Mathematical Foundations of Software Engineering

Course title:

Mathematical Foundations of Software Engineering

Course timing:	**May 24, 2017**
Mode of study:	**Lectures: 6 hours, Practice: 4 hours, Total: 10 hours**
Study materials:	Announced May 5th on http://edu.susu.ru

Prerequisites for entering the course:

Basics of Discrete Mathematics for Computing: Principles of counting. Logic and proof methods, including induction. Basic recurrence relations. Basics of algorithm complexity. Sets, relations (in discrete mathematics sense, as interrelation between elements of set), functions(in discrete mathematics sense, as interrelation between elements of two sets). Elementary graph theory. Elementary number theory.
Basics of C Programming Language: assignment, branching, loops. Code reading and understanding.

Course summary:

This course is devoted to application of the mathematical methods in software engineering. Introduction to main principles and formal methods used in software engineering process will be provided. Finite automata software verification technique will be introduced. Model checking software verification method, PROMELA verification modeling language and SPIN verification software package will be seen.
Course is lectured by *Assoc. Prof. Valentin Golodov* (South Ural State University (SUSU), Chelyabinsk, Russia). His research interest area includeserrorless computing, interval analysis, GPU computing.

Course outline:

#	Title	Duration	Summary
Lectures			
1	Introduction to mathematical foundations of software engineering	2hours	Mathematical foundations of software engineering:Boolean logic, first-order logic, models of first-order logic. Introduction to program verification, applications in Software Engineering. Completeness Theorem. Regular expressions, regular sets, finite-state machines, and applications in Software Engineering. Graph Theory, graph algorithms. Statecharts, Petri Nets and their role in Software Engineering.
2	Finite state machines	2 hours	Finite State Machines as technique for modeling the states and transitions of a software system.
3	Verification of software	2 hours	Model checking.PROMELA (Process or Protocol Meta Language).
Practice			
4	Verifying a model of	4 hours	Verification of model using model checking

	software		technique and SPIN model checker.

Reading:

Lectures:
http://www4.in.tum.de/publ/papers/mod96_boesswet_1997_Publication.pdf
http://www.slideserve.com/wynona/specification-techniques-and-formal-specifications
http://www.slideserve.com/mikayla-booth/requirements-techniques-cont
Practice:
PROMELA:https://en.wikipedia.org/wiki/Promela
SPIN:http://spinroot.com/spin/whatispin.html

Software:

SPIN software (http://spinroot.com/spin/Src/) will be provided as part of the virtual box image.
Virtual box program should be preinstalled on student's notebooks.

Course timetable:

Date	Time	Classes
May 24th	**9:00-12:00**	**Lectures**
	13:00-15:00	**Lectures**
	15:00-17:00	**Practice**