## CONTENTS

1. **INTRODUCTION** .................................................................................................................. 3
2. **STUDY AND RESEARCH AREAS** .................................................................................... 3
3. **MSC. STUDIES** ..................................................................................................................... 4
   3.1 **ADMISSION REQUIREMENTS** ................................................................................. 5
   3.2 **CURRICULUM FOR THE MSC. PROGRAM** ............................................................... 6
4. **PH.D. STUDIES** .................................................................................................................... 10
   4.1 **ADMISSION REQUIREMENTS** ................................................................................ 10
   4.2 **RESEARCH** ................................................................................................................. 11
   4.3 **STUDIES** .................................................................................................................... 11
5. **FACULTY RESEARCH INTERESTS** .................................................................................... 14
6. **RESEARCH GROUPS STRUCTURE** ................................................................................... 24
I. INTRODUCTION

This booklet aims at guiding the students in the department of computer science for the degrees of M.Sc. and Ph.D. studies. All the rules in this booklet are subject to the regulations of the School of Graduate Studies, as listed in the studies catalog of the Technion. In case of differences between the Hebrew and English versions of this catalog, the Hebrew version shall apply.

The booklet lists the admission requirements and curriculum for M.Sc. and Ph.D. studies, the list of faculty members and their research areas, as well as division into research areas.

The graduate studies catalogue of the department of computer science is also available on the Internet site of the department: www.cs.technion.ac.il. The site also includes detailed information (in Hebrew and English) on the department, courses and syllabi, theses submitted in recent years, and other relevant information. It is also advisable to contact the Graduate Studies coordinator in the department: Limor Gindin, in room 503, tel. 04-8294226, or by email: limorg@cs.technion.ac.il.

2. STUDY AND RESEARCH AREAS

The Computer Science Department offers study programs for the degrees of “Master in Science in Computer Science”, “Master in Science” and “Doctor of Philosophy”. Excellent students can apply for transfer to the Direct Ph.D. Track during their graduate studies.

The expertise and research areas of the department are:

Theory of Computer Science:
- Automata and Formal Languages
- Coding
- Complexity
- Computational Geometry
- Cryptology
- Distributed Computing
- Logic and Semantics
- Theory of Algorithms

Systems:
- Databases & Data Mining
- Distributed & Parallel Systems
- Hardware and Computer Architecture
- Networks, Communication & Systems
- Programming languages
- Operating Systems & Virtualization
- Software & Hardware verification
- Software Engineering
- Storage
- System Security

Artificial Intelligence:
- Learning
- Reasoning
Intelligent Systems and Scientific Computation:
Geometric Modeling
Graphics
Image Processing and Computer Vision
Robotics and Complex Systems
Scientific Computation and Numerical Algorithms

Interdisciplinary Research:
Bioinformatics (Computer Science and Biology)
Computational Linguistics and Natural Language Processing
Quantum Information Processing (Computer Science, Electrical Engineering, Physics and Chemistry)
World Wide Web, Electronic Commerce, and Computational Finance

In addition to the theoretic research in these areas, the department has research laboratories in the following areas:

Robotics
Computer Vision
Artificial Intelligence
Geometric Processing
Computer Graphics and Geometric Planning
Computer Communication Networks
Software Systems
Computer Systems
Natural Languages Processing
Cyber and Computer Security
Learning and Reasoning
Data and Knowledge
Information Storage and Memories
Bioinformatics
Quantum Information Processing

3. M.SC. STUDIES

The Department of Computer Science offers a study program for the degree of “Master of Science in Computer Science”. The studies are open for graduates of B.Sc. in Computer Science or other fields. The Department also offers a study program for the degree of “Master of Science” for graduates of B.Sc., which is not in computer science (such as Mathematics, Physics, and Electrical Engineering).

3.1 ADMISSION REQUIREMENTS

3.1.1 ADMISSION REQUIREMENTS FOR THE MASTER IN COMPUTER SCIENCE PROGRAM

Students who have graduated with a B.Sc. degree in Computer Science, or in any of the joint tracks of Computer Science and other Departments (e.g., Computer Engineering, Computer Science-Mathematics), may be admitted. Candidates who received their B.Sc. in another framework will be required to take supplementary courses as detailed below. Professional achievements of candidates with experience in industry or experience in research, as well as recommendation letters, will be taken into account by the Admissions Committee.
Candidates who wish to improve their academic standing towards admission may do so by taking courses as 'advanced studies'. For this, they have to meet the Vice Dean for Graduate Studies in order to decide what courses they should take, and set minimum acceptance grades for these courses.

The Department prefers internal students who receive a fellowship and who devote all their time to studies, research, and teaching.

