On the strongly generic undecidability of the halting problem

Alexander N. Rybalov
Omsk State University
rybalov@omskreg.ru

November 21, 2005
The halting problem (HP)

- Input: A Turing machine $M$

- Output:
  
  YES if $M$ halts on $\delta(M)$
  
  NO if $M$ does not halt on $\delta(M)$

here $\delta$ is some effective coding of Turing machines by binary strings

**Question 1** Is there an algorithm deciding HP?

**Theorem 1 (Classics)** HP is algorithmically undecidable.
Generic-case version of the HP

- Input: "Almost every" Turing machine $M$

- Output:

  YES if $M$ halts on $\delta(M)$

  NO if $M$ does not halt on $\delta(M)$

**Question 2** Is there an algorithm deciding HP for "almost all" inputs?

**Question 3** What does it mean "almost all"?
Asymptotic density of sets of programs

- $P$ is the set of all Turing machines
- $P_n$ is the set of all $n$-state machines
- $B$ is some set of Turing machines

**Definition 1**  *Asymptotic density of $B$ is*

$$
\mu(B) = \lim_{n \to \infty} \frac{|B \cap P_n|}{|P_n|}.
$$
The number of all $n$-state programs

Working alphabet is $\Sigma = \{0, 1, \square\}$. Machine can move the head to left and to right cell of the tape. Every $n$-state program contains $3n$ rules of type

$$(q_i, a) \rightarrow (q_j, b, s),$$

for every state $q_i$, $i = 1, \ldots, n$ and every symbol $a \in \Sigma$. Here $a, b \in \Sigma$, $s \in \{L, R\}$ and $q_j$ may be final state.

This follows that the number of all $n$-state programs is

$$|P_n| = (6(n + 1))^{3n}.$$
Generic sets of programs

Definition 2 A set $B$ of programs is called

- **generic if** $\mu(B) = 1$

- **negligible if** $\mu(B) = 0$

- **strongly negligible if** there are constants $0 < \sigma < 1$ and $C > 0$ such that for every $n$
  \[
  \frac{|B \cap P_n|}{|P_n|} < C \sigma^n,
  \]
  i.e. the sequence of the proportion of all $n$-state programs in $B$ exponentially fast converges to 0

- **strongly generic if** $P \setminus B$ is strongly negligible
Generic-case decidability and complexity of HP

**Question 4** Is there a generic set of Turing machines on which the HP is decidable?

**Theorem 2 (Hamkins, Miasnikov)** There is a generic set of Turing machines $B$ such that HP is polynomial time decidable on $B$.

**Question 5** What about strongly generic sets on which HP is decidable?

**Theorem 3 (Main result)** There is no strongly generic set of Turing machines on which HP is decidable.
How do we prove undecidability of classical HP?

Suppose HP is decidable, then

\[ \text{\textit{halt}}(x) = \begin{cases} 
1, & \text{if } x = \delta(M) \text{ and } M(x) \downarrow, \\
0, & \text{if } x = \delta(M) \text{ and } M(x) \uparrow.
\end{cases} \]

is computable function on \( \delta(P) \). Then the ”diagonal” function

\[ \text{\textit{diag}}(x) = \begin{cases} 
\text{not def}, & \text{if } x = \delta(M) \text{ and } M(x) \downarrow, \\
0, & \text{if } x = \delta(M) \text{ and } M(x) \uparrow.
\end{cases} \]

is computable on \( \delta(P) \) too. But the machine \( M \) computing \( \text{\textit{diag}} \) makes an error on \( \delta(M) \):

if \( M(\delta(M)) \downarrow \Rightarrow \text{\textit{diag}}(\delta(M)) = 0 \Rightarrow M(\delta(M)) \uparrow. \)

if \( M(\delta(M)) \uparrow \Rightarrow \text{\textit{diag}}(\delta(M)) \) is not defined \( \Rightarrow M(\delta(M)) \downarrow. \)
How to prove undecidability of HP on any strongly generic set?

Let \( S \) be a strongly generic set of programs. Suppose HP is decidable on \( S \), then

\[
\text{halt}(x) = \begin{cases} 
1, & \text{if } x = \delta(M) \text{ and } M(x) \downarrow, \\
0, & \text{if } x = \delta(M) \text{ and } M(x) \uparrow.
\end{cases}
\]

is computable function on \( \delta(S) \). Hence the function

\[
\text{diag}(x) = \begin{cases} 
\text{not def}, & \text{if } x = \delta(M) \text{ and } M(x) \downarrow, \\
0, & \text{if } x = \delta(M) \text{ and } M(x) \uparrow.
\end{cases}
\]

is computable on \( \delta(S) \) and computed by some machine \( M \). To get a contradiction we must give \( M \) the input \( \delta(M) \).

**Question 6** Should \( \delta(M) \) belong to \( \delta(S) \)? Should \( M \) be in \( S \)?
**Lemma 1** For any computable function \( f \) the set \( C(f) \) of all machines computing \( f \) is not strongly negligible.

**Idea of proof.** \( M \) has \( k \) states and computes \( f \). \( M^* \) has \( n > k \) states and program with the same transition rules as in \( M \) for first \( k \) states and arbitrary rules for \( n - k \) other states:

- **fixed 3\( k \) rules**
  \[
  \begin{cases}
  (q_1, 0) \to \ldots, \\
  \ldots \\
  (q_k, \square) \to \ldots,
  \end{cases}
  \]

- **arbitrary 3\((n - k)\) rules**
  \[
  \begin{cases}
  (q_{k+1}, 0) \to \ldots, \\
  \ldots \\
  (q_n, \square) \to \ldots.
  \end{cases}
  \]

\( M^* \) computes \( f \). \( A \) is the set of all such \( M^* \).

\[
\frac{|C(f) \cap P_n|}{|P_n|} \geq \frac{|A \cap P_n|}{|P_n|} = \frac{(6(n + 1))^{3(n-k)}}{(6(n + 1))^{3n}} = \frac{1}{(6(n + 1))^{3k}}.
\]

So \( C(f) \) is not strongly negligible.
Returning to HP.

- $C(diag)$ is not strongly negligible

- $P \setminus S$ is strongly negligible.

$\Rightarrow$ there is a machine $M$ computing $diag$ such that $M \in S$. That is all that we need to end proof of main theorem.
The end. Thank you.