Theoretical Computer Science
Languages and Grammars

Jonas Hübotter

Outline

Formal Languages

Definition 1

- The alphabet Σ is a finite set of symbols.

Formal Languages

Definition 1

- The alphabet Σ is a finite set of symbols.
- A finite sequence of symbols is called a word.

Formal Languages

Definition 1

- The alphabet Σ is a finite set of symbols.
- A finite sequence of symbols is called a word.
- ϵ is the empty word.

Formal Languages

Definition 1

- The alphabet Σ is a finite set of symbols.
- A finite sequence of symbols is called a word.
- ϵ is the empty word.
- Σ^* is the set of all words over an alphabet Σ.

Formal Languages

Definition 1

- The alphabet Σ is a finite set of symbols.
- A finite sequence of symbols is called a word.
- ϵ is the empty word.
- Σ^* is the set of all words over an alphabet Σ.
- $L \subseteq \Sigma^*$ is called a (formal) language.

.

Formal Languages

Definition 1

- The alphabet Σ is a finite set of symbols.
- A finite sequence of symbols is called a word.
- ϵ is the empty word.
- Σ^* is the set of all words over an alphabet Σ.
- $L \subseteq \Sigma^*$ is called a (formal) language.

.

Definition 2 (Operations on words)

- $|w|$ is the length of word w.

Formal Languages

Definition 1

- The alphabet Σ is a finite set of symbols.
- A finite sequence of symbols is called a word.
- ϵ is the empty word.
- Σ^* is the set of all words over an alphabet Σ.
- $L \subseteq \Sigma^*$ is called a (formal) language.

.

Definition 2 (Operations on words)

- $|w|$ is the length of word w.
- uv is the concatenation of two words u, v.

Formal Languages

Definition 1

- The alphabet Σ is a finite set of symbols.
- A finite sequence of symbols is called a word.
- ϵ is the empty word.
- Σ^* is the set of all words over an alphabet Σ.
- $L \subseteq \Sigma^*$ is called a (formal) language.

.

Definition 2 (Operations on words)

- $|w|$ is the length of word w.
- uv is the concatenation of two words u, v.
- $w^0 = \epsilon, w^{n+1} = ww^n$ defines repetition of a word w.

Operations on Formal Languages

Definition 3

Let A and B be formal languages.

- $AB = \{uv \mid u \in A, v \in B\}$ (concatenation).

Operations on Formal Languages

Definition 3

Let A and B be formal languages.

- $AB = \{uv \mid u \in A, v \in B\}$ (concatenation).
- $A^0 = \{\epsilon\}, A^{n+1} = AA^n$ (repetition).

Operations on Formal Languages

Definition 3

Let A and B be formal languages.

- $AB = \{uv \mid u \in A, v \in B\}$ (concatenation).
- $A^0 = \{\epsilon\}, A^{n+1} = AA^n$ (repetition).
- $A^* = \bigcup_{n \in \mathbb{N}_0} A^n$ (reflexive transitive closure).

Operations on Formal Languages

Definition 3

Let A and B be formal languages.

- $AB = \{uv \mid u \in A, v \in B\}$ (concatenation).
- $A^0 = \{\epsilon\}, A^{n+1} = AA^n$ (repetition).
- $A^* = \bigcup_{n \in \mathbb{N}_0} A^n$ (reflexive transitive closure).
- $A^+ = AA^* = \bigcup_{n \in \mathbb{N}} A^n$ (transitive closure).

.

Grammars

Definition 4

A grammar is a 4-tuple $G = (V, \Sigma, P, S)$

Grammars

Definition 4

A grammar is a 4-tuple $G = (V, \Sigma, P, S)$where

- V is a finite set of non-terminals (or variables)

Grammars

Definition 4

A grammar is a 4-tuple $G = (V, \Sigma, P, S)$where

- V is a finite set of non-terminals (or variables);
- Σ is an alphabet whose symbols are called terminals

Grammars

Definition 4

A grammar is a 4-tuple $G = (V, \Sigma, P, S)$where

- V is a finite set of non-terminals (or variables);
- Σ is an alphabet whose symbols are called terminals;
- $P \subseteq (V \cup \Sigma)^* \times (V \cup \Sigma)^*$ is a set of productions

Grammars

Definition 4

A grammar is a 4-tuple $G = (V, \Sigma, P, S)$where

- V is a finite set of non-terminals (or variables);
- Σ is an alphabet whose symbols are called terminals;
- $P \subseteq (V \cup \Sigma)^* \times (V \cup \Sigma)^*$ is a set of productions; and
- $S \in V$ is the initial variable.

