CSCI 303 Fall 2006
Homework 9

1. last updated fall 2006, Yuriy Brun, ybrun@usc.edu

For each of the following subsets of $\aleph$, prove whether it is decidable or undecidable:

a. $A = \{i : \beta_i(12) = 13\}$

Assume A is decidable, then let B_a be the program that decides A $(\beta_a = C_A)$. Then consider the following SBasic program:

"Input i
W = MY INDEX
Simulate B_a on input W
If B_a returns 1
K = 55
Else
K = 13
Print K"

There exists some j such that the above program is B_j. On input 12, B_j prints 13 if B_a prints 0 on input j and B_j prints 55 otherwise. Thus, if B_a claims that j $\in$ A, then $\beta_j(12) = 55$. Contradiction. If B_a claims that j $\notin$ A, then $\beta_j(12) = 13$. Contradiction. So A must be undecidable.

b. $B = \{i : \forall x \beta_i(x) \downarrow\}$

Assume B is decidable, then let B_b be the program that decides B $(\beta_b = C_B)$. Then consider the following SBasic program:

"Input i
W = MY INDEX
Simulate B_b on input W
L: If B_b returns 1
Goto L
Print i"

There exists some j such that the above program is B_j. On all inputs, B_j prints 100 if B_b prints 0 on input j and does not halt otherwise. Thus, if B_b claims that j $\in$ B, then $\beta_j(0)\uparrow$. Contradiction. If B_b claims that j $\notin$ A, then $\forall x \beta_j(x) \downarrow$. Contradiction. So B must be undecidable.

c. $C = \{i : (\exists a)\beta_i(i) = a \,\&\, a < i\}$

Assume C is decidable, then let B_c be the program that decides C $(\beta_c = C_C)$. Then consider the following SBasic program:

```
"Input i
W = MY INDEX
Simulate Bc on input W
If Bc returns 1
        K = W + 1
Else
        K = W – 1
Print K"
```

There exists some j such that the above program is B_j. On input i, B_j prints j + 1 if B_c prints 1 on input j and prints j – 1 otherwise. Thus, if B_c claims that j ∈ C, then $\beta_j(j) = j + 1 > j$. Contradiction. If B_c claims that j ∉ A, then $\beta_j(j) = j - 1 < j$. Contradiction. So C must be undecidable.

2. last updated fall 2006, Yuriy Brun, ybrun@usc.edu

Prove or disprove each of the following statements:

a. $\exists i \forall x \beta_i(x) = i$

The question asks, does there exist an SBasic program, which for all x, on input x prints its own index. The answer is yes:

```
"Input x
W = MY INDEX
Print W"
```

b. $\exists i \beta_i(i) = i^2$

The question asks, does there exist an SBasic program, which on input its own index prints its own index squared. The answer is yes:

```
"Input x
W = MY INDEX
L = W * W
Print L"
```

c. $\exists i \beta_i(i^2) = i$

The question asks, does there exist an SBasic program, which on input its own index squared prints its own index. The answer is yes:

```
"Input x
W = MY INDEX
Print W"
```

3. last updated fall 2006, Yuriy Brun, ybrun@usc.edu

Let $S = \{i : \beta_i(0)\downarrow\}$**.**

a. Is S **RE? Prove your answer.**

Consider the following program:

"A = 1

L: use an SBasic enumerator to write $B_1, B_2, \ldots, B_A$ in memory.

Simulate B_1 on input 0 for A steps. If it halts, print 1.

Simulate B_2 on input 0 for A steps. If it halts, print 2.

…

Simulate B_A on input 0 for A steps. If it halts, print A.

A = A + 1

Goto L."

The program recursively enumerates the elements of the set S, so S is recursively enumerable.

b. Is $\overline{S} = \aleph - S$ **RE? Prove your answer.**

We have shown in class that S is not decidable, and also that, for all sets $A \subseteq \aleph$, if A and $\overline{A}$ are both recursively enumerable, then A is decidable. Thus $\overline{S}$ has to be undecidable.

Proof that S is not decidable:

Assume S is decidable, then let B_a be the program that decides B $(\beta_a = C_S)$. Then consider the following SBasic program:

"Input i

W = MY INDEX

Simulate B_a on input W

L: If B_b returns 1

Goto L

Print i"

There exists some j such that the above program is B_j. On all inputs, B_j halts if B_a prints 0 on input j and does not halt otherwise. Thus, if B_a claims that j $\in$ S, then $\beta_j(0)\uparrow$. Contradiction. If B_a claims that j $\notin$ S, then $\beta_j(0)\downarrow$. Contradiction. So S must be undecidable.

Proof that if S and $\overline{S}$ are both recursively enumerable, then S is decidable:

On input i, alternate between simulating the program that recursively enumerates S and the program that recursively enumerates $\overline{S}$, until one of them prints i. Depending on which program printed i, i is in or not in S.