

# Teaching Statement

Erel Segal-Halevi

2017-01-28

This document is divided to two sections: the first is about my past as a teacher and the second is about my future plans.

## 1 Past Experience and Teaching Style

I studied in the science education department in the Technion (2004–2006), earning a diploma for teaching computer science in high-schools. During my studies, I experienced in teaching high-school students, who are not an easy audience. To keep them active, I had to learn techniques such as: asking leading questions, conducting in-class discussions, calling students to the whiteboard and letting students work in groups. I have found that these techniques are useful for college and university students too, and I apply them naturally while teaching.

I worked as a teaching assistant in the Technion (1997–1999) and in Jerusalem College of Technology (JCT, 2010). I taught both theory courses (discrete math, automata and formal languages) and programming courses (data structures, C++ workshop).

One of the challenges in teaching theory courses was that many students thought the courses were not relevant for their work as programmers, so they were not very motivated to learn. I found creative ways to show the relevance of theoretical concepts. For example, in the course on automata and formal languages, one of the lessons is about regular expressions. I spent a large part of the lesson in teaching regular expressions in PERL, showing that a regular expression can attain in one line, what a program without regular expressions attains in a whole page of code. In general, I always try to demonstrate the practical usefulness of the taught concepts for the students' future career.

One of the challenges in teaching programming courses was that grading the homework assignment was very time-consuming for the TAs, leaving too little time for helping the students improve their performance. I convinced and helped the TA-in-charge to install an *online judge* — an automatic grading software commonly used in programming olympiads.<sup>1</sup> This let students receive immediate feedback on the correctness of their output and correct their program accordingly. We, the TAs, still had to check the submissions in order to make sure they conform to programming standards, but at least we did not waste time on compilation and runtime errors.

From my experience in the Technion, I learned to appreciate the obligation to submit homework in pairs. This obligation greatly facilitates the connections between students. Such connections are an important asset for the students' future careers. I plan to apply such an obligation in my future courses, adding individual oral exams to prevent free-riding.

As a TA in the JCT, I was evaluated directly by my students. My grade for teaching automata and formal languages and C++ workshop was 9.64 and my grade for teaching data structures was 9.17. Verbal feedback sheets (in Hebrew) are attached at the end of this document.

---

<sup>1</sup><https://github.com/hit-moodle/onlinejudge/wiki/Installation>

## 2 Future Plans and Teaching Preferences

### 2.1 Basic courses

I am ready to teach both theoretical and practical courses. Below is a partial list of preferred courses, with some ideas I have for teaching them in an engaging way.

**A. Basic programming and object-oriented programming.** I would like to build the course around a task such as constructing a computer game. The teacher and students will construct the basic project together in class, teaching the required programming concepts along the way. As homework, each student or student-pair will customize their game. A game has both technical-physical aspects and artistic-creative aspects, so it may be appealing to diverse students. As a textbook, I will use some of the recently-published books teaching various programming-languages via game-writing, e.g: Java (McAllister and Fritz, 2014), Python (Matthes, 2015) or C# (Egges et al., 2013).

**B. Algorithms and data-structures.** Besides teaching the theoretical aspects of algorithms and data structures, I would like to have students participate in online programming contests, such as Google Code Jam<sup>2</sup> and Code Chef.<sup>3</sup> In these contests, there is no explicit requirement for using a particular algorithm or data-structure; there is not even an explicit requirement for runtime complexity. However, the inputs are large and there is a time-limit. Therefore, students who use naive solutions will quickly realize that such solutions are not good enough. This should motivate them to use more advanced algorithms and data-structures, such as those learned in the course.

**C. Machine learning,** I would like to teach both the theoretical foundations of learning theory (Mohri et al., 2012; Shalev-Shwartz and Ben-David, 2014) and the practical aspects of data science (Grus, 2015; Wickham and Golemund, 2017). Additionally, I would like to have students solve problems from Kaggle.<sup>4</sup> This is a website where individuals and organizations post their data-science challenges. Often, monetary prizes are offered for the best solutions. There are challenges of various levels, from beginner practice challenges to million-dollar challenges. All students will be given, as homework, some intermediate-level problems; the more advanced students will be able to pick more advanced challenges, and if they are very good, they will even make some money along the way.

### 2.2 New courses

I will be happy to construct new courses related to my research. Below are some suggestions to topics and relevant textbooks. Each of these topics is sufficiently interesting to fill an entire course, but it is also possible to combine two or more topics into a more general course.

1. Fair division algorithms (Robertson and Webb, 1998; Brams and Taylor, 1996).
2. Voting procedures (Brams, 2007).
3. Computational social choice (Brandt et al., 2016)
4. Auctions (Milgrom, 2000; Klemperer, 2004).
5. Algorithmic game theory (Nisan et al., 2007).
6. Text processing (Perkins, 2014; Morris, 2014; Ingersoll et al., 2013)

---

<sup>2</sup><https://code.google.com/codejam>

<sup>3</sup><https://www.codechef.com>

<sup>4</sup><https://www.kaggle.com/>

Besides these courses, I would like to suggest a course called **Israeli computer-science research**. Its goal will be to connect the students with leading Israeli researchers. Such connections may help them whether they work in the industry or in the academy. Each student or student-pair will have a task such as the following. (a) Choose a paper published in the last 3 years in a leading computer-science conference or journal, written by Israeli authors. (b) Read and understand the paper, possibly with the help of the instructor. (c) Contact the authors, meet them and interview them about the research presented in the paper — how it advanced after the publication and how they are planning to continue it. (d) Present the paper and the interview in front of the class. Teaching such a course will give me an opportunity to acquaint myself with various areas of computer science.

