Formal Languages and Automata Theory

Prof. Dr.-Ing. Sebastian Schlesinger

Berlin School for Economics and Law

September 25, 2024

Outline

What are Formal Languages?

- **Formal language** is a set of strings formed from an alphabet Σ.
- A language is defined by formal rules, usually grammar or automata.
- Applications: parsing, compilers, coding theory, and more.

What are Formal Languages?

- **Formal language** is a set of strings formed from an alphabet Σ.
- A language is defined by formal rules, usually grammar or automata.
- Applications: parsing, compilers, coding theory, and more.

Example

Alphabet: $\Sigma = \{a, b\}$

- String: $abba$
- Language: $L = \{ab, ba, abb, bba\}$

Formal Language Definitions

- **Alphabet** (Σ): A non-empty, finite set of symbols.
- **String** (w): A finite sequence of symbols from Σ.
- **Language** (L): A set of strings over Σ.
- Σ^*: The set of all possible strings over Σ, including the empty string ϵ.

Formal Language Definitions

- **Alphabet** (Σ): A non-empty, finite set of symbols.
- **String** (w): A finite sequence of symbols from Σ.
- **Language** (L): A set of strings over Σ.
- Σ^*: The set of all possible strings over Σ, including the empty string ϵ.

Operations on Languages

- Union: $L_1 \cup L_2$
- Concatenation: $L_1 L_2 = \{xy \mid x \in L_1, y \in L_2\}$
- Kleene Star: $L^* = \{w_1 w_2 \dots w_n \mid w_i \in L, n \geq 0\}$

Chomsky Hierarchy of Languages

- **Type 0**: Recursively Enumerable (Turing Machines)
- **Type 1**: Context-sensitive (Linear-bounded Automata)
- **Type 2**: Context-free (Pushdown Automata)
- **Type 3**: Regular (Finite Automata)

Chomsky Hierarchy of Languages

- **Type 0**: Recursively Enumerable (Turing Machines)
- **Type 1**: Context-sensitive (Linear-bounded Automata)
- **Type 2**: Context-free (Pushdown Automata)
- **Type 3**: Regular (Finite Automata)

Important Theorem: Language Inclusion

Regular $\subset$ Context-Free $\subset$ Context-Sensitive $\subset$ Recursively Enumerable

Regular Languages

- A language is **regular** if it can be described by a regular expression.
- Can be accepted by a finite automaton (DFA or NFA).
- Closure properties:
    - Union
    - Concatenation
    - Kleene Star

Regular Languages

- A language is **regular** if it can be described by a regular expression.
- Can be accepted by a finite automaton (DFA or NFA).
- Closure properties:
  - Union
  - Concatenation
  - Kleene Star

Theorem: Pumping Lemma for Regular Languages

If L is a regular language, then there exists a constant p such that any string $w \in L$ with $|w| \geq p$ can be split into three parts, $w = xyz$, such that:

- $|xy| \leq p$
- $|y| > 0$
- $xy^n z \in L$ for all $n \geq 0$

Context-Free Languages (CFL)

- Generated by context-free grammars (CFGs).
- Can be accepted by pushdown automata (PDA).
- Closure properties:
  - Union
  - Concatenation
  - Kleene Star

Context-Free Languages (CFL)

- Generated by context-free grammars (CFGs).
- Can be accepted by pushdown automata (PDA).
- Closure properties:
  - Union
  - Concatenation
  - Kleene Star

Theorem: Pumping Lemma for CFLs

If L is a context-free language, there exists a constant p such that any string $w \in L$ with $|w| \geq p$ can be split into five parts, $w = uvxyz$, such that:

- $|vxy| \leq p$
- $|vy| > 0$
- $uv^nxy^nz \in L$ for all $n \geq 0$

Context-Sensitive Languages

- Generated by context-sensitive grammars.
- Can be accepted by linear-bounded automata (LBA).
- Closure properties:
  - Union
  - Intersection
  - Complementation

Context-Sensitive Languages

- Generated by context-sensitive grammars.
- Can be accepted by linear-bounded automata (LBA).
- Closure properties:
  - Union
  - Intersection
  - Complementation

Theorem: Savitch's Theorem

Any context-sensitive language can be decided in deterministic space $O(n^2)$.

Recursively Enumerable Languages

- A language is recursively enumerable if there exists a Turing machine that can enumerate all strings in the language.
- Not all recursively enumerable languages are decidable.

Recursively Enumerable Languages

- A language is recursively enumerable if there exists a Turing machine that can enumerate all strings in the language.
- Not all recursively enumerable languages are decidable.

Theorem: Rice's Theorem

Any non-trivial property of the language recognized by a Turing machine is undecidable.

Important Theorems in Formal Languages

- **Myhill-Nerode Theorem**: Characterizes regular languages based on the equivalence of strings.
- **Kleene's Theorem**: Describes equivalence between regular expressions and finite automata.
- **Chomsky-SchÃ¼tzenberger Theorem**: Every context-free language can be represented using a Dyck language.
- **Rice's Theorem**: Undecidability of non-trivial properties of Turing machine languages.