The Church-Turing Thesis

- Various definitions of “algorithms” were shown to be equivalent in the 1930s
- **Church-Turing Thesis**: “The intuitive notion of algorithms equals Turing machine algorithms”
  - Turing machines serve as a precise formal model for the intuitive notion of an algorithm
- “Any computation on a digital computer is equivalent to computation in a Turing machine”

Dude, that’s pretty deep…

Closure Properties of Decidable Languages

- Decidable languages are closed under $\cup$, $\circ$, $\ast$, $\cap$, and complement
- Example: Closure under $\cup$
- Need to show that union of 2 decidable L’s is also decidable
  Let M1 be a decider for L1 and M2 a decider for L2
  A decider M for $L_1 \cup L_2$:
  - On input w:
    1. Simulate M1 on w. If M1 accepts, then ACCEPT w. Otherwise, go to step 2 (because M1 has halted and rejected w)
    2. Simulate M2 on w. If M2 accepts, ACCEPT w else REJECT w.
  - M accepts w iff M1 accepts w OR M2 accepts w
  i.e. $L(M) = L_1 \cup L_2$
Closure Properties of Decidable Languages

- Consider the proof for closure under $\cup$
  A decider $M$ for $L_1 \cup L_2$:
  On input $w$:
  1. Simulate $M_1$ on $w$. If $M_1$ accepts, then ACCEPT $w$. Otherwise, go to step 2 (because $M_1$ has halted and rejected $w$)
  2. Simulate $M_2$ on $w$. If $M_2$ accepts, ACCEPT $w$ else REJECT $w$. $M$ accepts $w$ iff $M_1$ accepts $w$ OR $M_2$ accepts $w$
i.e. $L(M) = L_1 \cup L_2$

Will this proof work for showing Turing-recognizable languages are closed under $\cup$? Why/Why not?

Uh...I dunno. Wait, will $M_1$ always halt?!

Closure for Recognizable Languages

- Turing-Recognizable languages are closed under $\cup$, $\circ$, $\ast$, and $\cap$ (but not complement! We will see this in a later lecture)
- Example: Closure under $\cap$
  Let $M_1$ be a TM for $L_1$ and $M_2$ a TM for $L_2$ (both may loop)
  A TM $M$ for $L_1 \cap L_2$:
  On input $w$:
  1. Simulate $M_1$ on $w$. If $M_1$ halts and accepts $w$, go to step 2. If $M_1$ halts and rejects $w$, then REJECT $w$. (If $M_1$ loops, then $M$ will also loop and thus reject $w$)
  2. Simulate $M_2$ on $w$. If $M_2$ halts and accepts, ACCEPT $w$. If $M_2$ halts and rejects, then REJECT $w$. (If $M_2$ loops, then $M$ will also loop and thus reject $w$)
  $M$ accepts $w$ iff $M_1$ accepts $w$ AND $M_2$ accepts $w$ i.e. $L(M) = L_1 \cap L_2$