

Théorie des Langages Rationnels

A Framework for Languages

Adrien Pommellet, LRE



January 23, 2023

Deciding Languages

A premature goal

Before giving meaning to a word in a language, a computer should be able to tell **whether said word belongs to the language** in the first place.

Deciding Languages

Languages and algorithms

Decidable language

A language L over an alphabet Σ is said to be decidable (or **recursive**) if there exists an algorithm A such that, on any input $w \in \Sigma^*$:

- If $w \in L$, then $A(w)$ answers true.
- If $w \notin L$, then $A(w)$ answers false.

We keep the **exact** definition of an algorithm vague on purpose: assume for now it is any method we can implement using a computer program, memory and speed issues notwithstanding.

Exercise 1. Are the following languages decidable?

- 1 The set of prime numbers.
- 2 The empty set \emptyset .
- 3 The set Σ^* of all words
- 4 The set of C source codes of programs that halt on a given input x .
- 5 The set of C source codes of programs that halt in less than ten seconds on a given input x .

Answer

Deciding Languages

A surprising result

Deciding Languages

Weaker decidability

Semi-decidable language

A language L over an alphabet Σ is said to be semi-decidable (or **recursively enumerable**) if there exists an algorithm A such that, on any input $w \in \Sigma^*$:

- If $w \in L$, then $A(w)$ answers true.
- If $w \notin L$, then $A(w)$ answers false or **does not end**.

Semi-decidable languages are said to be recursively enumerable because it is possible to design an algorithm that **enumerates** all the words in said language.

Exercise 2. Is the halting problem \mathcal{H} recursively enumerable?

Answer

Deciding Languages

Recursive or recursively enumerable?

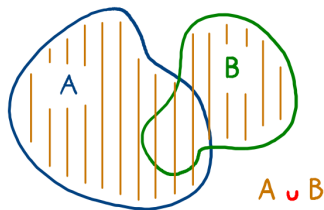
Theorem

If a language L is recursive, then it is recursively enumerable as well.

Set-theoretic Operations

Union

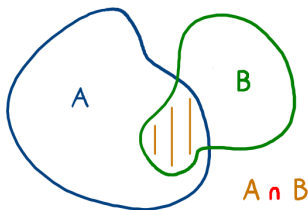
Given two languages A and B over an alphabet Σ , for all word $w \in \Sigma^*$, $w \in A \cup B$ if and only if $w \in A$ **or** $w \in B$.



Set-theoretic Operations

Intersection

Given two languages A and B over an alphabet Σ , for all word $w \in \Sigma^*$, $w \in A \cap B$ if and only if $w \in A$ **and** $w \in B$.



Set-theoretic Operations

Inclusion

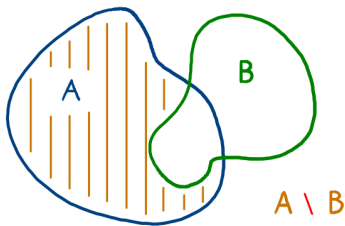
Given two languages A and B over an alphabet Σ , $A \subseteq B$ if for all $w \in \Sigma^*$, $w \in A$ **implies that** $w \in B$.



Set-theoretic Operations

Subtraction

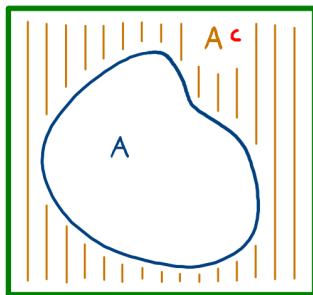
Given two languages A and B over an alphabet Σ , for all word $w \in \Sigma^*$, $w \in A \setminus B$ if and only if $w \in A$ **but** $w \notin B$.



Set-theoretic Operations

Complement

Given a language A over an alphabet Σ , for all word $w \in \Sigma^*$, $w \in A^c$ if and only if $w \notin A$. Note that $A^c = \Sigma^* \setminus A$. We also write $\bar{A} = A^c$.