A student who completed his B.Sc. in a recognized institute of higher learning in Israel with Computer Science as his major is not required to take supplementary courses.

Students who graduated from a college will be admitted according to the policy of the School of Graduate Studies, as stated from time to time. Furthermore, a graduate of a college may be admitted as a qualifying student, study 20 credits in this status, and must receive an average of at least 88. The subjects of study will be decided upon in coordination with the Vice Dean for Graduate Studies.

3.1.2 Admission Requirements for the Master of Science Program

Students who have graduated with a B.Sc. in scientific or engineering programs may be admitted. The student should secure a faculty member who will serve as an advisor. It is not obligatory to submit a research proposal upon admission. The study program for each student will be determined in coordination with the advisor and the Vice Dean for Graduate Studies, and will be approved by the Admissions Committee.

The program will include:
1. Graduate credits: 18 credits for students who completed a four-year program, 30 credits for graduates of a three-year program.
2. Supplementary study program as necessary.

3.1.3 Admission Requirements for External Students

A student may also be admitted as an external student (that is, without receiving a fellowship). The following three rules apply in this case:

An external student for M.Sc. studies must be present in the Department at least two days a week, for at least one year.

The student has an excellent academic record, with a GPA substantially higher than the admission threshold. Such a student will commit to work outside the department for at most three days a week (with the approval of the employer), and to work as a Teaching Assistant in the Department (regular load) if required to do so.

In case of a student with an exceptional academic record, the committee may consider admission even if the above conditions are not met.

3.2 Curriculum for the M.Sc. Program

During studies, the student should take a number of courses as described above, and perform research (with thesis or minor thesis) under the supervision of a faculty member. Students with a supplementary program should finish it as well. Course prerequisites are not imposed on graduate students, but students that take a course without its formal prerequisite, must understand that it is their own responsibility to obtain the knowledge necessary for understanding of the course material.

For the supervision of the research, the student should approach a faculty member in his field of interest. In special cases, and in coordination with, and with prior approval of the Graduate Studies
Committee, the advisor may be an adjunct teacher or a member of another department in the Technion. External students may neither choose adjuncts nor external faculty as advisors. The research project may be theoretical or involve advanced engineering. In special cases, there is an option of a 'minor thesis', in which case, eight additional credits are required.

### 3.2.1 Program for Graduates of a Three-Year Program B.Sc.

Students who graduated a three-year program are required to take courses worth 30 credits. The student must study at least six courses of the department of computer science from at least four different groups within the 11 groups of the optional courses (which are neither a project course, nor advanced topics, nor a seminar). Subjects listed in the following groups, with similar professions provided by the Department of Electrical Engineering. Listed undergraduate courses may be taken as long as the student does not pass the allowed number of undergraduate credit points. Excellent students can apply for transfer to the Direct Ph.D. Track during their graduate studies.

The group topics are as follows:

1. **Complexity of Computations**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>236306</td>
<td>Random graph</td>
<td>2.0</td>
</tr>
<tr>
<td>236307</td>
<td>Expander graphs and their applications</td>
<td>2.0</td>
</tr>
<tr>
<td>236308</td>
<td>Algebraic graph theory and combinatorial designs</td>
<td>3.0</td>
</tr>
<tr>
<td>236309</td>
<td>Introduction to coding theory</td>
<td>3.0</td>
</tr>
<tr>
<td>236313</td>
<td>Complexity theory</td>
<td>3.0</td>
</tr>
<tr>
<td>236315</td>
<td>Algebraic methods in computer science</td>
<td>3.0</td>
</tr>
<tr>
<td>236359</td>
<td>Algorithms 2</td>
<td>3.0</td>
</tr>
<tr>
<td>236374</td>
<td>Probabilistic methods and algorithms</td>
<td>3.0</td>
</tr>
<tr>
<td>236377</td>
<td>Distributed Graph Algorithms</td>
<td>3.0</td>
</tr>
<tr>
<td>236378</td>
<td>Principles of managing uncertain data</td>
<td>2.0</td>
</tr>
<tr>
<td>236508</td>
<td>Cryptography and complexity</td>
<td>2.0</td>
</tr>
<tr>
<td>236518</td>
<td>Communication complexity</td>
<td>2.0</td>
</tr>
<tr>
<td>236521</td>
<td>Approximation algorithms</td>
<td>2.0</td>
</tr>
<tr>
<td>236525</td>
<td>Introduction to network coding, bounds and construction</td>
<td>3.0</td>
</tr>
<tr>
<td>236760</td>
<td>Computational learning theory</td>
<td>2.0</td>
</tr>
</tbody>
</table>

