Introduction to Automata Theory

Max Randall
Chapman University

February 16, 2025

Abstract

Automata are abstract machines that model computations without memory. Before defining them formally, we consider some examples.

Contents

1 Introduction

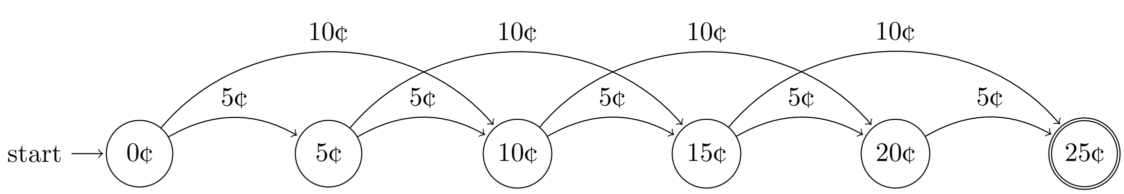
Automata are abstract machines that model computations without memory. Before defining them formally, we consider some examples.

2 Examples of Automata

2.1 Parking or Vending Machine

Specification: The machine requires 25 cents, paid in chunks of 5 or 10 cents.

Automaton: The state 25 is the accepting or final state. A word (i.e., a sequence of symbols 5 and 10) is accepted if it leads from the initial state (0) to the final state (25).



2.2 Variable Names

Specification: In defining a programming language, valid variable names should:

- Start with a letter ($\ell = a, b, c, \dots, z$).
- Be followed by any combination of letters (ℓ) or digits ($d = 0, 1, 2, \dots, 9$).
- End with a terminal symbol (t, e.g., $t =;$).

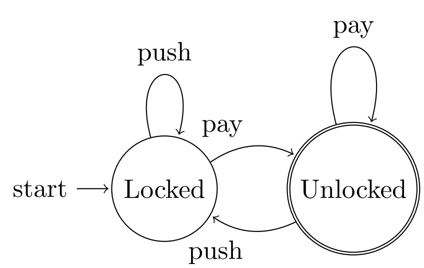
Automaton: Accepted words follow the pattern:

$$\ell(\ell + d)^* t$$

2.3 Turnstile

Specification: A money-operated turnstile:

- Starts in the **locked** state.
- From **locked**:
  - A **push** (u) keeps it **locked**.
  - A **pay** (p) moves it to the **unlocked** state.
- From **unlocked**:
  - A **pay** (p) keeps it **unlocked**.
  - A **push** (u) moves it back to **locked**.
- The **unlocked** state is the accepting state.



3 Homework

3.1 Exercise: Characterizing Accepted Words

Characterize all accepted words (i.e., describe exactly those words that are recognized).

The vending machine automaton accepts words consisting of payments in increments of 5 and 10 cents, reaching exactly 25 cents. The accepted words follow the pattern:

$$(10 + 5)^*$$

subject to the constraint that the sum of the numbers in the sequence equals 25.

3.2 Exercise: Turnstile Regular Expression

Characterize all accepted words and describe them using a regular expression.

The turnstile automaton can be characterized by the following regular expression:

$$(p + up)^* p (p + up)^*$$

where:

- p represents a **pay** action.
- u represents a **push** action.
- p^* means zero or more additional **pay** actions while the turnstile remains open.
- The entire sequence can repeat any number of times.

Example Accepted Words

p	(Pay once, unlocks, and stays open)
pp	(Pay twice, remains unlocked)
pup	(Pay, push to lock, then pay again to unlock)
$ppuppp$	(Multiple pays, a push to lock, then more pays)
$upupup$	(Invalid pushes in locked state, followed by pays to unlock)

4 Conclusion

Deterministic Finite Automata (DFAs) are a fundamental concept in automata theory, providing a mathematical model for recognizing patterns in strings. A DFA is defined as a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, Σ is an input alphabet, δ is the transition function mapping states and symbols to new states, q_0 is the initial state, and F is the set of accepting states. DFAs process input strings deterministically, meaning that for every state and input symbol, there is exactly one transition.

Formal languages are sets of words over an alphabet Σ. The set of all possible words is denoted Σ^*, which includes the empty word ε. The length of a word w is written as $|w|$, and the occurrence of a symbol a in w is denoted $|w|_a$. Example languages include L_1, which contains words with the substring "01," L_2, which consists of words whose lengths are powers of two, and L_3, which contains words with an equal number of 0s and 1s.

DFAs determine if a word belongs to a language by processing transitions. If the final state is in F, the word is accepted; otherwise, it is rejected.

References

[1] Hopcroft, J. E., Motwani, R., Ullman, J. D. *Introduction to Automata Theory, Languages, and Computation*. Pearson, 2007.