Computational Theory

Summary

Curtis Larsen

Utah Tech University—Computing

Fall 2025

Alphabets, Strings, and Languages

- An alphabet is a finite set of symbols: Σ.
- A string, w, is sequence of symbols from Σ.
- The infinite set of all possible strings is Σ^*.
- A string $w \in \Sigma^*$, has finite length.
- A language is a subset of possible strings. $L \subseteq \Sigma^*$.
- A language may be finite or infinite. However $L \cup \overline{L} = \Sigma^*$ and is infinite.
- A string is either in a language, $w \in L$, or not in a language, $w \notin L$ OR $w \in \overline{L}$.

Problems and Computability

- Computable problems can be defined in terms of determining whether or not a string is a member of a language. $w \in L?$
- A computing machine defines a process for determining "$w \in L?$".
- A generator defines a process to generate all $w \in L$.
- Languages are classified based on the computing machine structures needed to compute the membership of a string, or the generator structures needed to generate the members of a language.

Machines and Generators

- Deterministic Finite Automaton
  DFA = $(Q, \Sigma, \delta : Q \times \Sigma \to Q, q_0 \in Q, F \subseteq Q)$
- Nondeterministic Finite Automaton
  NFA = $(Q, \Sigma, \delta : Q \times \Sigma_\varepsilon \to \mathcal{P}(Q), q_0 \in Q, F \subseteq Q)$
- Regular Expression
- Context-free Grammar
  CFG = (V, Σ, R, S)
- Pushdown Automaton
  PDA = $(Q, \Sigma, \Gamma, \delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon), q_0 \in Q, F \subseteq Q)$

Machines and Generators

- Turing Machine (Deterministic)
  TM = $(Q, \Sigma, \Gamma, \delta, q_0 \in Q, q_{accept} \in Q, q_{reject} \in Q)$
  $\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$
- Nondeterministic Turing Machine
  NTM = $(Q, \Sigma, \Gamma, \delta, q_0 \in Q, q_{accept} \in Q, q_{reject} \in Q)$
  $\delta : Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$
- Multitape Turing Machines
  MTTM = $(Q, \Sigma, \Gamma, \delta, q_0 \in Q, q_{accept} \in Q, q_{reject} \in Q)$
  $\delta : Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R, S\}^k$
- Descriptions:
  - Formal
  - Implementation
  - High Level

Language Classes

- Not Turing-recognizable
- Turing-recognizable
- co-Turing-recognizable
- Not co-Turing-recognizable
- Decidable (Turing-decidable)
- Context-free
- Regular

Justify Existence?

- Not Turing-recognizable
- Turing-recognizable
- co-Turing-recognizable
- Not co-Turing-recognizable
- Decidable (Turing-decidable)
- Context-free
- Regular

Prove?

- Not Turing-recognizable
- Turing-recognizable
- co-Turing-recognizable
- Not co-Turing-recognizable
- Decidable (Turing-decidable)
- Context-free
- Regular

Prove Not?

- Not Turing-recognizable
- Turing-recognizable
- co-Turing-recognizable
- Not co-Turing-recognizable
- Decidable (Turing-decidable)
- Context-free
- Regular

Complexity Classes

- $\mathsf{TIME}(t(n))$
- $\mathsf{P} = \bigcup_k \mathsf{TIME}(n^k)$
- $\mathsf{NTIME}(t(n))$
- NP $=$ languages with polynomial time verifiers.
- $\mathsf{NP} = \bigcup_k \mathsf{NTIME}(n^k)$
- NP-HARD
- NP-COMPLETE

Complexity Class Proofs

- Prove membership in class.
- Prove not member of class.

You Did It!

Thanks for a great semester.