2. **Theory of Algorithms**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>236315</td>
<td>Algebraic methods in computer science</td>
<td>3.0</td>
</tr>
<tr>
<td>236357</td>
<td>Distributed algorithms A</td>
<td>3.0</td>
</tr>
<tr>
<td>236359</td>
<td>Algorithms 2</td>
<td>3.0</td>
</tr>
<tr>
<td>236377</td>
<td>Distributed Graph Algorithms</td>
<td>3.0</td>
</tr>
<tr>
<td>236521</td>
<td>Approximation algorithms</td>
<td>2.0</td>
</tr>
<tr>
<td>236715</td>
<td>Methods in analysis of algorithms</td>
<td>3.0</td>
</tr>
<tr>
<td>236719</td>
<td>Computational geometry</td>
<td>3.0</td>
</tr>
<tr>
<td>236755</td>
<td>Distributed algorithms</td>
<td>3.0</td>
</tr>
<tr>
<td>236760</td>
<td>Computational learning theory</td>
<td>2.0</td>
</tr>
<tr>
<td>236779</td>
<td>Foundations of Algorithms for Massive Datasets</td>
<td>2.0</td>
</tr>
<tr>
<td>238739</td>
<td>Discrete algorithmic geometry</td>
<td>2.0</td>
</tr>
</tbody>
</table>

3. **Logic and its Applications**

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>236026</td>
<td>Knowledge and games in distributed systems</td>
<td>2.0</td>
</tr>
<tr>
<td>236304</td>
<td>Logic for computer science 2</td>
<td>3.0</td>
</tr>
<tr>
<td>236342</td>
<td>Introduction to software verification</td>
<td>3.0</td>
</tr>
<tr>
<td>236345</td>
<td>Automatic verification of hardware and software systems</td>
<td>3.0</td>
</tr>
<tr>
<td>236356</td>
<td>Introduction to database theory</td>
<td>3.0</td>
</tr>
<tr>
<td>Course Code</td>
<td>Course Title</td>
<td>Credits</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>236368</td>
<td>Formal specification of complex systems</td>
<td>3.0</td>
</tr>
<tr>
<td>236378</td>
<td>Principles of managing uncertain data</td>
<td>2.0</td>
</tr>
</tbody>
</table>

4. Cryptology, Coding and Information

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>236309</td>
<td>Introduction to coding theory</td>
<td>3.0</td>
</tr>
<tr>
<td>236350</td>
<td>Network security</td>
<td>3.0</td>
</tr>
<tr>
<td>236379</td>
<td>Coding and algorithms for memories</td>
<td>3.0</td>
</tr>
<tr>
<td>236500</td>
<td>Cryptanalysis</td>
<td>3.0</td>
</tr>
<tr>
<td>236506</td>
<td>Modern cryptology</td>
<td>3.0</td>
</tr>
<tr>
<td>236508</td>
<td>Cryptography and complexity</td>
<td>2.0</td>
</tr>
<tr>
<td>236520</td>
<td>Coding for storage systems</td>
<td>2.0</td>
</tr>
<tr>
<td>236525</td>
<td>Introduction to network coding, bounds and construction</td>
<td>3.0</td>
</tr>
<tr>
<td>236990</td>
<td>Introduction to quantum information processing</td>
<td>3.0</td>
</tr>
</tbody>
</table>

5. Development of Software Systems

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>236268</td>
<td>Constructive computer architecture</td>
<td>5.5</td>
</tr>
<tr>
<td>236319</td>
<td>Programming languages</td>
<td>3.0</td>
</tr>
<tr>
<td>236321</td>
<td>Software engineering methods</td>
<td>3.0</td>
</tr>
<tr>
<td>236332</td>
<td>The internet of things (IoT) – technologies and implementations</td>
<td>2.0</td>
</tr>
<tr>
<td>236342</td>
<td>Introduction to software verification</td>
<td>3.0</td>
</tr>
<tr>
<td>236347</td>
<td>Program analysis and synthesis</td>
<td>3.0</td>
</tr>
<tr>
<td>236363</td>
<td>Databases</td>
<td>3.0</td>
</tr>
<tr>
<td>236368</td>
<td>Formal specification of complex systems</td>
<td>3.0</td>
</tr>
<tr>
<td>236369</td>
<td>Managing data on the world-wide web</td>
<td>3.0</td>
</tr>
<tr>
<td>236376</td>
<td>Operating systems engineering</td>
<td>4.0</td>
</tr>
<tr>
<td>236490</td>
<td>Computer Security</td>
<td>3.0</td>
</tr>
<tr>
<td>236491</td>
<td>Secure Programming</td>
<td>3.0</td>
</tr>
<tr>
<td>236496</td>
<td>Reverse Engineering</td>
<td>3.0</td>
</tr>
<tr>
<td>236700</td>
<td>Software design</td>
<td>3.0</td>
</tr>
<tr>
<td>236703</td>
<td>Object-oriented programming</td>
<td>3.0</td>
</tr>
<tr>
<td>236712</td>
<td>Agile software engineering</td>
<td>2.0</td>
</tr>
<tr>
<td>236780</td>
<td>Algorithms for dynamic memory management</td>
<td>2.0</td>
</tr>
</tbody>
</table>

