2017-11-30

- A **universal Turing machine (UTM)** is a Turing machine U that gets as input a description of a Turing machine $\lfloor M \rfloor$ and a word w. On such input, it simulates M on w.
- Given a TM $M = \langle \Sigma, \Gamma, Q, q_0, q_a, q_r, \delta \rangle$, there is a TM $M' = \langle \Sigma', \Gamma', Q', q_0', q_a', q_r', \delta' \rangle$ with 3 tapes that can simulate M on a given input w. How?
  - Store input w on tape 1.
  - Write the initial configuration of M on w on tape 2.
  - Write the next configuration on tape 3.
- Define $A_{\text{TM}} = \{(M, w) | w \in L(M)\}$.
- **Theorem**: A_{TM} is recognizable.
  - Let a recognizer for A_{TM} be U := On input (M, w): Run M on w. Accept if M accepts w. Reject if M rejects w. Loop if M does not halt on w.
- **Theorem**: A_{TM} is undecidable.
  - Let a decider for A_{TM} be H := On input (M, w): Run M on w. Accept if M accepts w. Reject if M does not accept w.
  - Construct D: On input $\lfloor M \rfloor$, run H on $(M, \lfloor M \rfloor)$. Accept if H rejects $(M, \lfloor M \rfloor)$. Reject if H accepts $(M, \lfloor M \rfloor)$.
  - Run D on $\lfloor D \rfloor \Rightarrow$ Contradiction.
- **Cantor's diagonalization**: If D is in the figure, a contradiction occurs at ?

$$\begin{array}{c|cccccccc}
 & \langle M_1 \rangle & \langle M_2 \rangle & \langle M_3 \rangle & \langle M_4 \rangle & \cdots & \langle D \rangle & \cdots \\
\hline
M_1 & \underline{accept} & reject & accept & reject & & accept & \\
M_2 & accept & \underline{accept} & accept & accept & \cdots & accept & \cdots \\
M_3 & reject & reject & \underline{reject} & reject & & reject & \\
M_4 & accept & accept & reject & \underline{reject} & & accept & \\
\vdots & & \vdots & & & \ddots & & \\
D & reject & reject & accept & accept & & \underline{\ ?\ } & \\
\vdots & & \vdots & & & & & \ddots
\end{array}$$

- **Theorem**: If L and L' are recognizable, then L is decidable.
  - Let M be a recognizer for L and M' be a recognizer for L.
  - Construct D := On input w: Run M and M' on w in parallel. Accept if M accepts w. Reject if M' accepts w.
- **Corollary**: A'_{TM} is not recognizable.
  - A_{TM} is recognizable but undecidable. Hence, A'_{TM} is not recognizable.
- Define the halting problem $\text{HALT} = \{(M, w) | M \text{ halts on } w\}$.
- **Theorem**: HALT is recognizable.
  - Let a recognizer for HALT be U := On input (M, w): Run M on w. Accept if M halts on w. Loop if M does not halt on w.

- **Theorem**: HALT is undecidable. (Intuition: $A_{\text{TM}} \leq_T$ HALT)
  - Let a decider for HALT be H := On input (M, w): Run M on w. Accept if M halts on w. Reject if M does not halt on w.
  - Construct D := On input (M, w): Run H on (M, w). Reject if H rejects (M, w). Otherwise, run M on w. Accept if M accepts w. Reject if M rejects w.
  - If H decides HALT, then D decides A_{TM}. A_{TM} is undecidable, so HALT is undecidable.
- Define $\text{HALT}_0 = \{\lfloor M \rfloor | M \text{ accepts } \lfloor M \rfloor\}$.
- **Theorem**: HALT_0 is recognizable.
  - Let a recognizer for HALT be U := On input $\lfloor M \rfloor$: Run M on $\lfloor M \rfloor$. Accept if M accepts $\lfloor M \rfloor$. Reject if M rejects $\lfloor M \rfloor$. Loop if M does not halt on $\lfloor M \rfloor$.
- **Theorem**: HALT_0 is undecidable.
  - Let a decider for HALT_0 be H := On input $\lfloor M \rfloor$: Run M on $\lfloor M \rfloor$. Accept if M accepts $\lfloor M \rfloor$. Reject if M does not accept $\lfloor M \rfloor$.
  - Construct D: On input $\lfloor M \rfloor$, run H on $\lfloor M \rfloor$. Accept if H rejects $\lfloor M \rfloor$. Accept if H accept $\lfloor M \rfloor$.
  - Run D on $\lfloor D \rfloor \Rightarrow$ Contradiction.
- Define $\text{HALT}_0' = \{\lfloor M \rfloor | M \text{ does not accept } \lfloor M \rfloor\}$.
- **Corollary**: HALT_0' is not recognizable.
  - HALT_0 is recognizable but undecidable. Hence, HALT_0' is not recognizable.
- **Theorem**: HALT_0' is undecidable.
  - Let a decider for HALT_0' be H := On input $\lfloor M \rfloor$: Run M on $\lfloor M \rfloor$. Accept if M does not accept $\lfloor M \rfloor$. Reject if M accept $\lfloor M \rfloor$.
  - Run H on $\lfloor H \rfloor \Rightarrow$ Contradiction.