Assignment #10 - Saaif Ahmed

Tuesday, December 10, 2019 7:44 PM

Problem 25.8

(f) Strings with more 0s than 1s.

Language : $\{0^{*\,n+1}\, 1^n\, 0^{k+1}\, 1^k \mid (n,k) \geq 0\}$

CFG:
$S \rightarrow 0$
$S \rightarrow 0S$
$S \rightarrow S10$
$S \rightarrow 10S$
$S \rightarrow 1S0$
$S \rightarrow 0S1$
$S \rightarrow S0$
$S \rightarrow S01$
$S \rightarrow 01S$

Assignment #10

Tuesday, December 10, 2019 7:53 PM

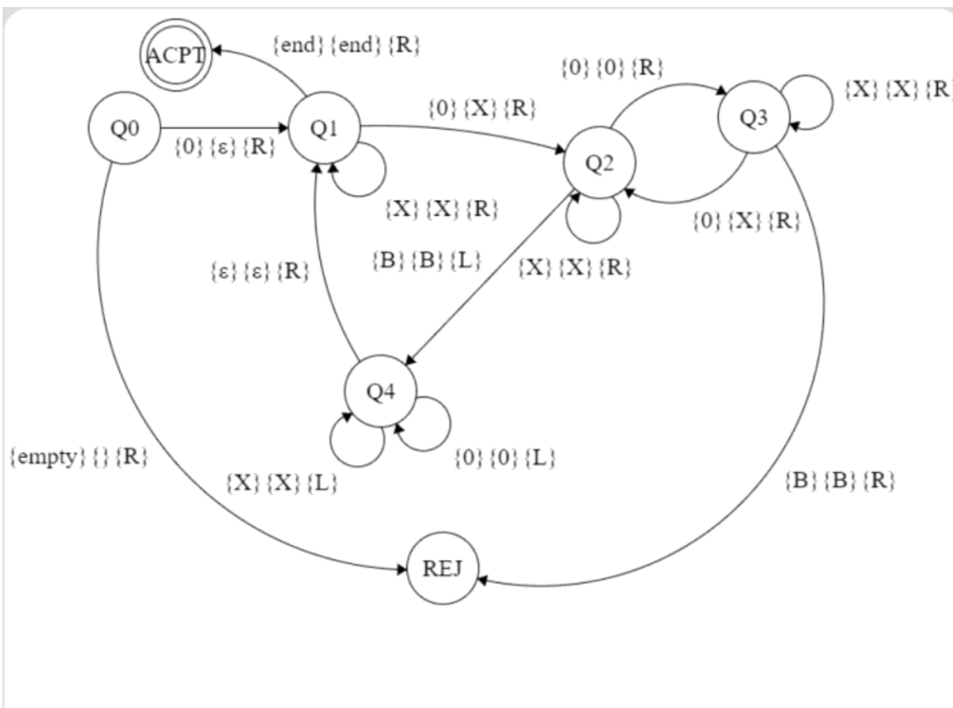
Problem 26.5

(d) $\mathcal{L} = \{0^{*2^n} \mid n \geq 0\}$

(i)

1: Accept String
2: Move to first unmarked 0 and replace it with ε
  If reached end of input REJECT and Halt
3: Move right to next unmarked 0 and replace with X
  If reached end of input ACCEPT and Halt
4: Move right to next unmarked 0 and Do nothing
  If reached end of input go to 7
5: Move right to next unmarked 0 and replace with X
  If reached end of input REJECT and HALT
6: Go to 4
7: Move left to ε
8: Go to 3

(ii) / (iii)



Assignment #10

Tuesday, December 10, 2019 8:10 PM

Problem 26.8

(f)

1: Accept String and verify valid input
2: Move right to end of input (#)
3: Move left to first unmarked bit and mark it
    If reach * then go to 8
4: Remember bit
5: Move right past # to first unmarked bit, write the remembered bit down by replacing what is on the tape, mark the bit with X
6: Move left to #
7: Go to 3
8: Return to *
9: Move right un-marking all bits until you reach #
10: Return to *
11: Move right to first unmarked bit, mark with an **P**
12: Move right past # to first bit marked with X, unmark it, mark it **Z**, remember bit
13: Move left to first **P** marked bit, replace with remembered bit
    If reach * then go to 15
14: go to 11
15: Move left to #
16: Move right to first bit marked **Z** and delete it
    if reached empty bit then HALT
17: go to 16

Assignment #10

Tuesday, December 10, 2019 10:37 PM

Problem 27.4

(b)

We describe a program used to resolve the twin primes conjecture

Code:

```
def not_infinite(array):
    try:
        count = 0
        for i in range len(array):
            count +=1
        return True
    except:
        return False



n = 0
twin_primes = []
while not_infinite(twin_primes):
    k = n+2

    if n is prime && k is prime:
        twin_primes.append( (n,k) )
    n+=1

print("Not Infinite")
```

We are given the ultimate debugger to prove the twin primes conjecture. This is how we use it on this piece of code to get our answer. The ultimate debugger will determine if the program halts. I have set it up such that the loop of finding twin prime pairs relies on generating an array of twin primes tuples. If the size of the set of all twin_prime numbers is indeed infinite then the last print statement will not appear. The ultimate debugger will tell us if program the program halts. If it halts then the set is finite. You can use another ultimate debugger on the try/except clause. The ultimate debugger will tell us if that for loop ever ends. If it doesn't end the debugger will run the except clause due to the error of infinite memory and the while loop will break.