6. Communication and Distributed Systems

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>236026</td>
<td>Knowledge and games in distributed systems</td>
<td>2.0</td>
</tr>
<tr>
<td>236322</td>
<td>Information storage systems</td>
<td>3.0</td>
</tr>
<tr>
<td>236334</td>
<td>Introduction to computer networks</td>
<td>3.0</td>
</tr>
<tr>
<td>236341</td>
<td>Internet networking</td>
<td>3.0</td>
</tr>
<tr>
<td>236350</td>
<td>Network Security</td>
<td>3.0</td>
</tr>
<tr>
<td>236351</td>
<td>Distributed systems</td>
<td>3.0</td>
</tr>
<tr>
<td>236357</td>
<td>Distributed algorithms A</td>
<td>3.0</td>
</tr>
<tr>
<td>236369</td>
<td>Managing data on the world-wide web</td>
<td>3.0</td>
</tr>
<tr>
<td>236370</td>
<td>Parallel and distributed programming</td>
<td>3.0</td>
</tr>
<tr>
<td>236377</td>
<td>Distributed Graph Algorithms</td>
<td>3.0</td>
</tr>
<tr>
<td>236490</td>
<td>Computer Security</td>
<td>3.0</td>
</tr>
<tr>
<td>236510</td>
<td>Database management systems implementation</td>
<td>3.0</td>
</tr>
<tr>
<td>236755</td>
<td>Distributed algorithms</td>
<td>3.0</td>
</tr>
</tbody>
</table>

7. Computer Systems

<table>
<thead>
<tr>
<th>Course Code</th>
<th>Course Title</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>236268</td>
<td>Constructive computer architecture</td>
<td>5.5</td>
</tr>
<tr>
<td>236278</td>
<td>Computational accelerators and accelerated systems</td>
<td>3.0</td>
</tr>
<tr>
<td>236322</td>
<td>Information storage systems</td>
<td>3.0</td>
</tr>
<tr>
<td>236334</td>
<td>Introduction to computer networks</td>
<td>3.0</td>
</tr>
</tbody>
</table>
236347  Program analysis and synthesis  3.0
236350  Network security  3.0
236363  Databases  3.0
236369  Managing data on the world-wide web  3.0
236376  Operating systems engineering  4.0
236379  Coding and algorithms for memories  3.0
236490  Computer Security  3.0
236491  Secure Programming  3.0
236496  Reverse Engineering  3.0
236510  Database management systems implementation  3.0
236780  Algorithms for dynamic memory management  2.0

8. Vision and Robotics

236200  Signal, image and data processing  4.0
236327  Digital image and signal processing  3.0
236330  Introduction to optimization  3.0
236372  Bayesian networks  3.0
236777  Deep learning and its applications  3.0
236781  Deep learning on computation accelerators  3.0
236790  Multigrid methods  2.0
236860  Digital image processing  2.0
236861  Geometric computer vision  3.0
236862  Sparse representations and applications in signal and image processing  2.0
236873  Computer vision  3.0
236875  Visual recognition  3.0
236927  Introduction to robotics  3.0

9. Geometry and Graphics

234325  Computer graphics 1  3.0
236324  Computer graphics 2  3.0
236329  Digital geometry processing  3.0
236373  Image synthesis  3.0
236716  Geometric models in CAD systems  3.0
236719  Computational geometry  3.0
238739  Discrete algorithmic geometry  2.0

10. Learning and artificial Intelligence

236200  Signal, image and data processing  4.0
236299  Introduction to natural language processing  3.0
236501  Bayesian networks  3.0
236501  Introduction to artificial intelligence  3.0
236756  Introduction to machine learning  3.0
236760  Computational learning theory  2.0
236777  Deep learning and its applications  3.0
236779  Foundations of Algorithms for Massive Datasets  2.0
236781  Deep learning on computation accelerators  3.0
236941  Introduction to neural networks  3.0

11. Bioinformatics

236522  Algorithms in computational biology  3.0
236523  Introduction to bioinformatics  2.5
The remaining courses should be of courses that yield expertise in the proposed research topic, according to a program decided jointly with the student's advisor. The student may take up to three seminars and advanced topics courses before the approval of his research program. The two following courses are obligatory. Students who did not study the courses:
236267 Computer architecture
236343 Theory of computation
in the framework of their B.Sc. studies are required to study them in the framework of the M.Sc. program (the credits will be counted).

