Witnessability of Undecidable Problems

Jaehui Hwang, Jungin Rhee, Gyeongwon Jeong

Sections

1. Introduction
2. Witnessibility and its Implications
3. Iterative Witness
4. Additional Analysis
   - Witnessability and Rice's Theorem
   - Constructions of Witness Functions
   - Parallelized Iterative Witness Algorithm

Introduction

Decision Problem and Undecidability

- A **decision problem** is a subset of $\mathbb{N}$.

- A decision problem $P \subseteq \mathbb{N}$ is called **decidable** (we say f decides P) if there exists a computable function $f : \mathbb{N} \to \{0, 1\}$ such that

$$f(x) = \begin{cases} 1 & \text{if } x \in P \\ 0 & \text{if } x \notin P. \end{cases}$$

- A decision problem $P \subseteq \mathbb{N}$ which is not decidable is called **undecidable**.

The Halting Problem

- The **Halting problem** H is a decision problem of determining whether, for a given arbitrary program and input, the program halts when run with that input.

$$H := \{(p, i) : \text{the program } p \text{ halts on the input } i \in \mathbb{N}\}$$

- However, (p, i) doesn't inherently belong to the set of natural numbers. How can we interpret H as a subset of $\mathbb{N}$?

The Halting Problem

- A program p is a finite sequence(string) over a finite alphabet Σ:

$$p = p_1 p_2 \cdots p_n, \text{ where } p_1, \cdots, p_n \in \Sigma$$

- Therefore, we can view a program p as a base-n integer(natural number), where $n = |\Sigma|$. The two terms "program" and "natural number" are used interchangeably throughout this slides.

- Moreover, since $\mathbb{N}^2$ and $\mathbb{N}$ have the same cardinality, there exists a bijection $\langle \cdot, \cdot \rangle$ from $\mathbb{N}^2$ to $\mathbb{N}$.

- Now, we can define the Halting problem H as a subset of $\mathbb{N}$:

$$H := \{\langle p, i \rangle : \text{the program } p \text{ halts on the input } i \in \mathbb{N}\}$$

Undecidability of the Halting Problem

Theorem 1

The Halting problem H is undecidable.

Proof.

Suppose H is decidable. Let $f : \mathbb{N} \to \{0, 1\}$ be a computable function such that $f(\langle p, i\rangle) = 1$ if and only if $\langle p, i\rangle \in H$. Consider the following program h:

$$h := \lambda i.\,(\textbf{if } f(\langle i, i\rangle) = 0 \textbf{ then } 0 \textbf{ else } (\textbf{while true skip}))$$

If $f(\langle h, h\rangle) = 1$, then h halts on the input h, so $f(\langle h, h\rangle)$ should be evaluated to 0. If $f(\langle h, h\rangle) = 0$, then h does not halt on the input h, so $f(\langle h, h\rangle)$ should not be evaluated to 0. Both cases result in a contradiction. □

Many-one Reduction

- One standard way to prove that a problem is undecidable is by many-one reduction.

Definition 2 (Many-one Reduction)

Given two problems $P, Q \subseteq \mathbb{N}$, one says P is **many-one reducible** to Q and writes $P \leq_m Q$, if there exists a computable function $g : \mathbb{N} \to \mathbb{N}$ such that

$$x \in P \quad \text{if and only if} \quad g(x) \in Q.$$

- Intuitively, P is many-one reducible to Q shows that Q is harder than P.
- Therefore, if $P \leq_m Q$ and P is undecidable, then Q is also undecidable.

Undecidable Problems in PL Theory

- Using Many-one reduction and the fact that the Halting problem is undecidable many problems in programming language theory are undecidable.
- One famous result about this is Rice's theorem:

  All non-trivial semantic properties of programs are undecidable.

- More concrete examples
  - Programming analysis [Landi 1992; Reps 2000]
  - Program verification [Abdulla and Jonsson 1996; Dima and Tiplea 2011]
  - Type systems [Hu and Lhoták 2020; Pierce 1992; Wells 1999]

Are We Doomed..?

- The undecidability of a problem implies no algorithm to solve the problem.
- Many problems, including problems in PL theory, are proven to be undecidable.
- Then.. are we doomed? Don't we have any chance to solve the problem?

Decidable Approximation

- In practice, people design decidable approximation, a.k.a. decidable heuristics, to handle undecidable problems.

Definition 3 (Decidable approximation)

Given an undecidable problem $P \subseteq \mathbb{N}$, any decidable problem $Q \subseteq \mathbb{N}$ is called a **decidable approximation** of P.

- Problem: There is no theoretical foundation for undecidable problems and its decidable approximations.

