Event-driven systems - course description

General information

Course name	Event-driven systems
Course ID	11.9-WE-AutD-E-dS-Er
Faculty	Faculty of Computer Science, Electrical Engineering and Automatics
Field of study	Automatic Control and Robotics / Computer Control Systems
Education profile	academic
Level of studies	Erasmus programme
Beginning semester	winter term 2017/2018

Course information

Semester	2
ECTS credits to win	3
Course type	obligatory
Teaching language	english
Author of syllabus	• dr hab. inż. Andrei Karatkevich, prof. UZ

Classes forms

The class form	Hours per semester (full-time)	Hours per week (full-time)	Hours per semester (part-time)	Hours per week (part-time)	Form of assignment
Lecture	15	1	-	-	Exam
Laboratory	15	1	-	-	Credit with grade

Aim of the course

- To provide knowledge on the ways and methods of formal specification of the event-driven systems
- To maintain the theoretcal basis necessary for understanging the ways of design and verification odf the event-driven systems

Prerequisites

Discrete process control

Scope

Informal introduction to the event-driven systems.

Mathematical foundations. Elements of the automata theory necessary for formal specification of an event-driven systems. Finite Stste Machine as a model of an event-driven systems. Deterministic and undeterministic automata.

Introduction to the temporal logic. Structore of the time, linear and branchung time. Operators and expressions of the temporal logic. Logic LTL, CTL, CTL*. Intuitive examples of specification of the simple event-driven systems using the temporal logic.

Reactive event-driven systems. General concept of HCFSM. Synchronous and asynchronous implementation of the event-driven systems.

Formal verification of the event-driven systems at the level of specification. System analysis by mean of studying of the specification which provided using LTL or CTL. "Safeness" and "liveness" propertries. "Liveness" and "safeness" properties. Counter-examples. Methods of model checking. Using a model checker (NuSMV is used as an example of such tool.)

Teaching methods

Lecture: conventional lecture
Laboratory: laboratory exercises

Learning outcomes and methods of theirs verification

Outcome description	Outcome symbols	Methods of verification	The class form
Is able to prepare a formal specification of a control device		• a quiz • an exam - oral, descriptive, test and other	• Lecture • Laboratory
Applies the appropriate mathematical tools in design of the event-driven systems		• a quiz • an exam - oral, descriptive, test and other	• Lecture • Laboratory
Knows the basic concepts of automata theory necessary to desing the simple control devices		• an exam - oral, descriptive, test and other	• Lecture
Verifies a model of an event-driven control system by means of a model checker		• a quiz • carrying out laboratory reports	• Laboratory

Assignment conditions

Lecture: The condition of pass is to obtain a positive assessment from the written examination.
Laboratory: a conditional of pass is to obtain positive grades from all laboratory exercises that are expected to be performed within the laboratory program.

Components of the final grade: lecture: 50% + laboratory: 50%

Recommended reading

1. Baier Ch., Katoen J.-P.: Principles of Model Checking, MIT Press, 2008.
2. Cavada R., Cimatti A., Keighren G., Olivetti E., Pistore M., Roveri M.: NuSMV 2.5 Tutorial (http://nusmv.fbk.eu/NuSMV/tutorial/index.html).
3. Clarke E. M., Jr., Grumnerg O., Peled D. A.: Model Checking, MIT Press, 1999.
4. Emerson E. A., Temporal and modal logic, Handbook of Theoretical Computer Science, Chapter 16, MIT Press, 1990.
5. Grobelna I.: Model Verification with NuSMV, Oficyna Wydawnicza Uniwersytetu Zielonogórskiego, Zielona Góra, 2011. (in Polish).
6. Pecol J.: Embedded Systems. A Contemporary Design Tool, Willey, 2008.

Further reading

1. Girault G., Volk R.: Petri Nets for Systems Engineering. A Guide to Modeling, Verification and Applications, Springer Verlag, Berlin, 2003.
2. Grumberg O., Veith H. (Eds.): 25 Years of Model Checking - History, Achievements, Perspectives. Lecture Notes in Computer Science 5000, Springer, 2008.
3. Kropf T., Introduction to Formal Hardware Verification, Springer, Berlin, 1999.
4. Øhrstrøm P., Hasle P. F. V.: Temporal logic: from ancient ideas to artificial intelligence, Springer, 1995

Notes

Modified by dr hab. inż. Andrei Karatkevich, prof. UZ (last modification: 29-03-2018 16:41)