**Language Deciders vs. Language Recognizers**

Every TM $T$ partitions the set of possible input strings over its input alphabet into three sets:

- $\text{Accept}(T) =$ the set of input strings for which $T$ halts by reaching its accept state;
- $\text{Reject}(T) =$ the set of input strings for which $T$ halts by reaching its reject state; and
- $\text{Loop}(T) =$ the set of input strings for which $T$ never halts.

For example, consider the following TM $M$ for strings over the alphabet $\Sigma = \{a, b\}$:

![Diagram of TM M]

It is easily checked that

- $\text{Accept}(M) =$ all strings of the form $a\Sigma^*a \cup b\Sigma^*b \cup \Sigma$;
- $\text{Reject}(M) =$ all strings of the form $a\Sigma^*b \cup \varepsilon$; and
- $\text{Loop}(M) =$ all strings of the form $b\Sigma^*a$.

Using this terminology, we can redefine recognizers and deciders as follows:

- A TM $T$ is a **recognizer** for a language $L$ if $\text{Accept}(T) = L$.
- A TM $T$ is a **decider** for a language $L$ if $\text{Accept}(T) = L$ and $\text{Loop}(T) = \emptyset$. Equivalently, a TM $T$ is a decider for $L$ if $\text{Accept}(T) = L$ and $\text{Reject}(T) = \overline{L}$.

We see that the above example $M$ is a recognizer for the language $a\Sigma^*a \cup b\Sigma^*b \cup \Sigma$ but it is not a decider for this language.

Recall these definitions:

- A language $L$ is **Turing-recognizable** if there is some TM that recognizes it.
- A language $L$ is **decidable** if there is some TM that decides it.

We will show that there are languages that are: (1) Turing-recognizable but not decidable, and (2) languages that are not even Turing-recognizable. A language in the first category has the peculiar property that any TM that recognizes it must fail to terminate for some input strings. A language in the second category has the even more peculiar property that there is no TM that accepts exactly those strings belonging to that language. (By the Church-Turing thesis, this means that there is no algorithm that is able to return with an accept result for exactly those strings belonging to such a language.)