English Language

- Alphabet $\Sigma=\{a,b,c,d,e,f,g....x,y,z\}$
- A sentence is nothing but string over alphabet.
- There can be infinitely many strings possible. Some of them may belong to the language.
- String 1 : I am a student .
- String 1 belongs to english language
- String 2: student i a am
- String 2 does not belong to english language.
- How do we say a given string is in the language or not? By checking grammar.(Set of rules)
- So we need grammar to define language. Grammar is a language generating device.

C++ Language

- Alphabet Σ={a,b..z,0,1..9,#,;,",(,),+,-,*,/,{,},[,],<,>,=.....}
- A program is nothing but string over alphabet.
- We can write infinitley many programs. Some of these may belong to c++ language.
- Which one?
- Program written using c++ syntax rules/gramnar.
- Compiler checks whether program belongs to c++ language or not.
- We also define language using automata which accepts string belonging to the language.
- Automata is a language accepting device.

Operations on Languages

- Union
- Intersection
- Complement
- Set Difference
- **Concatenation**
- **Closure**

Operation on Languages
Union

- $L_1 \cup L_2$ is defined as a set consiting of strings which are either in language L_1 or Language L_2.
- $L_1 \cup L_2 = \{ x \mid x \in L_1 \text{ or } x \in L_2 \}$
- $L_1 = \{a,ab,aab,aaab\}$
- $L_2 = \{a,aba,aabaa\}$
- $L_1 \cup L_2 = \{a,ab,aab,aaab,aba,aabaa\}$

Operation on Languages
Intersection

- $L_1 \cap L_2$ is defined as a set consiting of strings which are both in language L_1 and L_2.
- $L_1 \cap L_2 = \{ x \mid x \in L_1 \text{ and } x \in L_2 \}$
- $L_1 = \{a, ab, aab, aaab\}$
- $L_2 = \{a, aba, aabaa\}$
- $L_1 \cap L_2 = \{a\}$

Operation on Languages
Complement

- L’ is defined as a set consiting of strings which are not in language L
- $L' = \{ x \mid x \notin L \}$ or $L' = \Sigma^* - L$
- $L=\{ w \, \varepsilon \, \Sigma^* \mid$ w begins with a ‘0’$\}$
- $L'=\{ w \, \varepsilon \, \Sigma^* \mid$ w does not begin with a ‘0’$\}$

Operation on Languages
Difference

- $L_1 - L_2$ is defined as a set consiting of strings which are in language L_1 but not in L_2.
- $L_1 - L_2 = \{ x \mid x \in L_1 \text{ and } x \notin L_2 \}$
- $L_1 = \{a, ab, aab, aaab\}$
- $L_2 = \{a, aba, aabaa\}$
- $L_1 - L_2 = \{ab, aab, aaab\}$

Operation on Languages Concatenation

- L_1L_2 is defined as a set consiting of strings xy such that x belongs to Language L_1 and y belongs to Language $L_{2.}$
- $L_1L_2 = \{ xy \mid x \varepsilon L_1 \text{ and } y \varepsilon L_2 \}$
- $L_1 = \{a,b\}$
- $L_2 = \{c,d\}$
- $L_1L_2 = \{a,b\}\{c,d\} = \{ac,ad,bc,bd\}$
- $L_2L_1 = \{c,d\} \{a,b\} = \{ca,cb,da,db\}$
- Note $L_1L_2 \neq L_2L_1$ in general but
- $L_1L_2 = L_2L_1$ if $L_1 = L_2$

Operation on Languages Power of alphabet Σ^i as Concatenation

- Power of an alphabet Σ^i may be defined as concatenation of Σ
- $\Sigma=\{0,1\}$
- $\Sigma^0=\{\varepsilon\}$: set of strings of length 0
- $\Sigma^1=\{0,1\}$: set of strings of length $1=\Sigma$
- $\Sigma^2=\{00,01,10,11\}$: set of strings of length $2=\{0,1\}\{0,1\}=\Sigma\Sigma$
- $\Sigma^3=\{000,001,010,011,100,101,110,111\}$: set of strings of length $3=\Sigma\Sigma\Sigma=\Sigma^2\Sigma$ or $\Sigma\Sigma^2$
- $\Sigma^4=\Sigma^3\Sigma$ or $\Sigma\Sigma^3$, $\Sigma^5=\Sigma^4\Sigma$ or $\Sigma\Sigma^4$ and so on.....

Operation on Languages Closure

- Kleene closure L^*
- Positive closure L^+