Recap: Recognizable versus Decidable Languages

- A language $L$ is called **Turing-Recognizable** if there exists a TM $M$ such that $L(M) = L$
  - Note: $M$ need not halt on all inputs but it should halt and accept all and only those strings that are in $L$; it can reject strings by either going to $q_{\text{rej}}$ or by looping forever
- A TM is a **decider** if it halts on all inputs
- A language $L$ is **decidable** if there exists a **decider** $D$ such that $L(D) = L$

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 $M_1$ be a decider for $L_1$ and $M_2$ a decider for $L_2$
  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. $L(M) = L_1 \cup L_2$
Closure Properties

✦ 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 later)

✦ 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. $L(M) = L_1 \cap L_2$
Suppose you want a decider TM for deciding whether a DFA D accepts an input string w

How do we encode a given DFA D as input to a TM?

How does the TM decide if D accepts a given w?

(On-board solution: binary encoding of DFA/CFG/TM and three-tape decider TM)