Discrete Mathematics & Mathematical Reasoning
Course Overview

Colin Stirling

Informatics

Teaching staff

Lecturers:

- Colin Stirling, first half of course
- Kousha Etessami, second half of course

Course Secretary (ITO):

- Kendall Reid (kreid5@staffmail.ed.ac.uk)

Course web page on Learn and at

http://www.inf.ed.ac.uk/teaching/courses/dmmr/

Course web page on Learn and at

http://www.inf.ed.ac.uk/teaching/courses/dmmr/

Contains important links to

- Lecture schedule and slides
- Study guide (textbook reading)
- Weekly tutorial exercises
- Coursework
- Tutorial groups
- Discussion forum (piazza) not yet available
- Course organization
- ...

Lectures

- Monday 16.10-17.00 Here
- Tuesday 10.00-10.50 Here
- Thursday 16.10-17.00 Here

Lectures

- Monday 16.10-17.00 Here
- Tuesday 10.00-10.50 Here
- Thursday 16.10-17.00 Here

- 10 weeks of lectures in two halves of 5 weeks
- Lecture schedule and slides (like this one) on web page
- Study guide (textbook reading)

Textbook

- Kenneth Rosen, **Discrete Mathematics and its Applications**, 8th Edition, (Global Edition) McGraw-Hill, 2018 (or 7th Edition 2012)
- Available at Blackwells (may be a delay)
- For additional material see the course webpage

Tutorials

- See link on Learn for current assignment of tutorial groups (20 groups)

Tutorials

- See link on Learn for current assignment of tutorial groups (20 groups)
- Once times are decided, if you want to change group get in touch with the ITO to do this

Tutorials

- See link on Learn for current assignment of tutorial groups (20 groups)
- Once times are decided, if you want to change group get in touch with the ITO to do this
- Tutorials start week 2 (next week beginning 23 September)

Tutorials

- See link on Learn for current assignment of tutorial groups (20 groups)
- Once times are decided, if you want to change group get in touch with the ITO to do this
- Tutorials start week 2 (next week beginning 23 September)
- Tutorial attendance is mandatory

Tutorials

- See link on Learn for current assignment of tutorial groups (20 groups)
- Once times are decided, if you want to change group get in touch with the ITO to do this
- Tutorials start week 2 (next week beginning 23 September)
- Tutorial attendance is mandatory
- At tutorials discuss the weekly exercise sheets (available previous wednesday) on course web page

Tutorials

- See link on Learn for current assignment of tutorial groups (20 groups)
- Once times are decided, if you want to change group get in touch with the ITO to do this
- Tutorials start week 2 (next week beginning 23 September)
- Tutorial attendance is mandatory
- At tutorials discuss the weekly exercise sheets (available previous wednesday) on course web page
- You should answer the questions in advance and discuss solutions with tutor and fellow tutees

Tutorials

- See link on Learn for current assignment of tutorial groups (20 groups)
- Once times are decided, if you want to change group get in touch with the ITO to do this
- Tutorials start week 2 (next week beginning 23 September)
- Tutorial attendance is mandatory
- At tutorials discuss the weekly exercise sheets (available previous wednesday) on course web page
- You should answer the questions in advance and discuss solutions with tutor and fellow tutees
- Sample solutions are available the following week

Coursework

- There are two courseworks which are pen and paper exercises (similar to the weekly tutorial exercises) to be handed in to the ITO (Informatics Teaching Organisation, 6th floor of Appleton Tower)

Coursework

- There are two courseworks which are pen and paper exercises (similar to the weekly tutorial exercises) to be handed in to the ITO (Informatics Teaching Organisation, 6th floor of Appleton Tower)
- Each coursework is out of 50

Coursework

- There are two courseworks which are pen and paper exercises (similar to the weekly tutorial exercises) to be handed in to the ITO (Informatics Teaching Organisation, 6th floor of Appleton Tower)
- Each coursework is out of 50
- Courseworks will be available at least two weeks before their deadlines

Coursework

- There are two courseworks which are pen and paper exercises (similar to the weekly tutorial exercises) to be handed in to the ITO (Informatics Teaching Organisation, 6th floor of Appleton Tower)
- Each coursework is out of 50
- Courseworks will be available at least two weeks before their deadlines
- Coursework 1 due on Monday 21st Oct, Week 6

Coursework

- There are two courseworks which are pen and paper exercises (similar to the weekly tutorial exercises) to be handed in to the ITO (Informatics Teaching Organisation, 6th floor of Appleton Tower)
- Each coursework is out of 50
- Courseworks will be available at least two weeks before their deadlines
- Coursework 1 due on Monday 21st Oct, Week 6
- Coursework 2 due on Friday 22nd November, Week 10

