Decidability and Undecidability*

Xiaofeng Gao

Department of Computer Science and Engineering
Shanghai Jiao Tong University, P.R.China

CS363-Computability Theory

*Special thanks is given to Prof. Yuxi Fu for sharing his teaching materials.

Outline

Decidability and Undecidability

A predicate $M(\mathbf{x})$ is decidable if its characteristic function $c_M(\mathbf{x})$ given by

$$c_M(\mathbf{x}) = \begin{cases} 1, & \text{if } M(\mathbf{x}) \text{ holds,} \\ 0, & \text{if } M(\mathbf{x}) \text{ does not hold.} \end{cases}$$

is computable.

The predicate $M(\mathbf{x})$ is undecidable if it is not decidable.

An algorithm for computing c_M is called a decision procedure for $M(x)$.

Undecidability Result

Theorem. The problem '$x \in W_x$' is undecidable.

Proof. The characteristic function of this problem is given by

$$c(x) = \begin{cases} 1, & \text{if } x \in W_x, \\ 0, & \text{if } x \notin W_x. \end{cases}$$

Suppose $c(x)$ was computable. Then the function $g(x)$ defined below would also be computable.

$$g(x) = \begin{cases} 0, & \text{if } c(x) = 0, \\ \text{undefined}, & \text{if } c(x) = 1. \end{cases}$$

Let m be an index for g. Then

$$m \in W_m \text{ iff } c(m) = 0 \text{ iff } m \notin W_m.$$

Undecidability Result

Corollary. There is a computable function h such that both '$x \in Dom(h)$' and '$x \in Ran(h)$' are undecidable.

Proof. Let

$$h(x) = \begin{cases} x, & \text{if } x \in W_x, \\ \text{undefined}, & \text{if } x \notin W_x. \end{cases}$$

Clearly $x \in Dom(h)$ iff $x \in W_x$ iff $x \in Ran(h)$.

Undecidability Result (Halting Problem)

Theorem. The problem '$\phi_x(y)$ is defined' is undecidable.

Proof. If $y \in W_x$ were decidable then $x \in W_x$ would be decidable.

In this proof we have reduced the problem '$x \in W_x$' to the problem '$y \in W_x$'. The reduction shows that the latter is at least as hard as the former.

Methodology: Reduction

Many problems can be shown to be undecidable by showing that they are at least as difficult as $x \in W_x$

Thus we can reduce one problem to another to prove the undecidability property.

If a problem $M(\mathbf{x})$ would lead to a solution to general problem $x \in W_x$, then we say that $x \in W_x$ is reduced to $M(\mathbf{x})$.

The decidability of $M(\mathbf{x})$ implies the decidability of $x \in W_x$, from which we can conclude the undecidability of $M(x)$.

Undecidability Result

Theorem. The problem '$\phi_x = \mathbf{0}$' is undecidable.

Proof. Consider the function f defined by

$$f(x,y) = \begin{cases} 0, & \text{if } x \in W_x, \\ \text{undefined}, & \text{if } x \notin W_x. \end{cases}$$

By s-m-n theorem there is some total computable function $k(x)$ such that $\phi_{k(x)}(y) \simeq f(x,y)$.

It is clear that $\phi_{k(x)} = \mathbf{0}$ iff $x \in W_x$.

Further Discussion

Let g be the characteristic function of $\phi_x = \mathbf{0}$,

$$g(x) = \begin{cases} 1 & \text{if } \phi_x = \mathbf{0}; \\ 0 & \text{if } \phi_x \neq \mathbf{0} \end{cases}$$

Suppose that g is computable, then so is the function $h(x) = g(k(x))$. However, we have

$$h(x) = \begin{cases} 1 & \text{if } \phi_{k(x)} = \mathbf{0}, \text{ i.e. } x \in W_x \\ 0 & \text{if } \phi_{k(x)} \neq \mathbf{0}, \text{ i.e. } x \notin W_x \end{cases}$$

Thus h is not computable. Hence g is not computable, and the problem $\phi_x = \mathbf{0}$ is undecidable.

Undecidability Result

Corollary. The problem '$\phi_x = \phi_y$' is undecidable.

Let c be a number such that $\phi_c = \mathbf{0}$.

If $f(x,y)$ is the characteristic function of the problem $\phi_x = \phi_y$, then the function $g(x) = f(x,c)$ is the characteristic function of $\phi_x = \mathbf{0}$.

