Week 10: Exercises on Properties of Code

Multiple Choice Questions

Question 10M.1

Jessica writes the following program.

```
nat x = 0;
nat y = 2;
while (x < y) {
  x = x+1;
  y = y+2;
}
```

Bill writes the following program.

```
nat x = 0;
nat y = 2;
while (x < y) {
  x = x+2;
  y = y+1;
}
```

(a) Both programs halt.

(b) Both programs hang.

(c) Jessica's program halts and Bill's hangs.

(d) Jessica's program hangs and Bill's halts.

Solution (d).

Question 10M.2

Let P be a decision problem. To prove P undecidable, it suffices to

(a) Reduce P to a decidable problem.

(b) Reduce P to an undecidable problem.

(c) Reduce a decidable problem to P.

(d) Reduce an undecidable problem to P.

Solution (d).

Question 10M.3

Natalie, Harry and Xerxes each write a program with parameter `nat x`. Natalie's program is as follows.

```
nat y = x+2;
while (y < 100) {
  println("Hello");
  y = y+1;
}
```

Harry's program is as follows.

```
nat y = x;
while (y < 98) {
   println("Hello");
   println("Hello");
   y = y+2;
}
```

Xerxes' program is as follows.

```
nat y = x;
while (y < 98) {
   println("Hello");
   y = y+1;
}
```

(a) Natalie and Harry's program have the same semantics, but Xerxes' has a different semantics.

(b) Natalie and Xerxes' programs have the same semantics, but Harry's has a different semantics.

(c) Harry and Xerxes' programs have the same semantics, but Natalie's has a different semantics.

(d) All three programs have the same semantics.

Solution (b).

Question 10M.4

The body-code of a method

```
nat f (nat x){
  ...
}
```

is *sleepy* when `f(5)` and `f(6)` and `f(7)` all return even numbers.

(a) Sleepiness is semantic and decidable.

(b) Sleepiness is semantic and undecidable.

(c) Sleepiness is not semantic and is decidable.

(d) Sleepiness is neither semantic nor decidable

Solution (b).

Question 10M.5

Which of the following is true?

(a) Every undecidable property of code holds in some case.

(b) Every decidable property of code fails to hold in some case.

(c) Every semantic property of code holds in some case.

Solution (a).

Exercises

Exercise 10.1

Consider the following properties (*happy*, *joyous*, *glad* and *jolly*) of code. For each of these properties, answer the following questions, with explanation.

- Does it hold in some case?
- Does it fail to hold in some case?
- Is it semantic?
- Is it decidable?

1. A procedure

   ```
   void f (nat n)
   ```

   is *happy* when "a" has no more occurrences in the body code than "b".

   Solution

   The following code is happy:

   ```
   nat b = 3;
   ```

   The following code is not happy:

   ```
   nat a = 3;
   ```

   Happiness is not a semantic property because the procedure

   ```
   if (n > 3) {System.out.println("Hello");}
   ```

   is happy, the procedure

   ```
   if (n > 3) {System.out.println("Hello");}
   // a useless comment.
   ```

   is unhappy, and they have the same semantics. Happiness is a decidable property because we can write a program that goes through the text of the procedure, counts the number of `a`'s and `b`'s, then compares the counts to see whether the number of `b`'s is greater, and returns False if it is and True otherwise.

2. A procedure

   ```
   void f (nat n)
   ```

   is *joyous* when, if you apply it to any even number, it prints at least 3 characters.

   Solution

   The following code is joyous:

   ```
   System.out.println("aaa");
   ```

   The following code is not joyous:

   ```
   System.out.println("aa");
   ```

   because when you apply it to 0, it prints only two characters.

   Joyousness is a semantic property because it only depends on the input-output behaviour of a procedure. In other words, if we have two procedures that, for any argument n, have the same behaviour (printing the same characters and returning, or printing the same characters and not halting), then either they are both joyous or neither is.

   By Rice's Theorem, since joyousness is semantic and holds in some case and doesn't hold in some case, it is not decidable.

3. A procedure

```
void f (nat n)
```

is *glad* when either “a” has no more occurrences in the body code than “b”, or “b” than “c”, or “c” than “a”.

Solution

The following program is glad:

```
return;
```

Actually, *every* program is glad. For if the first and second conditions don’t hold then there are more `a`’s than `b`’s and more `b`’s than `c`’s. So there are more `a`’s than `c`’s, so there cannot be more `c`’s than `a`’s.

Gladness is semantic because it always holds. It is decidable for the same reason: given any procedure, just return True.

4. A pair of methods

```
nat f (nat n)
nat g (nat n)
```

is *jolly* when, for any integer n, either $\texttt{f}(n)$ hangs or $\texttt{g}(n)$ hangs, or they both terminate and the sum of the values returned is 33.

Solution

The following pair of methods is jolly.

```
return 26;
```

```
return 3+4;
```

The following pair of methods is not jolly:

```
return 3;
```

```
return 4;
```

because `f(0)` and `g(0)` both terminate, and the sum of the results is 7.

Jolliness is semantic because it only depends on the input-output behaviour of `f` and `g`. By Rice’s Theorem, since it is semantic, and holds in some case and doesn’t hold in some case, it is undecidable.

Exercise 10.2

Dave manages a software firm. His employees write programs that interact with a user by inputting and outputting text. Dave tells them that, as long as the user has only input basic English words, their code should only output basic English words. (Here, “basic” means no longer than 8 characters.) Dave wants a tool to check that his employees’ programs meet this requirement. Is that possible? Explain your answer.

Solution

No, it is not possible. This is a semantic property of code, since it only depends on the output as a function of the input. The program

```
println "Hello";
```

has this property and the program

```
println "Wonderful";
```

does not. So by Rice’s Theorem, it is undecidable.

Exercise 10.3 (Hard.)

Sketch a procedure which tests whether a given polynomial (with integer coefficients the first coefficient being 1) has an integer root. So for example, for $x^3 - 4x^2 + x + 6$ the answer should be "yes", because 3 is such a root, and for $x^2 - 2$ the answer should be no, because the only two roots, $\sqrt{2}$ and $-\sqrt{2}$, are irrational.

Solution

Let the polynomial be $P(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$. If α is a root, then we can write $P(x) = (x - \alpha)Q(x)$ for some polynomial Q of degree $n - 1$. But Q must also have integer coefficients, since the coefficients can be found by the polynomial division process, which never needs to divide coefficients. Hence a_0 is α times the constant term of Q, so the only possibilities for α are the factors of a_0. It suffices therefore to check them.