Discrete Mathematics
Past Papers by topic

Logic

Logic used to be taught in more detail in previous versions of the course. In the current version of the course, logic appears in conjunction with other topics (e.g. set theory).
Logic:

- [**2013P1Q4**] rules for quantifier, existential, negation, prove first-order logic statements
- [**2012P1Q3**] prove propositional statements, rules for negation and falsity, contradiction
- [**2012P1Q4 (c),(d)**] rules for universal quantifier, first-order logic statement
- [**2011P1Q3**] rules for implication and negation, proofs for propositional statements
- [**2011P1Q4 (c)**] rules for universal quantifier
- [**2010P1Q3**]
- [**2010P1Q4**] logic to natural language
- [**2009P1Q3 (a),(b)**] (define implication, disjunction rules, prove two propositional formulas)
- [**2007P2Q6 (a)**]
- [**2006P2Q6 (a)**] prove using truth tables

Number theory

Greatest common divisor and Diophantine equations:

- [**2018P2Q7 (a)**] quadratic modular equation
- [**2017P2Q7 (a)**]
- [**2017P2Q8 (a)**] common divisor properties
- [**2016P2Q8 (a)**]
- [**2016P2Q9 (a)**]
- [**2015P2Q7 (a)**] prove that if $\gcd(m, n) = 1$, then there is no natural greater than 2 that divides any number
- [**2015P2Q9 (a)**]
- [**2014P2Q7 (c)**] find linear combination and multiplicative inverse
- [**2008P2Q4**] gcd, rule induction
- [**2007P2Q3 (b)-(f)**] prove $\gcd(a, b) = \gcd(a \mod b, b)$, derive Euclid's algorithm, determine the algorithm's efficiency, solve linear Diophantine equation, find multiplicative inverse)
- [**2006P2Q4**] define gcd, show that the gcd divides all possible linear combinations and all possible linear combinations are a multiple of the gcd, natural linear combinations
- [**2003P1Q2 (a)-(c)**] find the gcd between two numbers, solve linear Diophantine equation, find multiplicative inverse
- [**2000P1Q8**] prove Stein's algorithm works

Binomial coefficients:

- [**2015P2Q7**] Hockey stick identity
- [**1995P1Q1**] Vandermande's convolution using generating functions or a combinatorial argument

Fibonacci numbers:

- [**2018P2Q9 (a)**]
- [**2004P1Q7**] divisibility properties of Fibonacci numbers

RSA encryption:

- [**2001P1Q7 (c),(d)**]
- [**1998P1Q7 (b)**]

Fermat's Theorem:

- [**2014P2Q7 (b)**] state and prove application
- [**2006P2Q3**] Mersenne primes, infinitely many pseudo-primes
- [**2005P1Q2 (a),(b)**] state, deduce for $n = 2$, show that 341 is a pseudo-prime
- [**2002P1Q7**] Fermat's theorem and Diffie-Hellman protocol, Montgomery multiplication
- [**2001P1Q7 (a),(b)**] Euler's theorem, Fermat's theorem to show that a number is composite
- [**1999P1Q7 (c),(d)**]

Euler's totient function:

- [**2005P1Q7 (b),(c),(d)**] group of co-primes, definition, bijection, show that ϕ is multiplicative
- [**1999P1Q7 (a),(b)**] define and prove Fermat-Euler

Division algorithm/Modulo:

- [**2016P2Q7 (a)**]
- [**2015P2Q8 (a)**] 1365 divides $n^{13} - n$
- [**2014P2Q7 (a)**] equivalence classes of modula
- [**2007P2Q3 (a)**] prove the division algorithm

Chinese Remainder theorem:

- [**2016P2Q7 (a)(ii)**]
- [**2007P2Q4**] Chinese remainder theorem and its generalisation
- [**2005P1Q7 (a)**]
- [**1998P1Q7 (a)**]

Fundamental theorem of arithmetic:

