Amplification of algorithmic problems: undecidable problems

Alexander Rybalov

Sobolev Institute of Mathematics, Omsk

September, 2016
Algorithm works correctly on all inputs.
Algorithm works correctly on **almost all** inputs but can ignore some.
**Definition**

Let $I$ be all inputs, $I_n$ — all inputs of size $n$. **Asymptotic density** of set $S \subseteq I$

$$
\mu(S) = \lim_{n \to \infty} \frac{|S \cap I_n|}{|I_n|}.
$$
Asymptotic density

**Definition**

Let $I$ be all inputs, $I_n$ — all inputs of size $n$. **Asymptotic density** of set $S \subseteq I$

$$
\mu(S) = \lim_{n \to \infty} \frac{|S \cap I_n|}{|I_n|}.
$$

**Remark**

$$
\frac{|S \cap I_n|}{|I_n|}
$$

is the probability to get an input from $S$ during random and uniform generation of inputs of size $n$. 
Generic sets

Definition

Set $S \subseteq I$ is called

- **generic** if $\mu(S) = 1$,
Generic sets

Definition

Set $S \subseteq I$ is called
- **generic** if $\mu(S) = 1$,
- **negligible** if $\mu(S) = 0$. 

Alexander Rybalov

Amplification of algorithmic problems: undecidable problems
Definition
Set $S \subseteq I$ is called **generically decidable** if there exists a decidable generic set $G \subseteq I$ such that $S \cap G$ is decidable.
### Definition

Set $S \subseteq I$ is called **generically decidable** if there exists a decidable generic set $G \subseteq I$ such that $S \cap G$ is decidable.

### Definition

Set $S \subseteq I$ is called **super undecidable** if it is not generically decidable.
Definition

Let $I$ and $J$ be input sets. Function $C : I \rightarrow P(J)$ is called cloning $I$ to $J$ if

1. $\forall x, y \in I \; x \neq y \rightarrow C(x) \cap C(y) = \emptyset$

2. There is an algorithm $E : I \times \mathbb{N} \rightarrow J$ such that for all $x \in I$

$$C(x) = \{ E(x, 0), E(x, 1), \ldots, \}$$
Definition

Let $I$ and $J$ be input sets. Function $C : I \rightarrow P(J)$ is called cloning $I$ to $J$ if

1. $\forall x, y \in I \ x \neq y \rightarrow C(x) \cap C(y) = \emptyset$

2. There is an algorithm $E : I \times \mathbb{N} \rightarrow J$ such that for all $x \in I$

$$C(x) = \{E(x, 0), E(x, 1), \ldots, \}$$

Definition

Cloning of $S \subseteq I$ is $C(S) = \bigcup_{x \in S} C(x)$.
Generic amplification

Definition
Let $I$ and $J$ be input sets. Function $C : I \rightarrow P(J)$ is called cloning $I$ to $J$ if
1. $\forall x, y \in I \ x \neq y \rightarrow C(x) \cap C(y) = \emptyset$
2. There is an algorithm $E : I \times \mathbb{N} \rightarrow J$ such that for all $x \in I$
   $$C(x) = \{E(x, 0), E(x, 1), \ldots, \}$$

Definition
Cloning of $S \subseteq I$ is $C(S) = \bigcup_{x \in S} C(x)$.

Definition
Cloning $C$ is non-negligible if $\mu(C(x)) > 0$ for all $x \in I$.
Generic amplification

Alexander Rybalov

Amplification of algorithmic problems: undecidable problems
How to amplify undecidability?

**Theorem (Miasnikov, Rybalov, 2008)**

Let $C : I \rightarrow P(J)$ be a non-negligible cloning and $S \subseteq I$ is undecidable. Then $C(S)$ is super undecidable.
Idea of proof

Alexander Rybalov

Amplication of algorithmic problems: undecidable problems
Idea of proof

Alexander Rybalov

Amplication of algorithmic problems: undecidable problems
Example 1: the Halting Problem

**Halting Problem**

- **Input:** A Turing machine $M$ over alphabet $\{0, 1\}$.
- **Output:**
  
  1, if $M$ halts on 0,
  
  0, if $M$ does not halt on 0.

**Theorem (Turing)**

The Halting Problem is algorithmically undecidable.
Representation of Turing machines

States

Non-final states $q_1, \ldots, q_n$. Final state $q_0$. Size of Turing machine is the number of non-final states.

Program

$$(q_1, 0) \rightarrow (q_{j_1}, t_1, D_1),$$
$$(q_1, 1) \rightarrow (q_{j_2}, t_2, D_2),$$
$$\ldots$$
$$(q_n, 1) \rightarrow (q_{j_{2n}}, t_{2n}, D_{2n})$$

$t_i \in \{0, 1\}, \ D_i \in \{R, L\} – shift.$
Normalized Turing machines

Definition

Turing machine $M$ is normalized, if for every state $q_i$ corresponding commands have the form

$$(q_i, 0) \rightarrow (q_j, t_1, D_1),$$
$$(q_i, 1) \rightarrow (q_k, t_2, D_2),$$

where $j, k \leq 2i + 1$.

