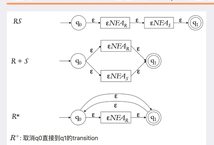
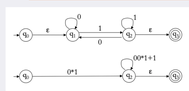
COMP218

Regular Language

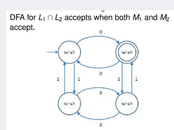
- Set Theory
  - $2^A = \{B | B \subseteq A\} = \{B : B \subseteq A\}$ → powerset: 所有子集的集合
- Language
  - Kleene Star Operation: $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup ... \cup \Sigma^{infinite} = \Sigma^0 \cup \Sigma^+ = \{\epsilon\} \cup \Sigma^+$
- Automata
  - DFA
  - NFA
  - epsilon-NFA
- Automata Conversion
  - epsilon-NFA —> NFA
    - 1. 原accept通过epsilon能到达的也是accept；2. 表格列出每个state经过每个transition及epsilon能到达的所有states。
    - epsilon-closure：CL(A)包括所有A能通过epsilon到达的states，包括A本身(0个epsilon)
  - NFA —> DFA
    - subset construction: 1. 从起始state开始分别写到达的state，出现新的state则续写表格，直到穷尽所有能到达的states；2. 所有包含原accept的也是accept。
- Regular Expression Conversion
  - RE —> epsilon-NFA

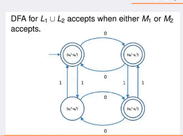


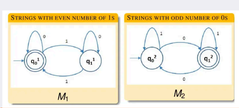
  - epsilon-NFA —> RE
    - epsilon-NFA —> GNFA
      - 1. simplify epsilon-NFA，q0只出不进，唯一的qa只进不出，锁哦于accept都用epsilon连接；2. transition全部写成RE。
    - GNFA —> 2-state GNFA



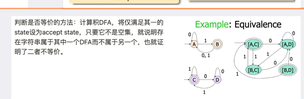
    - 2-state GNFA —> RE
      - 仅剩的q0 —> qa就是RE
- Regular Language
  - Cross-Product Construction



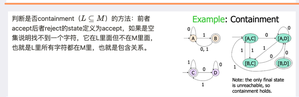




  - Decision
    - equivalence



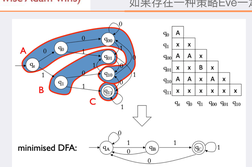
    - containment



  - Non-regularity
    - Closure Property
      - Y: complement，intersection, union，concatenation，Kleene Star，reversal
      - N：duplicate
    - Pumping Lemma
      - 是regularity的必要不充分条件，可以用来证明non-regularity。如果存在一种策略Eve一定会赢，说明L是non-regular

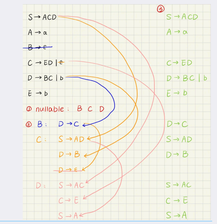
Adam	Eve
1 chooses n	chooses $z \in L$ ($\lvert z\rvert > n$)
2 writes $z = uvwxy$ ($\lvert vw\rvert \le n, \lvert v\rvert \ge 1$)	chooses i
	Eve wins if $uv^iwx^iy \notin L$
	(otherwise Adam wins)

  - DFA Minimization
    - 1. accept所在行全部distinguishable；2. 从上往下数次遍历，如果经过0/1到达distinguishable，那么它们也是distinguishable，直到没有新的出现；3. 连接所有undistinguishable的对，然后相互连接的分为一组；4. 将一组视为整体，重新写transition。

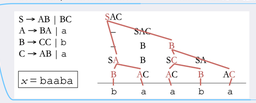


Context-Free Language

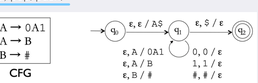
- Context-Free Grammar
  - variable/terminal
  - parse tree
    - 同一个string多个parsing：ambiguous
    - CYK Algorithm
      - remove epsilon
        - 1. 找出nullable variable(如果S是nullable则改写为S' —> S | epsilon)；2. 按照找到的顺序，依次进行移除并写出新的production。



      - remove unit-production
        - cycle: A —> B —> ... —> A, 删去所有production，B等全部改写为A
        - chain：A —> B —> ... —> a, 全部替换成A —> a, B —> a等
      - Chomsky Normal Form
        - 要么A → BC要么A → a (only allow S → epsilon)。每个terminal有单独的variable指定，然后全部改为A —> BC的形式

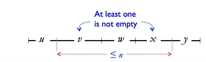


      - Probabilistic CFG —> Probabilistic CYK
- Pushdown Automata
  - non-deterministic，(read, pop/push)
  - CFG —> PDA



- Non-Context-Free
  - Pumping Lemma

Adam	Eve
1 chooses n	chooses $z \in L$ ($\lvert z\rvert > n$)
2 writes $z = uvwxy$	chooses i
($\lvert vwx\rvert \le n, \lvert vx\rvert \ge 1$)	Eve wins if $uv^iwx^iy \notin L$



  - Closure Property
    - Y: union, concatenation, star-operation
    - N: Intersection, complement
    - Regular Closure
      - $RL \cap CFL = CFL$

Turing Machine

- configuration
  - C1 yields C2
  - TM accept x
  - TM halts on x
    - total：halts on every input
      - recursive
- Universal Turing Machine
  - U模拟任何普通图灵机M对输入x执行的操作
  - U recognize L，but not decide L
- Recursive language = Decidable Problem = Total TM
  - DFA accept w
  - NFA accept w — 转换成DFA
  - RE accept w — 转换成NFA
  - DFA is minimal
  - L(D1) = L(D2) — 都minimize
  - L(D) is empty — minimize后跟空集等价
  - CFG generate w — CYK算法寻找parsing
- Recursively Enumerable Language = Recognizable Language = Semi-decidable Problem = Undecidable Problem
  - TM accept w
  - TM accept some input
  - TM halts on w — 用ATM证明
  - PCP：若干tile组合使上下匹配 — tile可以重复使用，无法穷尽所有的可能
  - CFG is ambiguous — 将PCP的tile改写成CFG
- Unrecognizable Language = Undecidable Problem
  - TM reject/loop on w — 补集如果也recursively enumerable那就是recursive了
  - TM accept no input — 补集
  - TM1 and TM2 accept same input — 令M = M1，M2为空，M1与M2是否等价转换成了M是否为空
  - CFG generate all inputs

Chomsky Hierarchy

- expressiveness最强，complexity最高
- recursively enumerable (Unrestricted Grammar)
- recursive
- context-sensitive (LBA)
- context-free (PDA)
- LR (k): unambiguous
- deterministic context-free
- regular (DFA, NFA)