At least six credits should be taken following the approval of the research proposal. Courses will be approved by the advisor or temporary advisor. It is possible to study up to six credits of undergraduate courses.

3.2.2 Program for Graduates of a Four-Year B.Sc. Program

The student must complete 18 credits of graduate courses, mostly in Computer Science courses or similar courses in the Department of Electrical Engineering courses. The courses will be chosen jointly with the advisor. The student is required to take at least two courses (six credits) after submitting a research proposal.

The two following courses are obligatory. Students who have not studied the courses:
236267 Computer architecture
236343 Theory of computation
in the framework of their B.Sc. studies are required to study them in the framework of the M.Sc. program.

3.2.3 Supplementary Program for Qualifying Students

A student will be credited with graduate credits for a supplementary course at the graduate level (prefix 236), if the student obtained a reasonable grade. A student may be exempted from taking a supplementary course if:

1. The student studied a similar, or a more advanced course, in the same field (possibly in another academic institution).
2. The student received an exemption from the teacher of the course.

A complete list of the supplementary courses will be compiled for each student at the time of his acceptance, according to the following list:

Mathematics Courses

<table>
<thead>
<tr>
<th>Number</th>
<th>Name</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>114031</td>
<td>Infinitesimal calculus 1M</td>
<td>5.5</td>
</tr>
<tr>
<td>114032</td>
<td>Infinitesimal calculus 2M</td>
<td>5.0</td>
</tr>
<tr>
<td>104166</td>
<td>Algebra A</td>
<td>5.5</td>
</tr>
<tr>
<td></td>
<td>or</td>
<td></td>
</tr>
<tr>
<td>104134</td>
<td>Modern algebra H</td>
<td>2.5</td>
</tr>
</tbody>
</table>

Computer Science Courses

<table>
<thead>
<tr>
<th>Number</th>
<th>Name</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>094412</td>
<td>Introduction to probability M</td>
<td>4.0</td>
</tr>
<tr>
<td>234129</td>
<td>Introduction to set theory and automata for CS</td>
<td>3.0</td>
</tr>
<tr>
<td>234125</td>
<td>Numerical algorithms</td>
<td>3.0</td>
</tr>
<tr>
<td>234141</td>
<td>Combinatorics for CS</td>
<td>3.0</td>
</tr>
</tbody>
</table>
A student has to pass the supplementary courses with an average of 88, and each course with a minimum grade of 80. Students may register to these courses even if they did not take their prerequisites. Upon successful completion of the supplementary courses, the vice-dean for graduate studies will approve the students transfer to the regular M.Sc. program (either for graduates of three-year or four-year B.Sc.).

4. PH.D. STUDIES

4.1 ADMISSION REQUIREMENTS

4.1.1 GENERAL

The prime goal of doctoral studies is the development of an independent research capability culminating in original research work. The student will prepare a detailed scientific thesis based on the research he conducted. The research must be innovative, advance the state of knowledge in the student's chosen research field, and be acceptable for publication in respectable scientific journals in the field.

4.1.2 DEFINITION OF TYPES OF STUDENTS

A candidate who fulfills the formal admission requirements for studying towards a Ph.D. degree as set down by the School of Graduate Studies, or has almost completed his Master's degree and is interested in studying for a Ph.D. degree, shall contact (at any time) the committee for graduate studies in order to enroll.

A candidate may be admitted either as a 'qualifying student' or as a 'regular student'. The former status allows the candidate one semester in which he may fulfill the conditions for admission as a regular student, as explained in Clause 31.02 of the School of Graduate Studies Regulations.

A student must be internal for at least one year during his studies.

4.1.3 REQUEST FOR ADMISSION

A candidate wishing to be admitted to studies will submit at least two letters of recommendation and two standard evaluation forms provided by the Department. In order to be accepted for doctoral studies, the candidate should contact a designated advisor and submit a proposal for a research topic. The CS Graduate Studies Committee will discuss the application for admission.

4.1.4 ADMISSIONS COMMITTEE

The admissions committee, composed of members of the Graduate Committee, will discuss the student's application and send the recommendation to the Graduate School.
4.1.5 CONDITIONS OF ADMISSION

In order to be admitted as a regular student for a Ph.D. degree, the candidate must fulfill the following conditions:

- Comply with all the formal requirements of the School of Graduate Studies.
- Be admitted by the Graduate Studies Committee, and comply with the requirements of that committee.
- Comply with other requirements (e.g., in case of a change in the field of research) that the Graduate Studies Committee may decide upon, while discussing the student's request.
- Be approved by the Dean of the School of Graduate Studies.