Thus g is not computable, neither is f.

Thus '$\phi_x = \phi_y$' is undecidable.

Undecidability Result

Theorem. Let c be any number. The followings are undecidable.
(a) Acceptance Problem: '$c \in W_x$', ($P_x(c) \downarrow$, or '$c \in Dom(\phi_x)$')
(b) Printing Problem: '$c \in E_x$'. ('$c \in Ran(\phi_x)$')

Proof. Consider the function f defined by

$$f(x,y) = \begin{cases} y, & \text{if } x \in W_x, \\ \text{undefined}, & \text{if } x \notin W_x. \end{cases}$$

By s-m-n theorem there is some total computable function $k(x)$ such that $\phi_{k(x)}(y) \simeq f(x,y)$.

It is clear that $c \in W_{k(x)}$ iff $x \in W_x$ iff $c \in E_{k(x)}$.

Rice's Theorem

Theorem. (Rice)
Suppose $\emptyset \subsetneq \mathscr{B} \subsetneq \mathscr{C}_1$. Then the problem '$\phi_x \in \mathscr{B}$' is undecidable.

Proof. Suppose $f_\emptyset \notin \mathscr{B}$ and $g \in \mathscr{B}$. Let the function f be defined by

$$f(x,y) = \begin{cases} g(y), & \text{if } x \in W_x, \\ \text{undefined}, & \text{if } x \notin W_x. \end{cases}$$

By s-m-n theorem there is some total computable function $k(x)$ such that $\phi_{k(x)}(y) \simeq f(x,y)$.

It is clear that $\phi_{k(x)} \in \mathscr{B}$ iff $\phi_{k(x)} = g$ iff $x \in W_x$. □

Partially Decidable Predicates

A predicate $M(\mathbf{x})$ of natural numbers is partially decidable if the function given by

$$f(\mathbf{x}) = \begin{cases} 1, & \text{if } M(\mathbf{x}) \text{ holds,} \\ \text{undefined}, & \text{if } M(\mathbf{x}) \text{ does not hold,} \end{cases}$$

is computable.

The function $f(\mathbf{x})$ is the partial characteristic function.

Partially Decidable Predicates

1. The halting problem is partially decidable. Its partial characteristic function is given by

$$f(x,y) = \begin{cases} 1, & \text{if } P_x(y)\downarrow, \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

2. Any decidable predicate is partially decidable: simply arrange for the decision procedure to enter a loop whenever it gives output 0.

3. For any computable function $g(\mathbf{x})$ the problem $\mathbf{x} \in Dom(g)$ is partially decidable, since it has the computable characteristic function $\mathbf{1}(g(\mathbf{x}))$.

Partially Decidable Predicates

4. The problem '$x \notin W_x$' is not partially decidable. For if f is its partial characteristic function, then

$$x \in Dom(f) \Leftrightarrow x \notin W_x.$$

The domain of its partial characteristic function differs from the domain of every computable function.

Partially Decidable Predicates

Theorem. A predicate $M(\mathbf{x})$ is partially decidable iff there is a computable function $g(\mathbf{x})$ such that $M(\mathbf{x}) \Leftrightarrow \mathbf{x} \in Dom(g)$.

Proof. g is essentially the partial characteristic function.

Partially Decidable Predicates

Theorem. A predicate $M(\mathbf{x})$ is partially decidable iff there is a decidable predicate $R(\mathbf{x}, y)$ such that $M(\mathbf{x}) \Leftrightarrow \exists y.R(\mathbf{x}, y)$.

Proof. "$\Leftarrow$" If $R(\mathbf{x}, y)$ is decidable and $M(\mathbf{x}) \Leftrightarrow \exists y.R(\mathbf{x}, y)$, then $g(\mathbf{x}) \simeq \mu y R(\mathbf{x}, y)$ is computable. Clearly $M(\mathbf{x}) \Leftrightarrow \mathbf{x} \in Dom(g)$. Thus $M(\mathbf{x})$ is partially decidable.

"$\Rightarrow$" Conversely suppose $M(\mathbf{x})$ is partially decided by program P. Let $R(\mathbf{x}, y)$ be

$$R(\mathbf{x}, y) \equiv P(\mathbf{x}) \downarrow \text{ in } y \text{ steps.}$$

Then $R(\mathbf{x}, y)$ is decidable and $M(\mathbf{x}) \Leftrightarrow P(\mathbf{x}) \downarrow \Leftrightarrow \exists y.R(\mathbf{x}, y)$.