Grammars

Definition 5 (Derivation)

A grammar induces a derivation relation $\rightarrow_G$ on words over $V \cup \Sigma$:

Definition 5 (Derivation)

A grammar induces a derivation relation $\to_G$ on words over $V \cup \Sigma$:

$$\forall \alpha, \alpha' \in (V \cup \Sigma)^*.\ \alpha \to_G \alpha' :\Leftrightarrow \exists \beta \to \beta' \in P.$$
$$\alpha = \alpha_1 \beta \alpha_2 \text{ and } \alpha' = \alpha_1 \beta' \alpha_2.$$

Grammars

Definition 5 (Derivation)

A grammar induces a derivation relation $\rightarrow_G$ on words over $V \cup \Sigma$:

$$\forall \alpha, \alpha' \in (V \cup \Sigma)^*.\ \alpha \rightarrow_G \alpha' :\Leftrightarrow \exists \beta \rightarrow \beta' \in P.$$
$$\alpha = \alpha_1 \beta \alpha_2 \text{ and } \alpha' = \alpha_1 \beta' \alpha_2.$$

A derivation of α_n from α_1 is denoted by $\alpha_1 \rightarrow_G \cdots \rightarrow_G \alpha_n$.

Grammars

Definition 5 (Derivation)

A grammar induces a derivation relation $\rightarrow_G$ on words over $V \cup \Sigma$:

$$\forall \alpha, \alpha' \in (V \cup \Sigma)^*.\ \alpha \rightarrow_G \alpha' :\Leftrightarrow \exists \beta \rightarrow \beta' \in P.$$
$$\alpha = \alpha_1 \beta \alpha_2 \text{ and } \alpha' = \alpha_1 \beta' \alpha_2.$$

A derivation of α_n from α_1 is denoted by $\alpha_1 \rightarrow_G \cdots \rightarrow_G \alpha_n$.

Definition 6 (Language of a Grammar)

Given the derivation of α_n from α_1, G produces α_n if $\alpha_1 = S$ and $\alpha_n \in \Sigma^*$.

Grammars

Definition 5 (Derivation)

A grammar induces a derivation relation $\rightarrow_G$ on words over $V \cup \Sigma$:

$$\forall \alpha, \alpha' \in (V \cup \Sigma)^*.\ \alpha \rightarrow_G \alpha' :\Leftrightarrow \exists \beta \rightarrow \beta' \in P.$$
$$\alpha = \alpha_1 \beta \alpha_2 \text{ and } \alpha' = \alpha_1 \beta' \alpha_2.$$

A derivation of α_n from α_1 is denoted by $\alpha_1 \rightarrow_G \cdots \rightarrow_G \alpha_n$.

Definition 6 (Language of a Grammar)

Given the derivation of α_n from α_1, G produces α_n if $\alpha_1 = S$ and $\alpha_n \in \Sigma^*$.
The language of G $L(G)$ is the set of all words produced by G.

Chomsky-Hierarchy

Definition 7

A grammar G is of

Chomsky-Hierarchy

Definition 7

A grammar G is of

- Type 0 always;

Chomsky-Hierarchy

Definition 7

A grammar G is of

- Type 0 always;
- Type 1 (context-sensitive) $\forall \alpha \to \beta \in P \setminus \{S \to \epsilon\}.\ |\alpha| \leq |\beta|$;

Chomsky-Hierarchy

Definition 7

A grammar G is of

- Type 0 always;
- Type 1 (context-sensitive) $\forall \alpha \to \beta \in P \setminus \{S \to \epsilon\}.\ |\alpha| \leq |\beta|$;
- Type 2 (context-free) if of type 1 and $\forall \alpha \to \beta \in P.\ \alpha \in V$;

Chomsky-Hierarchy

Definition 7

A grammar G is of

- Type 0 always;
- Type 1 (context-sensitive) $\forall \alpha \to \beta \in P \setminus \{S \to \epsilon\}.\ |\alpha| \leq |\beta|$;
- Type 2 (context-free) if of type 1 and $\forall \alpha \to \beta \in P.\ \alpha \in V$;
- Type 3 (right-linear) if of type 2 and $\forall \alpha \to \beta \in P \setminus \{S \to \epsilon\}.\ \beta \in \Sigma \cup \Sigma V$.

Word problem

Word Problem

given: a grammar G, a word $w \in \Sigma^*$
problem: $w \in L(G)$?

Word problem

Word Problem

given: a grammar G, a word $w \in \Sigma^*$
problem: $w \in L(G)$?

Automata are used to solve the word problem. Depending on the type of the grammar different automata are used.
For example, (Non-)Deterministic Finite Automata are used for right-linear grammars (regular languages) while Pushdown Automata are used for context-free grammars.