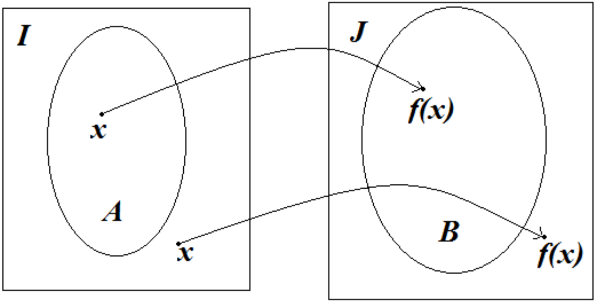
Generic reducibilities

Alexander Rybalov

Sobolev Institute of Mathematics, Omsk

September, 2016

Classical reducibility



Definition

$A \subseteq I$ reduces to $B \subseteq J$, if there is a computable function $f : I \to J$ such that $x \in A \Leftrightarrow f(x) \in B$ for all $x \in I$.

Properties of the classical reducibility

Denotation

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

Properties of the classical reducibility

Denotation

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

Properties

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

Properties of the classical reducibility

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,

Properties of the classical reducibility

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

$A \subseteq I$ 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,
3. if $A \leq B$ and B is decidable, then A is decidable – preserving of decidability.

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

Generic reducibility

Denotation

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

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

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. $\rho\Big(\{x \in I : f(x,0) \neq ?\}\Big) = 1.$

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

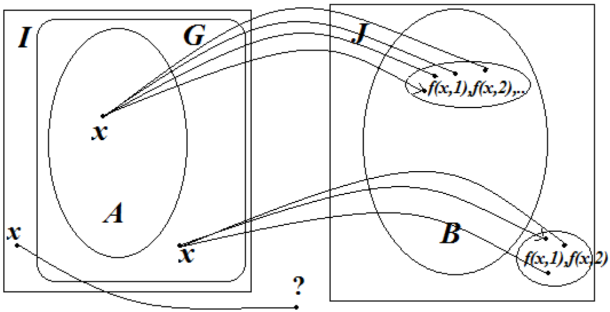
1. $\forall x \in I$ if $f(x, 0) \neq ?$, then $\forall n \in \mathbb{N}$ $f(x, n) \neq ?$.
2. $\rho\Big(\{x \in I : f(x, 0) \neq ?\}\Big) = 1$.
3. $\forall x \in I$ if $f(x, 0) \neq ?$, then $\rho\Big(\{f(x, n) : n \in \mathbb{N}\}\Big) > 0$.

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

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.

Transitivity

Theorem

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

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 $A \leq_{Gen} B$ 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?

- Compute $f(x, 0)$. If $f(x, 0) = ?$, output $h(x, n) = ?$ for all n.

Transitivity

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?

- 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 $A \leq_{Gen} B$ 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?

- 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 $g(f(x, k), 0) \neq ?$.
- Now $h(x, n) = g(f(x, k), n)$ for all n.

No reflexivity :(

Theorem

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

No reflexivity :(

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 *immune* if it is infinite but not containing infinite c.e. subsets.

No reflexivity :(

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 *immune* if it is infinite but not containing infinite c.e. subsets.

Definition

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

No reflexivity :(

Theorem

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

No reflexivity :(

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.

No reflexivity :(

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 $f(x, k) \notin W$.

No reflexivity :(

Theorem

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

- Suppose $W \leq_{Gen} W$ 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 $f(x,k) \notin W$.
- $\{f(x,0), f(x,1), \dots, \}$ is infinite c.e. subset of immune set $\mathbb{N} \setminus W$.

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?

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

How to construct non-generic simple set?

- $S_1, S_2, \ldots, S_k, \ldots$ – effective numeration of all c.e. sets.
- Step 0: $W = \varnothing$.
- 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: $W = \varnothing$.
- Step i: add to W an element $a \in S_i$ such that $|a|$ is not equal to sizes of elements of W.
- $W \cap S_i \neq \varnothing$ 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: $W = \varnothing$.
- Step i: add to W an element $a \in S_i$ such that $|a|$ is not equal to sizes of elements of W.
- $W \cap S_i \neq \varnothing$ 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?

Super undecidability and cloning

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 $C : \mathbb{N} \to P(\mathbb{N})$ such $W = C(S)$ 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 $A \leq_{Gen} K_0$.

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 $A \leq_{Gen} K_0$.

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 $A \leq_{Gen} K_0$.

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 $A \leq_{Gen} K_0$.

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 $f(x, k) = M_k$.