Coursework

- There are two courseworks which are pen and paper exercises (similar to the weekly tutorial exercises) to be handed in to the ITO (Informatics Teaching Organisation, 6th floor of Appleton Tower)
- Each coursework is out of 50
- Courseworks will be available at least two weeks before their deadlines
- Coursework 1 due on Monday 21st Oct, Week 6
- Coursework 2 due on Friday 22nd November, Week 10
- You will pick up your marked scripts from the ITO once marked

Grading

- Written examination in December: 85% OPEN BOOK EXAM

Grading

- Written examination in December: 85% OPEN BOOK EXAM
- Exam structure: compulsory part A of 5 questions 10 marks each; part B choose 2 out of 3 (25 marks each)

Grading

- Written examination in December: 85% OPEN BOOK EXAM
- Exam structure: compulsory part A of 5 questions 10 marks each; part B choose 2 out of 3 (25 marks each)
- Assessed coursework: 15%. Each coursework is $7\frac{1}{2}\%$

Grading

- Written examination in December: 85% OPEN BOOK EXAM
- Exam structure: compulsory part A of 5 questions 10 marks each; part B choose 2 out of 3 (25 marks each)
- Assessed coursework: 15%. Each coursework is $7\frac{1}{2}\%$
- To pass course need 40% or more overall
- No separate exam/coursework hurdle

Time management

- This is a 20 credit course (twice 10 credit courses)

Time management

- This is a 20 credit course (twice 10 credit courses)
- This means about 200 hours in total; 15 hours per week and 50 hours exam preparation

Time management

- This is a 20 credit course (twice 10 credit courses)
- This means about 200 hours in total; 15 hours per week and 50 hours exam preparation
- What should you do in a typical week?
  - 4 hours classes (3 lectures + 1 tutorial)

Time management

- This is a 20 credit course (twice 10 credit courses)
- This means about 200 hours in total; 15 hours per week and 50 hours exam preparation
- What should you do in a typical week?
  - 4 hours classes (3 lectures + 1 tutorial)
  - 4 hours tutorial preparation (for weekly exercise sheet)

Time management

- This is a 20 credit course (twice 10 credit courses)
- This means about 200 hours in total; 15 hours per week and 50 hours exam preparation
- What should you do in a typical week?
  - 4 hours classes (3 lectures + 1 tutorial)
  - 4 hours tutorial preparation (for weekly exercise sheet)
  - 2 hours per week on coursework

Time management

- This is a 20 credit course (twice 10 credit courses)
- This means about 200 hours in total; 15 hours per week and 50 hours exam preparation
- What should you do in a typical week?
  - 4 hours classes (3 lectures + 1 tutorial)
  - 4 hours tutorial preparation (for weekly exercise sheet)
  - 2 hours per week on coursework
  - 5 hours background; reading textbook and understanding slides

Time management

- This is a 20 credit course (twice 10 credit courses)
- This means about 200 hours in total; 15 hours per week and 50 hours exam preparation
- What should you do in a typical week?
  - 4 hours classes (3 lectures + 1 tutorial)
  - 4 hours tutorial preparation (for weekly exercise sheet)
  - 2 hours per week on coursework
  - 5 hours background; reading textbook and understanding slides
- Extra help at INFBASE

Questions about course administration?

Syllabus

- mathematical reasoning
- combinatorial analysis
- discrete structures
- algorithmic thinking
- applications and modelling

Foundations: proof

- Rudimentary predicate (first-order) logic: existential and universal quantification, basic algebraic laws of quantified logic (duality of existential and universal quantification)

- The structure of a well-reasoned mathematical proof; proof strategies: proofs by contradiction, proof by cases; examples of incorrect proofs (to build intuition about correct mathematical reasoning)

Foundations: sets, functions and relations

- Sets (naive): operations on sets: union, intersection, set difference, the powerset operation, examples of finite and infinite sets (the natural numbers). Ordered pairs, n-tuples, and Cartesian products of sets
- Relations: (unary, binary, and n-ary) properties of binary relations (symmetry, reflexivity, transitivity).
- Functions: injective, surjective, and bijective functions, inverse functions, composition of functions
- Rudimentary counting: size of the Cartesian product of two finite sets, number of subsets of a finite set, (number of n-bit sequences), number of functions from one finite set to another

Induction and recursion

- Principle of mathematical induction (for positive integers)
- Examples of proofs by (weak and strong) induction

