

Contents

Course description	1
1 Exercises 2019	2
Week 1	2
page 84, question 1,6	2
2 Questions 2018	4
1. Finite automata and regular languages	5
2. Pushdown automata and context-free languages	6
3. Turing machines	7
4. Decidability	8
5. Reducibility	9
6. NP-completeness proofs – examples.	10
7. Proof that SATISFIABILITY is NP-complete (do not assume that there is a known NP-Complete problem — use the proof in Sipser’s book).	11
8. Information-theoretic lower bounds (lower bounds proven by counting leaves in decision trees), especially the average case bounds for sorting by comparisons.	12
9. Adversary arguments – technique, examples.	13
10. Median problem – algorithm and lower bound.	14
11. Approximation algorithms	15
3 Questions 2017	16
Questions from last year	16
1. Finite automata and regular languages	17
2. Pushdown automata and context-free languages	18
3. Turing machines	18
4. Decidability	18
5. Reducibility	18
6. NP-completeness proofs – examples.	18
7. Proof that SATISFIABILITY is NP-complete.	18
8. Information-theoretic lower bounds (lower bounds proven by counting leaves in decision trees), especially the average case bounds for sorting by comparisons.	18
9. Adversary arguments – technique, examples.	18
10. Approximation algorithms	18
4 Lectures	19
Format	19
28-feb-2018 - TBD	19
22 march	19
assignment 2	19
assignment 3	19
assignment 5	19
10th of April	19
12th of April	20
CNE-SAT is NP-Complete	20
Subset-sum is NP-Complete	20
17th of April	20
1st of may	20

Course description

Chapter 1

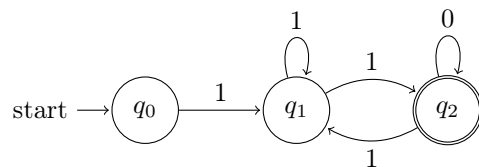
Exercises 2019

Week 1

page 84, question 1,6

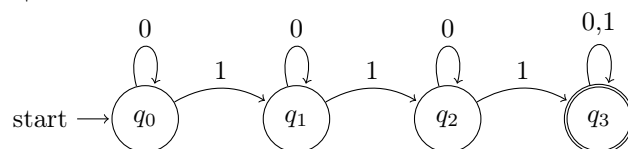
a

$w|w$ begins with a 1 and ends with a 0



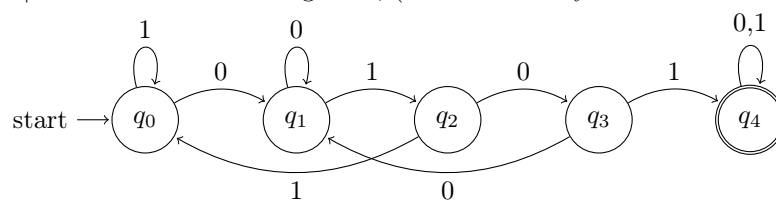
b

$w|w$ contains at least 3 1's



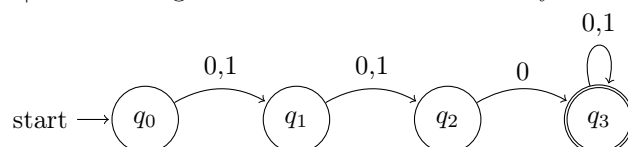
c

$w|w$ contains the substring 0101, (i.e. $w = x0101y$ for some x and y)



d

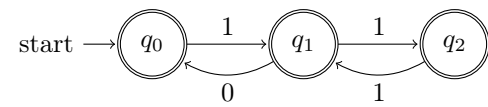
$w|w$ has at length of at least 3 and it's third symbol is 0



f

$w|w$ doesn't contain the substring 001

Accepts any string that doesn't contain the substring 001, and loops in rejecting state if this state is

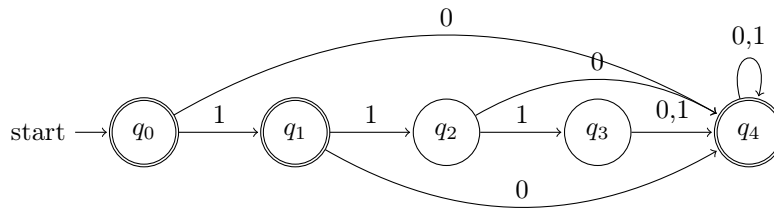


found, was unsure if the looping was needed but included it if it was the case.

h

w|w is any string except 11 and 111

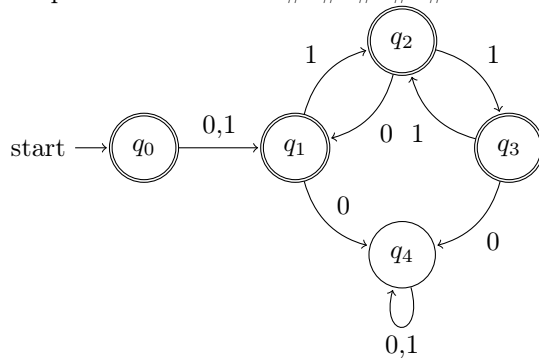
accepts any string that isn't 11, 111 including the empty string ϵ



i

w|w every odd position of w is a 1

accepts all states where #1#1#1#1#1 is followed also accepts the empty string ϵ



Chapter 2

Questions 2018

1. Finite automata and regular languages

Introduction What is it

Finite automata, regular languages

regular languages

Finite automata

Nondeterministic

NFA DFA

Pumping lemma for regular languages

2. Pushdown automata and context-free languages

What is a push down

What is a context free grammar $a \rightarrow 0a1$ $A \rightarrow B$ $B \rightarrow \#$ ambiguity grammar variables

What can it be used to programming languages Compiler

Pushdown automata in dept You use a stack.

