CS340: Theory of Computation | Sem I 2021-22

Lecture Notes 1: Introduction to Theory of Computation

Raghunath Tewari | IIT Kanpur

1 Introduction

- Course syllabus
    1. Homework assignments (20%): About 4 assignments throughout the semester. One week to solve the problems. Late submissions will be penalized.
    2. Quizzes (20%): 2 quizzes.
    3. Mid semester exam (20%):
    4. End semester exam (40%):
- Course textbook: Introduction to the Theory of Computation by Michael Sipser. 3rd edition.
- Course website: http://moodle.cse.iitk.ac.in
- TAs: Information posted on moodle.

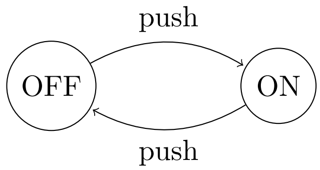
2 What is Theory of Computation?

- We first need to understand what is computation. Well defined step by step to solve a problem. Example: multiplication of two numbers, searching for a word in a dictionary, solving a crossword puzzle, (browsing the web?).
- Computational devices: calculator, vending machine, computer, phone, pen and paper, etc. There is a vast set of computational devices that we encounter in our day to day lives. Introduce mathematical models that characterize all such computational devices based on their fundamental property and study them. We will be studying power and limitations of these models.
- How do we come up with a solution? (Algorithms)
- What can be computed? Are there problems that cannot be computed? We will answer this question later in the semester.

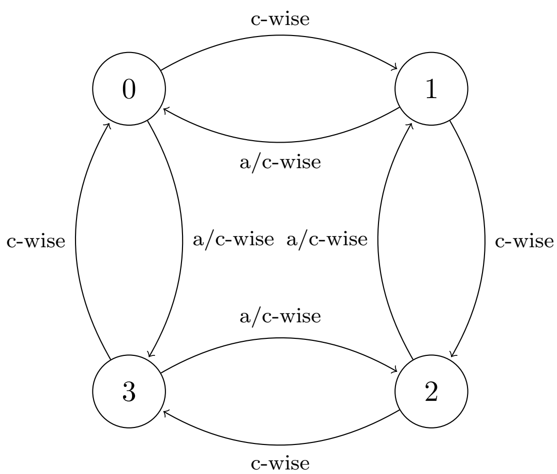
3 Introduction to Finite Automata

Informally, a finite automaton (plural is automata) is a system consisting of a set of states and interaction between them.
Examples,

1. A switch

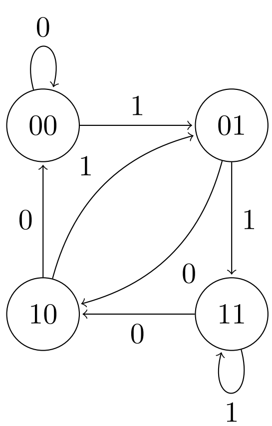


2. A fan regulator



3. Traffic lights

4. Checking if a binary number is divisible by 4. We will use the fact that a binary number is divisible by 4 if and only if the last two digits are 0.



Exercise 1. Read Chapter 0 from textbook.

Let us look at one more example of a finite state machine.

4 Another Example

Assume that in a chemistry lab there are 3 compounds, A, B and C, and they react as follows:

$$\begin{aligned} B + C &\longrightarrow 2A && (\alpha - \text{reaction}) \\ A + C &\longrightarrow 2B && (\beta - \text{reaction}) \\ A + B &\longrightarrow 2C && (\gamma - \text{reaction}) \end{aligned}$$

Properties:

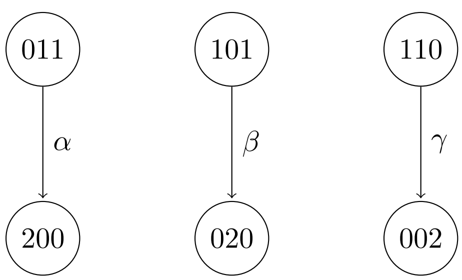
- The reactions are symmetric
- Total no. of units are the same

We say the lab is *dysfunctional* if only 1 compound is left.
At any given time the state of the lab is

- *static*, if it will/has become dysfunctional. Eg. only one compound is left, or, 1 unit of A and 1 unit of B is left.
- *intermediate*, if it can become dysfunctional. Eg. 2 units of A, 1 units of B, 1 units of C.
- *dynamic*, if it cannot become dysfunctional. Eg. 2 units of A and 1 unit of B.

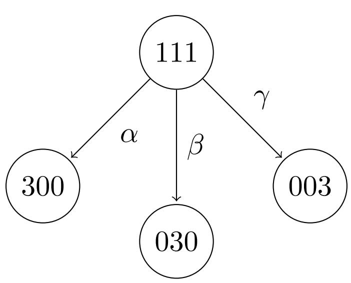
We define a *state* to be a sequence of 3 integers denoting the no. of units A, B and C resp.

