!!!

COMP4141 Slides Week 1 (W17A)

Brendan Mabbutt

UNSW

February 17, 2026

Admin Matters

Tutorials

- I will typically go through questions by demand
- There will also be time in the tutorials to work on the questions with others
- Starting week 2, there will be 20 minutes of the tutorial dedicated to a testing assignment (unfortunately, at the time of writing this, I do not know much about it).

Homework

- I will be the person to mark you homework and give you feedback on it
- Feel free to follow up in these tutorials about any question you have from my marking
- This will be done on formatif. There will be target grades (PS, CR, DN, HD) for the task so that you only complete the ones **up to** your target grade.
  - You must complete PS tasks before CR tasks.
  - You must complete CR tasks before DN tasks.
  - You must complete DN tasks before HD tasks.
- Please use LaTeX for your submission, though I don't believe your forced to.
- First deadline is Monday, the week after. Then if there are any issues, you are given the chance to resubmit the next Monday.

Need help in the course?

Ask a lot of questions!!!

- (please) **do** annoy me about it
- Ask on the Moodle forum(s)
- Use to textbook; *Introduction to the Theory of Computation* by Michel Sipser. I usually don't read textbooks by myself enjoyed this one :) Other textbook that you can find on Moodle are also good.
- Ask questions in the lectures

Github Repository

I will try to upload slide annotations on Github. Link:
`https://github.com/brendanmabbutt/COMP4141-W17A`

Advice for the course

- I may not speak for everyone but this course is light on workload but heavy on content. It is very important you spend time digesting the course content (e.g. reading the slides, attempting the tutorial question before the tutorial or reading through the textbook).
- Have fun with the course; the content can be very interesting.
- This is a very math (proof) heavy course so be sure to revise relevant MATH1081 content.

Automata Tutor

Will be needed this week (see here).

Extra

- If you have an ELS, please email me or talk to me about it in this tutorial if required
- have fun and work with others; that is really important in a problem solving crouse like this.

Alphabets, String & Languages

Alphabets & Strings

Alphabet

An *alphabet* is a non-empty finite set. The members of the alphabet are called symbols.

Strings over Σ

Base: ϵ is a string over Σ

Induction: If x is a string over Σ and a is a symbol from Σ, then ax is a string over Σ.

Language

A language over Σ is a subset of Σ^*.

Length and Concatenation

Definition (Length)

The length of a string is defined recursively.

Base: $|\epsilon| = 0$
Induction: $|ax| = 1 + |x|$

Definition (Concatenation)

Concatenation is defined recursively on the structure of the first string.

Base: $\epsilon \cdot x = x$ if x is a string over Σ.
Induction: If x and y are strings over Σ and $a \in \Sigma$, then

$$(ax) \cdot y = a(x \cdot y).$$

DFAs and NFAs

Deterministic Finite Automata (DFA)

Definition

A deterministic finite automaton (DFA) is a tuple

$$(Q, \Sigma, \delta, q_0, F)$$

where

- Q is a finite set of states,
- Σ is the input alphabet,
- $\delta : Q \times \Sigma \longrightarrow Q$ is the transition function,
- $q_0 \in Q$ is the start state,
- $F \subseteq Q$ is the set of accepting states.

Language of a DFA

Extended Transition Function and Acceptance

The extended transition function $\hat{\delta}$ is defined recursively:

Base: $\hat{\delta}(q, \epsilon) = q$
Induction: $\hat{\delta}(q, aw) = \hat{\delta}(\delta(q, a), w)$

Definition. A DFA accepts a string $w \in \Sigma^*$ iff

$$\hat{\delta}(q_0, w) \in F.$$

Non-Deterministic Finite Automata (NFA)

Definition

A non-deterministic finite automaton (NFA) is a tuple

$$(Q, \Sigma, \delta, q_0, F)$$

where

- Q is a finite set of states,
- Σ is the input alphabet,
- $\delta : Q \times \Sigma \longrightarrow 2^Q$ is the transition relation,
- $q_0 \in Q$ is the start state,
- $F \subseteq Q$ is the set of accepting states.

Runs and Language of an NFA

Run and Acceptance

A run of an NFA A on a word $w = a_1 \dots a_k$ is a sequence $q_0 q_1 \dots q_k$ of states in Q such that:

- q_0 is the initial state,
- for all $i = 1, \dots, k$, we have $q_i \in \delta(q_{i-1}, a_i)$.

The run is accepting if $q_k \in F$.

$$L(A) = \{ w \mid \text{there exists an accepting run of } A \text{ on } w \}.$$

Expressive Power of NFAs and DFAs

Theorem

A language L is $L(N)$ for some NFA N if and only if it is $L(D)$ for some DFA D.

ϵ-Nondeterministic Finite Automata

Definition

An ϵ-NFA is an NFA that allows state changes that do not consume input symbols.
The only difference in the formal definition is the transition relation:

$$\delta : Q \times (\Sigma \cup \{\epsilon\}) \to 2^Q.$$

ϵ-Closure

Definition

For $q \in Q$, define $E(q)$ to be the least set satisfying:

- $q \in E(q)$,
- if $s \in E(q)$ and $t \in \delta(s, \epsilon)$, then $t \in E(q)$.

For a set $S \subseteq Q$, define

$$E(S) = \bigcup_{q \in S} E(q).$$

P1

Problem 1

Let Σ be an alphabet containing symbol a, and, for $x \in \Sigma^*$, let $P(x)$ be the proposition that:

$$\text{For all strings } y \in \Sigma, |x.y| = |x| + |y|$$

Prove that $P(x)$ holds for all $x \in \Sigma^*$.

We prove $P(x)$ by induction on x

Base: Look at $x = \varepsilon$

Want: $|\varepsilon \cdot y| = |\varepsilon| + |y|$

By def $\varepsilon \cdot y = y$

By def $|\varepsilon| = 0$

$$|\varepsilon \cdot y| = |y|$$

$$|y| + |\varepsilon| = |y|$$

Inductive step:

Assume $\forall y$ that

$|x \cdot y| = |x| + |y|$ ... (IH)

want $|(ax) \cdot y| = |ax| + |y|$

Def $\forall b \in \Sigma,\ x, y \in \Sigma^*$

$$(bx) \cdot y = b(x \cdot y) \quad \cdots \text{①}$$

Def $\forall c \in \Sigma,\ z$

$$|cz| = 1 + |z| \quad \cdots \text{②}$$

LHS: $|(ax)\cdot y| = |a(x\cdot y)|$ (by 1)

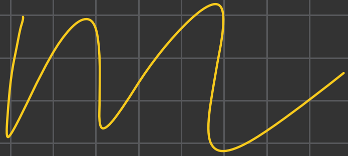
$= 1 + |x\cdot y|$ by 2

$= \underbrace{1 + |x|} + |y|$ by IH

$= |ax| + |y|$

Hi

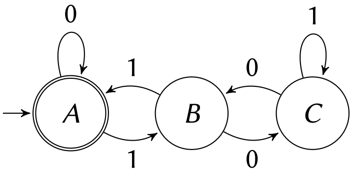
Hi



P2

Problem 2

Consider the following DFA over the alphabet $\Sigma = \{0, 1\}$:



(a) Which of the following words are in the language of the DFA:

(i) 010110101

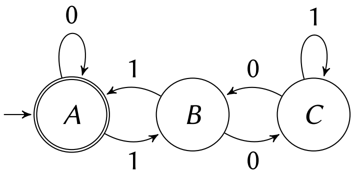
(ii) 110101010

(iii) ϵ

$$\hat{\delta}(q, ax) = \hat{\delta}(\delta(q, a), x)$$

Problem 2

Consider the following DFA over the alphabet $\Sigma = \{0, 1\}$:



$\delta(A, 0) = A$

(a) Which of the following words are in the language of the DFA:

(i) 010110101

(ii) 110101010

(iii) ϵ

$\hat{\delta}(A, 010110101)$

$= \hat{\delta}(A, 10110101)$

$$= \hat{\delta}(B, 0110101)$$

$$= \hat{\delta}(C, 110101)$$

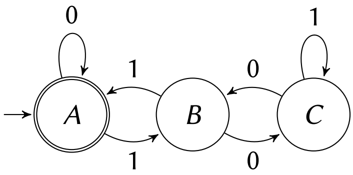
$$= \hat{\delta}(C, 0101$$

$$= \hat{\delta}(A, 01)$$

$$= B$$

Problem 2

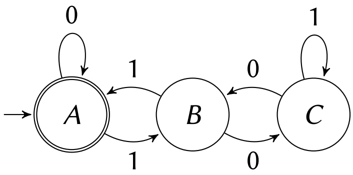
Consider the following DFA over the alphabet $\Sigma = \{0, 1\}$:



(b) Give a description of the language of the NFA.

Problem 2

Consider the following DFA over the alphabet $\Sigma = \{0, 1\}$:



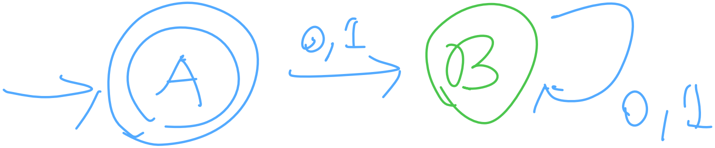
(c) Use the subset construction to construct a DFA that accepts the same language

P3

Problem 3

Let $\Sigma = \{0, 1\}$. Give DFAs over Σ that accept the following languages:

(a) $\{\epsilon\}$



Problem 3

Let $\Sigma = \{0, 1\}$. Give DFAs over Σ that accept the following languages:

(b) $\{w : w \text{ contains at least two 0s}\}$

Problem 3

Let $\Sigma = \{0, 1\}$. Give DFAs over Σ that accept the following languages:

(c) $\{w : w \text{ contains an odd number of 1s}\}$

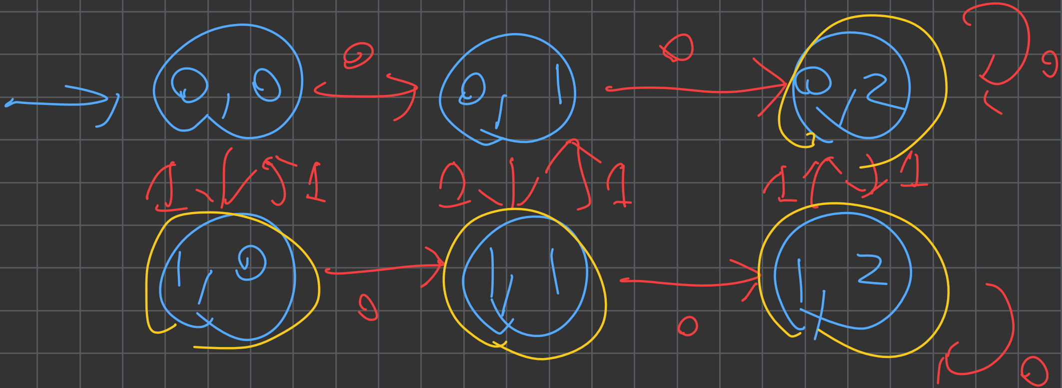
Problem 3

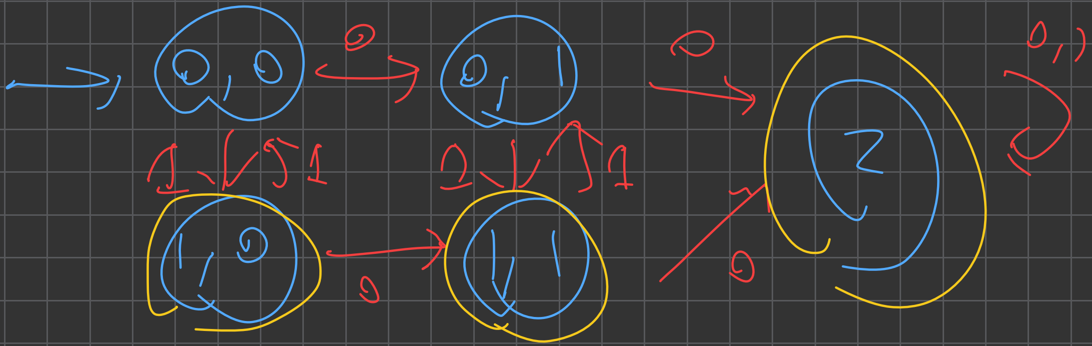
Let $\Sigma = \{0, 1\}$. Give DFAs over Σ that accept the following languages:

(d) $\{w : w$ contains at least two 0s or an odd number of 1s$\}$

State i, j (i is the parity of #1
j is number of 0s
(when $j = 2$ it's $\geq$)

(d) $\{ w : w$ contains an odd # of 1
or at least 2 0s $\}$





0, 0
0, 1
1, 0
1, 1
2
0
1
0, 1

Problem 3

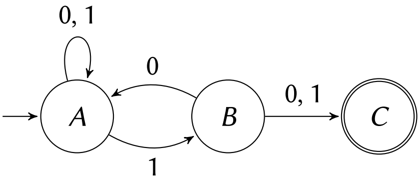
Let $\Sigma = \{0, 1\}$. Give DFAs over Σ that accept the following languages:

(e) $\{0^m 1^n : m, n \in \mathbb{N}\}$

P4

Problem 4

Consider the following NFA over the alphabet $\Sigma = \{0, 1\}$:



(a) Which of the following words are in the language of the ~~D~~FA: N

(i) 010110101

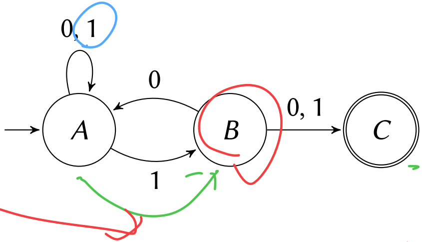
(ii) 110101010

(iii) ϵ

Problem 4

Consider the following NFA over the alphabet $\Sigma = \{0, 1\}$:

Run that does not accept



(a) Which of the following words are in the language of the ~~D~~FA: ABC

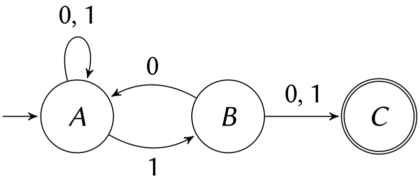
(i) 010110101 AAABC AAAAAAABAB N

(ii) 110101010

(iii) ϵ

Problem 4

Consider the following NFA over the alphabet $\Sigma = \{0, 1\}$:



(a) Which of the following words are in the language of the ~~D~~FA:
N

(i) 010110101

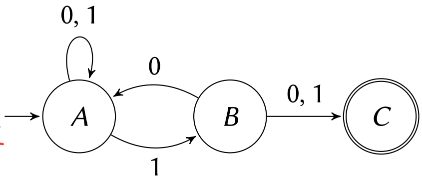
(ii) 110101010

(iii) ϵ

Problem 4

Consider the following NFA over the alphabet $\Sigma = \{0, 1\}$:

$N :=$

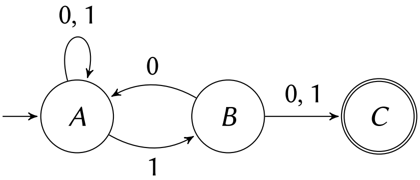


(b) Give a description of the language of the NFA.

$L(N) = \{ w : w$ has a 1 in the 2nd last position $\}$

Problem 4

Consider the following NFA over the alphabet $\Sigma = \{0, 1\}$:

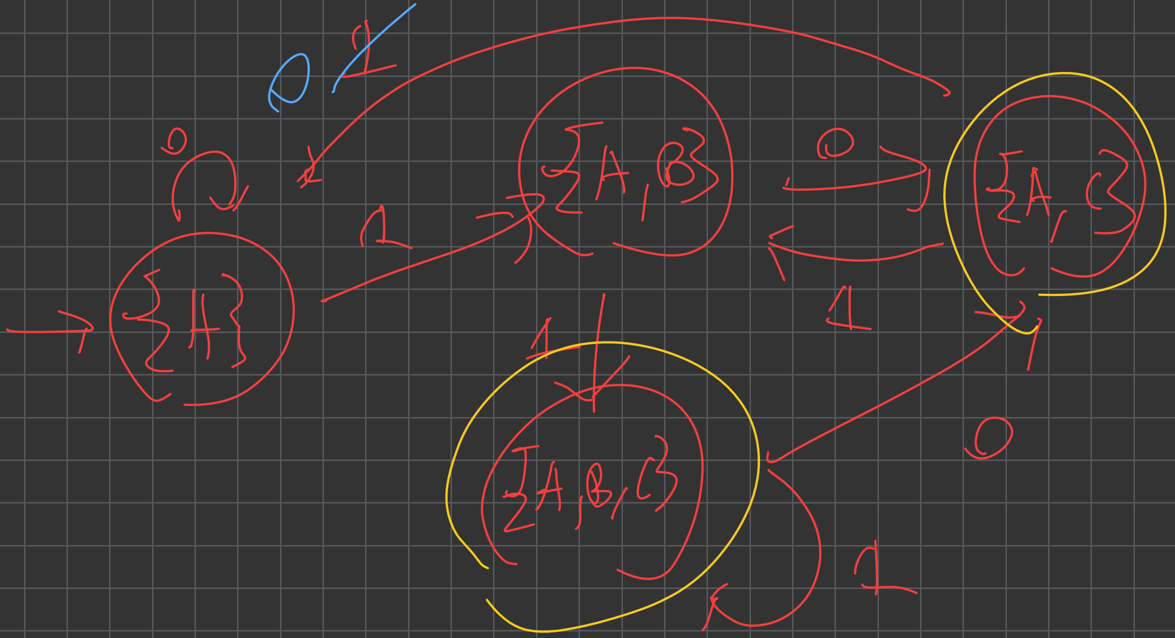


(c) Use the subset construction to construct a DFA that accepts the same language.

Make DFA with

$Q' = 2^Q$ where Q is

from of NFA



{A}
{A,B}
{A,C}
{A,B,C}
0
1

P5

Problem 5

Let $\Sigma = \{0, 1\}$. Give NFAs over Σ that accept the following languages. Try to minimize the number of states.

(a) $\{\epsilon\}$

Problem 5

Let $\Sigma = \{0, 1\}$. Give NFAs over Σ that accept the following languages. Try to minimize the number of states.

(b) $\{w : w \text{ contains at least two 0s}\}$

Problem 5

Let $\Sigma = \{0, 1\}$. Give NFAs over Σ that accept the following languages. Try to minimize the number of states.

(c) $\{w : w \text{ contains an odd number of 1s}\}$

Problem 5

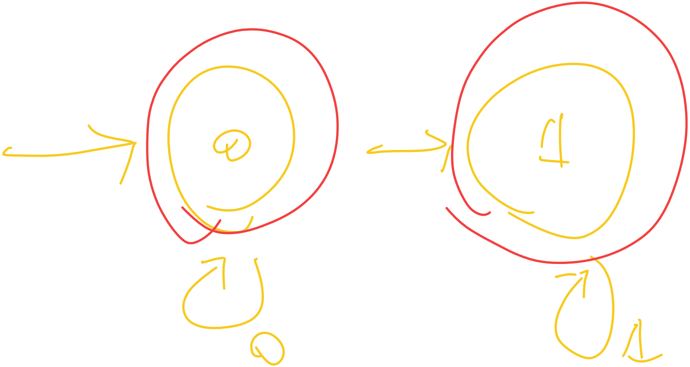
Let $\Sigma = \{0, 1\}$. Give NFAs over Σ that accept the following languages. Try to minimize the number of states.

(d) $\{w : w \text{ contains at least two 0s or an odd number of 1s}\}$

Problem 5

Let $\Sigma = \{0, 1\}$. Give NFAs over Σ that accept the following languages. Try to minimize the number of states.

(e) $\{0^m 1^n : m, n \in \mathbb{N}\}$



P6

Problem 6

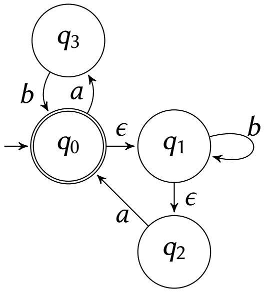
Let A be an NFA, and let A' be the NFA that results from swapping the accepting and non-accepting states of A.
Is it true that

$$L(A') = \Sigma^* \setminus L(A)\,?$$

P7 & P8

Problems 7 & 8 (ϵ-Nondeterministic Finite Automata)

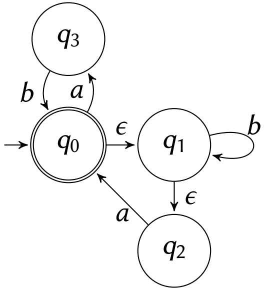
Assume the alphabet $\Sigma = \{a, b\}$. Consider the following ϵ-NFA.



7. List all strings of length ≤ 3 that this automaton accepts.

Problems 7 & 8 (ϵ-Nondeterministic Finite Automata)

Assume the alphabet $\Sigma = \{a, b\}$. Consider the following ϵ-NFA.



8. Using the subset construction, build a DFA that accepts the same language. Give the graphical representation of this DFA, clearly indicating the set of NFA states associated to each DFA state. Show only the reachable states.

Feedback

Please let me know how I can improve for next week here!

COMP4141 26T1 Feedback Form
(For Brendan Mabbutt's Tutorials)