Mathematical Foundations for Software Engineering

Course notes

Attila Matolcsy

25th August 2023 - Present

Contents

1 About

This document is Attila Matolcsy's own personal notes from the University of Gothenburg's Software Engineering and Management Bsc. Programme's Mathematical Foundations for SEM course.

These are all the notes from lessons, unprocessed.

WARNING

This document has typos in it, please use the main document that is cleaned up!

2 28^{th} August 2023

2.1 Introduction to the course

Lessons are not mandatory, nor the TA sessions. We join a TA group on Canvas, we are asked not to jump between them. The gropus should not have more than 8 persons / group.

Workload $\approx 200h$

Groups up to 3 are allowed but submissions must be on an individual basis.

FAQ is available on Canvas

You can send christian.berger@gu.se an email about problems that come up during class.

Course literature is available on Canvas, non of which is mandatory.

2.2 Logic

Logic = Meaning of mathematical statements $\wedge$ basis of reasoning

Applications = design of computing machines

0s & 1s.

Proofs = matheatical argument, essential for programs
= Security of systems

Theorem = Proven mathematical statements

Propositions = declerative statemnt that is either True or False

For e.g.:
- Stockholm is the capital of Sweden
- $4x5 = 20$
- $\pi \approx \frac{22}{7}$

Counter e.g.:
- attend my lectures
- $x + 1 = \pi$

Propositions are named by letters: p, q, r, s

field of propositional logic = the field that deals with propositionals

mathematical statements can be compined $\Rightarrow$ compoind propositions

Def1: $\neg p$ = not p

Def2: Conjuction: pq

p	q	$p \wedge q$	$\neg(A \wedge B)$
T	T	T	F
F	T	F	T
T	F	F	T
F	F	F	T

(From now on I will refer to Trues (Ts) as 1 and Falses (Fs) as 0)

Def 3: $\vee$ is logical or

p	q	$p \vee q$
1	1	1
1	0	1
0	1	1
0	0	0

Def 4: exclusive or

p	q	$p \oplus q$	$\neg (p \oplus q)$
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	1

Def 5: implication / conditional statements

p	q	$p \Rightarrow q$	$\neg p \vee q$	$\neg p$
1	1	1	1	0
1	0	0	0	0
0	1	1	1	1
0	0	1	1	1

Def 6: Biconditional state Example: ”There is a smallese regardless of values of the propositions

Contardiction: always false regardless of values of the propositions

contingency: $\neg$ (tantology) $\wedge \neg$ (contradiction)

equivalents:

$$\begin{aligned} p \wedge 1 &\equiv p \\ p \vee 1 &\equiv 1 \\ p \wedge 0 &\equiv 0 \\ p \vee 0 &\equiv p \end{aligned}$$

$p \wedge q \equiv p$

p	q	$p \wedge q$
1	1	1
0	0	0

negation laws:

$p \vee \neg p \equiv 1$

p	$\neg p$	$p \vee \neg p$
1	0	1
0	1	1

$p \wedge \neg p \equiv 0$

p	$\neg p$	$p \wedge \neg p$
1	0	0
0	1	0

3 29th August 2023

3.1 Binary counting

...	p	q	r	s
x increases from right to left: 2^x This is what the binary number represents:	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
This is our binary number:	$0\,(F)$	$1\,(T)$	$0\,(F)$	$1\,(T)$
We need to sum this: $0 + 4 + 0 + 1 = 5$	$0 \cdot 8 = 0$	$1 \cdot 4 = 4$	$0 \cdot 2 = 0$	$1 \cdot 1 = 1$

3.2 Predicate Logic

Propositional logic: everything is atomic

Predicate logic: Formalism of propositional logic is extended and more complicated expressions are possible to be ued for formal inference.

