# W13 L3 the Recursion Theorem

Friday, May 1, 2020 8:31

The Recursion Theorem says that it is possible to have a Turing machine $M$ which does:
$M =$ "on input $w$:
...
...
Calculate $\langle M \rangle$ This line is a valid high-level TM instruction!
..."

We can build up to the proof that this is possible in steps.

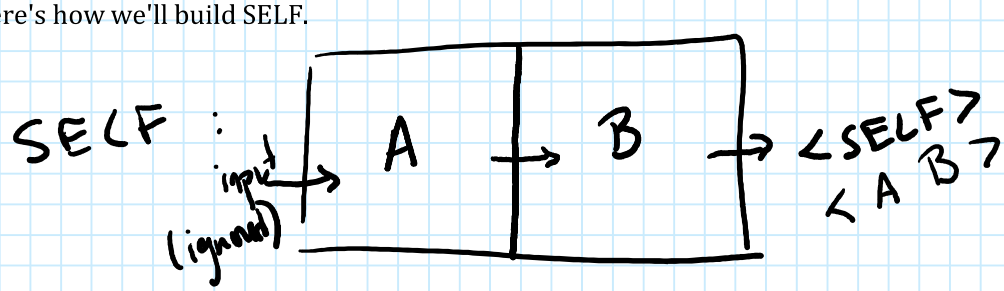
Lemma 1: There is a computable function $q: \Sigma^* \to \Sigma^*$ which on input $w$ outputs $q(w) = \langle P_w \rangle$ the encoding of a Turing machine which does the following:
$P_w =$"on input $x$: ignores $x$, and prints $w$ to the tape, and accepts."

Machine $Q$ which computes function $q$:
$Q =$ "on input $w$: build $P_w$ (as described above), write $\langle P_w \rangle$ to the tape, and accept."

Machine $Q$ is definitely a decider, so $q$ is a computable function.

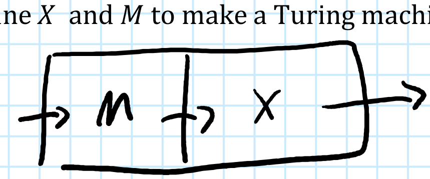
Lemma 2: There is a Turing machine SELF which ignores its input and prints ⟨SELF⟩ to the tape and halts.

Here's how we'll build SELF.




Part A is a piece that just uses lemma 1 to ignore its input, and print the encoding of the machine $P_{\langle B \rangle}$.

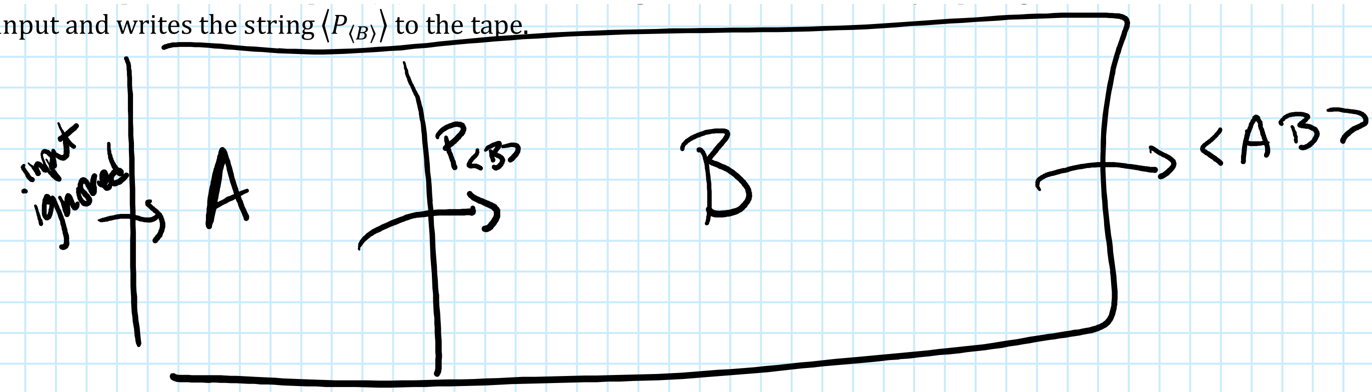
Part B we will construct directly:
$B =$ "on input $\langle M \rangle$ the encoding of a Turing machine:
1. Compute $q(\langle M \rangle) = \langle X \rangle$.
2. Combine $X$ and $M$ to make a Turing machine which does:




3. Print the encoding of this newly-built Turing machine, and then accept."

Overall, if we have a TM which runs $A$ and then $B$,
then in step 1 we will compute $\langle X \rangle$ where $X$ is a Turing machine which, on any input, ignores that input and writes the string $\langle P_{\langle B \rangle} \rangle$ to the tape.

input and writes the string $\langle P_{\langle B \rangle} \rangle$ to the tape.




This means that $\langle X \rangle = \langle A \rangle$.

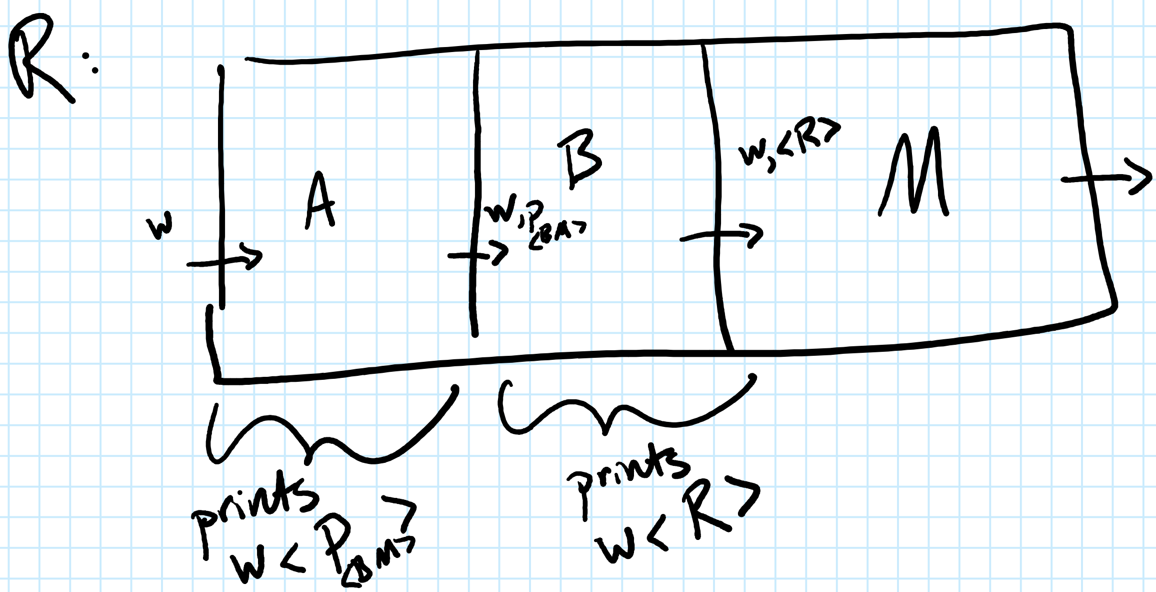
In step 2, we'll combine $A$ with $P_{\langle B \rangle}$ and get a Turing machine which prints the encoding of a Turing machine that does $A$ and then does $B$.

<u>Proof of the recursion theorem:</u>
We want to show how to build a general Turing machine that has "calculate its own description" as a middle step.

$M =$ "on input $w$:
  ...
  calculate $\langle M \rangle$
  ..."

Observe that if $M$ got an additional input which was $\langle M \rangle$ then this step would be easy.

So we'll use lemma 2 to build a machine $R$ which precomputes $\langle M \rangle$ and then pass it to $M$ as an additional input.




$R =$ "on input $w$:
1. Run $A$.
2. Run $B$.
3. Run $M$ on input $w$, using $B$'s output $\langle R \rangle$ on the line where "calculate $\langle M \rangle$" appeared."

This leads to some very nice, short, clean proofs of for example undecidability of $A_{TM}$.