### Finite State Machines (FSMs)

A FSM is described by:

* A finite set of **states**.
* A finite set of **input symbols**.
* A **transition function** which assigns a new state to every combination of state and input.
* A subset of states designated as **accepting states**.
* A state designed as the **starting state**.

The transition function defines a new state in terms of a current state and a current input symbol .

The end marker of a string is denoted by .

FSMs are used for **lexical analysis**.

### Pushdown Machines (PMs)

A PM is described by:

* A finite set of **input symbols**, including an **end marker**.
* A finite set of **stack symbols**, including a **bottom marker**.
* A finite set of **states**, including a **starting state**.
* A **control** which assigns an exit or transition to each combination of input symbol, stack symbol and state.
* A **starting stack**.

Each transition consists of a:

1. Stack operation:
   * PUSH: Push the symbol onto the stack.
   * POP: Pop the top stack symbol.
   * Leave the stack unchanged.
2. State operation:
   * STATE(S): Set the next state to state S.
3. Input operation
   * ADVANCE: Get the next input symbol.
   * RETAIN: Do not advance.

We design our parsers with single-state pushdown machines.