- [**2003P1Q7 (a)-(b)**] infinite primes, upper bound on the numbers with k given primes
- [**2001P1Q2 (a)-(c)**] proof, count number of divisors, smallest number with given number of factors

Set theory

Set theory:

- [**2018P2Q9 (b)**]
- [**2015P2Q7 (c)**]
- [**2014P2Q8 (a),(b),(c)**] find the correct predicate and prove the statement, bijections between functions and injections
- [**2011P2Q5**]
- [**2010P2Q5 (a)-(d)**] requires knowledge of rule induction
- [**2013P1Q3 (a)**] logic statements for set theory
- [**2009P1Q3 (c)**] big intersection and big union properties
- [**2002P1Q8**] intersections and intersection-closed

Relations:

- [**2018P2Q8 (a)**] predicate statement involving relations
- [**2018P2Q8 (b)**] prove injection, surjection
- [**2017P2Q8 (c)**] check if function is surjective
- [**2017P2Q9 (c)**]
- [**2016P2Q8 (c)**] show that one function is the inverse of the other
- [**2016P2Q9 (c)**]
- [**2015P2Q8**] id_A subset $g \circ f$ and $f \circ g$ subset id_B
- [**2013P2Q6**] rule induction for relations
- [**2012P2Q5 (a),(b)**] relational composition
- [**2012P1Q4 (a),(b)**] define transitive, find which of the relations are transitive, draw their graphs
- [**2011P2Q6 (a)-(c)**] membership, big intersection and big union
- [**2011P1Q4 (a),(b)**] prove or disprove statements about relations
- [**2009P1Q4**] proofs and counterexamples
- [**2008P2Q3**] define injection, bijection iff $f \circ g = \mathrm{id}_x$ and $g \circ f = \mathrm{id}_y$, $\mathcal{P}(X \times Y) \to (X \to \mathcal{P}(Y))$
- [**2006P2Q5 (a),(b)**] define injection, surjection and bijection, inverses
- [**1998P1Q2**] relations, equivalence classes, equivalence classes are disjoint, simple equivalence class
- [**1997P1Q8**] iterated relations

Equivalence relations:

- [**1993P10Q11 (a)**] elements that map to the same value form an equivalence relation

Discrete structures:

- [**2017P2Q9 (b)**]

Partial functions::

- [**1993P2Q3**]

Surjections:

- **[1993P10Q11 (b)]** number of surjections

Bijections:

- **[2018P2Q7 (b)]**
- **[2017P2Q8 (b)]** prove bijection cross product
- **[2015P2Q9]** prove bijection between $\mathcal{P}(X \cup Y)$ and $\mathcal{P}(X) \times \mathcal{P}(Y)$
- **[2013P2Q5]** bijections
- **[2007P2Q5]** bijections
- **[1999P1Q8 (a)]** bijection between equivalence classes and bijections

Countability:

- **[2012P2Q5 (c)]** is relational composition countable?
- **[2007P2Q6 (b)]** : diagonalisation
- **[2006P2Q5 (c)]** prove there is no injection between the power set of X and X.
- **[2005P1Q8]** various bijections, Russel's paradox, well-founded relation
- **[2004P1Q8]** various bijections
- **[2003P1Q8 (a)-(c)]** Schroder-Bernstein, enumerability properties and basic sets
- **[1999P1Q8 (b)-(e)]** Schroder-Bernstein, integers, rationals, reals, ML programs

Orders: *(lattices and well-founded relations are not define in the current version of the course, but some of these questions are good practice for set theory)*