Partially Decidable Predicates

Theorem. If $M(\mathbf{x}, y)$ is partially decidable, so is $\exists y.M(\mathbf{x}, y)$.

Proof. Let $R(\mathbf{x}, y, z)$ be a decidable predicate such that $M(\mathbf{x}, y) \Leftrightarrow \exists z.R(\mathbf{x}, y, z)$. Then $\exists y.M(\mathbf{x}, y) \Leftrightarrow \exists y \exists z.R(\mathbf{x}, y, z)$.

Use standard technique of coding the pair of numbers y, z such that $R(\mathbf{x}, y, z)$ reduces to the search for a single number u such that $\exists y.M(\mathbf{x}, y) \Leftrightarrow \exists u.R(\mathbf{x}, (u)_0, (u)_1)$.

The predicate $S(\mathbf{x}, y) \equiv R(\mathbf{x}, (u)_0, (u)_1)$ is decidable by substitution and so $\exists y.M(\mathbf{x}, y)$ is partially decidable.

Corollary. If $M(\mathbf{x}, \mathbf{y})$ is a partially decidable, so is $\exists \mathbf{y}.M(\mathbf{x}, \mathbf{y})$.

Partially Decidable Predicates

Example: $x \in E_y^{(n)}$ (n fixed) is partially decidable.

Proof: $x \in E_y^{(n)} \Leftrightarrow \exists z_1 \cdots \exists z_n \exists t (P_y(z_1, \cdots, z_n) \downarrow x \text{ in } t \text{ steps})$. The right one is decidable so $x \in E_y^{(n)}$ is partially decidable.

Example: $W_x \neq \varnothing$ is partially decidable.

Proof: $W_x \neq \varnothing$ iff $\exists y \exists t.(P_x(y) \downarrow \text{in } t \text{ steps})$. So $W_x \neq \varnothing$ is partially decidable.

Partially Decidable Predicates

Theorem. A predicate $M(\mathbf{x})$ is decidable iff both $M(\mathbf{x})$ and $\neg M(\mathbf{x})$ are partially decidable.

Proof: "$\Rightarrow$" If $M(\mathbf{x})$ is decidable, so is 'not $M(\mathbf{x})$', so both are partially decidable.

"$\Leftarrow$" Conversely, suppose that partial decidable procedures for $M(\mathbf{x})$ and 'not $M(\mathbf{x})$' are given by programs F, G. Then

$$F(x) \downarrow \Leftrightarrow M(\mathbf{x}) \text{ holds and } G(x) \downarrow \Leftrightarrow \text{'not } M(\mathbf{x})\text{' holds.}$$

Also, $\forall \mathbf{x}$, either $F(\mathbf{x}) \downarrow$ or $G(\mathbf{x}) \downarrow$ but not both.

Thus given $\mathbf{x}$, run the computation $F(\mathbf{x})$ and $G(\mathbf{x})$ simultaneously and go on until one of them stops. If $F(\mathbf{x})$ stops, then $M(\mathbf{x})$ holds; if $G(\mathbf{x})$ stops, then $M(\mathbf{x})$ not hold.

Partially Decidable Predicates

Corollary (Divergence Problem). The problem '$y \notin W_x$' ('$P_x(y) \uparrow$' or '$\phi_x(y)$ is undefined') is not partially decidable.

Proof: If this problem were partially decidable, since $P_x(y) \downarrow$ is partially decidable, then by the above theorem the Halting problem would be decidable.

Partially Decidable Predicates

Theorem. Let $f(\mathbf{x})$ be a partial function. Then f is computable iff the predicate '$f(\mathbf{x}) \simeq y$' is partially decidable.

Proof. If $f(\mathbf{x})$ is computable by $P(\mathbf{x})$, then

$$f(\mathbf{x}) \simeq y \Leftrightarrow \exists t.(P(\mathbf{x}) \downarrow y \text{ in } t \text{ steps}).$$

We are done by observing that '$P(\mathbf{x}) \downarrow y$ in t steps' is decidable.

Conversely let $R(\mathbf{x}, y, t)$ be such that

$$f(\mathbf{x}) \simeq y \Leftrightarrow \exists t.R(\mathbf{x}, y, t).$$

The equivalence gives rise to an algorithm.