Programming Languages and Computation

Week 11: Undecidability and Reductions

* 1. [Trick question.] Is it decidable whether God exists?

Solution

God either exists, or does not. In the first case, the program that always returns Yes/True/1 correctly decides the existence of God. In the second case, the program that returns No/False/0 correctly decides the problem.

* 2.

1. Is the set $\emptyset$ decidable?
2. Is the set $\mathbb{N}$ decidable?

Solution

The first is decided wrt the variable x by the program

```
x <- 0
```

The second is decided wrt the variable x by the program

```
x <- 1
```

** 3. Show that if $f : U \lesssim V$ and $g : V \lesssim W$ then $g \circ f : U \lesssim W$.

Solution

Recall that we have to check two things.

If f and g are computable, then so is their composition $g \circ f$. (This was an exercise in a previous sheet.)

By the definitions of $f : U \lesssim V$ and $g : V \lesssim W$ we have for any $n \in \mathbb{N}$ that

$$n \in U \iff f(n) \in V \iff g(f(n)) \in W$$

Hence, $g \circ f$ is a reduction from U to W.

*** 4. Prove that the set

$$\mathsf{ZERO} = \{ \ulcorner S \urcorner \mid [\![\mathtt{S}]\!]_{\mathtt{x}}(0)\downarrow \}$$

is undecidable by reduction from HALT.

Solution

We prove this by reduction from the halting problem. Suppose we want to decide the halting problem for a program S and input n. We construct the following program $G_{S,n}$: On input m,

1. Ignore the input m.
2. Simulate the program S on input n.
3. If that simulation terminates, output 0 (or any other number).

This code transformation is computable: we can write a program that given the source code $\ulcorner S \urcorner$ and n computes the source code $\ulcorner G_{S,n} \urcorner$.

Furthermore, we have that

$$\langle \ulcorner S \urcorner, n \rangle \in \mathsf{HALT} \iff \llbracket G_{S,n} \rrbracket_{\mathtt{x}}(0)\downarrow \iff \ulcorner G_{S,n} \urcorner \in \mathsf{ZERO}$$

The first equivalence follows by the structure of the program $G_{S,n}$. If simulating S on n terminates, execution always reaches step 3, and hence $G_{S,n}$ always outputs 0. If simulating S on n does not terminate, then $G_{S,n}$ never terminates, so that $\llbracket G_{S,n} \rrbracket_{\mathtt{x}}(0)\uparrow$.

The second equivalence is by definition.

Hence, if we could decide ZERO then we could decide HALT. So we cannot decide ZERO.

*** 5. Prove that the predicate ZERO is undecidable, but using Rice's theorem instead.

Solution

The set ZERO is exactly of the form

$$\{\ulcorner S \urcorner \mid \llbracket S \rrbracket_{\mathtt{x}} \in \mathscr{ZERO}\}$$

where

$$\mathscr{ZERO} = \{f : \mathbb{N} \rightharpoonup \mathbb{N} \in \mathscr{PR} \mid f(0)\downarrow\}$$

Furthermore, the everywhere-undefined function $\emptyset : \mathbb{N} \rightharpoonup \mathbb{N}$ is *not* in $\mathscr{ZERO}$, whereas the identity function $\mathrm{id}_{\mathbb{N}} : \mathbb{N} \rightharpoonup \mathbb{N}$ is in $\mathscr{ZERO}$. Both are computable.

Therefore, Rice's theorem applies to show that ZERO is undecidable.

**** 6. Show that the predicate

$$U = \{\langle \ulcorner S \urcorner, \ulcorner T \urcorner \rangle \mid \llbracket S \rrbracket_{\mathtt{x}}(0) \simeq \llbracket T \rrbracket_{\mathtt{x}}(0)\}$$

is undecidable, by reduction from HALT.

Solution

By reduction from the Halting Problem.

Construct the code transformation $F : \textbf{Stmt} \times \mathbb{N} \to \textbf{Stmt} \times \textbf{Stmt}$ given by

$$F(\mathtt{D}, n) = (\mathtt{x\ :=\ n;\ D;\ x\ :=\ 0}, \mathtt{x\ :=\ 0})$$

This code transformation is computable. Its reflection is computed as follows: On input m,

1. Write $m = \langle \ulcorner \mathtt{D} \urcorner, n \rangle$.
2. Construct the program $S_{\mathtt{D},n}$, which is given by

```
x <- n; D; x <- 0
```

3. Return $\langle \ulcorner S_{\mathtt{D},n} \urcorner, \ulcorner \mathtt{x\ :=\ 0} \urcorner \rangle$.

Now we have

$$\langle \ulcorner S_{\mathtt{D},n} \urcorner, \ulcorner \mathtt{x\ :=\ 0} \urcorner \rangle \in U \iff \langle \ulcorner \mathtt{D} \urcorner, n \rangle \in \mathsf{HALT}$$

As HALT is not decidable, neither is U.

*** 7. Is it possible to prove that the predicate U from the previous exercise is undecidable using Rice's theorem? If so, prove it. If not, explain why not.

Solution

The set U is not of the form $\{\ulcorner S \urcorner \mid [\![S]\!]_{\mathtt{x}} \in \mathscr{F}\}$ for a set of computable functions $\mathscr{F} \subseteq \mathscr{PR}$. Therefore, Rice's theorem does not *immediately* apply to it.

However, there may be a proof that this is undecidable that *uses* Rice's theorem to arrive at some intermediate result, from which the undecidability of U follows.

**** 8. [Trick question.] Is the predicate

$$V = \{\ulcorner S \urcorner \mid \forall n \in \mathbb{N}.\ [\![\mathtt{S}]\!]_{\mathtt{x}}(n) \downarrow\}$$

from the last lecture semi-decidable? Why or why not?

Solution

It is indeed *not* semi-decidable.

However, you may intuitively notice that semi-decidability usually amounts to running simulations, which try to decide whether $\exists n. \phi(n)$ for some decidable property $\phi(n)$ of natural numbers by trying all possible $n \in \mathbb{N}$ in parallel, hoping that one might succeed.[1]

In this case, the property $\phi(n) = [\![\mathtt{S}]\!]_{\mathtt{x}}(n) \downarrow$ is asked to hold of *all* natural numbers. Thus, a positive answer seems to somehow require trying all numbers in parallel at the same time, and noting that the property succeeds for all numbers. Algorithmically, this seems to require an infinite number of simulations! While this is indeed true, you do not currently have the tools to show this.

[1] See also `https://risingentropy.com/the-arithmetic-hierarchy-and-computability/`.