- **[2012P2Q5 (d)]** well-founded
- **[2011P2Q6 (d)]** down-closed
- **[2010P2Q6]** preorders, down-closed, complete primes
- **[2009P2Q6]** least upper bounds
- **[2004P1Q2]** partial order, well-founded relation, product ordering
- **[2002P1Q2]** well-founded relation, minimal elements, application to problem with strings
- **[2001P1Q8]** product order, least upper bounds, lattices
- **[2000P1Q2]** partial orders on partitions
- **[2000P1Q7]** separated elements in partial orders
- **[1998P1Q8]** partial order/total order
- **[1997P1Q7]** tree-like partial orders
- **[1996P1Q7]** topological sort on partially-ordered sets, isomorphism between orderings
- **[1994P2Q3]** partial order, total order, well-order
- **[1994P11Q10 (a),(b)]** well-ordered relations
- **[1993P11Q11]** partial, total, well-order, closure

Induction

Rule induction:

- [**2018P2Q10 (a)**]
- [**2016P2Q7 (b)**]
- [**2016P2Q10**] prove or disprove inequalities between the counts of letters in strings
- [**2016P2Q9**] inductively defined total cover relation
- [**2014P2Q9 (a)**] show that no multiple of 5 is in the set
- [**2012P2Q6**]
- [**2008P2Q4**] rule induction for multiples of $n - m$, below a value
- [**2009P2Q5**] strings and there are more occurrences of a than b

Induction:

- [**2017P2Q7 (b)**] count the number of ways to form n using 1s and 2s
- [**2017P2Q9 (a)**]
- [**2016P2Q8 (b)**]
- [**2016P2Q9 (b)**]
- [**2015P2Q8 (b)**] prove that a natural is either even or odd.
- [**2014P2Q9 (b),(c)**] principle of induction for set of strings
- [**2013P1Q3 (b),(c)**] induction over lists, induction to prove the correctness of reverse and append
- [**2006P2Q6 (b)**]
- [**1994P11Q10 (c),(d)**] inductively defined relation

Counting:

- [**2017P2Q7 (b)**] count the number of ways to sum to n using 1s and 2s
- [**2009P1Q4 (c),(d)**] count irreflexive symmetric relations, count symmetric and antisymmetric relations
- [**2003P1Q7**] (c): (inclusion/exclusion, find primes < 100)
- [**2001P1Q2 (b)**] count the number of divisors for a number
- [**1997P1Q7 (b)**] counting tree-like partial orders
- [**1997P1Q2**] counting functions between two sets
- [**1996P1Q1**] count the number of bipartite graphs
- [**1995P1Q8**] principle of inclusion/exclusion, count surjections
- [**1994P2Q1**] derive the recurrence relation for Stirling numbers
- [**1994P10Q11**] count the number of invalid strings
- [**1993P2Q1**] deriving the Catalan numbers
- [**1993P10Q11 (b)**] number of surjections

Linear recurrences:

- [**1996P1Q8**]

- [**1994P2Q1**] Stirling numbers recurrence
- [**1994P10Q11**]
- [**1994P11Q10 (c),(d)**] inductively defined relation
- [**1993P2Q2**]

Bipartite matchings and flows: *(out of syllabus - part of the Algorithms course)*

- [**1995P1Q7**]
- [**1995P10Q13**]
- [**1994P2Q2**]

Formal languages

Finite automata/Regular expressions:

- [**2018P2Q9 (c)**]
- [**2018P2Q10 (b)**] show that the language of all regular expressions is not regular
- [**2017P2Q10**] decide whether the given languages are regular or not
- [**2016P2Q10 (c)**] prove r and s is regular if r, s are regular)
- [**2016P2Q10 (b)**] prove or disprove regular language/ remove one character and check if language is regular
- [**2015P2Q9**] DFA for language of palindromes
- [**2015P2Q10**] DFA accepting a^n for $n = 1, 2, n \equiv 4 \pmod 6$ or $n = 7 \pmod 6$, define regular, ultimately periodic sequences)
- [**2014P2Q10**] odd/even language, pumping lemma variant,
- [**2013P2Q8**] describe reg exp to DFA, reg exp for specific DFA, state pumping lemma, use pumping lemma to prove non-regular language (and some regular)
- [**2012P2Q8**] properties of the matches relation
- [**2011P2Q8**] reg exp for all strings over $\{a, b, c\}$ with at least one occurrence of each symbol, given NFA find DFA, find reg exp, show that if DFA accepts a string then it accepts one of length < num states, pumping lemma to prove language not regular (or use intersection with a^*b^*)
- [**2010P2Q9**] def deterministic, complement of DFA, give counterexample for complement of NFA, design NFAs for $\{a, aa, aaa\}$, complement of $\{a, aa, aaa\}$, all strings with length mult of 3 or 5, all strings matching $(aa|b)^*(bb|a)^*$, all strings matching $(\emptyset^*)^*$
- [**2009P2Q9**] L-equivalence classes for regular languages, upper bound on equivalence classes for concrete language, show that a^nb^n has an infinite number of equivalence classes
- [**2007P2Q8**] pumping lemma for regular languages, palindromes over $\Sigma = \{a\}$ is regular, palindromes over $\{a, b\}$ is non-regular, a^*b^* is regular, same number of as and bs is non-regular, finite language is regular
- [**2006P2Q8**] difference of regular languages is regular, if DFA accepts all strings of length < num states then it accepts all strings, describe algorithm for deciding if two languages are equivalent
- [**2005P2Q9 (a)**] intersection of two regular languages is regular.
- [**2004P2Q9**] complement of regular language is regular, palindromes is non-regular, multiple of 3s is regular, primes is non-regular

- **[2003P2Q9]** all strings without bb, def reg exp, if no occurrence of $\emptyset$ then language non-empty, complement of reg exp is reg exp
- **[2014P2Q1]**0 DFA and RE for languages with ever number of one of $\{a, b, c\}$, DFA for intersection, small variant of pumping lemma
- **[2002P2Q9]** $a^m b^n$ regular, $a^m b^n$ with $m \leq n$ non-regular, $a^m b^n$ with $m + n \leq 4$ regular, complement of regular is regular, complement of non-regular language is non-regular, (f) choose b^n
- **[2001P2Q7]** prove pumping lemma, $L_1 = ww$ non-regular, $L_2 = wvw$ is regular
- **[2000P2Q7]** def reg exp, def accept by DFA, prove DFA to reg exp, example construction from DFA to reg exp
- **[1999P2Q7]** Modifying one character from a string of a RL results into an RL, prove that a DFA accepts a string that is shorter than the number of states in the DFA, Kleene's theorem, give algorithm to check if RE accepts string
- **[1998P2Q7]** Define the pumping lemma, $a^m b^{2n}$ is reg, $a^p b^{2q}$ for p, q prime not reg, no infinite subset of $a^n b^n$ is regular, no infinite set of ww is regular, every finite subset of ww is regular.
- **[1997P2Q7]** Explain how to check if you regular languages are equivalent.
- **[1996P2Q8]** filter odd length strings from regular language, filter palindromes from regular language, find regular subset of Pal (e.g. 1111)
- **[1995P2Q27]** twice as many 0s as 1s, prefix language, any finite language, $(r|s)^*$ vs $(r^*)^*$
- **[1995P3Q3]** Equivalence between regular expressions and DFAs
- **[1994P3Q3]** define accept language, show that regular expressions can simulate DFAs, give example
- **[1993P6Q12]** set of strings that are not palindromes, union of countably many languages, set of strings where $3 \nmid \#a$ and $3 \nmid \#b$, all strings that are of the form ww,
- **[1993P5Q12]** definition of reg exp, $L(t|sr)$ subset $L(r)$, then it contains $L(s^*t)$ (empty string not in s), iff condition is empty string in s.

Context-free languages (out of syllabus):

- **[2005P2Q9 (b)]** example of regular context free grammar
- **[1995P4Q3]** construct a pushdown automaton for accepting the propositional calculus language
- **[1994P4Q3]** define CFLs, union of CFLs is CFL, prove not CF.