Discrete Mathematics with Applications I

COMPSCI&SFWRENG 2DM3

McMaster University, Fall 2019

Wolfram Kahl

2019-09-03

What is This Course About?

- Calendar description:

  > Functions, relations and sets; the language of predicate logic, propositional logic; proof techniques, counting principles; induction and recursion, discrete probabilities, graphs, and their application to computing.

- Logic, sets, functions, relations, counting, trees, graphs: **Discrete Mathematics**
- Calculus is the mathematics of **continuous** phenomena: physical sciences, traditional engineering
- **Discrete Mathematics** is
  - the math of data— **whether complex or big**
  - the math of reasoning— **logic**
  - the math of AI— **machine reasoning**
  - **the math of software**

Goals and Rough Outline

- Understand the mechanics of mathematical expressions and proof
  — starting in a familiar area: **Reasoning about integers**
- Develop skill in **propositional calculus**
  - "**propositional**": statements that can be true or false,
    not numbers

Does Superman Exist?

(Textbook p. 37, p. 89)

If Superman were able and willing to prevent evil, he would do so. If Superman were unable to prevent evil, he would be impotent; if he were unwilling to prevent evil, he would be malevolent. Superman does not prevent evil. If Superman exists, he is neither impotent nor malevolent. Therefore, Superman does not exist.

"**Propositions**": statements that can be true or false, e.g.:

- Superman is able to prevent evil.
- Superman prevents evil.
- Superman is impotent.
- Superman exists.

Raymond Smullyan posed many puzzles about an island that has two kinds of inhabitants:

- **knights**, who always tell the truth, and
- **knaves**, who always lie.

You encounter two people A and B.
What are A and B if

1. A says "We are both knaves."?
2. A says "At least one of us is a knave."?
3. A says "If I am a knight, then so is B."?
4. A says "We are of the same type."?
5. A says "B is a knight" and
   B says "The two of us are opposite types."?

Portia's Suitor's Dilemma

(Textbook p. 86)

Portia has a **gold** casket and a **silver** casket
and has placed a picture of herself in one of them.
On the caskets, she has written the following inscriptions:

Gold: ***The portrait is not in here***

Silver: ***Exactly one of these inscriptions is true.***

Portia explains to her suitor that each inscription may be *true* or *false*, but that she has placed her portrait in one of the caskets in a manner that is consistent with the truth or falsity of the inscriptions.

If the suitor can choose the casket with her portrait,
she will marry him.

Goals and Rough Outline

- Understand the mechanics of mathematical expressions and proof
  — starting in a familiar area: **Reasoning about integers**

- Develop skill in **propositional calculus**
  - "**propositional**": statements, not numbers
  - "**calculus**": **formalised** reasoning, **calculation**

Portia's Suitor's Dilemma

Portia has a gold casket and a silver casket
and has placed a picture of herself in one of them.
On the caskets, she has written the following inscriptions:

G: ***The portrait is not in here*** $G \equiv \neg gc$

S: ***Exactly one of these inscriptions is true.*** $S \equiv (S \equiv \neg G)$

If the suitor can choose the casket with her portrait,
she will marry him.

- **Formalisation** is the first step towards solution

gc :≡ The portrait is in the gold casket
G :≡ The inscription on the gold casket is true
S :≡ The inscription on the silver casket is true

Portia's Suitor's Dilemma

Portia has a gold casket and a silver casket and has placed a picture of herself in one of them. On the caskets, she has written the following inscriptions:

G: ***The portrait is not in here*** $G \equiv \neg gc$

S: ***Exactly one of these inscriptions is true.*** $S \equiv (S \equiv \neg G)$

If the suitor can choose the casket with her portrait, she will marry him.

- **Formalisation** is the first step towards solution

gc :≡ The portrait is in the gold casket
G :≡ The inscription on the gold casket is true
S :≡ The inscription on the silver casket is true

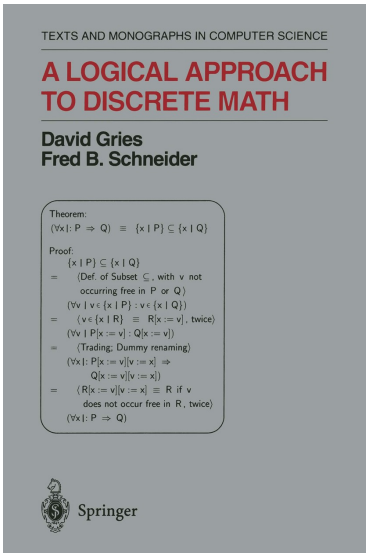
$S \equiv S \equiv \neg G$ — This is Def. S: Formalisation of silver casket inscr.
$=$ ⟨ (3.2) Symmetry of $\equiv$ ⟩
$\neg G$
$=$ ⟨ Def. G: Formalisation of gold casket inscription ⟩
$\neg\neg gc$
$=$ ⟨ (3.12) Double negation ⟩
gc

