

Introduction to Computer Science Theory

Alvin Lin

August 2017 - December 2017

What is theory?

- Greek *theoria*: “a looking at, viewing, beholding”
- National Academy of Sciences: “... a well-substantiated explanation of some aspect of the natural world, based on a body of facts that have been repeatedly confirmed through observation and experiment.
- A simple explanatory framework

Criteria for a Good Theory

- Makes falsifiable predictions.
- Well-supported by many independent strands of evidence.
- Consistent with pre-existing theories.
- Can be adapted and modified to account for new evidence.
- Parsimonious (Occam’s razor)

Automata Theory

- The foundational theory of computer science.
- Based on simple, abstract, and mathematically well-defined machines.
- Uses proofs, theorems, et cetera to extend the theory.

Why should you care about the theory of computing?

- Any scientific inquiry should be guided by and phrased in terms of theory.
- To gain an understanding of what computers can and cannot do, akin to the role physics plays in engineering.
- To learn how to think and express yourself formally and abstractly, using English, not a programming language. This is use when communicating with clients and other nontechnical people, writing patents or scientific papers, or porting algorithms to other systems.
- Programming is itself a formal problem. Understanding more broadly how formal methods work results in better programming.

Sets

A collection of objects (a.k.a. hashes, associated arrays). Examples:

- The set of natural numbers

Define \mathbb{N} :

Basis : $1 \in \mathbb{N}$

Recursion : $\forall n \in \mathbb{N}(n + 1 \in \mathbb{N})$

- $\{1, -1, 2, -2, \dots, 345, -345\}$
- $\{x \in \mathbb{N} \mid x > 517\}$
- 2^S where S is a set.

$$2^{\{0,1,2\}} = \{\emptyset, \{0\}, \{1\}, \{2\}, \{0, 1\}, \{0, 2\}, \{1, 2\}, \{0, 1, 2\}\}$$

- $S = \{1, \{1, \{1, \{\dots\}, \dots\}\}\}$ is a set that contains itself.
- Is $\{S \mid S \text{ is a set and } S \in S\}$ a set? Yes.
- Is $\{S \mid S \text{ is a set and } S \notin S\}$ a set? No.

The complement of a set is denoted \bar{A} or as A' .

$$\bar{A} = \mathbb{U} - A$$

Languages

A **language** is a set of strings involving symbols from some **alphabet**. An **alphabet** is a finite set of indivisible objects, usually denoted by Σ . Examples:

- $\{0, 1\}$
- $\{a, b, c, d, e, \dots, x, y, z\}$

A string over Σ is finite (possibly empty) sequence of elements of Σ . Strings of alpha $\{0, 1\}$ are:

- ϵ
- 0
- 1
- 10010011101
- 00000

ϵ denotes the **null string**: the empty sequence of elements of Σ . Sometimes this is denoted \wedge or λ . The following strings are not strings over $\{0, 1\}$:

- 1021
- 1111111111...

If x is a string of Σ , $|x|$ denotes the length of x , the number of alphabet symbols in the string.

- $|1| = 1$
- $|10001010101| = 11$
- $|00000| = 5$
- $|\epsilon| = 0$

The set of all strings over Σ is denoted by Σ^* . For example:

$$\{0, 1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, \dots\}$$

A language of Σ is a subset of Σ^* .

$$\begin{aligned} \{a, b, c\}^* &= \{\epsilon, a, b, c, aa, ab, ac, ba, bb, bc, \dots\} \\ L^* &= \{\epsilon\} \cup L \cup L \circ L \cup L \circ L \circ L \cup \dots \\ &= \cup_{i=0}^{\infty} L^i \text{ where } L^0 = \{\epsilon\} \end{aligned}$$

This operation is known as the Kleene star.

A language over Σ is a subset of Σ^* . Example:

- $\{1, 010, 111000, 110\}$
- \emptyset
- ϵ
- $\{0, 1\}^*$
- $\{x \in \{0, 1\}^* \mid |x| \text{ is even}\}$

Because languages are sets of strings, we can construct new languages using set operations (like union, intersection, and complementation).

$$L_1, L_2$$

$$L_1 \cup L_2 \quad L_1 \cap L_2$$

If L is a language of Σ , then \bar{L} (the complement of L) is $\Sigma^* - L$.

$$L = \{0, 00, 000, 0000, \dots\}$$

If $\Sigma = \{0\}$:

$$\bar{L} = \{\epsilon\}$$

If $\Sigma = \{1, 0\}$:

$$\bar{L} = \{\epsilon, 1, 01, 10, 11, \dots\}$$

Operations on Strings

Let $x, y \in \Sigma^*$:

- xy is the **concatenation** of x and y . For example, if $x = ab$ and $y = aabb$, then $xy = abaabb$. For all strings x , $x\epsilon = \epsilon x = x$.

- For i an integer, x^i is the concatenation of i x 's. For example, if $x = abb$, then $x^3 = abbabbabb$. For all strings x , $x^0 = \epsilon$.
- x is a **substring** of y if and only if there exist $w, z \in \Sigma^*$ such that $wxz = y$.
- x is a **prefix** of y if and only if there exists a $z \in \Sigma^*$ such that $xz = y$.
- x is a **suffix** of y if and only if there exists a $z \in \Sigma^*$ such that $zx = y$.

Operations on Languages

Operations on strings can be extended to operations on languages. Let $L, L_1, L_2 \subseteq \Sigma^*$:

- $L_1 \circ L_2 = \{xy \mid x \in L_1, y \in L_2\}$ is the concatenation of languages L_1 and L_2 .
What is $\{ab, a\} \circ \{a, ba, aaaa\}$

$$\{aba, abba, abaaaa, aa, aba, aaaaa\}$$

- For i an integer, L^i is the concatenation of i L 's.
- $L^* = \cup_{i=0}^{\infty} L^i$ and $L^+ = \cup_{i=1}^{\infty} L^i$.

You can find all my notes at <http://omgimanerd.tech/notes>. If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech