1.4
Languages

Languages

Definition

A language L is a set of strings over Σ. In other words $L \subseteq \Sigma^*$.

Standard set operations apply to languages.

- For languages A, B the concatenation of A, B is $AB = \{xy \mid x \in A, y \in B\}$.
- For languages A, B, their union is $A \cup B$, intersection is $A \cap B$, and difference is $A \setminus B$ (also written as $A - B$).
- For language $A \subseteq \Sigma^*$ the complement of A is $\bar{A} = \Sigma^* \setminus A$.

Languages

Definition

A language L is a set of strings over Σ. In other words $L \subseteq \Sigma^*$.

Standard set operations apply to languages.

- For languages A, B the concatenation of A, B is $AB = \{xy \mid x \in A, y \in B\}$.
- For languages A, B, their union is $A \cup B$, intersection is $A \cap B$, and difference is $A \setminus B$ (also written as $A - B$).
- For language $A \subseteq \Sigma^*$ the complement of A is $\bar{A} = \Sigma^* \setminus A$.

Exponentiation, Kleene star etc

Definition

For a language $L \subseteq \Sigma^*$ and $n \in \mathbb{N}$, define L^n inductively as follows.

$$L^n = \begin{cases} \{\epsilon\} & \text{if } n = 0 \\ L \bullet (L^{n-1}) & \text{if } n > 0 \end{cases}$$

And define $L^* = \cup_{n \geq 0} L^n$, and $L^+ = \cup_{n \geq 1} L^n$

Exercise

Problem

Answer the following questions taking $A, B \subseteq \{0,1\}^*$.

1. *Is* $\epsilon = \{\epsilon\}$? *Is* $\emptyset = \{\epsilon\}$?
2. *What is* $\emptyset \bullet A$? *What is* $A \bullet \emptyset$?
3. *What is* $\{\epsilon\} \bullet A$? *And* $A \bullet \{\epsilon\}$?
4. *If* $|A| = 2$ *and* $|B| = 3$, *what is* $|A \bullet B|$?

Exercise

Problem

Consider languages over $\Sigma = \{0, 1\}$.

1. *What is* $\emptyset^0$*?*
2. *If* $|L| = 2$*, then what is* $|L^4|$*?*
3. *What is* $\emptyset^*$*,* $\{\epsilon\}^*$*,* ϵ^**?*
4. *For what* L *is* L^* *finite?*
5. *What is* $\emptyset^+$*,* $\{\epsilon\}^+$*,* ϵ^+*?*

Languages and Computation

What are we interested in computing? Mostly functions.

Informal definition: An algorithm $\mathcal{A}$ computes a function $f : \Sigma^* \to \Sigma^*$ if for all $w \in \Sigma^*$ the algorithm $\mathcal{A}$ on input w terminates in a finite number of steps and outputs $f(w)$.

Examples of functions:

- Numerical functions: length, addition, multiplication, division etc
- Given graph G and s, t find shortest paths from s to t
- Given program M check if M halts on empty input
- Posts Correspondence problem

Definition

A function f over Σ^* is a boolean if $f : \Sigma^* \to \{0,1\}$.

Observation: There is a bijection between boolean functions and languages.

- Given boolean function $f : \Sigma^* \to \{0,1\}$ define language $L_f = \{w \in \Sigma^* \mid f(w) = 1\}$
- Given language $L \subseteq \Sigma^*$ define boolean function $f : \Sigma^* \to \{0,1\}$ as follows: $f(w) = 1$ if $w \in L$ and $f(w) = 0$ otherwise.

Languages and Computation

Definition

A function f over Σ^* is a boolean if $f : \Sigma^* \rightarrow \{0, 1\}$.

Observation: There is a bijection between boolean functions and languages.

- Given boolean function $f : \Sigma^* \rightarrow \{0, 1\}$ define language $L_f = \{w \in \Sigma^* \mid f(w) = 1\}$
- Given language $L \subseteq \Sigma^*$ define boolean function $f : \Sigma^* \rightarrow \{0, 1\}$ as follows: $f(w) = 1$ if $w \in L$ and $f(w) = 0$ otherwise.

Languages and Computation

Definition

A function f over Σ^* is a boolean if $f : \Sigma^* \rightarrow \{0,1\}$.