Contributions of the Paper

- Shows the existence of universal ways to identify the precision issues of many algorithms.
- Shows even if problems encountered in PL theory and formal methods may be undecidable, improving any given decidable approximation is computable. This provides a theoretical foundation for justifying why research efforts targeting those problems are promising.
- Many mathematical methods used in undecidability proofs are commonly regarded as ways to prove negative results (e.g., undecidability). However, the results show that they also give ways to improve any given decidable approximation (thus, they also have positive effects).

Witnessibility and its Implications

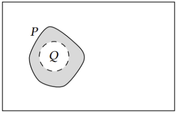
Imprecision Witness

- Consider an undecidable problem $P \subseteq \mathbb{N}$ and its decidable approximation $Q \subseteq \mathbb{N}$.

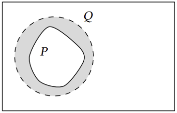
- The symmetric difference

$$P \triangle Q = (P - Q) \cup (Q - P)$$

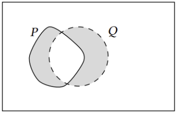
is called the **imprecision** of the approximation, and any element in $P \triangle Q$ is called an **imprecision witness** of Q.



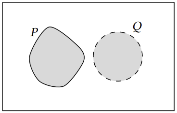
(a) Under-approximation.



(b) Over-approximation.



(c) Non-disjoint.



(d) Disjoint.

Witnessability of an Undecidable Problem

Definition 4 (Witnessability)

We say an undecidable problem $P \subseteq \mathbb{N}$ is **witnessable** if there exists a partial computable function w_P, such that for any decidable approximation $Q \subseteq \mathbb{N}$ and any program $q \in \mathbb{N}$ where ϕ_q decides Q, $w_P(q)$ is defined and $w_P(q) \in P \triangle Q$. The function w_P is called a **witness function** of P.

- For any decidable approximation Q of a witnessable problem P, we can always find an imprecision witness of Q using the witness function of P.

Partial Computable Function

- For each program $p \in \mathbb{N}$, we define a function ϕ_p:

$$\phi_p(x) = \begin{cases} n & \text{if running } p \text{ on } x \text{ terminates without error and returns } n \in \mathbb{N} \\ \uparrow & \text{(undefined) otherwise} \end{cases}$$

- We call each of these functions $\phi_1, \phi_2, \cdots$ a **partial computable function**.

- Given a partial computable function ϕ and an input x, the notation $\phi(x)\downarrow$ means ϕ is defined on x and $\phi(x)\uparrow$ means ϕ is undefined on x.

- The **domain** of a partial computable function ϕ is the set of inputs on which ϕ is defined: $\{x \in \mathbb{N} : \phi(x)\downarrow\}$.

- If the domain of ϕ is $\mathbb{N}$, ϕ is called a **(total) computable function**.

Witnessability of an Undecidable Problem

Definition 4 (Witnessability)

We say an undecidable problem $P \subseteq \mathbb{N}$ is **witnessable** if there exists a partial computable function w_P, such that for any decidable approximation $Q \subseteq \mathbb{N}$ and any program $q \in \mathbb{N}$ where ϕ_q decides Q, $w_P(q)$ is defined and $w_P(q) \in P \triangle Q$. The function w_P is called a **witness function** of P.

- For any decidable approximation Q of a witnessable problem P, we can always find an imprecision witness of Q using the witness function of P.
- "ϕ_q decides Q" means that ϕ_q is a total computable function.
- The domain of w_P is $\{q \in \mathbb{N} : \phi_q \text{ is a total computable function}\}$.

Main Theorems about Witnessability

Theorem 5

The diagonal halting problem $K = \{i : \phi_i(i) \downarrow\}$ is witnessable.

Proof.

For any decidable approximation $Q \subseteq \mathbb{N}$ and a natural number q such that ϕ_q is the characteristic function of Q, we construct a 2-ary partial computable function f using the universal function(interpreter) for all 1-ary partial computable functions(which interprets q as ϕ_q)

$$f(q,x) = \begin{cases} \uparrow & \text{if} \quad \phi_q(x) = 1 \\ 0 & \text{if} \quad \phi_q(x) = 0, \end{cases}$$

Define $w_p(q)$ as follows :

$$w_p(q) = \psi(j,q) = \phi_j^2(q,x) = f(q,x),$$

where j is an index such that $f(q,x) = \phi_j^2(q,x)$ for all $q, x \in \mathbb{N}$. Then, $w_P(q) \in K \triangle Q$ for all q, by case analysis. □

Main Theorems about Witnessability

Theorem 6

If P is witnessable, then its complement P^C is also witnessable.

Proof.

Similar to the previous theorem, construct f as follows:

$$f(q,x) = \begin{cases} \uparrow & \text{if} \quad \phi_q(x) = 0 \\ 0 & \text{if} \quad \phi_q(x) = 1, \end{cases}$$

and define w_{P^C} using f, then w_{P^C} is a witnessable function of P^C. □

Main Theorems about Witnessability

Theorem 7

If P_1 is witnessable and P_1 is many-one reducible to P_2, P_2 is also shown to be witnessable.

Proof idea : Let $P_1 \leq_m P_2$, and $w_{P_1}(x)$ is a witnessable function of P_1. Define $w_{P_2}(x)$ as follows:

$$w_{P_2}(x) = f(w_{P_1}(\psi(j, x, k))),$$

where f is a reduction from P_1 to P_2, and k is an index for f, and j is index for function $h(i_1, i_2, x) = \phi_{i_1}(\phi_{i_2}(x))$. Then $w_{P_2}(x)$ is a witnessable function of P_2.

Non-witnessable Problems

Theorem 8

There exists a non-witnessable undecidable problem.

Theorem 9

There are $2^{\aleph_0}$ witnessable undecidable problems, and non-witnessable undecidable problems.

Proof idea : Construct a specific algorithm to suggest corresponding witnessable(and non-witnessable) undecidable problem for each element in $\mathcal{P}(\mathbb{N})$, **using many-to-one reduction**.

Iterative Witness

Iterative Witness

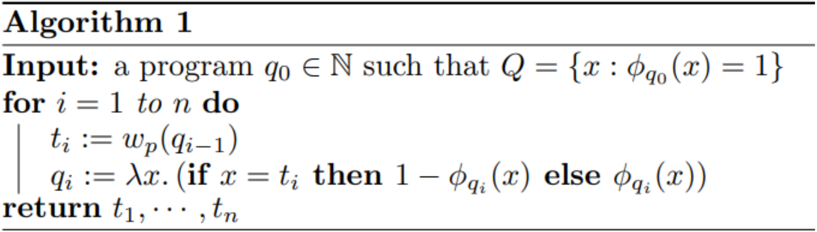
- Given a witnessable problem P and its decidable approximation Q, the witness function w_P gives only one imprecision witness of Q.
- Is there a way to find multiple imprecision witnesses for Q?
- Let q_0 be a program that decides Q, then $t_1 := w_P(q_0)$ is an imprecision witness of Q.
- Consider the following program q_1:

$$q_1 := \lambda x.\,(\textbf{if } x = t_1 \textbf{ then } 1 - \phi_{q_0}(x) \textbf{ else } \phi_{q_0}(x)),$$

then $w_P(q_1)$ is another imprecision witness of Q different from $w_P(q_0)$.

Iterative Witness Algorithm

- By repeating this process, we can obtain n imprecision witnesses $t_1, \cdots, t_n$ of Q for arbitrarily large n:



Algorithm 1

```
Input: a program $q_0 \in \mathbb{N}$ such that $Q = \{x : \phi_{q_0}(x) = 1\}$
for $i = 1$ to $n$ do
    $t_i := w_p(q_{i-1})$
    $q_i := \lambda x.\,(\textbf{if } x = t_i \textbf{ then } 1 - \phi_{q_i}(x) \textbf{ else } \phi_{q_i}(x))$
return $t_1, \cdots, t_n$
```

Additional Analysis

Outline

1. Introduction

2. Witnessibility and its Implications

3. Iterative Witness

4. Additional Analysis
    - Witnessability and Rice's Theorem
    - Constructions of Witness Functions
    - Parallelized Iterative Witness Algorithm

Witnessability and Rice's Theorem

- Rice's theorem states the following:

  All non-trivial semantic properties of programs are undecidable.

- And the paper mentions (without concrete proof) that:

  All non-trivial semantic properties of programs are witnessable.

- We completed the proof of these statements.

Rice's Theorem

- Let $\mathbf{P}$ denote the collection of all partial computable functions, i.e. $\mathbf{P} := \{\phi_i : i \in \mathbb{N}\}$.
- A **property** is a subset of $\mathbf{P}$.
- For each property $F \subseteq \mathbf{P}$, we can define a decision problem D_F by

$$D_F := \{i \in \mathbb{N} : \phi_i \in F\}.$$

Theorem 10 (Rice's Theorem)

For any property $F \subseteq \mathbf{P}$, D_F is decidable if and only if $F = \emptyset$ or $F = \mathbf{P}$.

