Theory of Computation: Assignment 8 Solutions

Arjun Chandrasekhar

1. (a) We will show that we can enumerate all possible finite length homework assignments. We'll use the following procedure:

   1. List all assignments with 0 characters
   2. List all assignments with just 1 character
   3. List all assignments with 2 characters
   4. ...

   Eventually, any assignment with a finite number of symbols will be listed out.

   (b) AFSOC the set of infinite assignments is countable, i.e. we can list out the assignments $A_1, A_2, \ldots$. We will create a new assignment A^* as follows:

   1. The first symbol of A^* is different from the first symbol of A_1
   2. The second symbol of A^* is different from the second symbol of A_2
   3. ...

   In general, the i-th symbol of A^* is different from the i-th symbol of A_i.

   Because our assignments are allowed to be infinitely long, and because every symbol comes from the English alphabet, A^* is a valid assignment.

   Note that, A^* disagrees with every other assignment A_i that was part of our enumeration. This means that A^*, which is a perfectly valid assignment, is not part of the enumeration. But then our enumeration was not complete - it didn't contain every possible assignment. This is a contradiction. We conclude that the set of infinite assignments is not actually countable.

2. AFSOC $\overline{\text{HALT}}$ is decidable. There is a machine $\overline{H}$ that decides $\overline{\text{HALT}}$. We will construct a strange machine S that does the following:

   1. S takes a TM description $\langle M \rangle$ as input
   2. S runs $\overline{H}$ on $\langle M \rangle\langle M \rangle$. That is, it uses $\overline{H}$ to check if M loops on its own source code.
   3. S "does the opposite" of $\overline{H}$.
      a. If $\overline{H}$ accepts $\langle M \rangle\langle M \rangle$, then S immediately halts.
      b. If $\overline{H}$ rejects $\langle M \rangle\langle M \rangle$, then S immediately goes into a loop.

   Figure 1 gives a diagram of this machine.

   Let's see what happens when S receives its own source code $\langle S \rangle$ as input

   1. S runs $\overline{H}$ on $\langle S \rangle\langle S \rangle$. That is, it uses $\overline{H}$ to check if S loops on its own source code $\langle S \rangle$
   2. S "does the opposite" of $\overline{H}$
      a. If $\overline{H}$ accepts $\langle S \rangle\langle S \rangle$, then S immediately halts.
      b. If $\overline{H}$ accepts $\langle S \rangle\langle S \rangle$, then S goes into a loop

   If $\overline{H}$ says that S loops on $\langle S \rangle$, then S immediately halts. If $\overline{H}$ says that S halts on $\langle S \rangle$, then S goes into a loop. Either S is literally contradicting itself, or $\overline{H}$ is not deciding $\overline{\text{HALT}}$ correctly. Either way, we have reachced a contradiction. We conclude that $\overline{\text{HALT}}$ is undecidable.

   Figure 2 gives a flowchart for how S ends up contradicting itself.

   Figure 3 show how we can interpret the above proof as a form of $\overline{\text{HALT}}$.

