

* Before the break (recap): Möbius inversion.

Key points:

- μ is defined as the multiplicative inverse of the ~~the~~ edge function $\zeta \in \mathcal{A}(P)$

- We can derive a formula for μ (not proved in class), as follows:

$$\mu([x,x]) = 1 \quad \forall x \in P$$

$$\mu([x,y]) = - \left(\sum_{z \in [x,y)} \mu([x,z]) \right) \quad \text{if } x \not\leq y$$

(x strictly less than y)

- Consequences: $\mu * \zeta = \zeta * \mu = \delta$.

If p & q are vertex functions such that

$p = \zeta * q$, then:

$\mu * p = \mu * \zeta * q$ (multiply by μ on the left)

$\Rightarrow \mu * p = \delta * q = q$

$\Rightarrow \boxed{q = \mu * p}$ [Möbius inversion]

* See worksheet for practice on how to use the inversion formula to obtain the inclusion/exclusion principle

* Summary:

- Set up a subset poset of all the possible "properties"

Eg. $S = \{3, 5, 7\}$ properties = being divisible by 3, 5, 7 respectively.

$P = \mathcal{P}(S)$ with subset relation.

- Set $p(A) = \#$ items satisfying all properties in A and perhaps some properties not in A

(vertex functions)

$q(A) = \#$ items satisfying all properties in A and not satisfying any property of $S \setminus A$

Then observe:

$$p(A) = \sum_{A \subseteq B} q(B) = \sum_{A \subseteq B} \zeta([A,B]) \cdot q(B)$$

$$p(A) = (\zeta * q)(A)$$

$\Rightarrow q(A) = (\mu * p)(A)$.

Möbius inversion

* Finally \therefore If $\mu \in P$ is a SUBSET POSET

then we have an easier formula for μ :

$$\mu([A, B]) = (-1)^{|B \setminus A|} \quad \text{direct formula [proof omitted]}$$

So then the previous equation becomes

$$g(A) = (\mu * p)(A) = \sum_{A \subseteq B} \mu([A, B]) \cdot p(B) \quad \leftarrow \text{one-sided convolution expansion}$$

$$g(A) = \sum_{A \subseteq B} (-1)^{|B \setminus A|} \cdot p(B)$$

(Come ask me in office hrs if you are curious about further applications of Möbius inversion!)

We will not cover them in class due to lack of time

③

* New topic: "Machines"

We'll start with regular expressions

** Background

Def: An alphabet is a finite set of "letters" or "symbols", typically denoted by Σ .

E.g. ① "English" alphabet = $\{a, b, c, \dots, x, y, z\}$

② We'll usually use the binary alphabet:

$$\Sigma = \{0, 1\}$$

Def: A word or string on Σ is a finite ordered list of elements of Σ , not necessarily distinct.

Either: $w = a_1 a_2 \dots a_n$ where $a_i \in \Sigma$, or w is the empty word, denoted ϵ .

* Note: To avoid confusion, we ~~assume~~ assume that $\epsilon \notin \Sigma$ as a symbol (similarly, Σ itself should not be a symbol in Σ , etc.)

Def: A language on Σ is simply a set of words on Σ ; not necessarily finite.

④

Def: Given an alphabet Σ , we denote by Σ^* the set of all possible words on Σ , including the empty word.

Thus a language on Σ is just any subset $L \subseteq \Sigma^*$.

Examples: $\Sigma = \{0, 1\}$

Some words on Σ : $\epsilon, 111, 0, 10, 001100111$, etc.

$\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, \dots\}$

Examples of languages:

$L = \emptyset \leftarrow 0$ elements.

$L = \{\text{all strings starting with } 1\} \leftarrow$ infinite

$L = \{\epsilon\} \leftarrow 1$ element

$L = \{001, 00, 10\}$

$L = \{\epsilon, 10, 11110000111\}$

$L = \{\text{all words in which } 0\text{s and } 1\text{s alternate}\} \leftarrow$ infinite

⑤

* Basic operations on languages & words. (fixed Σ) ⑥

- Concatenation of words.

Given words w_1 and w_2 , their concatenation₁ is the word $w = w_1 w_2$ of w_1 with w_2 .

E.g.: $w_1 = 010$, $w_2 = 110$
 $w_1 w_2 = 010110$

More precisely: $w_1 = a_1 a_2 \dots a_m$
 $w_2 = b_1 b_2 \dots b_n$ } if $a_i, b_j \in \Sigma$

$w_1 w_2 = a_1 a_2 \dots a_m b_1 b_2 \dots b_n$.

* Note: The order matters!

- Concatenation of languages.

If L_1 and L_2 are languages, then their concatenation (in that order) is:

$L_1 \circ L_2 = \{w_1 w_2 \mid w_1 \in L_1 \text{ and } w_2 \in L_2\}$.

E.g.: $L_1 = \{01, 110\}$
 $L_2 = \{110, 11\}$

$L_1 \circ L_2 = \{01110, 0111, 110110, 11011\}$

Note: If $w_1 = \epsilon$, then $w_1 w_2 = w_2$
If $w_2 = \epsilon$ then $w_1 w_2 = w_1$. } concatenation with ϵ keeps the other word as-is

E.g. $L_1 = \{\epsilon, 101\}$, $L_2 = \{00\}$

$$L_1 \circ L_2 = \{\underbrace{00}_{\epsilon}, 10100\}$$

E.g. $L_1 = \{00, 11\}$, $L_2 = \emptyset$.

$$L_1 \circ L_2 = \emptyset.$$

- Union / intersection of languages

$L_1 \cup L_2$, $L_1 \cap L_2$ are just unions / intersections of sets.

E.g. $L_1 = \{\epsilon, 101\}$, $L_2 = \{00\}$

$$L_1 \cup L_2 = \{\epsilon, 101, 00\}$$

$$L_1 \cap L_2 = \emptyset.$$

- Star of a language

If L is a language, then L^* consists of zero or more concatenations of (possibly different) words of L

$$L^* = \{\epsilon\} \cup \{w_1 w_2 \dots w_k \mid w_i \in L\}$$

E.g. $L = \{00, 11\}$

$$L^* = \{\epsilon, 00, 11, 0011, 0000, 1100, 1111, 110011, \dots\}$$