ASSIGNMENT 10

Xinhao Luo

Tuesday 12$^{\text{th}}$ May, 2020

1 DMC Problem 25.8(f)

i) $S \to A0A$

ii) $A \to 0A1|1A0|AA|0A|\varepsilon$

2 DMC Problem 26.5(d)

i) Pseudo Code

1) Check if the string is valid (not empty, no 1 included), and mark the first bit with ✗
2) Back to ✗, check if all bits after ✗are marked
   - If True, ACCEPT
   - Otherwise, move back to ✗
3) Move right and find the next non-marked bit, mark it
   - If ␣, REJECT
4) Move right the next non-mark bit, and goto step 3
   - If ␣, go to step 2

ii) Machine Code

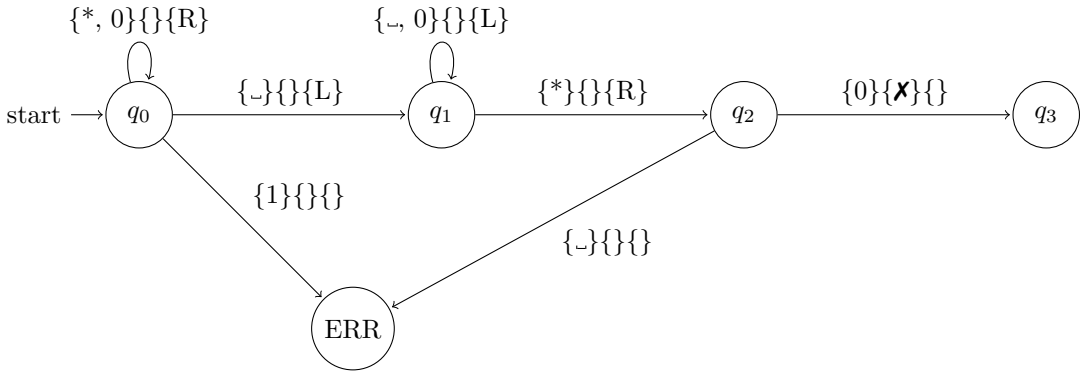

Figure 1: Step 1

1)

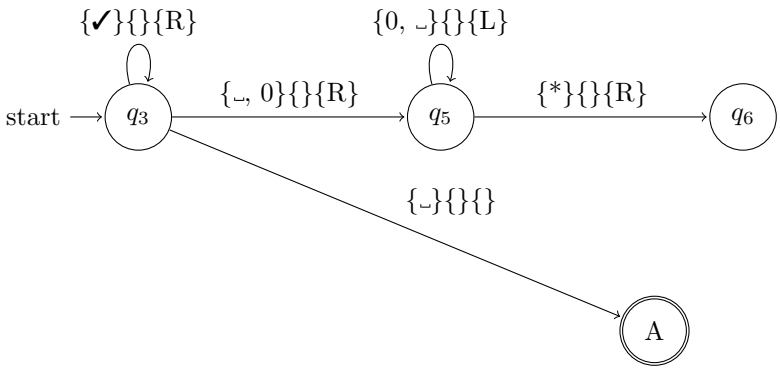

Figure 2: Step 2

2)

iii)