A candidate who does not comply with the first condition may be accepted as a qualifying student for a period of one semester. During this time he must fulfill all the requirements which will allow a change of status to that of a regular student.

4.1.6 DIRECT STUDIES

Outstanding students studying for M.Sc. may transfer to the direct study program towards a Ph.D. If the conditions for this transfer are satisfied (according to Clause 24.07 of the School of Graduate Studies Regulations), the student needs to apply, with a recommendation of his advisor and an additional recommendation, to the Graduate Studies Committee.

4.2 RESEARCH

GENERAL

The prime goal of doctoral studies is the development of an independent research capability culminating in original research work. The student will prepare a detailed scientific thesis based on the research he conducted. The research must be innovative, advance the state of knowledge in the student's chosen research field, and be acceptable for publication in respectable scientific journals in the field.

4.2.1 ADVISOR AND RESEARCH TOPIC

The student has to secure an advisor and define a research topic within one semester of his admission. An external student is not allowed to choose an adjunct teacher as an advisor.

4.2.2 CANDIDACY EXAM

Towards the end of the first year, the student has to submit a research plan describing his up-to-date achievements and his plans for the continuation of his research. This plan has to be approved by the advisor and the Graduate Studies Committee, and serves as a basis for the candidacy exam.

4.3 STUDIES

4.3.1 STUDIES CREDITS

As of winter semester 2011, courses requirements for Ph.D. students at the department are:
1. At least 12 credits of graduate computer science courses (or joint courses for undergraduate and graduate studies).
2. Ph.D. Direct Track students will be required to obtain six credits in addition to their M.Sc. Studies sum of credits.

4.3.2 Supplementary Courses

Candidates for a Ph.D., whose previous degrees are not in Computer Science, will be required to take supplementary courses, selected from the obligatory courses in the three-year program for B.Sc.

A candidate may be exempted from the above-mentioned courses if:
- The student has already successfully completed equivalent courses.
- The courses were prerequisites for advanced courses that he has already completed.

The Graduate Studies Committee will determine, on a case-by-case basis, the supplementary courses, as well as the number of graduate credits the student has to complete (4–24). These courses will be selected in coordination with the student's advisor.
5. FACULTY RESEARCH INTERESTS

Ailon, Nir
Associate Processor


Almagor, Shaull
Senior Lecturer

Formal verification of software and hardware, Model checking, Temporal Logic, Automata, Dynamical systems, Synthesis and Planning.

Attiya, Hagit
Professor

Distributed computation and theoretical computer science; in particular: fault-tolerance; timing-based and asynchronous algorithms.

Bar-Yehuda, Reuven
Professor Emeritus

Combinatorial optimization: graph algorithms; scheduling algorithms, computational geometry.

Baram, Yoram
Professor Emeritus

Statistical learning theory, Pattern recognition; Classification; Regression, Neural networks; Associative memory; Non-linear network dynamics; Virtual reality and feedback control aids for movement disorders.

Barequet, Gill
Professor

Discrete and computational geometry; geometric computing; combinatorics; computer-aided geometric design; computer graphics and visualization.

Belinkov, Yonatan
Senior Lecturer

Natural language processing; machine learning for language understanding and generation; neural network representations; interpretability and robustness of machine learning models.
Ben-Chen, Mirela
Associate Professor

Shape analysis and understanding, 3D geometry processing, deformation and animation, fluid simulation on surfaces, vector field analysis and design, numerical algorithms for geometric data, computer graphics.

Biham, Eli
Professor

Cryptology and cryptanalysis; symmetric cryptography, quantum cryptography and quantum computation.

Bronstein, Alexander
Professor

3D acquisition and processing, deformable shape analysis and modelling, computer vision, machine learning, numerical geometry.

Bruckstein, Alfred M.
Professor

Image and signal processing, image analysis and synthesis; pattern recognition; applied geometry; robotics, especially ant robotics; estimation theory; neural coding.

Bshouty, Nader H.
Professor

Computational learning theory.

Censor-Hillel, Keren
Associate Professor


Cohen, Reuven
Professor

El-Yaniv, Ran  
Professor

Statistical learning theory, data clustering and compression, applications to information retrieval, web mining, human-computer interaction, biological sequence analysis, texture analysis and synthesis, and music analysis and synthesis; online algorithms: design, theoretical analysis and practical experimentation, computational finance: Portfolio selection algorithms.

Elad, Michael  
Professor

Signal and image processing, and computer vision; Mathematical methods for image representation; Numerical methods in image processing.

Elber, Gershon  
Professor

Computer aided geometric design; computer graphics.

Etzion, Tuvi  
Professor

Coding theory; combinatorial algorithms and designs; digital sequences in coding and communication.

