Formal Languages and Computability 7

Ragnar Björn Ingvarsson, rbi3

15. október 2024

3

We could achieve this by first creating 1# on the left side. Then, we could subtract the left number from the right number, check if the right number is all zeroes, and if not, increment the left number by 2 and repeat. Then, we could have an overflow checker and if it gets ticked, we reject.

This works on the basis that each perfect square is a sum of some sequential odd numbers,

$$1 = 1$$

$$4 = 1 + 3$$

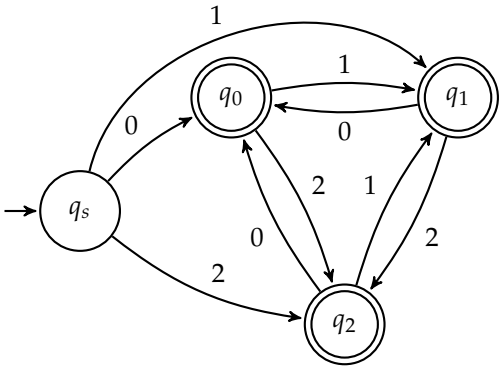
$$9 = 1 + 3 + 5$$

$$16 = 1 + 3 + 5 + 7$$

and so on.

4

a) We can show that A is regular by creating a NFA that describes the language.



Here we have a NFA that only accepts a string where from a recent 0,1,2 we can only accept numbers that are not the most recent number.

b) This can be solved by first reading the tape and determining if the string is in A and reject it if so, as per the question on ed, `https://edstem.org/eu/courses/1535/discussion/134577`. Then, we could create three spaces on the left side of the input string, for counting the occurrences of each symbol with a binary number. Then we run through the string, replacing each read symbol with say, an X, and incrementing the corresponding counter. At the end, we have our three binary numbers representing the number of symbols of each kind and we can compare the three to ensure that no symbol occurrs more than the other two combined plus 1.