Languages are sets of strings. From an OOP perspective:

**Basic type:** char character over an alphabet \( \Sigma \).

We will think of string as a basic type, but it’s “really” capital sigma.

\[
\text{String} = \text{list of char} \quad \text{A string is a (finite)} \quad \text{list of chars (in } \Sigma \).
\]

*Example:*

\[
\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \quad x = \langle 2, 7 \rangle \text{ is the list } (2, 7)
\]

13,596 is not a string over \( \Sigma \), but \( 13 \) over \( \Sigma' = \Sigma \cup \{, \} \).

Often we can say \( \Sigma \) is “big enough for all the chars we’ll use.”

Or we can assume \( \Sigma = \{0, 1, \} \) (or \( \{a, b, \} \))

or \( \{a, b, \} \).

**ASCII** = \{96 printable vs typewriter chars

\begin{itemize}
  \item 82 control codes
  \item 128 other “upper ASCII” chars
\end{itemize}

We can re-code a char in ASCII as a byte in \{0,1\}.

**UNICODE** uses 16 bits per “wide” char. UTF-8 blends ASCII & UNICODE

\[
\text{language} = \text{set of string} = \text{set of list of char}
\]

“Second Order” Object

\[
\text{Class} = \text{set of language}
\]

“Third Order” (Star Wars 17)

“Will feature Jedi vs. the Third Order.”
Examples of Languages:

\[ \Sigma = \{0, 1\} : \quad L_0 = \{10, 11, 101, 111, 1011, 1101\} \]

\[ \Omega = \{0, \ldots, 9\} : \quad L_{\Omega} = \{2, 3, 5, 7, 11, 13\} \]

This language expresses the property of being a prime number \( \leq 13 \), but the idea of "prime number" is simpler and yields an infinite language.

- **Primes:** \[ \{ 3, 10, 11, 101, 111, 1010, 1101, 10001, \ldots \} \]

Numbers and Strings (and languages) will often be interchangeable in 1st order.

Natural Numbers \( \mathbb{N} = 0, 1, 2, 3, 4, \ldots \) \( (\mathbb{N}^+ \) starts with 1).

Standard Binary Notation: \[ 0, 1, 10, 11, 100, 101, 110, 111, 1000, \ldots \]

Kids strings, other than 0, that have leading 0s (standard order on \( \Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, \ldots\} \)

Under the latter, Primes could be encoded as \[ \{0, 1, 01, 11, 011, \ldots\} \]

To represent Rational Numbers like \( 17/355 \), use strings like "17/355" or "10001/101011111"

Encoding other objects as Strings. E.g. Graphs.

A graph \( G = (V, E) \) consists of a set \( V \) of nodes and a set \( E \subseteq V \times V \) of edges.

A typical edge looks like \( (a, b) \) where \( a \in V \) and \( b \in V \). \( E \) as a relation \( G \) is undirected if \( E \) is symmetric, i.e. \( (a,b) \in E \Rightarrow (b,a) \in E \).
By convention, undirected graphs generally don't have self-loops \((a,a) \in E\), but directed graphs often do. E.g., \(G_2\):

We can also label edges by changing e.g., \((1,3)\) to \((1, L, 3)\) where \(L\) is the label.

Using chars (and later E/S) as labels defines finite automata (ch. 1).

Example: \(G =\)

To encode \(G\) as a string, we can use an

**Edge List**

If I want a binary string encoding of \(G\), I could convert \(\text{ASCII}\) to bytes, or I could do an adjacency matrix:

Finally, "unroll" \(A_G\) (in row-major order) as the

We can now specify properties of graphs via languages of strings, e.g.

\[ L_\Delta = \{ X_G : G \text{ (an undirected graph) has at least one triangle}\} \]

\(G_{above}\) is in \(L_\Delta\).
Operations on Languages: Some are for sets in general & others are specific to strings.

**Cartesian Product:** $A \times B = \{ (a, b) : a \in A \land b \in B \}$

Example:

$A = \{ 0, 01, 10 \}$  $B = A$  $A \times A = \{ (a, b) : a \in A \land b \in A \}$

$A \times A = \{ (0, 0), (0, 01), (0, 10), (01, 0), (01, 01), (01, 10), (10, 0), (10, 01), (10, 10) \}$

Always, $|A \times B| = |A| \cdot |B|$ so $|A \times A|$ here $= 3 \times 3 = 9$.

Specific to set of strings: **Concatenation of Languages**

Defn:

$A \cdot B = \{ x \cdot y : x \in A \land y \in B \}$.

So $A \cdot A = \{ x \cdot y : x \in A \land y \in A \}$

For the above $A$, $A \cdot A = \{ 0.0, 0.01, 0.10, 01.0, 01.01, 01.10, 10.0, 10.01, 10.10 \}$. $A \times A$ but:

$\{ 00, 001, 010, 0101, 0110, 100, 1001, 1010 \}$

Unlike lists, sets don't allow repeats. So as a language, it is $\{ 00, 001, 010, 0101, 0110, 100, 1001, 1010 \}$, $|C| = 8$.

The lecture was also meant to include these other set operations:

Union: $A \cup B = \{ x : x \in A \text{ or } x \in B \}$

Difference of sets: $A \setminus B = \{ x : x \in A \text{ but } x \notin B \}$

Intersection: $A \cap B = \{ x : x \in A \text{ and } x \in B \}$

Symmetric Difference: $A \Delta B = \{ x : x \in A \text{ xor } x \in B \}$

The text uses — for difference and union — $A \setminus B$ or $\cup$ or $\sqcup$ and $A \cup B$.