Assignment #10

Tuesday, December 10, 2019 11:58 PM

Problem 27.20

- **(a)**
  Answer: You cannot say anything about B
- **(b)**
  Answer: B is decidable
- **(c)**
  Answer: A is undecidable
- **(d)**
  Answer: You cannot say anything about A

Assignment #10

Wednesday, December 11, 2019 12:01 AM

Problem 27.45

(a)

Answer: There is no possible solution. Any combination of d1 and d2 will always have the bottom string be at least 1 bit larger than the top.

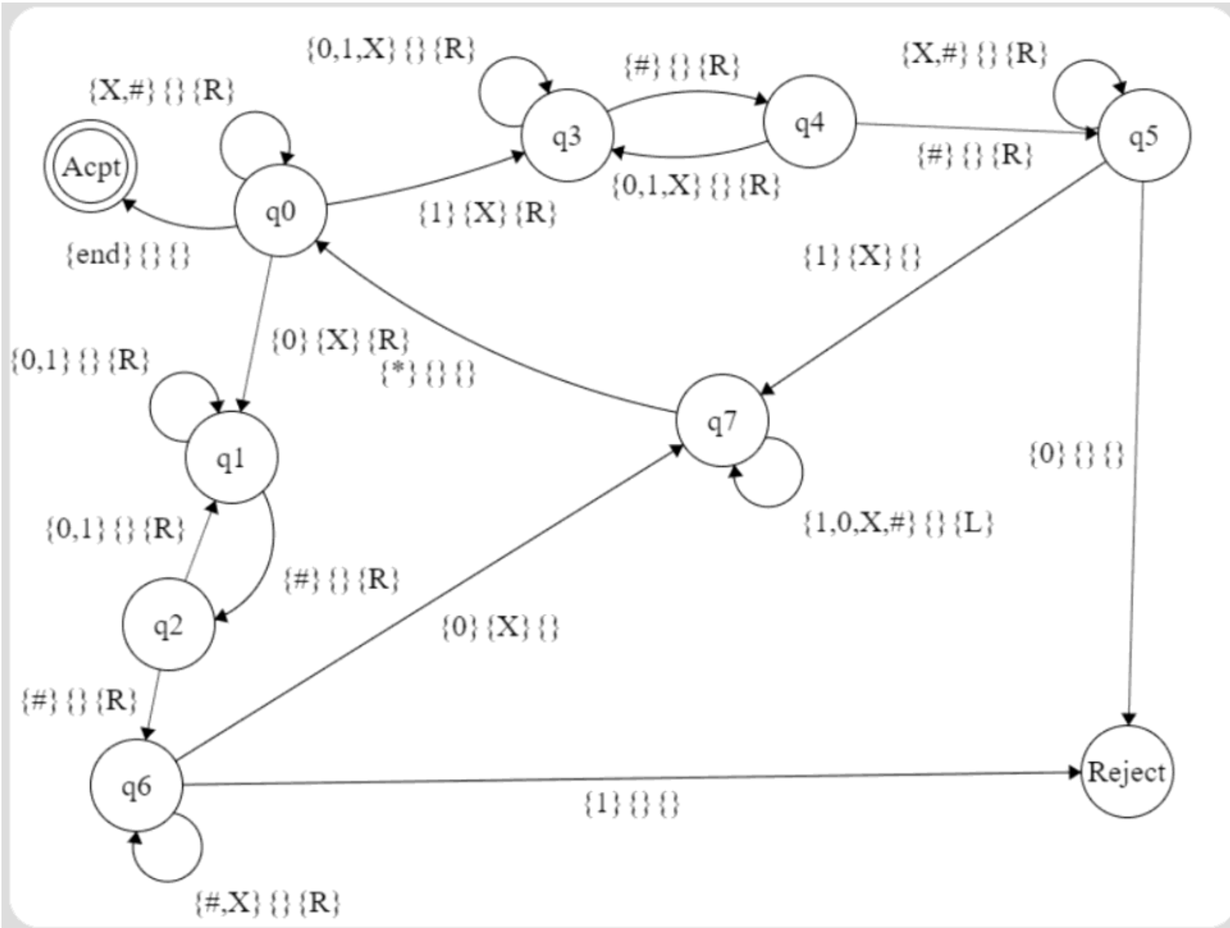
(b)

Answer: d1, d1, d3, d4, d4, d2, d2

Assignment #10

Wednesday, December 11, 2019 12:07 AM

Problem 27.46



Pseudo Code:

1. Verify all dominoes have equal top and bottoms
2. Very input is a1#...#aN##b1#...#bN