ?\}\right) = 1$.
3. $\forall x \in I$ if $f(x, 0) \neq ?$, then $\rho\left(\{f(x, n) : n \in \mathbb{N}\}\right) > 0$.
4. $\forall x \in I$ if $f(x, 0) \neq ?$, then
   - $x \in A \Rightarrow \forall n \in \mathbb{N} \ f(x, n) \in B$.
   - $x \notin A \Rightarrow \forall n \in \mathbb{N} \ f(x, n) \notin B$. 
Generic reducibility
Preserving of generic decidability

**Theorem**

If $A \leq_{Gen} B$ and $B$ is generically decidable, then $A$ is generically decidable.
Preserving of generic decidability

**Theorem**

If $A \leq_{Gen} B$ and $B$ is generically decidable, then $A$ is generically decidable.

Generic algorithm, deciding $x \in A$?
Preserving of generic decidability

Theorem
If $A \leq_{Gen} B$ and $B$ is generically decidable, then $A$ is generically decidable.

Generic algorithm, deciding $x \in A$?

- Computes $f(x, 0)$. If $f(x, 0) = ?$ output ?
Preserving of generic decidability

**Theorem**

If $A \leq_{Gen} B$ and $B$ is generically decidable, then $A$ is generically decidable.

Generic algorithm, deciding $x \in A$?

- Computes $f(x, 0)$. If $f(x, 0) = ?$ output ?
- If $f(x, 0) \neq ?$, computes $f(x, 1), f(x, 2), \ldots$ until generic algorithm for $B$ outputs on some $f(x, k)$ defined answer.
Preserving of generic decidability

**Theorem**

If $A \leq_{Gen} B$ and $B$ is generically decidable, then $A$ is generically decidable.

Generic algorithm, deciding $x \in A$?

- Computes $f(x, 0)$. If $f(x, 0) = ?$ output ?
- If $f(x, 0) \neq ?$, computes $f(x, 1), f(x, 2), \ldots$ until generic algorithm for $B$ outputs on some $f(x, k)$ defined answer.
- It will be correct answer for $A$. 

Alexander Rybalov

Generic reducibilities
Transitivity

**Theorem**

If $A \leq_{Gen} B$ and $B \leq_{Gen} C$, then $A \leq_{Gen} C$. 

Alexander Rybalov

Generic reducibilities
### Theorem

If $A \leq_{Gen} B$ and $B \leq_{Gen} C$, then $A \leq_{Gen} C$.

$A \leq_{Gen} B$ by $f$, $B \leq_{Gen} C$ by $g$. How to construct a generic reduction $h$ of $A$ to $C$?
Transitivity

Theorem

If $A \leq_{\text{Gen}} B$ and $B \leq_{\text{Gen}} C$, then $A \leq_{\text{Gen}} C$.

$A \leq_{\text{Gen}} B$ by $f$, $B \leq_{\text{Gen}} C$ by $g$. How to construct a generic reduction $h$ of $A$ to $C$?

- Compute $f(x, 0)$. If $f(x, 0) = \text{?}$, output $h(x, n) = \text{?}$ for all $n$.  
  
Alexander Rybalov

Generic reducibilities
Transitivity

Theorem

If $A \leq_{\text{Gen}} B$ and $B \leq_{\text{Gen}} C$, then $A \leq_{\text{Gen}} C$.

$A \leq_{\text{Gen}} B$ by $f$, $B \leq_{\text{Gen}} C$ by $g$. How to construct a generic reduction $h$ of $A$ to $C$?

- Compute $f(x, 0)$. If $f(x, 0) = ?$, output $h(x, n) = ?$ for all $n$.
- If $f(x, 0) \neq ?$ computes $f(x, 1), f(x, 2), \ldots$ until $g(f(x, k), 0) \neq ?$. 
Transitivity

Theorem

If $A \leq_{\text{Gen}} B$ and $B \leq_{\text{Gen}} C$, then $A \leq_{\text{Gen}} C$.

$A \leq_{\text{Gen}} B$ by $f$, $B \leq_{\text{Gen}} C$ by $g$. How to construct a generic reduction $h$ of $A$ to $C$?

