Languages and Computation (COMP2049/AE2LAC)

Revision

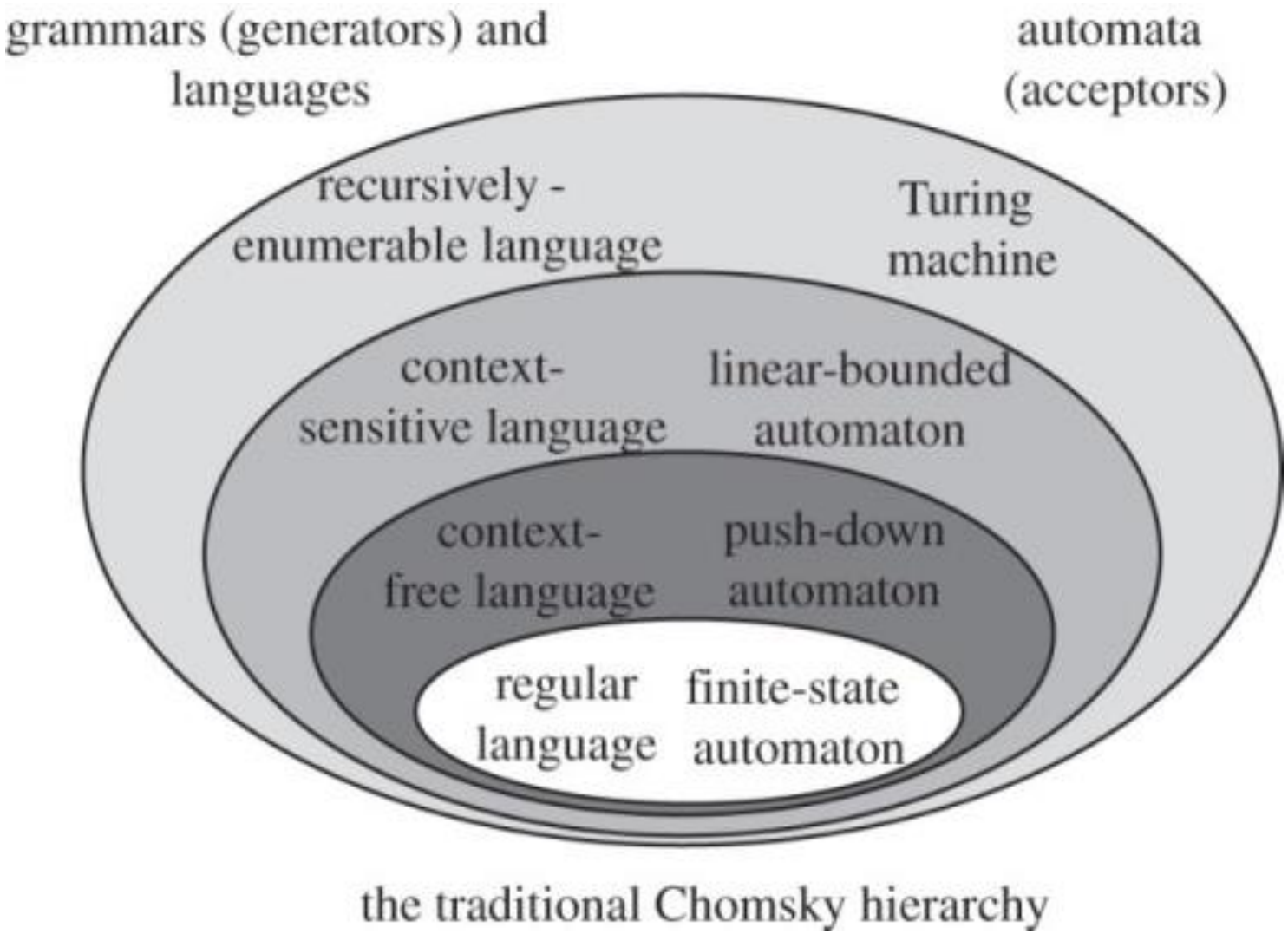
Dr. Tianxiang Cui

tianxiang.cui@nottingham.edu.cn

Examination

- 29th May 2018 (9:30 – 11:30) TB405
  - **Please double check this**
- Answer all THREE questions
- Total marks available: 75
- Regular short answer questions

