Contents

Course description

Chapter 1

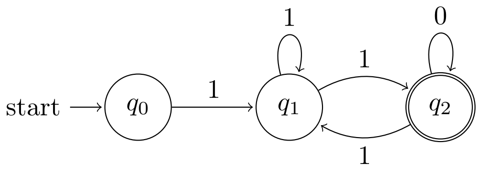
Exercises 2019

Week 1

page 84, question 1.7

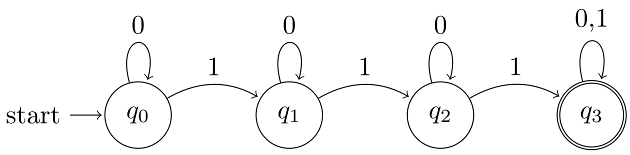
a

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



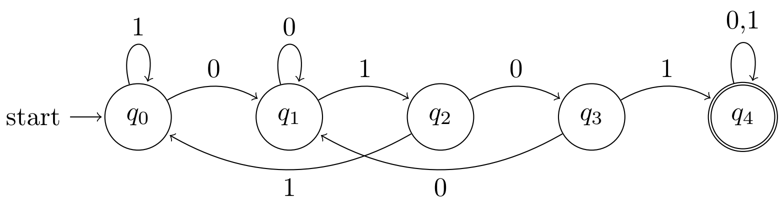
b

w|w contains at least 3 1's



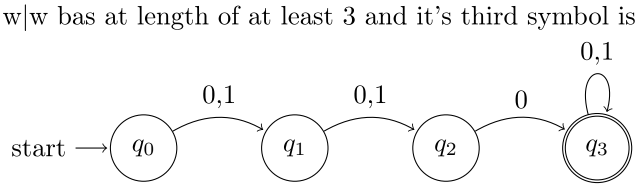
c

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



d

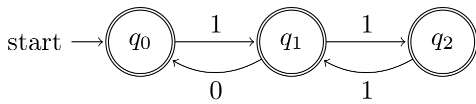
w|w bas at length of at least 3 and it's third symbol is 0



f

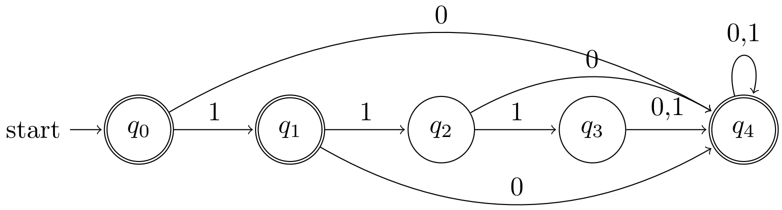
w|w doesn't contain the substring 001

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.



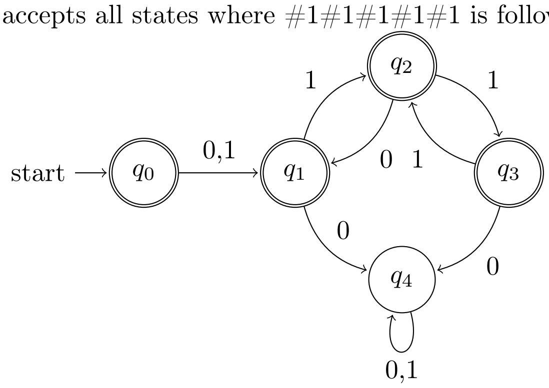
h

w|w is any string except 11 and 111
accepts any string that isn't 11, 111 including the empty string ϵ



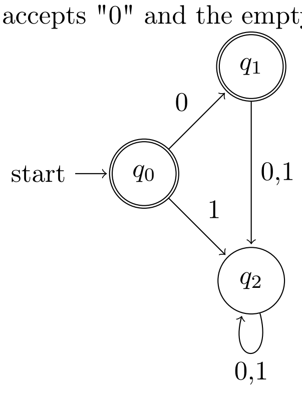
i

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



k

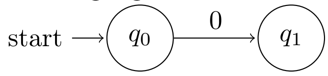
w|w is only the empty string and 0
accepts "0" and the empty string ϵ



page 84, question 1.7

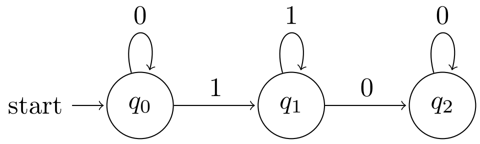
d

The language 0 with two states,



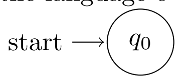
e

the language 0*1*0* with 3 states



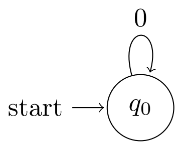
g

the language ϵ with 1 state



h

The language 0* with 1 state



Solve the following problem,

A man is travelling with a wolf (w) and a goat (g). He also brings along a nice big cabbage (c). He encounters a small river which he must cross to continue his travel. Fortunately, there is a small boat at the shore which he can use. However, the boat is so small that the man cannot bring more than himself and exactly one more item along (from w, g, c). The man knows that if left alone with the goat, the wolf will surely eat it and the goat if left alone with the cabbage will also surely eat that. The man's task is hence to device a transportation scheme in which, at any time, at most one item from w, g, c is in the boat and the result is that they all crossed the river and can continue unharmed.

a

Describe a solution to the problem which satisfies the rules of the "game". You may use your answer to (b) to find a solution.

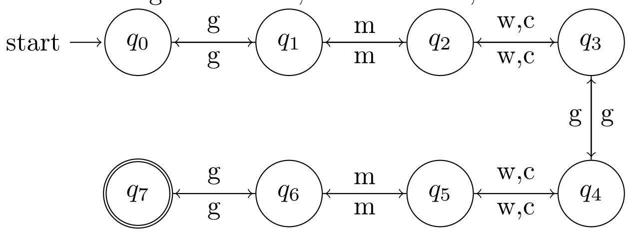
- First you carry the goat to the other side, and go back empty.
- The you ferry the wolf to the other side, and swap with the goat and bring the goat back.
- you then swap the goat with the cabbage and bring it to the other side.
- lastly you head back empty and bring the goat.
- you now have all the items on the other side of the river.

b

The string all of the valid moves are

$$\begin{aligned}(g(m(x(g(y(m(g)*)*)*)*)*)*)*\\x,y=(w|c)\\x\neq y\end{aligned}\tag{1.1}$$

This is due to the fact that it's legal moves in the game, like the man can bring the goat to the other side and bring it back too, it's bad move, but it's still a valid move.



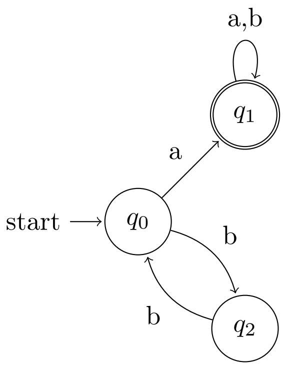
Week 2

page 86, question 1.16

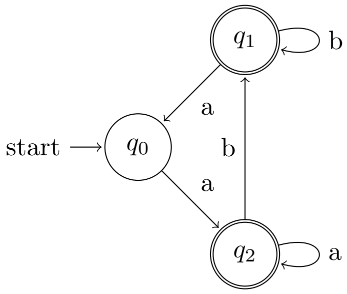
* is any number
! is one or more

a

it accepts (bb)*a!(a|b)*
the language 0*1*0* with 3 states



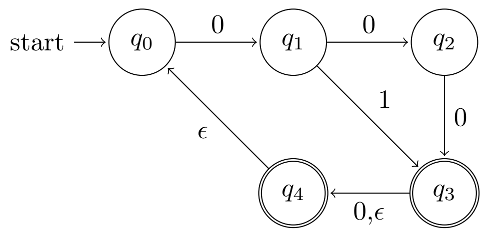
b



page 86, question 1.17

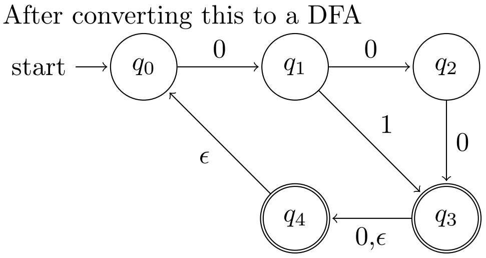
a

The first task is to make a NFA recognizing (01u001u010)*



b

After converting this to a DFA



page 86, question 1.18

Predefined terms

$$x = (0,1)^* \tag{1.2}$$

$$y = (0|1) \tag{1.3}$$

a

$1x0$

b

$x1x1x1x$

c

$x0101x$

d

$yy0x$

e

$(0(yy)*)|1y(yy)^*)$

f

$0^*(1+10)^*$

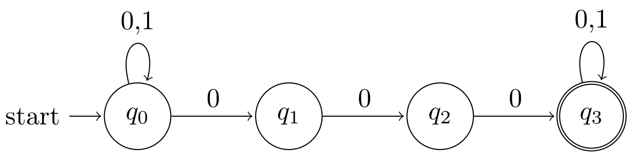
g

$yyyyy$

page 86, question 1.19 a

Convert the following regex to a NFA via lemma 1.55

$$(0 \cup 1)^*000(0 \cup 1)^* \tag{1.4}$$



page 86, question 1.20

a

$$a^*b^* \tag{1.5}$$

member aa, ab, aabb, aab, abbbb not member ba, abab,

b

member abab, abababab, not member aaba baba

c

member ab, aabb, not memeber ba, aabba

d

member aaa, aaaaaa not memeber a, aaaa

e

member
not memeber

f

member
not memeber

g

member ab, b not memeber abb, aab

h

member
not memeber

page 86, question 1.21 b

The solution is

$$((a|b)(a, bb) * b(a)?)* \tag{1.6}$$

(a or b, followed by any number of (a,bb)* followed by a single b (as it matches uneven numbers of b and followed by 0 or 1 a*

page 86, question 1.29

a

For the task a we have the sting

$$0^n1^n2^n \tag{1.7}$$

the first contradiction is that when we have a string eg. 000111222 and we split it so we have the floowing, x = 000, y = 111, z = 222 and we pump y so we have the string xyyz this violates the first condition of the pumping lemma, as we'll have more 1s then 0s and 2s.
the string y only contains 1s which also causes a contradiction.
and the 3rd case if we have the string x = 00, y = 011 and z = 1222 where we'll get out of order letters so we'll again reach a contradiction.

b

For assignment b i find it a bit odd, i may misunderstand the exercise but w/e
We want to pump the language

$$a_2 = \{www|w \in \{a,b\}^*\} \tag{1.8}$$

But from my understanding is the * a klein star? eg then one w and www are equivalent as {a,b} is all possibly strings in the library w. ? or is it understand such that w is equal to either a or b but any number of them eq $\{a|b\}^*$,
How i choose to intercept it for now is that w is equal to either the string a* or b* and if this is the case then some of the same argumentation as for task a is valid.
www can give us the string 111000111 and we can split it as follows. 111|1000|11111, This means that zyyx gives of a out of order 1 and we therefor get a contradiction wrt. to the pumping lemma.

Chapter 2

Questions 2018

1. Finite automata and regular languages

Introduction What is it
Finite automata, regular languages
regular languages aboveabove Finite atomata
Nondetermanistic
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 -> 0a1 A -> B B -> # ambiguity grammar variables
What can it be used to programming languages Compiler
Pushdown atomata in dept You use a stack.
Pumping lemma Proof
Application

3. Turing machines

Introduction What is a turning machine
Multitape
Nondetermanistic turning machine Faster but less powerfull?

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. qlique, 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 -> BC You can have a rule about a terminal set for our alphabet. $A- > a \in \sum \epsilon$ You're only allowed to produce Epsilon from the start symbol. $S- > \epsilon$ 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

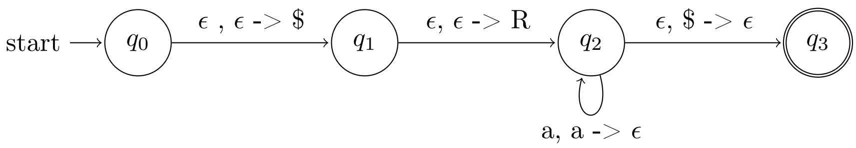
Chapter 4

Lectures

Format

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

28-feb-2018 - TBD



22 march

assignment 2

We can reconnize that it is two regular languages, and that when we concatinate them we get a reglang sigma star is regular,
we can apply theom 1.49 regular langs are closed under concat.

assignemnt 3

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

assignemnt 5

a) prove by counter exsample
$a^ib^ic^i|i \geq 0 \cup a, b, c^*$ b) prove by counter exsample $a^ib^ic^i|i \geq 0 \cup \varnothing$

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:

$$\forall A \in NP : A \leq_p SAT, \tag{4.1}$$

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

$$N = (Q, \sum, R, \alpha, q_o, q_a ccept, q_r eject). \tag{4.2}$$

Let $W = W_1 W_z * W_n$ be input to A,
Crate (in polytime) a boolean formula F which is satisfiable if W is accepted by N,
Look at accepting breach 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, Δan alg with approx p
Vertex cover has a 2-approx alg.