- Compute $f(x, 0)$. If $f(x, 0) =$?, output $h(x, n) =$? for all $n$.
- If $f(x, 0) \neq$? computes $f(x, 1)$, $f(x, 2)$, ... until $g(f(x, k), 0) \neq$?.
- Now $h(x, n) = g(f(x, k), n)$ for all $n$. 

Alexander Rybalov
Generic reductions
Theorem

There is a computably enumerable (c.e.) set $W \subseteq \mathbb{N}$ such that $W \not\leq_{\text{Gen}} W$. 
Theorem

There is a computably enumerable (c.e.) set $W \subseteq \mathbb{N}$ such that $W \not\leq_{\text{Gen}} W$.

Definition

Set $A \subseteq \mathbb{N}$ is \textit{immune} if it is infinite but not containing infinite c.e. subsets.
Theorem

There is a computably enumerable (c.e.) set $W \subseteq \mathbb{N}$ such that $W \not\leq_{Gen} W$.

Definition

Set $A \subseteq \mathbb{N}$ is \textit{immune} if it is infinite but not containing infinite c.e. subsets.

Definition

Set $B \subseteq \mathbb{N}$ is \textit{simple} if $B$ is c.e. and $\mathbb{N} \setminus B$ is immune.
Theorem

If $W \subseteq \mathbb{N}$ is simple and non-generic, then $W \not\leq_{\text{Gen}} W$. 
Theorem

If \( W \subseteq \mathbb{N} \) is simple and non-generic, then \( W \not\leq_{\text{Gen}} W \).

- Suppose \( W \not\leq_{\text{Gen}} W \) by some generic reduction \( f \).
Theorem

If $W \subseteq \mathbb{N}$ is simple and non-generic, then $W \not\leq_{Gen} W$.

- Suppose $W \not\leq_{Gen} W$ by some generic reduction $f$.
- $\mathbb{N} \setminus W$ is not negligible, so there is $x \notin W$ such that $f(x, 0) \neq$? and for all $f(x, k) \notin W$. 

Alexander Rybalov

Generic reducibilities
Theorem

If $W \subseteq \mathbb{N}$ is simple and non-generic, then $W \not\leq_{Gen} W$.

- Suppose $W \not\leq_{Gen} W$ by some generic reduction $f$.
- $\mathbb{N} \setminus W$ is not negligible, so there is $x \notin W$ such that $f(x, 0) \neq ?$ and for all $f(x, k) \notin W$.
- $\{f(x, 0), f(x, 1), \ldots, \}$ is infinite c.e. subset of immune set $\mathbb{N} \setminus W$. 

Alexander Rybalov

Generic reducibilities
How to construct non-generic simple set?

- $S_1, S_2, \ldots, S_k, \ldots$ – effective numeration of all c.e. sets.
How to construct non-generic simple set?

- $S_1, S_2, \ldots, S_k, \ldots$ – effective numeration of all c.e. sets.
- Step 0: $W = \emptyset$. 

$W \cap S_i \neq \emptyset$ for every infinite c.e. $S_i$.

For every $n$ there is only one element of size $n$ in $W$, so $\rho(W) = \lim_{n \to \infty} \frac{|N_n \cap W|}{|N_n|} = \lim_{n \to \infty} \frac{1}{2^{n}} = 0$. 

Alexander Rybalov

Generic reducibilities
How to construct non-generic simple set?

- $S_1, S_2, \ldots, S_k, \ldots$ – effective numeration of all c.e. sets.
- Step 0: $W = \emptyset$.
- Step $i$: add to $W$ an element $a \in S_i$ such that $|a|$ is not equal to sizes of elements of $W$.
How to construct non-generic simple set?

- $S_1, S_2, \ldots, S_k, \ldots$ – effective numeration of all c.e. sets.
- Step 0: $W = \emptyset$.
- Step $i$: add to $W$ an element $a \in S_i$ such that $|a|$ is not equal to sizes of elements of $W$.
- $W \cap S_i \neq \emptyset$ for every infinite c.e. $S_i$.
How to construct non-generic simple set?