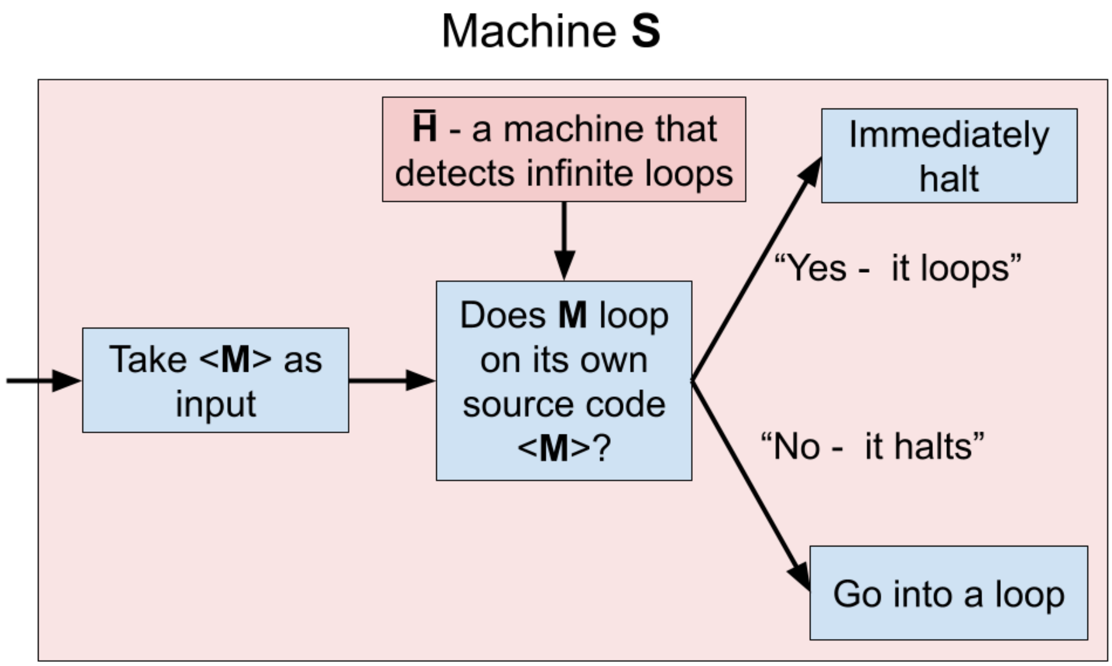

Figure 1: A strange machine

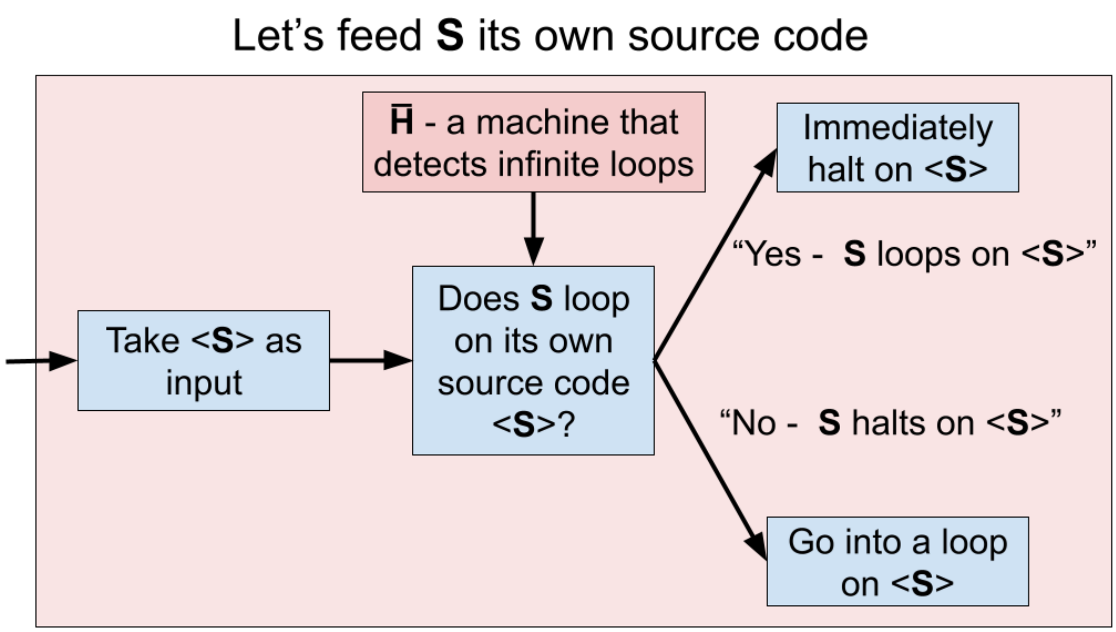

Figure 2: S receives its own source code as input.

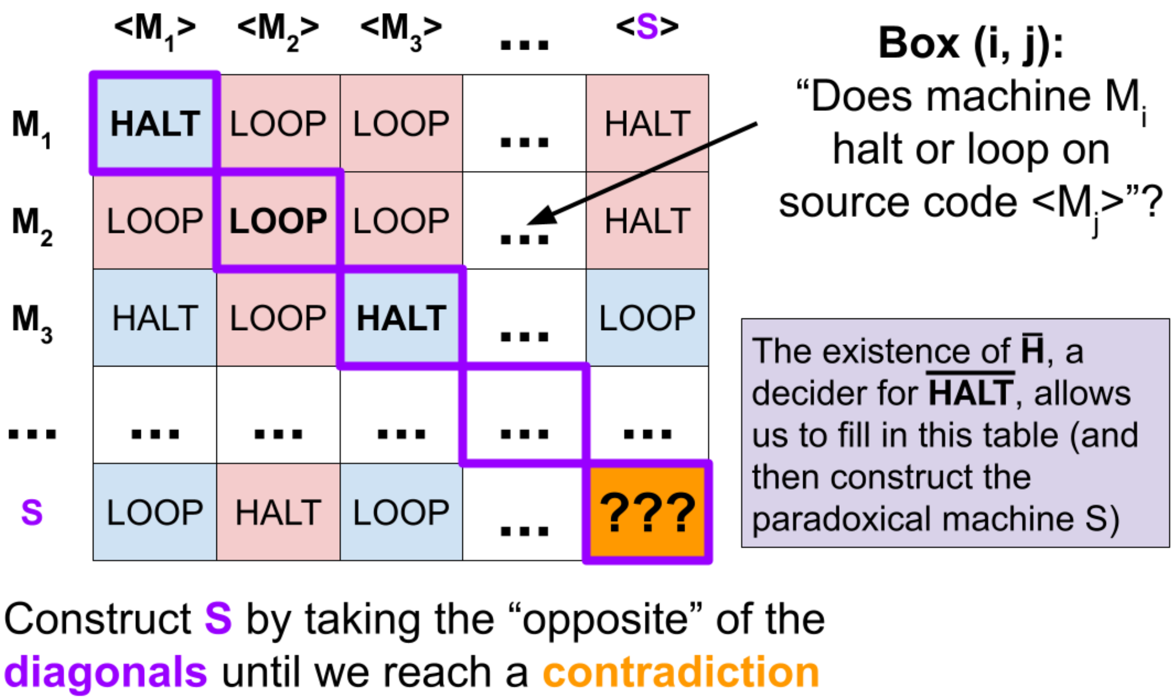

Figure 3: Diagonalizing $\overline{\text{HALT}}$