Case: 2 units in the lab

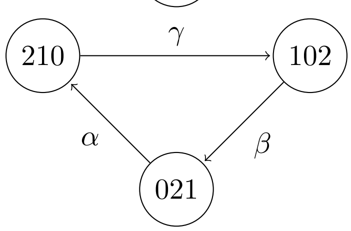


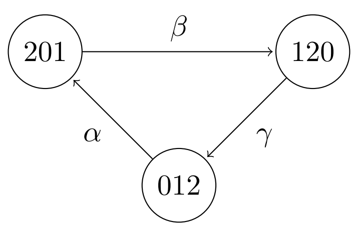
All states are static.

Case: 3 units in the lab



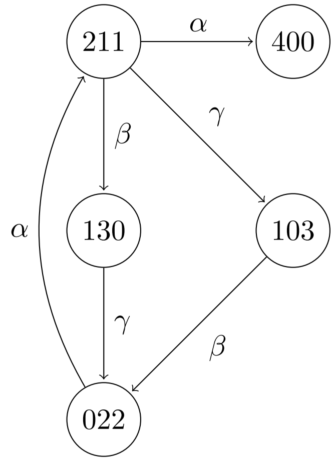
All states are static.





All states are dynamic.

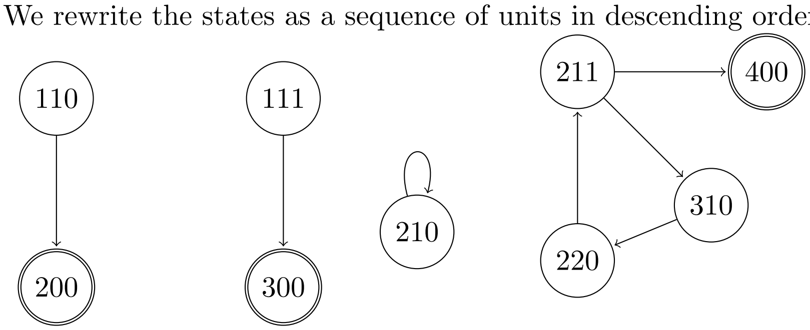
Case: 4 units in the lab



This state is static

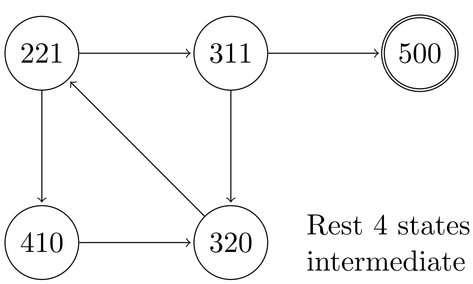
Rest 4 states are intermediate

<u>Note:</u> The lab becoming dysfunctional is dependent on the "set" of numbers of each compound, and not on the "ordered triplet". This is because of the symmetry of the three reactions. For example, the state **211** and **121** can be considered the same.
We rewrite the states as a sequence of units in descending order of the numbers.



2 units 3 units 4 units

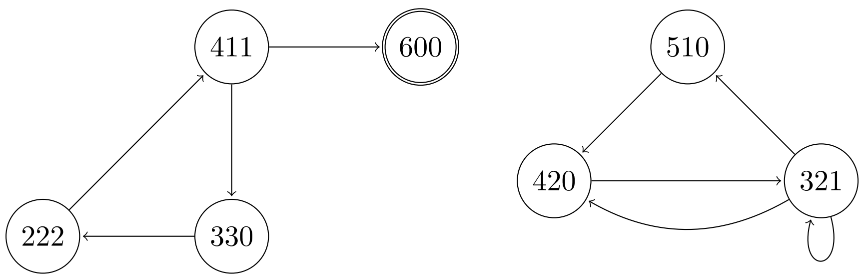
Case: 5 units in the lab



This state is static

Rest 4 states are intermediate

Case: 6 units in the lab



- State 600 is static.
- States 411, 330, 222 are intermediate.
- States 510, 420, 321 are dynamic.

Exercise 2. 1. Draw the diagram for 7 units.

2. Read chapter 0 from textbook.

We will now setup a mathematical framework to define and study various computational models.

5 Basic Definitions

- An *alphabet* is a finite, non-empty set symbols. Convention: Σ, Γ. Egs: $\{0,1\}$, $\{a, b, \ldots, z\}$.
- A *string* is a finite sequence of symbols chosen over some fixed alphabet. Eg. 0110.
- The *empty string* is the string having 0 symbols. Denoted as ϵ.
- *Length of a string* is the number of symbols in it. Eg. $|0110| = 4$, $|\epsilon| = 0$.
- For an alphabet Σ,
$$\Sigma^i = \{w | w \text{ is a string over } \Sigma \text{ of length } i\}.$$
- $\Sigma^* = \bigcup_{i \geq 0} \Sigma^i$ and $\Sigma^+ = \bigcup_{i > 0} \Sigma^i$
- L is a *language* over Σ if $L \subseteq \Sigma^*$. Note that L can be Σ^* as well.