**Simple computers**

- The basic question: What is a computer?
- We’ll start by looking at very simple computers.
  - Input: anything
  - Output: YES or NO
    (ACCEPT or REJECT)
- Each such computer **recognizes a language**.

---

**Languages**

- **What is a “language”?**
  - [Merriam-Webster] “a systematic means of communicating ideas or feelings by the use of conventionalized signs…”
  - Examples: English, Spanish, Mandarin, Swahili, …
  - More examples: Pascal, Scheme, C, Perl, …
- **For our purposes, a language is a set of strings.**
  - English = (“Hello.”, “Come here.”, “Programming is fun.”, “In God we trust.”, “Don’t tase, bro!”, …)
  - Perl = \{p : p is a syntactically correct Perl program\}
  - L = \{“a”, “b”, “aa”, “bb”, “ab”, “ba”\}

---

**Precise definitions**

- A language is a set of **strings**.
- A string is a sequence of characters/symbols.
  - Note: sequence (ordered) versus set (unordered)
  - When writing a string, omit commas.
  - “abc” instead of (a,b,c).
- A character/symbol is an element of an **alphabet**.
- An alphabet is any **nonempty finite** set.
  - Alphabets usually denoted by Greek caps: \(\Sigma\) or \(\Gamma\).
### Operations on strings

- Let \( w, x \) be strings over alphabet \( \Sigma \).
  - \( |w| = \text{length of } w = \text{number of chars in } w \).  
  - \( wx = w \circ x = \text{concatenation: } w \text{ followed by } x \).
  - \( w^R = \text{reverse of } w \).
  - Special: \( \varepsilon = \text{empty string} \).

### Terminology

- We speak of a string or language *over* a certain alphabet.
  - Thus, “abracadabra” is a string over the alphabet \{a, b, c,..., y, z\}.
  - “καλημερα” is a string over the alphabet \{α, β, γ,..., ι, ω\}.
  - \{a, b, aa, bb, ab, ba\} is a language over the alphabet \{a, b\}. It’s also a language over the alphabet \{a,..., z\}.

### Prefixes and suffixes

- Any “initial segment” of a string \( w \) is called a *prefix* of \( w \).
- Let us make this precise and mathematical: “initial segment” is imprecise, though intuitive.
- **Definition:** The string \( x \) is a prefix of the string \( w \) iff \( w = xy \) for some string \( y \).
- Even more precisely: … iff \( \exists \ y \ (w = xy) \).
- How do you think we should define suffixes?
Practice

- Is $c$ a prefix of $caccba$?
  - Yes, if $x = c$, take $y = accba$ to get $caccba = xy$.
- Is $caccba$ a prefix of $caccba$?
  - Yes, if $x = caccba$, take $y = e$ to get $caccba = xy$.
- Is $\varepsilon$ a prefix of $caccba$?
  - Yes, if $x = \varepsilon$, take $y = caccba$ to get $caccba = xy$.
  - Same reasoning shows that $\varepsilon$ is a prefix of any string.
- Is $\varepsilon$ a prefix of $\varepsilon$?
  - Sure!

More practice

- If $x$ is a prefix of $w$, is $x^R$ a prefix of $w^R$?
  - No; take $w = abcdef$, $x = ab$
  - Then $w^R = fedcba$, $x^R = ba$; not a prefix! But...
  - It seems $x^R$ is a suffix of $w^R$. Can you prove this?
- Can you express $(xy)^R$ in terms of $x^R$ and $y^R$?
  - $(xy)^R = y^R x^R$
- If $x$ is a string what do $x^2$, $x^3$, etc. denote?
  - $x^2 = xx$, $x^3 = xxx$, etc. Thus, $(abc)^2 = abcabc$. 

On to Finite Automata