Theorem

For every Turing machine there exists normalized Turing machine, computing the same function.
Theorem (Hamkins, Miasnikov)

The Halting Problem for Turing machines with one-way tape is generically decidable.
**Theorem (Hamkins, Miasnikov)**

The Halting Problem for Turing machines with one-way tape is generically decidable.

---

**Theorem (Rybalov)**

The Halting Problem for normalized Turing machines is super undecidable.
Cloning of Turing machines

Cloning

Cloning $C$ adds new states $q_{n+1}, \ldots, q_{n+m}$ to machine $M$ with program

\[(q_1, 0) \rightarrow (q_{j_1}, t_1, D_1), \]
\[(q_1, 1) \rightarrow (q_{j_2}, t_2, D_2), \]
\[\ldots \]
\[(q_n, 1) \rightarrow (q_{j_{2n}}, t_{2n}, D_{2n}) \]

and new arbitrary normalized commands for new states:

\[(q_{n+1}, 0) \rightarrow (q_{k_1}, u_1, E_1), \]
\[(q_{n+1}, 1) \rightarrow (q_{k_2}, u_2, E_2), \]
\[\ldots \]
\[(q_{n+m}, 1) \rightarrow (q_{k_{2m}}, u_{2m}, E_{2m}) \]

Theorem

For every machine $M$ the clone $C(M)$ is not negligible.
Cloning of Turing machines

**Cloning**

Cloning $C$ adds new states $q_{n+1}, \ldots, q_{n+m}$ to machine $M$ with program

\[(q_1, 0) \rightarrow (q_{j_1}, t_1, D_1),\]
\[(q_1, 1) \rightarrow (q_{j_2}, t_2, D_2),\]
\[\ldots\]
\[(q_n, 1) \rightarrow (q_{j_{2n}}, t_{2n}, D_{2n})\]

and new arbitrary normalized commands for new states:

\[(q_{n+1}, 0) \rightarrow (q_{k_1}, u_1, E_1),\]
\[(q_{n+1}, 1) \rightarrow (q_{k_2}, u_2, E_2),\]
\[\ldots\]
\[(q_{n+m}, 1) \rightarrow (q_{k_{2m}}, u_{2m}, E_{2m})\]

**Theorem**

For every machine $M$ the clone $C(M)$ is not negligible.
Example 2: First order theory of Arithmetic

Theory $Th(\mathbb{N})$

- Input: A first-order formula $\Phi$ of the language $\{+, -, \times, 0, 1\}$.
- Output:
  - 1, if $\Phi$ is true over $\mathbb{N}$,
  - 0, otherwise.

Theorem (Church)

$Th(\mathbb{N})$ is undecidable.
∀x_1 ∀x_2 ∃x_3 (((x_1 = x_2) ∨ (x_2 = x_1 + x_3)) & (x_1 = x_2x_3)) ∨ (x_3 ≠ x_2x_1)

Alexander Rybalov

Amplification of algorithmic problems: undecidable problems
A formula representation is **normalized** if for every variable $x_i, i > 1$ there exists variable $x_{i-1}$ which is the same leaf as $x_i$, or to the left leaf.

\[
\forall x_1 \forall x_2 \exists x_3 (((x_1 = x_3) \lor (x_3 = x_1 + x_2)) \land (x_1 = x_2 x_3)) \lor (x_2 \neq x_3 x_1)
\]

\[
\forall x_1 \forall x_3 \exists x_2 (((x_1 = x_2) \lor (x_2 = x_1 + x_3)) \land (x_1 = x_2 x_3)) \lor (x_3 \neq x_2 x_1)
\]

**Theorem**

For every formula there exists equivalent normalized formula.
Theorem

$Th(\mathbb{N})$ is super undecidable.
Cloning of formulas

For $\Phi = Q_1 x_1 \ldots Q_t x_t \phi$ the clone $C(\Phi)$ is the set of formulas

$$\left\{ Q_1 x_1 \ldots Q_t x_t Q_{t+1} x_{t+1} \ldots Q_{3n} x_{3n} \left( \phi \lor \left( (x_1 \neq x_1) \land \psi \right) \right) \right\},$$

where $\psi$ is arbitrary quantifier-free normalized formula on $x_1, \ldots, x_n$ and quantifiers $Q_{t+1}, \ldots, Q_n$ are arbitrary.
Theorem

For every formula $\Phi$ the set $C(\Phi)$ is not negligible.
Example 3: Word problem for semigroups

\[ S = \langle a_1, \ldots, a_m | R \rangle \] – a semigroup with undecidable word problem.

**Theorem (Myasnikov, Rybalov)**

Word problem in the semigroup \( S^+(x) = \langle A, x | R, xa_i = x, xx = x \rangle \) is super undecidable

**Cloning of words**

\[ C(w_1, w_2) = (w_1xv, w_2xu), \]

where \( v \) and \( u \) are arbitrary words over alphabet \( \{ a_1, \ldots, a_m, x \} \).