Etsion, Yoav  
Associate Professor


Filmus, Yuval  
Senior Lecturer

Computational complexity, Proof complexity, Analysis of Boolean Functions, Combinatorics.

Fischer, Eldar  
Associate Professor

Efficiency of calculations: especially property testing, statistical deductions, and probabilistically checkable proofs; combinatorics: especially graph theory, regularity theorems in combinatorial structures, and applications to algorithms; logic in computer science: logical characterization of properties for which there exist efficient algorithms or desirable combinatorial aspects.
Francez, Nissim
Professor Emeritus

Primary: formal semantics of natural language; type-logical grammar; computational linguistics; A-calculus and proof theory. Secondary: semantics of programming languages; program verification; concurrent and distributed programming; logic programming.

Friedman, Roy
Professor

Distributed systems; group communication; wide-area applications; middleware, CORBA and .NET; clustering; distributed multimedia applications; mobile computing.

Geiger, Dan
Professor

Probabilistic networks, Bayesian learning, Computational Genetics.

Gil, Joseph
Associate Professor

Software engineering, in particular: aspects related to the object-oriented paradigm, programming languages and parsing.

Gotsman, Craig (Chaim)
Professor

Computer graphics; animation; rendering; geometric modeling; computational geometry.

Grumberg, Orna
Professor

Computer-aided verification of software and hardware; model checking; formal verification; temporal logics; modularity; abstraction; distributed model checking, sat-based model checking, games, 3-valued logics.

Heymann, Michael
Professor Emeritus

Specification and Control of Discrete-Event and Hybrid Systems; Systems and Control Theory; Robotics; Optimization Theory.

Ishai, Yuval
Professor

Cryptography, Complexity theory.
Itai, Alon  
Professor Emeritus


Itzhaky, Shachar  
Senior Lecturer

Software synthesis; High-level and automated programming; Functional programming languages; Formal proof assistants.

Kaminski, Michael  
Professor

Non-monotonic logic; complexity of algebraic computations; finite automata theory; temporal logic.

Kantorowitz, Eliezer  
Associate Professor (Ret.)

Software engineering; user interfaces; component oriented programming; internet programming; components with semantic interfaces; statistical estimation of the number of software faults

Katz, Shmuel  
Professor Emeritus

Program verification; formal specification methods; aspect-oriented software development; distributed systems; programming methodology; temporal logic; partial orders; programming languages; software engineering.

Kimelfeld Benny  
Associate Professor

Database systems and theory, information extraction, information retrieval, data mining, probabilistic and inconsistent databases.

Kimmel, Ron  
Professor

Image processing, computer vision, medical image analysis, computer graphics, differential geometry, scientific computing, machine learning.
Kohavi, Zvi
Professor Emeritus
Failure-Tolerant design, testing and fault diagnosis of digital systems; Switching and finite-automata theory; Reliability.

Kushilevitz, Eyal
Professor
Cryptography; Machine learning; computational complexity and communication complexity; randomized distributed protocols.

Lempel, Abraham
Professor Emeritus
Application of discrete mathematics to problems in computer science and information theory; Imaging and compression technology.

Lindenbaum, Michael
Professor
Image processing and computer vision, especially the statistical analysis of visual tasks.

Litman, Ami
Associate Professor (Ret.)
Interconnection networks; parallel computation on fixed connection networks; systolic systems; layout of networks; digital systems, VLSI.

Makowsky, Johann
Professor Emeritus
Logic and complexity; complexity over the reals; algebraic combinatorics.

Markovitch, Shaul
Professor

Mendelson, Avi
Professor
Computer Architecture - heterogeneous systems, system-on-a-chip, power management, fault-tolerance, GPGPU; Operating Systems - for system-on-chip and heterogeneous systems; Real-Time systems - WCET for single processor and multi-processors, operating system, mix-criticality power management for RT systems.
Mor, Tal  
Associate Professor  
Theoretical quantum information processing (computing, cryptography, information, communication); implementation (hardware) of quantum information processing; modern cryptology.

Moran, Shlomo  
Professor Emeritus  
Algorithmic aspects of bioinformatics (with emphasis on phylogenetics); combinatorics and graph theory.

Naor, Seffi (Joseph)  
Professor  
Theory of algorithms and applications; Randomness and computation; approximation and online algorithms; combinatorial optimization; randomized algorithms; communication networks; parallel computation.

Ornan, Uzzi  
Visiting Professor  
Natural language processing, machine translation, information retrieval, processing of Hebrew in all levels (phonology, morphology, syntax, semantics), speech recognition.

Paz, Azaria  
Professor Emeritus  
Theory of automata, deterministic and probabilistic; Theory of algorithms and integer algorithms; Theory of Bayes networks and Theory of Graphoids

Petrank, Erez  
Professor  
Concurrent Algorithms: theory and practice; Memory Management (in particular for modern parallel architectures); Operating Systems, Programming Languages.