Rice's Theorem

Theorem 10 (Rice's Theorem)

For any property $F \subseteq \mathbf{P}$, D_F is decidable if and only if $F = \emptyset$ or $F = \mathbf{P}$.

- This theorem is an immediate consequence of the following **Lemma 11**.

Lemma 11

For any property $F \subseteq \mathbf{P}$, if $F \neq \emptyset$ and $F \neq \mathbf{P}$ then $K \leq_m D_F$.

- The diagonal halting problem K is undecidable, so $K \leq_m D_F$ implies D_F is also undecidable.

Rice's Theorem

Lemma 11

For any property $F \subseteq \mathbf{P}$, if $F \neq \emptyset$ and $F \neq \mathbb{P}$ then $K \leq_m D_F$.

Proof.

WLOG, assume that *no-halt* $\notin F$ where *no-halt* being the partial computable function that is undefined everywhere. Since $F \neq \emptyset$, there exists $a \in \mathbb{N}$ such that $\phi_a \in F$. Now for any $i \in \mathbb{N}$, define $g(i)$ as a following program:

$$g(i) := \lambda x. (\phi_i(i); \phi_a(x))$$

If $i \in K$, since $\phi_i(i)$ terminates, $\phi_{g(i)} = \phi_a \in F$, thus $g(i) \in D_F$. If $i \notin K$, then since $\phi_i(i)$ not terminates, $\phi_{g(i)} =$ *no-halt* $\notin F$, thus $g(i) \notin D_F$. Therefore, g is a many-one reduction from K to D_F. □

Witnessability and Rice's Theorem

Theorem 12

For any non-trivial property $F \subseteq \mathbf{P}$, D_F is witnessable.

Lemma 11

For any property $F \subseteq \mathbf{P}$, if $F \neq \emptyset$ and $F \neq \mathbf{P}$ then $K \leq_m D_F$.

- The diagonal halting problem K is *witnessable* by **Theorem 5**, so $K \leq_m D_F$ implies D_F is also *witnessable* by **Theorem 7**.

Theorem 5

The diagonal halting problem $K = \{i : \phi_i(i) \downarrow\}$ is witnessable.

Theorem 7

If P_1 is witnessable and P_1 is many-one reducible to P_2, P_2 is also shown to be witnessable.

Outline

1. Introduction

2. Witnessibility and its Implications

3. Iterative Witness

4. Additional Analysis
   - Witnessability and Rice's Theorem
   - Constructions of Witness Functions
   - Parallelized Iterative Witness Algorithm

Constructions of Witness Functions

- One drawback of the definition of witnessable is that even if a problem is proved to be witnessable, it doesn't provide information about what the witness function for that problem might be.
- Thus, we tried to explicitly construct witness functions for several witnessable problems.

Witness Function of the Diagonal Halting Problem

- We first construct the witness function of the diagonal halting problem K, by following the proof of **Theorem 5** which demonstrates that K is witnessable. The witness function w_K we constructed is as follows:

$$w_K(q) := \lambda x.\,(\textbf{if } \phi_q(x) = 0 \textbf{ then } 0 \textbf{ else } (\textbf{while true skip}))$$

Theorem 13

For any $q \in \mathbb{N}$ such that $Q = \{x : \phi_q(x) = 1\}$ is decidable, $w_K(q) \in K \triangle Q$.

Proof.

Suppose $w_K(q) \in K$. Then $\phi_{w_K(q)}(w_K(q)) \downarrow$, so the program

$$\textbf{if } \phi_q(w_K(q)) = 0 \textbf{ then } 0 \textbf{ else } (\textbf{while true skip})$$

terminates. To do so we should have $\phi_q(w_K(q)) = 0$, which implies $w_K(q) \notin Q$. Suppose, on the other hand, $w_K(q) \notin K$. Then $\phi_{w_K(q)}(w_K(q)) \uparrow$, so the program above runs forever. To do so we should have $\phi_q(w_K(q)) = 1$, which implies $w_K(q) \in Q$. Therefore, $w_K(q) \in K \triangle Q$ in both cases. □

Witness Functions from Rice's Theorem

- Furthermore, we construct the witness function of D_F for any non-trivial property F. Assuming *no-halt* $\notin F$ and $\phi_a \in F$ for some $a \in \mathbb{N}$, the witness function w_F we constructed is as follows:

$$\begin{aligned} w_F(q) := {} & \textbf{let } g \equiv \lambda x.\,(\lambda i.\,(\phi_i(i);\, \phi_b(x))) \\ & \textbf{in } g(\lambda x.\,(\textbf{if } \phi_q(g(x)) = 0 \textbf{ then } 0 \textbf{ else } (\textbf{while true skip}))) \end{aligned}$$

Theorem 14

For any $q \in \mathbb{N}$ such that $Q = \{x : \phi_q(x) = 1\}$ is decidable, $w_F(q) \in D_F \triangle Q$.