Basic number theory and some cryptography

- Integers and elementary number theory (divisibility, GCDs and the Euclidean algorithm, prime decomposition and the fundamental theorem of arithmetic)
- Modular arithmetic (congruences, Fermat's little theorem, the Chinese remainder theorem)
- Applications: public-key cryptography

Counting

- Basics of counting
- Pigeon-hole principle
- Permutations and combinations
- Binomial coefficients, binomial theorem, and basic identities on binomial coefficients
- Generalizations of permutations and combinations (e.g., combinations with repetition/replacement)
- Stirling's approximation of the factorial function

Graphs

- Directed and undirected graph: definitions and examples in Informatics
- Adjacency matrix representation
- Terminology: degree (indegree, outdegree), and special graphs: bipartite, complete, acyclic, ...
- Isomorphism of graphs; subgraphs
- Paths, cycles, and (strong) connectivity
- Euler paths/circuits, Hamiltonian paths (brief)
- Weighted graphs, and shortest paths (Dijkstra's algorithm)
- Bipartite matching: Hall's marriage theorem

Trees

- Rooted and unrooted trees
- Ordered and unordered trees
- (Complete) binary (k-ary) tree
- Subtrees
- Examples in Informatics
- Spanning trees (Kruskal's algorithm, Prim's algorithm.)

Discrete probability

- Discrete (finite or countable) probability spaces
- Events
- Basic axioms of discrete probability
- Independence and conditional probability
- Bayes' theorem
- Random variables
- Expectation; linearity of expectation
- Basic examples of discrete probability distributions, the birthday paradox and other subtle examples in probability
- The probabilistic method: a proof technique

Questions about course syllabus?

My proof

Colin's proof that $1 = 2$

$a = b$	Premise
$a^2 = ab$	Multiply both sides by a
$a^2 - b^2 = ab - b^2$	Subtract b^2 from both sides
$(a-b)(a+b) = b(a-b)$	Algebra
$a+b = b$	Divide both sides by $a-b$
$2b = b$	Replace a by b because $a = b$
$2 = 1$	Divide both sides by b

Reasoning 1

Given the following two premises

- All students in this class understand logic
- Colin is a student in this class

Reasoning 1

Given the following two premises

- All students in this class understand logic
- Colin is a student in this class

Does it follow that

- Colin understands logic

Reasoning 2

Given the following two premises

- Every computer science student takes discrete mathematics
- Helen is taking discrete mathematics

Reasoning 2

Given the following two premises

- Every computer science student takes discrete mathematics
- Helen is taking discrete mathematics

Does it follow that

- Helen is a computer science student

Reasoning 3

Given the following three premises

- All hummingbirds are richly coloured
- No large birds live on honey
- Birds that do not live on honey are dull in colour

Reasoning 3

Given the following three premises

- All hummingbirds are richly coloured
- No large birds live on honey
- Birds that do not live on honey are dull in colour

Does it follow that

- Hummingbirds are small (that is, not large)?

Recall propositional logic from last year

Propositions can be constructed from other propositions using logical connectives

Recall propositional logic from last year

Propositions can be constructed from other propositions using logical connectives

- Negation: $\neg$
- Conjunction: $\wedge$
- Disjunction: $\vee$
- Implication: $\rightarrow$
- Biconditional: $\leftrightarrow$

Recall propositional logic from last year

Propositions can be constructed from other propositions using logical connectives

- Negation: $\neg$
- Conjunction: $\wedge$
- Disjunction: $\vee$
- Implication: $\rightarrow$
- Biconditional: $\leftrightarrow$

The truth of a proposition is defined by the truth values of its elementary propositions and the meaning of connectives

Recall propositional logic from last year

Propositions can be constructed from other propositions using logical connectives

- Negation: $\neg$
- Conjunction: $\wedge$
- Disjunction: $\vee$
- Implication: $\rightarrow$
- Biconditional: $\leftrightarrow$

The truth of a proposition is defined by the truth values of its elementary propositions and the meaning of connectives

The meaning of logical connectives can be defined using truth tables

Examples of logical implication and equivalence

- $(p \wedge (p \rightarrow q)) \rightarrow q$
- $(p \wedge \neg p) \rightarrow q$
- $((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$
- $\vdots$

Examples of logical implication and equivalence

- $(p \wedge (p \rightarrow q)) \rightarrow q$
- $(p \wedge \neg p) \rightarrow q$
- $((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$
- $\vdots$
- $(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$
  Contraposition
- $\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q)$ De Morgan
- $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$ De Morgan
- $\neg(p \rightarrow q) \leftrightarrow (p \wedge \neg q)$
- $\vdots$