Preliminaries

Mathematical Preliminaries

Sets

集合根据元素数量可分为 finite set 和 infinite set

Universal set: all possible elements

集合的操作：$\cup$ (Union), $\cap$ (Intersection), $-$ (Difference), $\overline{A}$ (Complement)

De Morgan's Law:

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$
$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$

Null set: $\varnothing$

Subset: $A \subseteq B$, 如果 $A \neq B$ 即为 proper subset，记为 $A \subset B$

Disjoint Sets: $A \cap B = \varnothing$

集合中元素的数量称为集合的势（Cardinality），记为 $|A|$

Power sets: a set of sets, 是所有子集的集合，记为 $P(S), 2^S$ ，有 $|2^S| = 2^{|S|}$

Cartesian Product: 笛卡尔积，$|A \times B| = |A| \cdot |B|$

Functions

给定集合 A, B ，函数 f ，函数将每个 A 中的元素映射到至多一个 B 中的元素，记为

$$f : A \to B$$

total function: $A =$ domain

injective function: $\forall a, a' \in A, a \neq a' \to f(a) \neq f(a')$

surjective function: $\forall b \in B, \exists a \in A, f(a) = b$

bijective function: total, injective and surjective

Big O notation: 参见 Asymptotic

Relations

Given two sets, A, B , a **relation** R is any subset of $A \times B$

Equivalence Relations: Reflexive, Symmetric and Transitive

对于一个等价关系 R ，可定义 Equivalence Class

$$[x]_R = \{y : xRy\}$$

两个等价类之间的关系只有相等或 disjoint

对于 A 上的关系 R ，有

- Partial order: Reflexive, Transitive and **Antisymmetric**
- Total order: partial order and $\forall a, b \in A$, either aRb or bRa, also called **linear order**

Graphs

a directive graph $G =< V, E >$

- Walk: a sequence of adjacent edges
- Path: a walk where no edge is repeated
- Simple path: a path where no node is repeated
- Cycle: a walk from a node to itself
- Simple cycle: only the base node is repeated

A **tree** is a directed graph that has no **cycle**.

Proof Techniques

归纳法 or 归谬法

鸽笼原理

String and Languages

Alphabet and string

Alphabet: 符号的有限集合，记为 Σ

String: alphabet 中的符号组成的序列（有穷？）

空串即没有符号的 string ，记为 λ 或是 ϵ

Σ^*: the set of all possible strings from alphabet Σ (including λ)

Σ^+: the set of all possible strings from alphabet Σ except λ , $\Sigma^+ = \Sigma^* - \{\lambda\}$

string 有几种运算，对于 string $w = a_1 a_2 \ldots a_n, v = b_1 b_2 \ldots b_m$

- concatenation: $wv = a_1 a_2 \ldots a_n b_1 b_2 \ldots b_m$
- reverse: $w^R = a_n \ldots a_2 a_1$
- length: The length of a string x is the number of symbols contained in the string x, denoted by $|x|$. $|w| = n, |wv| = |w| + |v|, |\lambda| = 0$
- substring: s is a substring of x if there exist strings y and z such that $x = ysz$ (y, z can be empty string)
- prefix: when $x = sz(y = \epsilon)$, s is called a prefix of x
- suffix: when $x = ys(z = \epsilon)$, s is called a suffix of x
- $w^n = \underbrace{ww \cdots w}_{n}, w^0 = \lambda$

解决字符串相关的问题时，一般根据其**长度**进行分情况讨论

Languages

Language 是 string 的集合，即 Σ^* 的子集

> e.g. $L = \{a^n b^n : n \geqslant 0\}$ ，这个 Language 不能通过正则描述

由于 Language 本质是集合，故集合的操作同样适用于 Language（Union，Intersection，Difference，Complement），$\overline{L} = \Sigma^* - L$

除此之外还有一些 Language 独有的操作

- reverse: $L^R = \{w^R : w \in L\}$
- concatenation: $L_1 L_2 = \{xy : x \in L_1, y \in L_2\}$
- $L^n = \underbrace{LL \cdots L}_{n}, L^0 = \{\lambda\}$
- Kleene Closure: $L^* = \bigcup_{i=0}^{\infty} L^i$
- Positive Closure: $L^+ = \bigcup_{i=1}^{\infty} L^i = L^* - \{\lambda\}$