Mathématiques pour l’informatique

Marco Saerens

# Chapter 1: Logic and proofs

Definitions + theorems + exercices

## 1.1 Propositional logic

## 1.3 Propositional equivalences

## 1.4 Predicates and quantifiers

## 1.5 Nested quantifiers

## 1.6 Rules of inference

## 1.7 Normal forms (only what is explained in section 1.3)

## 1.8 Introduction to proofs

Direct proof, proof by contraposition, proof by contradiction.

## Proofs:

* Proving any logical equivalence statement
* Any proposition can be transformed in disjunctive normal form
* *A* is a valid argument if and only if (*p*1∧ *p*2∧ *p*3∧ …∧ *pm*) → *q* is a tautology

# Chapter 5: Induction and recursion

Definitions + exercices

## 5.1 Mathematical induction

# Chapter 6: Counting

Definitions + theorems + exercices

## 6.1 The basics of counting

## 6.2 The pigeonhole principle

**Not**: Theorems.

## 6.3 Permutations and combinations

## 6.4 Binomial coefficient and identities

**Not**: Other identities involving binomial coefficients (not theorem 3-4 (Vandermonde))

## 6.5 Generalized permutations and combinations

**Not**: Distributing objects into boxes

## Proofs:

* the number of different subsets of a finite set *S* is 2|*S*|
* the generalized pigeonhole principle
* formula for the number of permutations
* formula for the number of combinations
* formula for the number of combinations with repetitions (generalized combinations)
* formula for the number of permutations with indistinguishable objects
* the binomial theorem
* Pascal’s identity

# Chapter 8: Advanced counting techniques

Definitions + theorems + exercices

## 8.1 Applications of recurrence relations

## 8.2 Solving linear recurrence relations

Linear homogeneous and non-homogeneous equations and transformations

## 8.5 Inclusion-exclusion

**Only**: For three sets

## + State-space approach (slides only)

## + Some expressions for computing series

## Proofs:

* solution for solving linear homogeneous recurrence relations of degree two when all roots are distinct
* same, but when the roots are repeated
* solution for solving linear nonhomogeneous recurrence relations
* solution for solving linear homogeneous recurrence relations by state-space approach
* proof of formula for computing |*A*∪*B*∪*C*|
* series computation

# Chapter 10: Graphs

Definitions + theorems + algorithms + exercices

## 10.1 Graphs and graph models

## 10.2 Graph terminology and special types of graphs

**Not**: Theorem 5 (Hall’s marriage theorem)

## 10.3 Representing Graphs and Graph Isomorphism

## 10.4 Connectivity

**Not**: An inequality for vertex connectivity and edge connectivity

## 10.5 Euler and Hamilton paths

**Not**: Theorem 3 (Dirac’s theorem) and Theorem 4 (Ore’s theorem)

## 10.6 Shortest-path problems

## 10.7 Planar graphs

**Not**: No proof of theorem – only Euler’s formula

## 10.8 Graph coloring

## 10.9 The PageRank algorithm (slides only)

## Proofs:

* an undirected graph has an even number of vertices of odd degree
* formula for counting paths between vertices
* necessary and sufficient conditions for Euler circuits
* any sub-path of a shortest path is itself a shortest path
* a node cannot appear more than once on a shortest path
* dijkstra’s algorithm (the two parts)
* the PageRank score corresponds to the probability of finding the random walker in each node in the long run

# Chapter 13: Dynamic programming

Definitions + theorems + algorithms + exercices

## 13.1 Dynamic programming recurrence expression and its derivation

## 13.2 Application to edit-distance computation

## 13.2 Bellman-Ford algorithm for computing the shortest path distance

## Proofs:

* dynamic programming formula
* Bellman-Ford formula for computing the shortest path distance