SYLLABUS / FIŞA DISCIPLINEI

1. Information on the study programme / Date despre programul de studii

1.1. Institution / Instituţia de învăţământ superior	Universitatea de Vest din Timişoara
1.2. Faculty / Facultatea	Matematică şi Informatică
1.3. Department / Departamentul	Computer Science (Informatică)
1.4. Study program field	Computer Science (Informatică)
1.5. Study cycle/ Ciclul de studii	Bachelor / licenţă
1.6. Study programme / Programul de studii / calificarea*	Computer Science / Informatică în limba engleză / Database administration / *Administrator baze de date - 252101; Computer network administration / Administrator de retea de calculatoare - 252301; Analyst / Analist - 251201; Research assistant in computer science / Asistent de cercetare în informatica - 214918; Teacher in secondary schools / Profesor în învatamântul gimnazial - 233002; Programmer / Programator - 251202; Software systems designers / Proiectant sisteme informatice - 251101*

2. Information on the course / Date despre disciplină

2.1. Title of the course / Denumirea disciplinei	Formal Languages and Automata Theory						
2.2. Teacher in charge of the course / Titularul activităţilor de curs	Madalina Erascu						
2.3. Teacher in charge of the seminar / Titularul activităţilor de seminar	Madalina Erascu						
2.4. Study year / Anul de studii	1	2.5. Semester / Semestrul	2	2.6. Examination type / Tipul de evaluare: E(xam)/C(olloquim)	E	2.7. Course type / Regimul disciplinei: M(andatory)/ E(lective)/ F(acultative)	M

3. Estimated study time (number of hours per semester) /Timpul total estimat (ore pe semestru al activităţilor didactice)

3.1. Attendance hours per week / Număr de ore pe săptămână	4	out of which / din care: 3.2 lecture/ curs	2	3.3. seminar/laborator	2
3.4. Attendance hours per semester / Total ore din planul de învăţământ	56	out of which: 3.5 lecture / curs	28	3.6. seminar/laborator	28
Distribution of the allocated amount of time / Distribuţia fondului de timp*					**hours/ ore**
Individual study /Studiu după manual, suport de curs, bibliografie şi notiţe					36

Supplementary documentation at library or using electronic repositories / Documentare suplimentară în bibliotecă, pe platformele electronice de specialitate	30
Preparing for laboratories, homework, reports etc. /Pregătire seminarii/laboratoare, teme, referate, portofolii şi eseuri	30
Exams / Examinări	2
Tutoring / Tutorat	2
3.7. Total number of hours of individual study / Total ore studiu individual	80
3.8. Total number of hours per semester / Total ore pe semestru	160
3.9. Number of credits (ECTS) / Număr de credite	6

4. Prerequisites (if it is the case) / Precondiţii (acolo unde e cazul)

4.1. curriculum / de curriculum	Not the case
4.2. skills / de competenţe	Basic mathematical knowledge, problem solving and programming skills

5. Requirements (if it is the case) / Condiţii (acolo unde e cazul)

5.1. for the lecture / de desfăşurare a cursului	Classroom with blackboard and video projector
5.2. for the seminar, laboratory / de desfăşurare a seminarului/laboratorului	Classroom with blackboard, computers and video projector

6. Acquired skills / Competenţe specifice acumulate

Professional skills / Competenţe profesionale	Introduce concepts in automata theory and theory of computation: ● Identify different formal language classes and their relationships ● Design grammars and recognizers for different formal languages ● Prove or disprove theorems in automata theory using their properties ● Determine the decidability and intractability of computational problems
Transversal skills / Competenţe transversale	The ability to communicate knowledge about different notions from formal languages and automata theory and their usage.

7. Objectives of the course / Obiectivele disciplinei (reieşind din grila competenţelor specifice

acumulate)

7.1. General objective / Obiectivul general al disciplinei	Knowledge and understanding of basic notions from the formal languages theory: grammars and automata
7.2. Specfic objectives / Obiectivele specifice	*Knowledge objectives:* (1) Identify different formal language classes and their relationships (2) Design grammars and recognizers for different formal languages *Habilitation objectives*: (1) Prove or disprove theorems in automata theory using its properties (2) Determine the decidability and intractability of computational problems *Attitudinal objectives*: (1) to argue the importance of formal languages, automata, decidability results to an IT specialist.

8. Content / Conţinuturi*

8.1. Lecture / Curs	**Teaching strategies / Metode de predare**	**Remarks, details / Observaţii**
C1. (2h) Course overview. Introduction to Automata Theory & Formal Languages	Lecture, conversation, illustration	References: 1. M. Erascu - slides 2. Introduction to automata theory, languages and computation - Authors: JE Hopcroft, R Motwani and JD Ullman, Publisher:Addison Wesley/Pearson; 2rd Edition
C2. (2h) Finite Automata	Lecture, conversation, illustration	Same as above
C3. (2h) Regular Expressions	Lecture, conversation, illustration	Same as above
C4. (2h) Regular Language Properties	Lecture, conversation, illustration	Same as above
C5. (2h) Context Free Grammars and Languages	Lecture, conversation, illustration	Same as above
C6. (2h) Pushdown Automata	Lecture, conversation, illustration	Same as above
C7. (2h) Midterm		
C8-9. (4h) Context-Free Language Properties	Lecture, conversation, illustration	Same as above

C10-11. (2h) Turing Machines	Lecture, conversation, illustration	Same as above
C12-C13. (4h) Undecidability	Lecture, conversation, illustration	Same as above
C14. (2h) Course & Finals Review	Lecture, conversation, illustration	Same as above

