Proof of step 4: The language Halting, \( H \), is not recursive

In the following proof we argue by contradiction that if \( H \) is recursive then \( SA \) is recursive. But \( SA \) is not recursive. Therefore, \( H \) is not recursive.

Proof:

Assume that \( H \) is recursive and \( T_H \) is the corresponding Turing Machine. Then, we can construct a composite Turing Machine \( T_{SA} \) for the language \( SA \) as shown in figure below.

Note that the Turing Machine \( T_1 \) converts the input of \( T_{SA} \) into an input for the Turing Machine \( T_H \). Thus, \( T_1 \) can be designed in such a way that it will copy an input \( w \) to the right of it after changing \( w \) to its encoding \( e(w) \). This TM, \( T_1 \), can be designed in such a way that it does not loop. If \( T_H \) does not loop then we have a TM for the language \( SA \) which does not loop. Thus, \( SA \) is recursive. This is a contradiction.

Explanation of the input \( < w, e(w) > \) to the box \( T_H \) above. Note that the input to box \( T_H \) should be \( < w, e(w) > \) and not \( < e(w), e(w) > \). The explanation is given below.

Definition of the language SelfAccepting, \( SA \), is as follows:

\( SA = \{ w \epsilon \{0,1\}^* \text{ and } w=e(T) \text{ for some TM } T, \text{ and } T \text{ accepts } w \} \)

Thus, the input to box \( T_{SA} \) is any string \( w \) of 0’s and 1’s. But \( T_{SA} \) accepts \( w \) only when it is a code for some TM \( T \) and \( T \) accepts \( w \). Any other inputs of 0’s and 1’s to \( T_{SA} \) are not accepted. Since \( w \) already represents a TM code, it need not be coded again. However, when \( w \) is used as an input to \( T \), this \( w \) needs to be coded for input. Note that the TM code itself is the input.

Following is an example:

Assume that \( T \) is a TM that accepts \( \{0,1\}^* \), that is, it accepts all strings. Thus, TM \( T \) has only one move that goes to halt state from the start state.
on input delta. Obviously, this TM T accepts its own encoding e(T) because it accepts everything.

TM T:
- Move: \((q_1, \Delta) = (h, \Delta, S)\) We assumed \(q_1\) is the start state. Thus,
- \(e(T) = 001001010101011\)

Now \(w = e(T)\) is the input to the TM T. This needs to be coded. We have two symbols in the input, namely, 0 and 1. We use the following encoding for the characters 0 and 1.
- \(S(0) = 00\)
- \(S(1) = 000\)

Based on the above encoding \(e(w)\) becomes:
- \(001001000100010010001001000100010010001001000100010010010010010001\)

Therefore, \(<e(T), e(W)>\) is \(<w, e(W)>\) which is
- \(00100101010111100100100010010001001000100100010010001001000100100010001\)