Pinter, Ron  
Professor  
Bioinformatics; High performance computing; Programming languages; Compiler technology; Automated design of integrated circuits; Information organization and retrieval; Data integration; Algorithmic number theory.

Raz, Danny  
Professor  
Theory and applications of management related problems in IP networks; active networks, network location problems, theory of network management, QoS routing, wireless networks, and other optimization problems.
Rivlin, Ehud
Professor
Robot vision; robot navigation; motion planning; visual servoing; active vision; object recognition; artificial intelligence; image understanding, image processing; image databases.

Romano, Yaniv
Senior Lecturer
Data science and machine learning; reproducibility, reliability, and fairness in modern machine learning; data-driven inference; deep neural representations; computational imaging.

Rosenfeld, Nir
Senior Lecturer
Behavioral machine learning; Learning with humans in the loop; Machine learning for decision support; Learning with the presence of human actors; Social and behavioral modeling; Societal implications of using predictive methodologies.

Roth, Ronny
Professor
Error-correcting codes; coding for magnetic and optical recording; application of coding theory to complexity; information theory; digital communication.

Rothblum, Ron
Senior Lecturer
Cryptography and Computational Complexity.

Rottenstreich, Ori
Senior Lecturer
Computer networks, theory and algorithms for networks, hash based data structures.

Salzman, Oren
Senior Lecturer
Robotics, Algorithmic motion planning, Foundations of robotic planning, Computational challenges in robotics.

Schwartz, Roy
Senior Lecturer
Design and analysis of algorithms, combinatorial optimization, approximation algorithms, the geometry of metric spaces and its applications, submodular optimization, and randomized algorithms.
Schuster, Assaf
Professor
Parallel and distributed computing; peer-to-peer computing large scale data mining; scalable model checking; high-performance computer architecture; shared memory consistency models; java memory model; fault tolerance; distribute shared memory; non-stop systems.

Shachnai, Hadas
Professor
Design and analysis of algorithms for combinatorial optimization problems, in particular, algorithms for packing, scheduling and resource allocation problems arising in Information and Communication services; parameterized algorithms and their usage in approximation; randomized algorithms; parallel computation.

Shlomi, Tomer
Associate Professor

Shmueli, Oded
Professor
Database systems: theoretical aspects of query processing in relational databases, xml databases and logic based databases (datalog); system issues: physical storage, concurrency control, recovery, replication and distribution; querying the WWW; electronic commerce; automated negotiation.

Sidi, Avram
Professor Emeritus
Theory and application of scalar and vector extrapolation methods; numerical integration; numerical linear algebra; numerical solution of integral equations; padé and other related rational approximations.

Talgam-Cohen, Inbal
Senior Lecturer
Algorithmic game theory; Theory of computation; Optimization; Internet economics; Market design; Auctions.

Tsafrir, Dan
Associate Professor
Operating systems, parallel systems, security, performance evaluation.
Ungarish, Marius  
Professor Emeritus

Simulations of rotating fluids (incompressible, compressible, two-phase, liquid metals); two-phase flows; gravity currents; computational fluid dynamics, implementation of parallel computers, numerical methods.

Vizel Yakir  
Senior Lecturer

Formal verification of hardware and software systems; Model Checking; SAT/SMT solving; Abstraction techniques; Security Verification; Hardware-Software Co-Verification; Machine learning for verification.

Yaakobi, Eitan  
Associate Professor

Information and coding theory with applications to non-volatile memories, associative memories, data storage and retrieval, and voting theory.

Yadgar, Gala  
Senior Lecturer

Operating systems, file systems, storage systems and devices, large scale data centers, cache management and content distribution, codes for storage systems.

Yahav, Eran  
Associate Professor

Program analysis, abstract interpretation, program verification, program synthesis, concurrent and distributed systems, programming languages, and software engineering.

Yavneh, Irad  
Professor

Multigrid computational methods; scientific computing; computational physics; geophysical fluid dynamics; image processing and analysis; numerical analysis.

Zaks, Shmuel  
Professor Emeritus

Theory of distributed computing; atm and optical networks; combinatorial and graph algorithms; combinatorics and graph theory; discrete mathematics.
6. RESEARCH GROUPS STRUCTURE

- RESEARCH GROUPS
  - THEORY OF COMPUTER SCIENCE
  - SYSTEMS
  - ARTIFICIAL INTELLIGENCE FOUNDATIONS
  - INTELLIGENT SYSTEMS AND SCIENTIFIC COMPUTATION
  - INTER-DISCIPLINARY RESEARCH