## References

- Brams, S. J. (2007). *Mathematics and Democracy: Designing Better Voting and Fair-Division Procedures*. Princeton University Press, first edition edition.
- Brams, S. J. and Taylor, A. D. (1996). *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press.
- Brandt, F., Conitzer, V., Endriss, U., Lang, J., and Procaccia, A. D., editors (2016). *Handbook of Computational Social Choice*. Cambridge University Press, 1 edition.
- Egges, A., Fokker, J. D., and Overmars, M. H. (2013). *Learning C# by Programming Games*. Springer, 2013 edition.
- Grus, J. (2015). *Data Science from Scratch: First Principles with Python*. O’Reilly Media, 1 edition.
- Ingersoll, G. S., Morton, T. S., and Farris, D. (2013). *Taming Text: How to Find, Organize, and Manipulate It*. Manning Publications, 1 edition.
- Klemperer, P. (2004). *Auctions: Theory and Practice (The Toulouse Lectures in Economics)*. Princeton University Press.
- Matthes, E. (2015). *Python Crash Course: A Hands-On, Project-Based Introduction to Programming*. No Starch Press, 1 edition.
- McAllister, W. and Fritz, S. J. (2014). *Programming Fundamentals Using Java: A Game Application Approach (Computer Science)*. Mercury Learning & Information, hardcover w/ dvd edition.
- Milgrom, P. (2000). Putting Auction Theory to Work: The Simultaneous Ascending Auction. *Journal of Political Economy*, 108(2):245–272.
- Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012). *Foundations of Machine Learning (Adaptive Computation and Machine Learning series)*. The MIT Press.
- Morris, M. (2014). *Text Processing in Java*. Colloquial Media Corporation.
- Nisan, N., Roughgarden, T., Tardos, E., and Vazirani, V. V. (2007). *Algorithmic game theory*. Cambridge University Press.
- Perkins, J. (2014). *Python 3 Text Processing with NLTK 3 Cookbook*. Packt Publishing - ebooks Account.
- Robertson, J. M. and Webb, W. A. (1998). *Cake-Cutting Algorithms: Be Fair if You Can*. A K Peters/CRC Press, first edition.
- Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 1 edition.
- Wickham, H. and Grolemund, G. (2017). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data*. O’Reilly Media, 1 edition.



מכון לב  
Lev Institute

אב תשע"א  
אוגוסט 2011

מכון טל  
Tal Institute

מרצה נכבד/ה

שלום רב,

מכון נוה  
Naveh Institute

**הנדון: סקר שביעות רצון מההוראה, סמסטר ב' תשע"א**

מכון לוסיג  
Lustig Institute

מצ"ב תוצאות סקר שביעות רצון מההוראה שנערך לקראת סיום סמסטר ב'.

הסטודנטים נתבקשו לסמן את מידת הסכמתם להיגדים שלהלן על פני סולם בן שבע דרגות מ-4 עד 10, כאשר הדרגה 10 מסמנת את מידת ההסכמה הגבוהה ביותר, והדרגה 4 את מידת ההסכמה הנמוכה ביותר.

- |  |
|--|
| 1. המרצה מלמד/ת בצורה מעניינת                    |
| 2. המרצה מלמד/ת בצורה מאורגנת                    |
| 3. המרצה מתייחס/ת בצורה נאותה לסטודנטים          |
| 4. המרצה שומר/ת על לוח הזמנים                    |
| 5. המרצה זמין/נה מחוץ לשעות ההוראה               |
| 6. באופן כללי הייתי מרוצה מאיכות ההוראה של המרצה |

בחישוב התוצאות לא נלקחו בחשבון תגובות הסטודנטים שסימנו שמידת השתתפותם בקורס הייתה מועטה או שכלל לא השתתפו בקורס.

בהערות הסטודנטים נתבקשו להתייחס לנושאים כגון:

- מידת התיאום בין המרצה והמתרגל,
- מידת יכולתו של המרצה להרצות בפני סטודנטים רבים באולמות גדולים,
- האם המרצה מספק חומר קריאה שממנו ניתן ללמוד אם סטודנט נעדר משיעור,
- האם המרצה משתמש באמצעי הוראה כגון מצגות, moodle, פורומים או דואל כדי להקל על הלמידה

לידיעתכם: הסטודנטים אינם מחויבים להשתתף בסקר. כתוצאה מכך קיים לעתים פער משמעותי בין מספר הסטודנטים הרשומים בקורס לבין מספר התגובות שנתקבלו בסקר. עם זאת, אני ממליץ להתייחס לתוצאות ולהערות בכובד ראש הראוי. בטוחני שהן עשויות לסייע בידך להמשיך ולקדם את טיב ההוראה שלך למען הסטודנטים שלנו.

בתודה על שיתוף הפעולה  
ובברכה,

פרופ' מנחם שטיינר  
רקטור



