Notation and Basic Concepts

Jeffrey Shallit
School of Computer Science
University of Waterloo
shallit@uwaterloo.ca
https://cs.uwaterloo.ca/~shallit
Basic objects

Here are some of the objects we’ll study in this course:

- A **symbol**: a letter, like `a` or a number like `0`. This is an “atomic” notion that we won’t define further.

- A **string** or **word**: a finite ordered list of symbols, like `boat` or `10110`.
  - Strings will always be **finite** in this course.
  - A special string is `ε`, the empty string. (Some books write `λ` or `Λ` instead.)
  - The length of a string `x` is written `|x|`.

- An **alphabet** is a finite nonempty set of symbols.
  - Examples of alphabets: `{0, 1}` and `{a}`
  - We speak of a string being defined **over** an alphabet.
Languages

- A *language* is a set of strings, like \{\texttt{over, under}\} or \{0, 01, 010, 0101, 01010, \ldots\}.

  - Languages can be finite (have only finitely many elements) or infinite (have infinitely many elements).
  - The empty set is denoted \(\emptyset\).
  - Don’t confuse the empty string with the empty set!
Basic operations on strings

- **Concatenation**: join two strings by juxtaposing them.
  - Example: `house` concatenated with `boat` gives the word `houseboat`.
  - Written like multiplication: either with a · symbol `x · y` or just by writing one next to the other: `xy`.
  - Concatenation obeys the associative law: `(xy)z = x(yz)`.
  - The identity element for concatenation is the empty string; that is, `x · ϵ = ϵ · x = x` for all strings `x`.
  - In general, concatenation does *not* obey the commutative law: `house · boat ≠ boat · house`.
Since concatenation is like multiplication, we can raise a string to a power by repeated concatenation.

$x^n$ is our shorthand for

\[ xx \cdots x. \]

Example: $(\text{mur})^2 = \text{murmur}$.

Note: $x^0 = \epsilon$, the empty string.

Power obeys the usual rule $x^{m+n} = x^m x^n$.

But doesn’t obey the rule $(xy)^n = x^n y^n$ in general.

The formal (recursive) definition of power:

\[ x^0 = \epsilon \]

\[ x^n = x \cdot x^{n-1} \text{ for } n \geq 1 \]
Prefix, suffix, substring

- We say $x$ is a **prefix** of $z$ if $z = xy$ for some string $y$.
  - *dog* is a prefix of *dogmatic*

- We say $x$ is a **suffix** of $z$ if $z = wx$ for some string $w$.
  - *ape* is a suffix of *grape*

- We say $x$ is a **substring** of $z$ if $z = wxy$ for some strings $w, y$.
  - *cat* is a substring of *concatenate*
  - Warning! In other books and papers, **substring** is sometimes called **subword** or **factor** and sometimes means something entirely different...

- In all three cases, $x, y, w, z$ are all allowed to be empty.
  - A general principle: the empty string is a string like any other string and is usually not treated differently
Throughout the course, I’ll try to use consistent notation.

- Single symbols will be denoted by letters at the front of the alphabet, like \( a, b, c \).
- Strings (words) will be denoted by letters near the end of the alphabet, like \( s, t, u, v, w, x, y, z \).
- Natural numbers \( \{0, 1, \ldots \} \) will be denoted by \( i, j, k, l, m, n \).
- Alphabets will be denoted by upper-case Greek letters, like \( \Sigma \) and \( \Delta \) and \( \Gamma \).
- Languages will be denoted by upper-case Roman letters, like \( L, A, B \).

So just by looking at a line of text, you can usually tell what kinds of objects are being discussed.
More operations on strings

- **Reversal**: Reverses the order of letters in a string.
  - Written $x^R$
  - Example: $(\text{stressed})^R = \text{desserts}$

- Formal (recursive) definition:
  - $\epsilon^R = \epsilon$
  - If $x = za$, for $a$ a single symbol, then $x^R = a \cdot z^R$

Here’s one of the most basic results about reversal of strings: $(xy)^R = y^Rx^R$ for all strings $x, y$.

How can we prove this? As we saw, the basic tool for proving things like this is **induction on the length of the string**. But which string?