**AT Project**

**The course project is worth 15% of the final** [**grade**](javascript:void(0))**.**

You may work in groups of one, two, three, or four. The complexity of your task does not depend on the group size, which means that forming a group of three or four will reduce your workload. A group should submit one programand one final report, presentation and all members of the group will get the same grade.

**Task selection:** First, you need to form a group, select a task for your project and post it on the Canvas in the Discussion field (Include names of group members and topic you chose). **Do not** **choose the topic which is already chosen.**

**Each group must have a unique project.**

**Project proposal (October 28 1 point):** You should submit a one-page proposal by 11:59pm on Sunday, October 28. It should describe the selected problem, intended approach to solving it, and methods for evaluating your results.

[**Final**](javascript:void(0)) **report, slides, codes, and reports (November 15, 9 points):** The final report should include the description of your task, [summary](javascript:void(0)) of results, main conclusions, and discussion of any surprising discoveries. The report is due by 11:59pm on Thursday, November 15.

[**Demonstration**](javascript:void(0)) **and presentation (5 points):** The project concludes with a 10-15 min presentation held during the last class period (Friday, November 16). The members of the group will get the grade depending on the presentation.

**Submitting your assignment**

* Submission via Canvas Assignment.
  + It is your responsibility to submit your report, code, and presentation in a timely fashion.
* All files should be zipped together.
* A report that presents the performance evaluation of your solution.
  + The report should be properly formatted (an academic format style, such as ACM or IEEE being preferred) and contain quantitative data along with you analysis of these data.

**Late Submission Policy**

* **Late work will be not accepted.**

**Some project Ideas**

1. A membership algorithm for CFG (CYK algorithm)
2. Automata in Compilers
3. The Automation Process of an ATM
4. Automata and DNA Computing

T. Krasińskiand, S. Sakowski, “A theoretical model of the Shapiro finite state automaton built on DNA,” *Theoretical and Applied Informatics*, vol. 18, pp. 161-174, 2006.

T. Krasiński and S. Sakowski, *Extended Shapiro Finite State Automaton Built on DNA*. Uniwersytet \Lódzki. Wydzia\l Matematyki i Informatyki, 2008.

T. Krasinski, S. Sakowski, and T. Poplawski, “Autonomous push-down automaton built on DNA,” *Informatica*, vol. 36, no. 3, p. 263+, Sep-2012.

Y. Benenson, B. Gil, U. Ben-Dor, R. Adar, and E. Shapiro, “An autonomous molecular computer for logical control of gene expression,” *Nature*, vol. 429, no. 6990, May 2004.

1. Cellular automata as NFAs/DFAs (Game of Life simulator known as Golly).
2. The Church-Turing Thesis. Undecidable problems for Recursively Enumerable Languages
3. Implementation and visualization of procedure Context-Free Grammar (CFG) to Push-Down Automata (PDA).
4. Implementing the procedure NFA into DFA
5. LL-parser
6. Implementing the Pumping Lemma procedure
7. Mealy and Moore machines
8. Natural language processing. Computational Linguistics – FSM Lexicon.
9. NP complexity and how it relates to automata. The halting problem.
10. Langton’s Ant. Langton’s Ant is a two-dimensional universal Turing Machine.