- **Calculation** — no case analysis, easy to check

Goals and Rough Outline

- Understand the mechanics of mathematical expressions and proof
  — starting in a familiar area: **Reasoning about integers**
- Develop skill in **propositional calculus**
  - "**propositional**": statements, not numbers
  - "**calculus**": **formalised** reasoning, **calculation**
- Develop skill in **predicate calculus**
  - "predicate": statement about some subjects.
- Make first acquaintance with **reasoning about programs**
- . . . *skill development takes time and effort* . . .
- Become familiar with **basic theories of discrete mathematics**
  - Sets, Functions, Relations
  - Sequences, Graphs
- Encounter mechanised discrete mathematics

Textbook: "LADM"



"This is a rather extraordinary book, and deserves to be read by everyone involved in computer science and — perhaps more importantly — software engineering. I recommend it highly [. . .]. If the book is taken seriously, the rigor that it unfolds and the clarity of its concepts could have a significant impact on the way in which software is conceived and developed."

— Peter G. Neumann

First Tool: CalcCheck

- CalcCheck: A proof checker for the textbook logic
- CalcCheck analyses textbook-style presentations of proofs
- **You can check your proofs before handing them in!**
- **Will be used in exams!**
  — with proof checking turned off. . .
  . . . but syntax checking left on
  — We help with syntax problems
- **Will be used in exams**
  — **as far as possible. . .**
  You need to be able to do both:
  - Write formalisations and proofs using CalcCheck
  - Write formalisations and proofs by hand on paper

Organisation

- Schedule
- Grading
- Exams
- Avenue
- Course Page: `http://www.cas.mcmaster.ca/~kahl/CS2DM3/2019/`

— See the Outline (on course page and on Avenue)

— Read the Outline!

Rough Timeline

• Introduction to Calculational Reasoning	Parts of Chapters 1, 15	
Boolean Expressions and Propositional Logic	Chapters 1–5	≈ 4 weeks
Quantification and Predicate Logic	Chapters 8–9	
• Predicates and Programming	Chapter 10	≈ 1 week
• Sets	Chapter 11	≈ 3 weeks
Relations and Functions	Chapter 14	
• Induction and Sequences	Chapters 12–13	≈ 1.5 weeks
• Graphs, Counting	Chapters 19, 16	≈ 2.5 weeks

You will always need everything you have learned so far!

Schedule

- **Lectures:** Tuesday, Wednesday, Friday 12:30–13:20, CNH-104

  — attend! **— take notes!**
- **Office hour:** *preliminary:* Wed.. 14:00–15:00, and by appointment

  — or catch me after lecture
- **Studying and Homework:** About 2 hours per lecture

  — includes reading the textbook

  — and writing proofs in CalcCheck$_{\text{Web}}$
- **Tutorials** (starting **Thursday, September 5**):

T8	Th	11:30–12:20	JHE-210	T1	Mo	15:30–16:20	JHE-210
T5	Fr	11:30–12:20	KTH-109	T3	Tu	10:30–11:20	JHE-210
T7	Mo	12:30–13:20	JHE-210	T4	Tu	13:30–14:20	JHE-210
T2	Mo	13:30–14:20	JHE-210	T6	We	13:30–14:20	JHE-210

- **TA office hours** (starting **Wednesday, September 4**): Mo-Fr, TBA

Grading

- **Homework**, from one lecture to the next — in total: **8%**
  - The weakest $\frac{1}{8}$ are dropped
  - MSAFs for homework are not processed
- **Weekly assignments** — in total: **12%**
  - The two weakest assignment marks are dropped
  - MSAFs for assignments are not processed
- **2 Midterm Tests**, closed book, **on computers in the UTS labs**, **each:**
  - **12%** if not better than your final
  - **20%** if better than your final

  — in total at least: **24%**
  — in total up to: **40%**
- **Final** (closed book, 2.5 hours, **possibly on computers**) **40%–56%**

= **100%**

- Possible **bonus assignments** — only count if you passed the course

Exams

- Exercise questions, assignment questions, and the questions on midterm tests, and on the final —

  — will be somewhat similar. . .
- All exams are **closed book**.
- **You need to be able and prepared to do both:**
  - Write formalisations and proofs using CALCCHECK
  - Write formalisations and proofs by hand on paper
- **Know your stuff!** — . . . and not only in the exams . . .

  — . . . and not only for this term . . .

  — . . . similar to learning a new language

The Language of Discrete Mathematics

The mathematical foundations of Computing and Software involve **language skills and knowledge**:

- **Vocabulary:** Commonly known concepts and technical terms
- **Syntax/Grammar:** How to produce complex statements and arguments
- **Semantics:** How to relate complex statements with their meaning
- **Pragmatics:** How people actually use the features of the language

Conscious and fluent use of the **language of (discrete) mathematics** is the foundation for **precise specification and rigorous reasoning** in **Computer Science and Software Engineering**.