MTL783: Theory of Computation 1st Semester, 2025-2026

Lecture 1 — 24/07/2025

Lecturer: Prof Minati De *Scribe: Team 1*

Scribed by:

1. Keshav Rai (2022MT61968)
2. Siddharth Saini (2022MT11283)
3. Parth Bhardwaj (2022MT11257)
4. Tushar Goyal (2022MT11266)
5. Ryan Azim Shaikh (2022MT11923)

1 Foundational Definitions

Definition 1.1 (Alphabet)**.** *An* alphabet Σ *is any finite, non-empty set of symbols.*

Definition 1.2 (String)**.** *A* string *over* Σ *is a finite sequence* $w = a_1 a_2 \dots a_n$ *with each* $a_i \in \Sigma$*. Its length is* $|w| = n$*. The unique string of length zero is called the* empty string *and denoted by* λ*.*

Definition 1.3 (Concatenation)**.** *If* u *and* v *are strings over* Σ*, their* concatenation uv *is the string formed by appending* v *after* u*. One has* $|uv| = |u| + |v|$*.*

Definition 1.4 (Language)**.** *A* language *over* Σ *is any subset* $L \subseteq \Sigma^*$*.*

Definition 1.5 (Iteration)**.** *For a string* $w \in \Sigma^*$ *and integer* $k \geq 0$*, define*

$$w^0 = \lambda, \qquad w^k = \underbrace{w\, w \cdots w}_{k \text{ times}}.$$

Then $w^* = \{\, w^k \mid k \geq 0\}$ *and* $w^+ = w^* \setminus \{\lambda\}$*.*

Examples

- Alphabets: $\Sigma_1 = \{a, b, c\},\ \Sigma_2 = \{0, 1\},\ \Sigma_3 = \{a, b, c, \dots, z\}$.
- Strings over $\{a, b\}$: $u = ab,\ w = abab,\ \lambda$ (empty string).
- The set of all strings over Σ is Σ^*.
- Any language L satisfies $L \subseteq \Sigma^*$.

2 Classes of Abstract Machines

Finite Automata (FA)

Machines with finite control only; recognise exactly the *regular languages*.

Push-Down Automata (PDA)
FA equipped with a stack; recognise exactly the *context-free languages*.

Linear Bounded Automata (LBA)
Turing machines whose tape usage is linearly bounded by the input; recognise exactly the *context-sensitive languages*.

Turing Machines (TM)
Unbounded-tape machines; capture the full power of algorithmic (recursively enumerable) computation.

3 Automata–Grammar Correspondence

Chomsky Type	**Grammar**	**Equivalent Automaton**
Type-3	Right-linear (regular)	FA
Type-2	Context-free	PDA
Type-1	Context-sensitive	LBA
Type-0	Unrestricted	TM

4 Further Reading

1. M. Sipser, *Introduction to the Theory of Computation*, 3rd ed.
2. D. C. Kozen, *Automata and Computability*.
3. P. Linz, *An Introduction to Formal Languages and Automata*.
4. M. Smid and A. Maheshwari, *Intro to Theory of Computation* (notes).