Decidability

• “the power of algorithms to solve problems.” p 165

• What are the limits of algorithmic solvability?

• How can we tell if two Regular Expressions define the same language?
  – or, can we?

• A language is **decidable** if some TM decides it
Find an algorithm to determine whether a given polynomial Diophantine equation with integer coefficients has an integer solution.

Diophantine equation is an indeterminate polynomial equation that allows the variables to be integers only.

Indeterminate: an equation for which there is an infinite set of solutions.

Algorithm = “a process according to which it can be determined by a finite number of operations”
Hilbert

• Undecidable
• But *is* Turing Recognizable

• Take a question (yes/no)
  – turn it into a language where answer is yes
  – encode in a string
  – build TM
  – If always halts: decidable!
Decidability

• Acceptance Problem (DFA): Does a given DFA, $B$, accept a given string $w$?

• In terms of languages (because we have defined computation as accept/reject a language):
  – $A_{DFA} = \{ <B, w> | B$ is a DFA that accepts $w \}$
  – For ALL input pairs $<B, w>$ can a single TM be constructed that will decide $<B, w> \in A_{DFA}$
    • can we build one TM that will work for all DFAs?
    • is there an *algorithmic* way to solve this problem?
Theorem

• $A_{DFA}$ is decidable
  – given $<B, w>$ we can decide if $<B, w> \in A_{DFA}$ or $<B, w> \not\in A_{DFA}$

• Proof Idea:
  – Use a TM, $M$, to simulate $B$ with input $w$
  – Keep track of current state and current position on the input string
  – Update according to the DFA’s $\delta$
Also...

- $A_{NFA}$ and $A_{Regular \ Expression}$ are also decidable
  - why?
Emptiness testing

• Does a finite automata accept any strings at all?
  – $E_{DFA} = \{ <A> \mid A \text{ is a DFA and } L(A) = \emptyset \}$

• Theorem: $E_{DFA}$ is decidable

• Proof Idea:
  – is it possible to reach an accept state from $q_0$?
Equivalence testing

• Do two DFAs recognize the same language?
  – \( \text{EQ}_{\text{DFA}} = \{ \langle A, B \rangle \mid A \text{ and } B \text{ are DFAs and } L(A) = L(B) \} \)

• Theorem: \( \text{EQ}_{\text{DFA}} \) is decidable
  – Proof:
Question

• Can we tell if two Regular Expressions define the same language?

– why or why not?
CFGs

• $A_{\text{CFG}} = \{<G, w> \mid G \text{ is a CFG that generates } w\}$
• $A_{\text{CFG}}$ is decidable

• Could enumerate all strings produced by $G$: could be infinite, though
• Proof Idea
Equivalence of CFGs

- $EQ_{CFG} = \{<G, H> \mid G \text{ and } H \text{ are CFL and } L(G) = L(H)\}$
  - not decidable