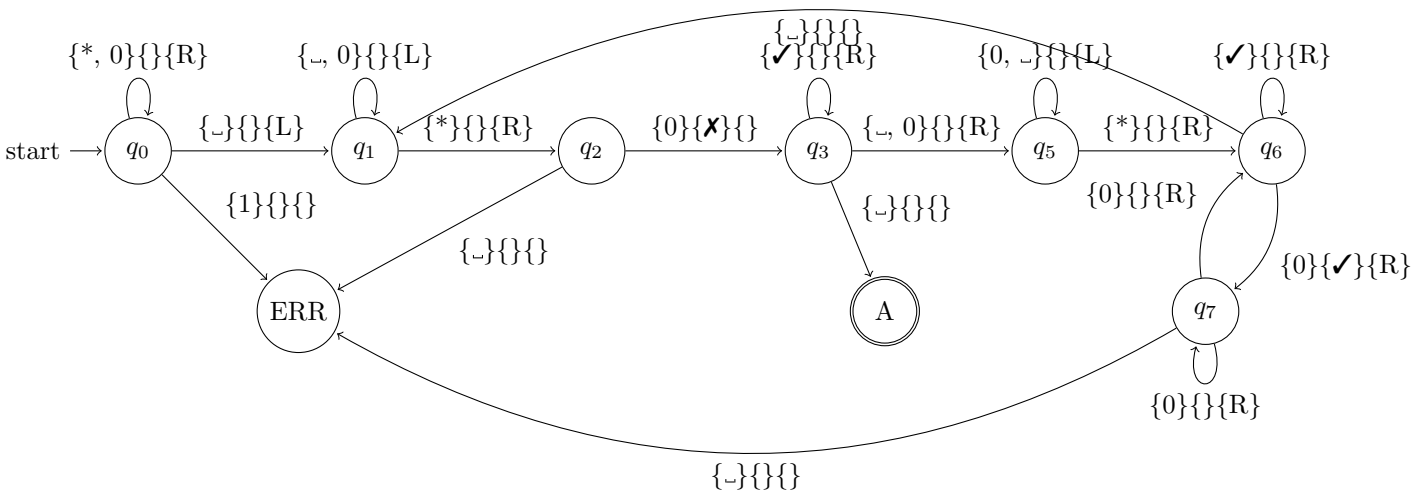

Figure 3: Final Version

3 DMC Problem 26.8(f)

1) Move right to the first non-marked bit, mark it and remember it.
    - if ␣, return to *; erasing all marks and halt
2) Move right to the last non-marked bit
    - If there is none, return to *, erasing all marks and halt
    - Otherwise, remember it, replace it with the bit from step 1 and mark it
3) Move left to the first marked bit
    - Replace the bit with the bit remembered in step 2 and goto step 1

4 DMC Problem 27.4(b)

Define function "Debugger(FUNCTION[, INPUT])", where INPUT is an optional input for the function, and it will return True when given function will halt with given input, False otherwise.

```
def findNext(i):
    if i is Prime and i + 2 is prime:
        return i
    else:
        findNext(i + 1)

def test():
    for j in Prime Number Set:
        if Debugger(findNext, j) is False:
            break
```

If statement "Debugger(test)" return True, then it proves that the number of twin prime is infinite; otherwise there will be finite number of pairs.

5 DMC Problem 27.20

a) B is also undecidable

b) Unsure about B

c) Unsure about A

d) A is decidable

6 DMC Problem 27.45

a) $3 \cdot d_1 + 4 \cdot d_2$

b) $d_1 \cdot 2 + d_3 + d_4 \cdot 2 + d_2 \cdot 2$

7 DMC Problem 27.46

If each domino has the same content on both top and bottom, it doesn't really matter what sequence or number of domino used, as it will always be a solution.
Input Format: $\alpha_1\#\alpha_2...\#\#\delta_1\#\delta_2...$
The sketch of the Turing machine is shown below:
INPUT: <S>The encoded domino strings.

1. Check that <S>is a valid encoding of dominoes set
2. Back to beginning; find and mark (✓) the first unmarked bit from the beginning and remember it
   - If #, goto step 8
   - If ␣, REJECT
3. Move right to find the double pun, and stop at the first pun
4. Move right and find the first unmarked bit
   - If the bit does not match the one in step 2, continue to step 5
   - If matched, mark (✓) it and move to step 2
5. Move right and mark the rest of the string with ✗until a pun is met
6. Move to beginning, and find the first non-marked bit
7. Move right and mark the rest of the string with ✗until a pun is met, then goto step 2
8. Move left, and check if string before before a pun or * was met has all marked (✓)
   - If true, ACCEPT
   - Otherwise, go back to step 2