Set-theoretic Operations

Common properties

Given three languages A , B and C over an alphabet Σ , these properties hold:

$$A^{\complement\complement} = A$$

$$A \cup A^{\complement} = \Sigma^*$$

$$(A \cup B)^{\complement} = A^{\complement} \cap B^{\complement}$$

$$(A \cap B)^{\complement} = A^{\complement} \cup B^{\complement}$$

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

Other Operations

Concatenation of languages

Some operations on words can be extended to languages.

Concatenation

Given two languages L_1 and L_2 , we define their concatenation as the language $L_1 \cdot L_2 = \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$.

Exponentiation of languages

Exponentiation

Given an integer $k \in \mathbb{N}$ and a language L , we define the languages:

- $L^0 = \{\varepsilon\}$.
- $L^k = \underbrace{L \cdot \dots \cdot L}_{k \text{ times}}$.

L^k is the set of all words obtained by concatenating k words of L .

Note that $L^k \neq \{u^k \mid u \in L\}$. We can concatenate k **different** words together, we don't always have to repeat the **same** word k times.

As an example, if $L = \{ab, cd, e\}$, then $abcde \in L^3$.

Other Operations

Prefixes, suffixes, factors

Prefix of a language

Let L be a language on Σ . The language $\text{Pref}(L)$ is the set of all words that are the prefix of **at least one** word of L .

Formally, $\text{Pref}(L) = \{w \mid x \in L, w \in \text{Pref}(x)\}$. We define $\text{Suff}(L)$ and $\text{Fact}(L)$ in a similar manner.

Note that $w \in \text{Pref}(L)$ does not have to be a prefix of **every** word in L .

As an example, $\text{Pref}(\{a, bc\}) = \{\varepsilon, a, b, bc\}$.

Practical Application

Exercise 3. Are the following properties true in the general case, assuming a generic language L ? If they're not, use **counter-examples** to falsify them.

$$\text{Fact}(L) = \text{Pref}(L) \cup \text{Suff}(L)$$

$$\text{Pref}(L) = \text{Pref}(\text{Pref}(L))$$

$$\text{Pref}(\text{Fact}(L)) = \text{Pref}(L)$$

$$\text{Pref}(\text{Fact}(L)) = \text{Fact}(L)$$

$$\text{Fact}(L) = \text{Pref}(\text{Suff}(L))$$

Answer

Other Operations

Kleene star

Kleene star

Given a language L , we introduce $L^* = L^0 \cup L_1 \cup L^2 \cup \dots = \bigcup_{n \geq 0} L^n$, that is, the set of words than we can obtain by concatenating a **finite** number of words (possibly **none**) of L .

In a similar manner, we define $L^+ = \bigcup_{n \geq 1} L^n$ as the set of words than we can obtain by concatenating **at least one** word of L .

Other Operations

Properties of the Kleene star

For any language L , the following properties hold:

- $\varepsilon \in L^*$, even if $\varepsilon \notin L$. We can always pick 0 words of L to concatenate.
- However, $\varepsilon \in L^+$ if and only if $\varepsilon \in L$.
- $\emptyset^* = \varepsilon$. Again, we pick 0 words to concatenate.
- $\Sigma^* = \Sigma^*$. A word is indeed a finite concatenation of letters, and our initial notation makes sense.

Practical Application

Exercise 4. Are the following properties true in the general case, assuming three generic languages L_1, L_2, L_3 ?

$$\{ab\} \cup \{ba\} = \{abba\}$$

$$\{a\}^n = \{a^n\}$$

$$\{a\}^* = \{a^n \mid n \geq 0\}$$

$$\{a, b\}^n = \{a^n, b^n\}$$

$$\{a, b\}^3 = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$$

$$(L_1 \cup L_2)^2 = L_1^2 \cup L_1L_2 \cup L_2L_1 \cup L_2^2$$

$$L_1 \cdot (L_2 \cup L_3) = (L_1 \cdot L_2) \cup (L_1 \cdot L_3)$$

Answer

See you next class!