Elements of Theory:

- **Char**: A symbol that can be told apart from other symbols. For example, 'a', 'i', Spanish 'ch', 'rr', Chinese 'chars' can even be composed of multiple other chars.

- **Alphabet** = a (finite) set of symbols. For example, the binary alphabet is \{0, 1\} or \{a, b\}.

- **ASCII alphabet**: \{32 control codes, \^, " # ... A ... Z, a ... z...\} mapped over \{0, 1\}^8 i.e. 00000000 -- 11111111

- **String** = a (finite) list of char, repetitions allowed. A string = list\langle\text{char}\rangle

Main operation: for concatenation, eg 'ab', 'ba' = 'abba'.

- **empty string** = "\" We will denote it by \(\varepsilon\) (alternative \(\lambda\) / \(\text{epsilon}\) / \(\text{lambda}\))

For any string \(x\), \(\varepsilon \cdot x = x\cdot \varepsilon = x\).

- **Language** = a set of strings.

Example: \(\emptyset\) is the empty language. It is not the same as \(\{\varepsilon\}\).

- **Often infinite**!

- **Language** = set\langle\text{string}\rangle
Languages have associated operations too:

- All set operations $\cup$, $\cap$, $\sim$ (complementation).
- **Concatenation of languages**
  
  $A \cdot B = \{ x \cdot y : x \in A \text{ and } y \in B \}$

**Example:**

- $A = \{ "01", "010" \}$
- $B = \{ "11", "011" \}$

  $A \cdot B = \{ 01 \cdot 11, 01 \cdot 011, 010 \cdot 11, 010 \cdot 011 \}$

  $= \{ 0111, 011011, 010011 \}$

  Different from **Cartesian Product** $A \times B$.

  Added: $A \times B = \{ (x, y) : x \in A \text{ and } y \in B \}$.

  In this case, $A \times B = \{ (01, 11), (01, 011), (010, 11), (010, 011) \}$

  As ordered pairs, these remain different.

  Always $|A \times B| = |A| \cdot |B|$ but as above, $|A \cdot B| < |A| \cdot |B|$ can happen (when $\cdot$ might confuse with length of a string, we can write $\|A\| \text{ for cardinality}$ instead.)