Observation: There is a bijection between boolean functions and languages.

- Given boolean function $f : \Sigma^* \rightarrow \{0,1\}$ define language $L_f = \{w \in \Sigma^* \mid f(w) = 1\}$
- Given language $L \subseteq \Sigma^*$ define boolean function $f : \Sigma^* \rightarrow \{0,1\}$ as follows: $f(w) = 1$ if $w \in L$ and $f(w) = 0$ otherwise.

Language recognition problem

Definition

For a language $L \subseteq \Sigma^*$ the language recognition problem associate with L is the following: given $w \in \Sigma^*$, is $w \in L$?

- Equivalent to the problem of "computing" the function f_L.
- Language recognition is same as boolean function computation
- How difficult is a function f to compute? How difficult is the recognizing L_f?

Why two different views? Helpful in understanding different aspects?

Language recognition problem

Definition

For a language $L \subseteq \Sigma^*$ the language recognition problem associate with L is the following: given $w \in \Sigma^*$, is $w \in L$?

- Equivalent to the problem of "computing" the function f_L.
- Language recognition is same as boolean function computation
- How difficult is a function f to compute? How difficult is the recognizing L_f?

Why two different views? Helpful in understanding different aspects?

Language recognition problem

Definition

For a language $L \subseteq \Sigma^*$ the language recognition problem associate with L is the following: given $w \in \Sigma^*$, is $w \in L$?

- Equivalent to the problem of "computing" the function f_L.
- Language recognition is same as boolean function computation
- How difficult is a function f to compute? How difficult is the recognizing L_f?

Why two different views? Helpful in understanding different aspects?

How many languages are there?

The answer my friend is blowing in the slides.

Recall:

Definition

An set X is countable if there is a bijection f between the natural numbers and A.

Theorem

Σ^* is countable for every finite Σ.

The set of all languages is $\mathbb{P}(\Sigma^*)$ the power set of Σ^*

Theorem (Cantor)

*$\mathbb{P}(\Sigma^*)$ is **not** countable for any finite Σ.*

How many languages are there?

The answer my friend is blowing in the slides.

Recall:

Definition

An set X is countable if there is a bijection f between the natural numbers and A.

Theorem

Σ^* is countable for every finite Σ.

The set of all languages is $\mathbb{P}(\Sigma^*)$ the power set of Σ^*

Theorem (Cantor)

*$\mathbb{P}(\Sigma^*)$ is **not** countable for any finite Σ.*

Cantor's diagonalization argument

Theorem (Cantor)

$\mathbb{P}(\mathbb{N})$ *is not countable.*

- Suppose $\mathbb{P}(\mathbb{N})$ is countable infinite. Let $S_1, S_2, \ldots,$ be an enumeration of all subsets of numbers.
- Let D be the following diagonal subset of numbers.

$$D = \{i \mid i \notin S_i\}$$

- Since D is a set of numbers, by assumption, $D = S_j$ for some j.
- **Question:** Is $j \in D$?

Consequences for Computation

- How many C programs are there? The set of C programs is countable since each of them can be represented as a string over a finite alphabet.
- How many languages are there? Uncountably many!
- Hence some (in fact almost all!) languages/boolean functions do not have any C program to recognize them.

Questions:

- Maybe interesting languages/functions have C programs and hence computable. Only uninteresting languors uncomputable?
- Why should C programs be the definition of computability?
- Ok, there are difficult problems/languages. what languages are computable and which have efficient algorithms?

Consequences for Computation

- How many *C* programs are there? The set of *C* programs is countable since each of them can be represented as a string over a finite alphabet.
- How many languages are there? Uncountably many!
- Hence some (in fact almost all!) languages/boolean functions do not have any *C* program to recognize them.

Questions:

- Maybe interesting languages/functions have *C* programs and hence computable. Only uninteresting languors uncomputable?
- Why should *C* programs be the definition of computability?
- Ok, there are difficult problems/languages. what languages are computable and which have efficient algorithms?

Easy languages

Definition

A language $L \subseteq \Sigma^*$ is finite if $|L| = n$ for some integer n.

Exercise: Prove the following.

Theorem

The set of all finite languages is countable.

THE END

...

(for now)