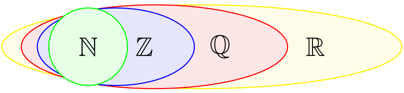
Predicate logic contains:

- all components from propositional logic
- terms (E.g.: Alice likes BOS [Underlined is a term])
- quantifiers:
  - something is always True
  - something is sometimes True
  - something is never True
- Predicate symbols: P, Q, R
- functions: f, g
- quantifiers: $\forall, \exists$
- identity: $=$

Example: "There is a smallest number."

Collection of al persons, ideas, symbols, data structures that affect the logical argument under consideration

Elements of the inverese of the discourse are called individuals



...

Order of args is important

unary predicate: "x is a cat" binary predicate: "y is mother of y"

unary predicates descripbe properties of object: $p\,(x) \Rightarrow$ x has property P

interpret as: of a predicate (P) in a set of objects (called A) is a set og those elements of A (AKA subset) that have property p:

$\{\alpha \in A \quad | \quad P\,(\alpha)\}$

$\{(\alpha, \beta) \in A \times A \quad | \quad Q\,(\alpha, \beta)\}$

Atomic formula

1. predicate name + arguments
2. an identity $t_1 = t_2$

Atomic formulas are statements that can be combined woth logical connectioves

$M\left(j,p\right) \rightarrow \neg M\left(p,j\right)$

If Jane is Paul's mother, then Paul is not Jame's mother.

If all arguments of a predicate are individual constants, the resulting formula must be Ⓣ or Ⓕ

Example:

	Bob	Jane	Alice	Paul
Bob	F	F	F	F
Jane	F	F	T	T
Alice	F	F	F	F
Paul	F	F	F	F

$M(a,b)$

Inly method that assigns truth values to all possible combinations of individuals is called assignment of the predicate

Universal quantifier: $\forall$

something is true for all infividuals

Existential quantifier: $\exists$

something is for at least one individual

$\forall \times A$ $\forall =$ for all
$\times$ is bound by the quantifier
A is the scope

Everone gets a break once in a while

$B(x) :=$ "x gets a break in a while"
$\forall \times B\left(x\right)$

Some people don't eat meat

$p(x) :=$ "x eats meat"
$\exists \times \left(\neg P\left(x\right)\right)$

$\exists x\left(\forall y k\left(x,y\right)\right)$

$\neg \exists x A\left(x\right) \equiv \forall x \neg A$

4 Sick break

Between the previous and next notes I was unable to make notes due to being sick

5 5^{th} September 2023

FSM: Finite state machines

$M = (S, I, O, f, g, S_0)$

S = states
I = input alphatbeth
O = output alphatbeth
f = transition function
g = function that assigns each pair of state and input a corresponding output.
s_0 = starting state

$f : I \times S \to s$ $g : I \times S \to s$

Example:

FSM $S - \{s_0, s_1, s_2, s_3\}$
$I = \{0, 1\}$
$O = \{0, 1\}$

state	f: input 0	f: input 1	g: input 0	g: input 1
s_0	s_1	s_0	1	0
s_1	s_2	s_0	1	1
s_2	s_1	s_2	0	1
s_3	s_2	s_1	0	0

6 7th September 2023

{"4","20","3"}
$\phi = \{ \}$
$\lambda =$ ""

Exercise 1 Find a phrase-structure grammar for each of these languages.

1. The set consisting of the bit strings 0, 1, and 11.

No, you can't just write {1}+

2. All sets of bit strings containing only 1s.

3. All sets of bit strings eith 0s in the beginning and 1s at the end.

4. All sets of bit strings that starts with a s 0 followed nu am ebem mu,ber of 1s or none

7 11th September 2023

Regex	$R = a$	matches aa
	$R = aa$	matches aa
	$R = a+$	matches more than 0 a's
	$R = a\|b$	matches a or b
	$R = a + \|b+$	matches sequencis of a or b
	$R = (a\|b)+$	matches

8 12th September 2023

Graph Theory

- Graph
- Vertices
- Edges
- Directed Graph
- Indegree ($-$)
- Outdegree ($+$)