Recommended bibliography / Bibliografie:

[1] JE Hopcroft, R Motwani and JD Ullman. Introduction to automata theory, languages and computation Addison Wesley/Pearson; 3rd Edition

[2] Dexter C. Kozen. Automata and Computability (Undergraduate Texts in Computer Science). Springer (August 1997)

[3] Azadeh Farzan. Introduction to Theory of Computation. Lecture notes. https://www.cs.toronto.edu/~azadeh/teaching/teaching.html

8.2. Seminar, lab / Seminar, laborator	**Teaching/learning strategies / Metode de predare/ invăţare**	**Remarks, details / Observaţii**
S1-2. (4h) Detect the language generated by a grammar; detect the grammar which generates a given language.	Questioning, dialogue, collaborative learning	Based on the notions presented in the lecture, the students will be able to access the homework from the course website (https://merascu.github.io/links/FLAT.html). They have to prepare it. It will then will be discussed in the class.
S3-4. (4h) Design of Finite Automaton: deterministic finite automaton (DFA) or nondeterministic finite automaton (NFA).	Same as above	Same as above
S5-6. (4h) Construction of regular expressions and of regular languages generated by then. Construction of eps-NFA.	Same as above	Same as above
S7-8. (4h) Usage of Pumping lemma	Same as above	Same as above
S9-10. (4h) Design of Context Free Grammars from given languages		
S11-12. (4h) Design of Pushdown Automata	Same as above	Same as above

S13-14. (4h) Turing Machines and Undecidability	Same as above	Same as above

Remarks: Students could also choose projects from the list below (this list is subject to change - the actual list of subjects will be listed on the course website):

1. Evaluation of arithmetical expressions using two stacks.
2. Generation and evaluation of arithmetical expressions using polish notation
3. Simulation of an DFA
4. Evaluation of regular expressions and construction of an DFA equivalent
5. DFA minimization
6. Conversion to Chomsky normal form
7. Simulation deterministic pushdown automata.

Students who decide to work on a project must announce the lecturer on the chosen topic. The project consists of a program solving the problem as well as documentation. The lecturer must be consulted on what the documentation must contain. Projects are presented to the lecturer.

Recommended bibliography / Bibliografie

[1] JE Hopcroft, R Motwani and JD Ullman. Introduction to automata theory, languages and computation Addison Wesley/Pearson; 3rd Edition

[2] Dexter C. Kozen. Automata and Computability (Undergraduate Texts in Computer Science). Springer (August 1997)

[3] Azadeh Farzan. Introduction to Theory of Computation. Lecture notes. https://www.cs.toronto.edu/~azadeh/teaching/teaching.html

9. Correlations between the content of the course and the requirements of the IT field / Coroborarea conţinuturilor disciplinei cu aşteptările reprezentanţilor comunităţii epistemice, asociaţiilor profesionale şi angajatorilor reprezentativi din domeniul aferent programului

The content of the lecture is consistent with the one of similar courses from other universities. It covers the fundamental aspects necessary for the familiarity with issues of formal languages and automata. The content is not very useful for ordinary IT companies, but students can train their algorithmic thinking and programming languages through the proposed projects. Nevertheless, the lecture trains the ability of thinking and problem solving, tasks which are indispensables for a programmer.

10. Evaluation / Evaluare*

Activity / Tip de activitate	10.1. Evaluation criteria / Criterii de evaluare**	10.2. Evaluation methods / Metode de evaluare***	10.3. Weight in the averaged mark / Pondere din nota finală
10.4. Lecture / Curs	Knowledge and application of notions from C1-C7.	Midterm	30%
	Knowledge and application of	Written exam in the	40%

	notions from C1-C14.	exam session	
10.5. Seminar/ lab	The ability to learn and apply concepts presented during the lectures.	Homeworks and activity (oral examination)	20%
10.6. Presentation	The ability to either implement an algorithm in a certain programing language for certain notions presented during the lecture (from a project list which will be posted after the first week on the website) or to a more theoretical topic with algorithmic solution. These must be presented in oral form (15 minutes presentation + 5 minutes questions) in front of your colleagues.		10%

10.6. Minimal knowledge for passing / Standard minim de performanţă

Minimal knowledge for passing (grade 5): acquiring fundamental understanding of the knowledge of automata theory and formal languages.
Criteria for the maximal grade, that is 10, is to fulfill 10.6. 1-2 weeks before the actual presentation you must discuss it with your teacher.
The final grade is computed as a weighted average of the grades given for the components specified in 10.4 and 10.5. 10.6 contributes to the final grade, however it does not have negative impact on the grade if this activity is not fulfilled. The exam is passed if the average is equal or greater than 4.1 (not necessary as each note to be greater than 4.1). The start at Midterms and Final Exam is 0. If the grade is greater than equal to 4.1 means 5, greater than equal to 5.1 means 6, …, greater than equal to 9.1 means 10.
At each exam sessions (including reexamination and improvements), the score is computed by the same rule. Midterm can not be retaken.
There is no mandatory presence requirement, however, note that your seminar grade is based on your activity during the semester (solving exercises at the whiteboard, tests, etc).
Note: Students may attend office hours (2 modules / week according to the schedule set out at the beginning of the semester) where the lecturer (course/seminar) answers questions students and provides further explanations related to course content, applications from seminary themes.

Date/ Data completării

16.02.2019

Signature (lecture) /
Semnătura titularului de curs
Madalina Erascu

Signature (seminar)
Semnătura titularului de seminar
Madalina Erascu

Signature (director of the department)
Semnătura directorului de departament
Conf.dr. Victoria Iordan