

CPSC 3400 - Homework 6

DFAs, Regular Expressions and Turing machines

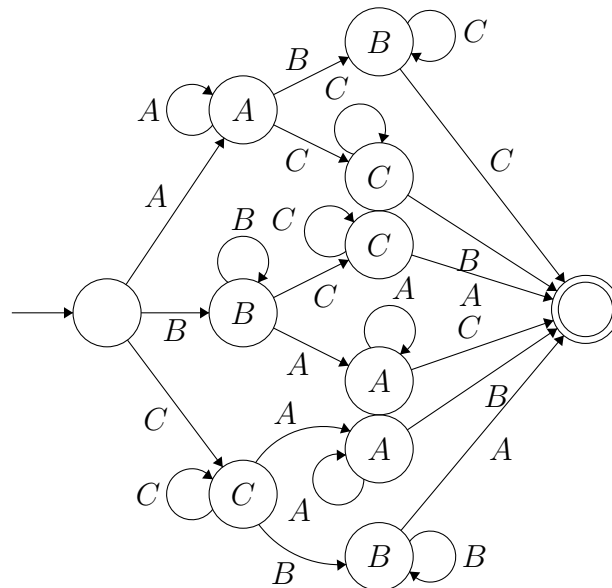
Minh-Hieu Nguyen

May 28, 2022

Part I. DFAs

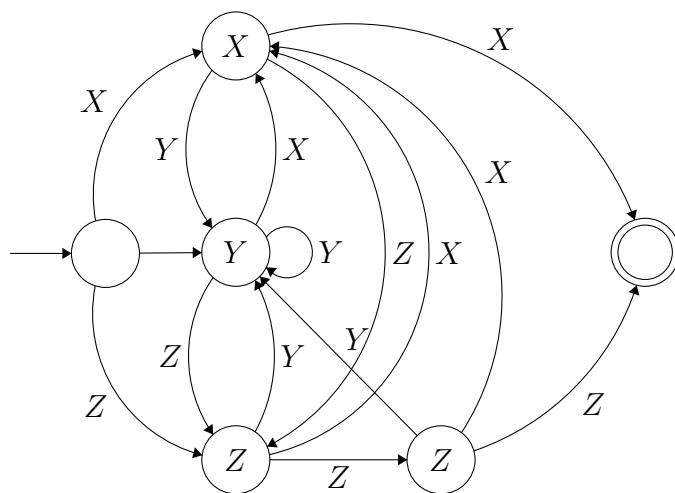
Draw DFAs for the following language specifications.

1. All strings on $\Sigma = \{A, B, C\}$ that contain each letter (A , B , and C) at least once.



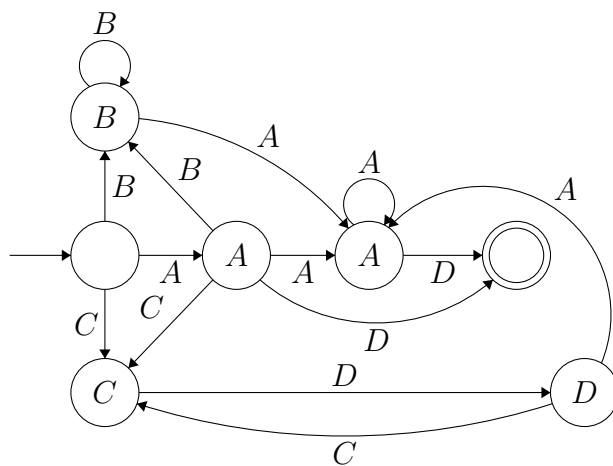
2. All strings on $\Sigma = \{X, Y, Z\}$ that contain two consecutive X s or three consecutive Z s (or both).

Solution



3. All strings on $\Sigma = \{A, B, C, D\}$ that match the Python regular expression $\sim (A?(B|CD)*A+D)\$$

Solution



Part II. Regular Expressions

For each item, write a *single* regular expression that matches that item. Note that in ALL cases, the entire string must match without additional characters.

1. A string of digits that contains only digits and contains exactly two fives. Examples of acceptable strings include: "15445", "55", "05563"

The string should be rejected if it contains anything other than a digits.

Solution

- Answer: $\wedge([0-4,6-9])^*(5)([0-4,6-9])^*(5)([0-4,6-9])^*\$$
- Explanation:
 - $([0,4,6-9])^*$: Accept all strings that contain digits in range from 0 to 9 except 5. It also accept empty strings. (1)
 - (5) : Accept only a digit-5 string. (2)

2. A regular expression that matches a time expressed in the form "1:45 PM".

The hours part must be a number from 1 to 12, the minutes range from 00 to 59, and the time must indicate either AM or PM (uppercase only and preceded by exactly one space).

Solution

- Answer: $\wedge((1[0-2] | [1-9]))(:)([0-5][0-9]) ([AP]M)\$$
- Explanation:
 - $(1[0-2] | [1-9])$: Accept all digit strings that are either in range from 10 to 12 or in range from 0 to 9 without leading zeroes.
 - $(:)$: Accept the separator, ":", between hours and minutes.
 - $([0-5][0-9])$: Accept all digit strings from 00 to 59 (with leading zeroes).
 - $([AP]M)$: Accept "AM" or "PM" with one preceding space to indicate the time meridiem.

3. A regular expression that matches a string representing a comma separated list of variable names such as: `hello, get_max, sum3`

- A variable name consists of letters, digits, and underscores but cannot start with a digit.
- There is exactly one space after every comma. No other spaces are allowed.
- Commas and spaces are not allowed before the first name and after the last name.
- An empty string is considered a match.

Solution

- Answer: `^$|^(([a-zA-Z_]\w*)(,))*([a-zA-Z_]\w*)$`
- Explanation:
 - `^$`: Accept empty string.
 - `([a-zA-Z_]\w*)`: Accept variable names that match with the requirements. `\w` is equivalent to `[a-zA-Z0-9_]`.
 - `(([a-zA-Z_]\w*)(,))*`: As the previous one, but additionally accept previous variable names followed by exactly one comma and one space, if there are more than one variable name.

Part III. Turing Machines

Design a Turing machine on the input alphabet $\{x, y, z\}$ that removes all z characters from the input such there are no gaps. If the input string is $xzzyxzy$, the output should be $xyxy$.

- Hint: The final string does not need to reside on the same part of the tape where it started.

Solution