The Chomsky Hierarchy



the traditional Chomsky hierarchy

Basics

- Symbol
- Word(string)
  - Concatenation of words
- Alphabet
  - Powers of an alphabet
  - Kleene star of an alphabet
- Language
  - Union
  - Concatenation
  - Closure

Deterministic Finite Automata

- The language of a DFA
  - A language is regular *iff* it is the set of strings accepted by some DFA
- Accepting the union/intersection/difference of two Languages
- Language distinguishable
- The Pumping Lemma
  - To prove a language is not regular by showing the language cannot be accepted by a DFA
  - E.g., $L = \{a^i b^i \mid i \geq 0\}$
- Equivalence classes of DFA
- Minimizing the number of states in DFA
  - Table-filling algorithm

Regular Languages

- **Definition**:
- The set of regular languages $\boldsymbol{R}$ over an alphabet $\sum$ is defined recursively as follows:
- *Basis Clause*:
  - The empty language $\varnothing$ is the element of $\boldsymbol{R}$
  - For any symbol $s \in \sum$, the language $\{s\}$ is the element of $\boldsymbol{R}$
- *Inductive Clause*:
  - For every two languages L_1 and L_2 in $\boldsymbol{R}$, the three languages $L_1 \cup L_2$, $L_1 L_2$, and L_1^* are elements of $\boldsymbol{R}$

Regular Expressions

- A regular language has an explicit formula
  - A **regular expression** for a language is a slightly more user-friendly formula
- Parentheses **()** replace curly braces **{}**, and are used only when needed, and the union symbol is replaced by **+**

Regular language	***Regular Expression***
$\emptyset$	$\emptyset$
$\{\varepsilon\}$	ε
$\{a,b\}^*$	$(a+b)^*$
$\{aab\}^*\{a,ab\}$	$(aab)^*(a+ab)$

Exercise

- Write regular expression for the following language over the alphabet $\sum = (a,b,c)$

1. Words that contain at least one a
2. Words that contain exactly one a and one b (but any number of c's)
3. Words such that all c's appear before all a's

Nondeterministic Finite Automata

- We can add nondeterminism to DFA – NFA
  - But this does not increase its power
- Convert NFA into DFA
  - Eliminate ε-transitions
  - Subset construction
- Kleene's Theorem

1. For every alphabet $\sum$, every regular language over $\sum$ can be accepted by a finite automaton
2. For every finite automaton A, the language $L(A)$ is regular

Context-Free Grammars

- Nonterminal, terminal, starting symbol, production
- Rules, called productions, that describe how symbols called nonterminals, can be replaced by nonterminals and terminals until only terminals left

$$\textit{nonterminal} \rightarrow \textit{terminals and nonterminals}$$

- Derivation tree
  - Left-most
  - Right-most
- Ambiguity
- Disambiguating grammars
  - Operator precedence
  - Associativity

Pushdown Automata

- Similar to a finite automaton, but with unlimited memories
- At any time instance, the state of the computation of a PDA is given by an Instantaneous Description (ID)/configuration
  - Describe the sequence of moves for a given input word
- The Language of a PDA
  - Acceptance by final state
  - Acceptance by empty stack

PDA and CFG

- The CFGs and the PDAs describe the same class of languages
- Translating a CFG into a PDA
- Deterministic PDA
  - A PDA that has no choice – $|\delta(q, x, z)| + |\delta(q, \varepsilon, z)| \leq 1$
  - The set of languages accepted by the DPDAs is a strict subset of the languages accepted by PDAs: L(DPDA) $\subset$ L(PDA) = CFL

Turing Machine

- TM – a general model of computation
- Church–Turing Thesis – any real-world computation can be translated into an equivalent computation involving a TM
- Language of a TM – use TM to process a given input string
- Use TM to Compute a function
- Combine TMs
- Variants of TM
  - Multitape Turing Machines
  - Nondeterministic Turing Machines
- Universal Turing Machines
  - Encoding function

Enumerability, Decidability

- Recursively enumerable languages are those that can be accepted by a TM
- Recursive languages are those that can be decided by a TM
- Unrestricted grammars – correspond to recursively enumerable languages
  - Similar to CFGs, except that production rules must have at least one variable on the left hand side
  - Less restrictive than CFG rules
- Context-Sensitive Grammar (CSG): unrestricted grammar in which no production is length-decreasing – Length of RHS of production must be greater than or equal to the length of LHS
- Linear-Bounded Automata (LBA)

Good Luck With Your Exam!