

# Lecture #10: Introduction to Turing Machines

## Questions for Review

### Introduction to Turing Machines

1. What is a **Turing machine**? Give both an “informal definition” of this as well as a “formal definition” as a 7-tuple.
2. Why are Turing machines of interest?
3. What is the **input alphabet**  $\Sigma$  and the **tape alphabet**  $\Gamma$  of a Turing machine? How are they related?
4. What information does a **configuration** of a Turing machine include? How is this generally represented as a string?
5. What is the **start configuration** (or **initial configuration**) for a Turing machine  $M$  and input string  $\omega$ ?
6. What information do **moves** (or **transitions**) of a Turing machine depend on? Define the **transition function** for a Turing machine as precisely as you can.
7. What does a **transition table** for a Turing machine look like?
8. Describe — as precisely as you can — how the configuration of a Turing machine is changed when a transition is applied. Make sure to include information about how “moves left” and “moves right” are applied. How are the strings representing a transition changed when each kind of move is applied?
9. Explain the meaning of the notation  $\vdash$  and  $\vdash^*$ .
10. What does it mean for a Turing machine to **accept** an input string? What does it mean for a Turing machine to **reject** an input string? What does it mean for a Turing machine to **loop** on an input string?
11. What does it mean for a Turing machine to **recognize** a language? What does it mean for a language  $L \subseteq \Sigma^*$  to be **Turing-recognizable** (or **recognizable**)?
12. What does it mean for a Turing machine to **decide** a language? What does it mean for a language  $L \subseteq \Sigma^*$  to be **Turing-decidable** (or **decidable**)?

13. What is a **Turing machine that computes a function** — and what kinds of functions can these compute? How are these different from Turing machines that recognize languages?

## Turing Machine Design

As previously noted, “Turing machine design” is not a significant part of this course. However, it will be useful to know at least a little bit about this in order to solve problems concerning “computability” later on in this course.

14. Describe the design technique, **refinement**. How can this be used to design a Turing machine to recognize (or decide) a given language, or to compute a given function?
15. What is a **high-level** description of a Turing machine?
16. What is an **implementation-level** description of a Turing machine?
17. What is a **formal** description of a Turing machine?
18. How are three kinds of “descriptions” of a Turing machines related? How can they be useful for Turing machine design?