Outline

1. Introduction

2. Witnessibility and its Implications

3. Iterative Witness

4. Additional Analysis
    - Witnessability and Rice's Theorem
    - Constructions of Witness Functions
    - Parallelized Iterative Witness Algorithm

Recap: Iterative Witness

- Let q_0 be a program that decides Q, then $t_1 := w_P(q_0)$ is an imprecision witness of Q.

- Consider the following program q_1:

$$q_1 := \lambda x. (\textbf{if } x = t_1 \textbf{ then } 1 - \phi_{q_0}(x) \textbf{ else } \phi_{q_0}(x)),$$

  then $w_P(q_1)$ is another imprecision witness of Q different from $w_P(q_0)$.

- Until now, to obtain $w_P(q_1)$, we need to compute $t_1 = w_P(q_0)$ first.

- However, if we have the explicit definition of w_P, we can define q_1 as follows (even without knowing t_1):

$$\begin{aligned} q_1 := {} & \textbf{let } w_P \equiv (\text{the explicit definition of } w_P) \\ & \textbf{in } \lambda x. (\textbf{if } x = w_P(q_0) \textbf{ then } 1 - \phi_{q_0}(x) \textbf{ else } \phi_{q_0}(x)) \end{aligned}$$

Parallelized Iterative Witness Algorithm

- Using this idea, we can compute $t_1 := w_P(q_0)$ and $t_2 := w_P(q_1)$ in parallel:

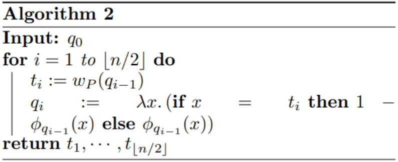
$$\begin{aligned} q_0 \quad &\Longrightarrow \quad t_1 := w_P(q_0) \\ q_1 \quad &\Longrightarrow \quad t_2 := w_P(q_1) \end{aligned}$$

where

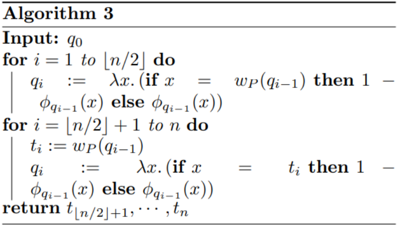
$$\begin{aligned} q_1 := \ &\textbf{let } w_P \equiv (\text{the explicit definition of } w_P) \\ &\textbf{in } \lambda x.\, (\textbf{if } x = w_P(q_0) \textbf{ then } 1 - \phi_{q_0}(x) \textbf{ else } \phi_{q_0}(x)). \end{aligned}$$

Parallelized Iterative Witness Algorithm

- We extend this idea to suggest the following parallel algorithms (for 2 cores) to compute n impricision witnesses of Q with an input $q_0 \in \mathbb{N}$ such that q_0 decides Q:



```
Algorithm 2
Input: $q_0$
for $i = 1$ to $\lfloor n/2 \rfloor$ do
    $t_i := w_P(q_{i-1})$
    $q_i := \lambda x.\,(\textbf{if } x = t_i \textbf{ then } 1 - \phi_{q_{i-1}}(x) \textbf{ else } \phi_{q_{i-1}}(x))$
return $t_1, \cdots, t_{\lfloor n/2 \rfloor}$
```



```
Algorithm 3
Input: $q_0$
for $i = 1$ to $\lfloor n/2 \rfloor$ do
    $q_i := \lambda x.\,(\textbf{if } x = w_P(q_{i-1}) \textbf{ then } 1 - \phi_{q_{i-1}}(x) \textbf{ else } \phi_{q_{i-1}}(x))$
for $i = \lfloor n/2 \rfloor + 1$ to $n$ do
    $t_i := w_P(q_{i-1})$
    $q_i := \lambda x.\,(\textbf{if } x = t_i \textbf{ then } 1 - \phi_{q_{i-1}}(x) \textbf{ else } \phi_{q_{i-1}}(x))$
return $t_{\lfloor n/2 \rfloor + 1}, \cdots, t_n$
```

- Both algorithms have the same time complexity in terms of the number of calls to the witness function w_P ($n/2$ times for each algorithm)
- This idea can be extended for k cores in such a way that the number of times the witness function needs to be called on each core becomes n/k.

References

- Shuo Ding and Qirun Zhang. 2023. Witnessability of Undecidable Problems. Proc. ACM Program. Lang. 7, POPL, Article 34 (January 2023), 21 pages.