- $S_1, S_2, \ldots, S_k, \ldots$ – effective numeration of all c.e. sets.
- Step 0: $W = \emptyset$.
- Step $i$: add to $W$ an element $a \in S_i$ such that $|a|$ is not equal to sizes of elements of $W$.
- $W \cap S_i \neq \emptyset$ for every infinite c.e. $S_i$.
- For every $n$ there is only one element of size $n$ in $W$, so

$$\rho(W) = \lim_{n \to \infty} \frac{|\mathbb{N}_n \cap W|}{|\mathbb{N}_n|} = \lim_{n \to \infty} \frac{1}{2^n} = 0.$$
Super undecidability and cloning

Question
Can every super undecidable set be constructed by cloning?

Answer
No!

Theorem
\( W \) is super undecidable. There is no effective non-negligible cloning \( C : \mathbb{N} \rightarrow \mathbb{P}(\mathbb{N}) \) such that \( W = C(S) \) for some c.e. set \( S \subseteq \mathbb{N} \).
Question
Can every super undecidable set be constructed by cloning?

Answer
No!
Question
Can every super undecidable set be constructed by cloning?

Answer
No!

Theorem
- $W$ is super undecidable.
Question
Can every super undecidable set be constructed by cloning?

Answer
No!

Theorem
- \( W \) is super undecidable.
- There is no effective non-negligible cloning \( C : \mathbb{N} \rightarrow P(\mathbb{N}) \) such that \( W = C(S) \) for some c.e. set \( S \subseteq \mathbb{N} \).
Completeness

**Theorem**

Set $K_0$ of all Turing machines halting on 0 is generically complete in the class of all c.e. sets. For every c.e. set $A$ $A \leq_{Gen} K_0$. 

How to construct generic reduction $f$ of c.e. set $A$ to $K_0$?

There is a classical reduction $h$ of $A$ to $K_0$.

Fix $x \in \mathbb{N}$. Compute $h(x)$ some Turing machine. Enumerate effectively machines from the clone of $h(x)$: $M_0, M_1, M_2, \ldots$.

Now define $f(x, k) = M_k$. 

Alexander Rybalov

Generic reducibilities
Completeness

**Theorem**

Set $K_0$ of all Turing machines halting on 0 is generically complete in the class of all c.e. sets. For every c.e. set $A$, $A \leq_{Gen} K_0$.

How to construct generic reduction $f$ of c.e. set $A$ to $K_0$?
Completeness

Theorem

Set \( K_0 \) of all Turing machines halting on 0 is generically complete in the class of all c.e. sets. For every c.e. set \( A \), \( \leq_{Gen} K_0 \).

How to construct generic reduction \( f \) of c.e. set \( A \) to \( K_0 \)?

- There is a classical reduction \( h \) of \( A \) to \( K_0 \).
Completeness

**Theorem**

Set $K_0$ of all Turing machines halting on 0 is generically complete in the class of all c.e. sets. For every c.e. set $A$, $A \leq_{Gen} K_0$.

How to construct generic reduction $f$ of c.e. set $A$ to $K_0$?

- There is a classical reduction $h$ of $A$ to $K_0$.
- Fix $x \in \mathbb{N}$. Compute $h(x)$ – some Turing machine.
Completeness

**Theorem**

Set $K_0$ of all Turing machines halting on 0 is generically complete in the class of all c.e. sets. For every c.e. set $A$, $A \leq_{\text{Gen}} K_0$.

How to construct generic reduction $f$ of c.e. set $A$ to $K_0$?

- There is a classical reduction $h$ of $A$ to $K_0$.
- Fix $x \in \mathbb{N}$. Compute $h(x)$ – some Turing machine.
- Enumerate effectively machines from the clone of $h(x)$: $M_0, M_1, M_2, \ldots$. 
Completeness

Theorem

Set $K_0$ of all Turing machines halting on 0 is generically complete in the class of all c.e. sets. For every c.e. set $A$, $A \leq_{\text{Gen}} K_0$.

How to construct generic reduction $f$ of c.e. set $A$ to $K_0$?

- There is a classical reduction $h$ of $A$ to $K_0$.
- Fix $x \in \mathbb{N}$. Compute $h(x)$ – some Turing machine.
- Enumerate effectively machines from the clone of $h(x)$: $M_0, M_1, M_2, \ldots$.
- Now define $f(x, k) = M_k$. 

Alexander Rybalov

Generic reducibilities