Final Review

Turing Machines

- Deterministic TM
  - Configuration; initial configuration
  - Computation
  - Halting
  - Accepting
  - Rejecting
  - Description
    - Graph
    - Shorthand
  - Deciding = the D class of languages
  - Semideciding = the SD class of languages
  - Computing functions

TM Extensions

- Multiple tapes
  - Multi-tape TM is equivalent to deterministic TM
- Nondeterministic TM
  - Accepting, Rejecting
  - Deciding, Semideciding, Computing functions
  - Nondeterministic TM are equivalent with deterministic TM
    - For deciding, semideciding, computing functions
- One-way tape TM
  - One-way tape TM is equivalent to deterministic TM
  - PDA with two stacks can simulate a TM
- TM can simulate real computers

Universal TM

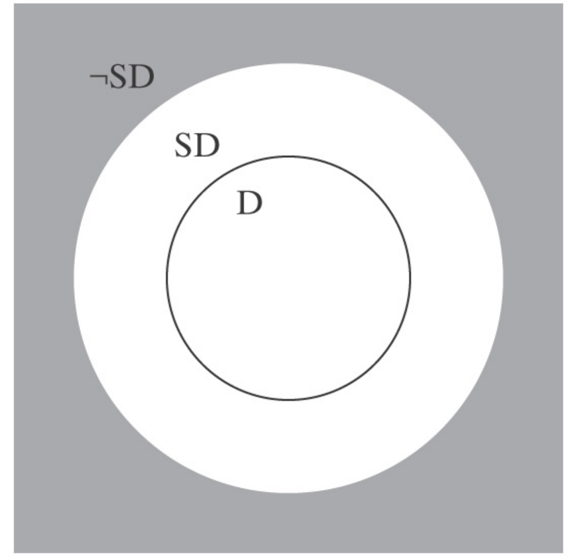
- TM encoding
  - States, tape alphabet, transitions
- Encoding multiple inputs
- Enumerating TMs
- Universal TM
  - Specification
    - On input <*M,w*>, simulate *M* on *w*
  - Construction
- Church-Turing thesis

The Halting Problem

- Halting Problem for TM is in SD \ D
  - Semidecidable (in SD)
  - Not decidable (not in D)
    - Trouble
    - Diagonalization

- $H \in D \Rightarrow D = SD$

D and SD



- $D \subset SD$
- $SD \setminus D \neq \varnothing$ (H is here)
- SD is countable

∀ $\neg SD$ is uncountable

- D is closed under complement
- SD is not closed under complement
- $L \in D$ iff $L, \neg L \in SD$

∀ $\neg H \in \neg SD$

Enumeration

- TM enumerates
- Turing-enumerable language
- $L \in SD$ iff L is Turing enumerable

- TM lexicographically enumerates
- Lexicographically Turing-enumerable language
- $L \in D$ iff L is lexicographically Turing enumerable

Reduction

- Reduction:
  - $L_1 \leq L_2$
    - L_1 is reducible to L_2
    - L_2 is harder than L_1
- Using reduction for undecidability
  - Prove that L_2 is not in D
    - Find suitable L_1 not in D
    - Show that $L_1 \leq L_2$
- $H, H_{\varepsilon}, H_{ANY}, A, A_{\varepsilon}, A_{ANY} \in SD \setminus D$

- Rice's Theorem:
  - Any nontrivial property of SD is undecidable.
- Practical implications on programs

Non-SD languages

- Proving a language L in not in SD
  - $\neg L \in SD \setminus D$
  - Reduction from non-SD language

$\forall$ $\neg H$, $H_{\neg ANY}$, EqTMs, H_{ALL}, A_{ALL}, $TMreg$, $A_{anbn} \notin SD$

Unrestricted grammars

- Unrestricted grammars
- Equivalence with SD
  - Grammar → TM
  - TM → Grammar
- Decision problems
  - Undecidability follows from SD

Non-TM problems

- Post Correspondence Problem (PCP)
- PCP $\in$ SD \ D
  - $L_a \leq$ MPCP $\leq$ PCP

- Problems of context-free languages
  - $CFG_{ALL} \notin D$
    - Reduction from H
    - Computation histories
  - $CFG_{=}$, $PDA_{MIN} \notin D$
    - Reduction from CFG_{ALL}
  - IntEmpty, $CFG_{UNAMBIG} \notin D$
    - Reduction from PCP