Pumping lemma Proof

Application

3. Turing machines

Introduction What is a Turing machine

Multitape

Nondeterministic Turing machine Faster but less powerful?

4. Decidability

Introduction If a language defined by a DFA is decidable.

Examples of Decidability and undecidability

5. Reducibility

What is it and how to use it.

6. NP-completeness proofs – examples.

what is np_completeness

Why we use it to reduce,

Proff of np complete. clique, subset sum, Hamiltonian circuit,

7. Proof that SATISFIABILITY is NP-complete (do not assume that there is a known NP-Complete problem — use the proof in Sipser's book).

Cook-levin theorem - insipset, præsentation use slides on homepage.

8. Information-theoretic lower bounds (lower bounds proven by counting leaves in decision trees), especially the average case bounds for sorting by comparisons.

As is average case.

9. Adversary arguments – technique, examples.

10. Median problem – algorithm and lower bound.

11. Approximation algorithms

Chapter 3

Questions 2017

Questions from last year

1. Finite automata and regular languages
2. Pushdown automata and context-free languages
3. Turing machines
4. Decidability
5. Reducibility
6. NP-completeness proofs – examples.
7. Proof that SATISFIABILITY is NP-complete.
8. Information-theoretic lower bounds (lower bounds proven by counting leaves in decision trees), especially the average case bounds for sorting by comparisons.
9. Adversary arguments – technique, examples.
10. Approximation algorithms.

1. Finite automata and regular languages

Introduction

Types of Automata

DFA - NDFA -

2. Pushdown automata and context-free languages

CFG(context free grammar) regular language defined by DFA ambiguity inherited ambiguity Chomsky normal form You're allowed to have a rule that turns A into two sub rules. $A \rightarrow BC$ You can have a rule about a terminal set for our alphabet. $A \rightarrow a \in \Sigma$ You're only allowed to produce Epsilon from the start symbol. $S \rightarrow \epsilon$ theorem Any CFL is generated by a CFG in Chomsky normal form, pushdown automata PDA(s) NFA with a stack.

3. Turing machines

4. Decidability

5. Reducibility

6. NP-completeness proofs – examples.

7. Proof that SATISFIABILITY is NP-complete.

8. Information-theoretic lower bounds (lower bounds proven by counting leaves in decision trees), especially the average case bounds for sorting by comparisons.

9. Adversary arguments – technique, examples.

10. Approximation algorithms

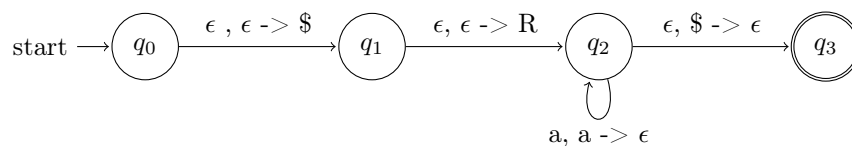
Chapter 4

Lectures

Format

Titles should be listed as (date - topic(s)) for easier lookup

28-feb-2018 - TBD



22 march

assignment 2

We can recognize that it is two regular languages, and that when we concatenate them we get a regular language. We can apply theorem 1.49 regular languages are closed under concatenation.

assignment 3

d)
use p from pumping lemma
 $(xy)^{3p}(x)^p$
q

assignment 5

a) prove by counter example
 $a^i b^i c^i | i \geq 0 \cup a, b, c^*$ b) prove by counter example $a^i b^i c^i | i \geq 0 \cup \emptyset$

10th of April

- Polynomial time reductions
- NP-completeness
- Examples of proofs
 - 3-SAT
 - CLIQUE
 - Vertex cover
 - Independent set

12th of April

CNE-SAT is NP-Complete

Cook-lenin thm

SAT is NP-Complete

Show that:

$$\forall A \in NP : A \leq_p SAT, \quad (4.1)$$

$A \in NP$, Let N be a polytime NTM which accepts it in time $d_1 n^k + d_2$

$$N = (Q, \Sigma, R, \alpha, q_o, q_{accept}, q_{reject}). \quad (4.2)$$

Let $W = W_1 W_2 \dots W_n$ be input to A ,

Construct (in polytime) a boolean formula F which is satisfiable if W is accepted by N ,

Look at accepting branch of computation tree

Look at sequence of configurations.

Subset-sum is NP-Complete

17th of April

Hamiltonian circuit is NP-complete

Conclusion on NP-complete

Information on theoretic lower bound technique.

1st of may

Approximation algorithms

δ -TSP has 2-approximation algs

For general TSP and a fixed P , Δ an alg with approx p

Vertex cover has a 2-approx alg.