- natural number $\mathbb{N}$ in this course include 0
- Strings are finite i.e $S=101110\ldots$ is never allowed
- if Σ is $\{0,1\}$, then Σ^* is the set of all finite strings over Σ i.e $\{101, 011, 01110, 01110, 1, \ldots\}$
- Σ is finite as well
- The Hamming distance is only defined when $|x|=|y|$ and $d_H(x,y)$ is the number of different character in x and y
- A language is a set of strings over Σ and a subset of Σ^*
- $\bar{L}=\Sigma^* \setminus L$
- Language concatenation : $L_1=\{1,11\}, L_2=\{0,00\}$, then $L_1 \cdot L_2=\{10,100,110,1100\}$
- The empty set $\emptyset$ concatenate with any L is empty set i.e $L\cdot\emptyset=\emptyset\cdot L=\emptyset$
- $|L_1\cdot L_2|$ may not $=|L_1|\times|L_2|$ due to duplication
- $L^n=\begin{cases}\{\varepsilon\} & n=0\\ L\cdot L^{n-1} & n>0\end{cases}$
- $\emptyset^0=\{\varepsilon\}$ but $\emptyset^n=\emptyset$, $n>0$
- $L^*=L^0 \cup L^1 \cup \cdots$ i.e $\varepsilon \in L^* \forall L$ and $\emptyset^*=\{\varepsilon\}$
- format of L, $L=\{w\in\{0,1\}^* \mid w=1^p, p \text{ is prime}\}$
- $M=(Q,\Sigma,\delta,s,F)$, $s\in Q$, $F\subseteq Q$
- $L(M)$ is the Language of M eg. $L(M)=\{w\in\Sigma^* \mid \ldots\}$
- Regular language: can be recognized by some DFSA
- Theorem: For any DFSA M, there is an NFSA that accepts the same language i.e $L(M)=L(N)$ and vice-versa.
- pumping lemma
  1. we can write x as uvw
  2. $|v|>0$
  3. $|uv|<n_0$
  4. $uv^kw \in L, \forall k\in\mathbb{N}$
- Steps to prove a language is not regular
  1. Assume for a contradiction that L is regular
  2. Choose n_0
  3. find $x\in L$ and $|x|>n_0$
  4. write $x=uvw$ and $|v|>n_0$, $|uv|<n_0$
  5. show $uv^kw \notin L$
  6. Contradiction

- given any DFSA M, we can build a regular expression R s.t. $L(R)=L(M)$
- Proving two L are the same we need to show $L_1=L_2$ iff $L_1\subseteq L_2$ and $L_2\subseteq L_1$, apply to regex as well
- if R_1, R_2 differ by only the subexpressions S_1 and S_2, and $S_1\equiv S_2$ then $R_1\equiv R_2$
- GFG $G=(V,\Sigma,P,S)$
- A context-free language is generated by some CFG
- PDA $M=(Q,\Sigma,\Gamma,\delta,s,F)$
- $\forall L$, L can be expressed as $L(G)$ for some CFG $G\Leftrightarrow$ L can be expressed as $L(M)$ for some non-deterministic PDA M
- if a language is an ambiguous CFL, any PDA that recognizes it must be non-deterministic
- $T(X)$= True : we set X to be true
- $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
- $x\rightarrow y\rightarrow z : x\rightarrow(y\rightarrow z)$
- if every row P is 1 and Q is 1, then $P\Rightarrow Q$
- $P\Leftrightarrow Q$
- $P\leftrightarrow Q$ LEQV $P\wedge Q\vee\neg P\wedge\neg Q$
- 2 normal form
  ① Conjuctive Normal Form (CNFs) : $(\ \vee\ )\wedge(\ \vee\ )\wedge(\ \vee\ )$
  ② Disjunctive Normal Form (DNFs): $(\ \wedge\ )\vee(\ \wedge\ )\vee(\ \wedge\ )\cdots$
- complete set: $\{\neg,\vee,\wedge\}, \{\neg,\vee\}, \{\neg,\wedge\}$
- incomplete set: $\{\rightarrow\}, \{\vee,\wedge\}$
  - To prove a set is complete, we use structural induction to show that we can use it to generate every propositional formula.

  Practically, we just need to show our set can generate ¬ and either ∨ or ∧

  To show a set of connectives is not complete, we need to find a property of every formula generated using only these formulas, and then show that this property is not true for every formula.
- P NAND Q : only 1 1 $\Rightarrow$ 0 and $\Rightarrow$ 1 otherwise $\neg x$: x NAND x $x\vee y$: $\neg x$ NAND $\neg y\Leftrightarrow(x$ NAND $x)$ NAND $(y$ NAND $y)$
- P xnor Q : two the same $\Rightarrow$ 0, otherwise $\Rightarrow$ 1 if all input is F, every step is F
- $\Box$: $\Box P=0$ $\neg x: x\rightarrow\Box x$ $x\vee y: (x\rightarrow\Box x)\rightarrow y$

- first-order language $\mathcal{LN}$
  1. infinite # of variable
  2. finite # of constant
  3. finite # of predicates
- Atomic formula e.g. $A(0,1,1)$, $E(y,x)$, $A(0,1,0)$
- Free and bound variables
- Interpretation:
  - Structure: Domain: the value that the constant can take; predicates def; value of constant
  - Valuation: The value that we give to the free variables
- $\neg \exists x, P \Leftrightarrow \forall x, \neg P$
- $\neg \forall x, P \Leftrightarrow \exists x, \neg P$

Now, that's an easy example, so let's see a slightly harder one:

$$P(x) \vee \exists x, ((\forall x Q(x) \wedge R(x)) \rightarrow \neg \exists x, S(x))$$

free

can be rewritten as:

$\Leftrightarrow P(x) \vee \exists x, ((\forall x Q(x) \wedge R(x)) \rightarrow \neg \exists x, S(x))$
$\Leftrightarrow P(x) \vee \exists t, ((\forall y Q(y) \wedge R(t)) \rightarrow \neg \exists z, S(z))$
$\Leftrightarrow P(x) \vee \exists t, ((\forall y Q(y) \wedge R(t)) \rightarrow \forall z, \neg S(z))$
$\Leftrightarrow \exists t, P(x) \vee (\forall y Q(y) \wedge R(t)) \rightarrow \forall z, \neg S(z))$
$\Leftrightarrow \exists t, \exists y (P(x) \vee (Q(y) \wedge R(t)) \rightarrow \forall z, \neg S(z))$
$\Leftrightarrow \exists t \exists y \forall z, (P(x) \vee ((Q(y) \wedge R(t)) \rightarrow \neg S(z))$

note: $(\forall x, a) \rightarrow b$
$\Leftrightarrow \neg (\forall x, a) \vee b$
$\Leftrightarrow (\exists x, \neg a) \vee b$
$\Leftrightarrow \exists x (\neg a) \vee b$
$\Leftrightarrow \exists x\ a \rightarrow b$

$$\forall x, \exists y, (P(x) \vee Q(y))?$$

$\Leftrightarrow \exists y, \forall x, (P(x) \vee Q(y))$

$\forall x, y, \exists z, (C(x) \vee y \vee z) \wedge (\neg x \vee \neg y \vee \neg z)$

$\exists z, \forall x, y, (C(x) \vee y \vee z) \wedge (\neg x \vee \neg y \vee \neg z)$