Peruvian Computing Society (SPC)
School of Computer Science
Sillabus 2022-I

1. COURSE

CS211. Theory of Computation (Mandatory)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Mandatory
2.6 Modality	:	Face to face
2.7 Prerrequisites	:	CS1D2. Discrete Structures II. (2^{nd} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE

This course emphasizes formal languages, computer models and computability, as well as the fundamentals of computational complexity and complete NP problems.

5. GOALS

- That the student learn the fundamental concepts of the theory of formal languages.

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Assessment**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Assessment**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Assessment**)

7. SPECIFIC COMPETENCES

a1) Apply demonstration techniques (direct method, contrapositive, induction and contradiction) to demonstrate properties in discrete structures and algorithms.

a30) Apply basic concepts of set theory, relationships and functions.

a31) Apply the computational concept of non-determinism in problem solving.

b11) Understand the difference between an NP-difficult problem and one that has a polynomial solution.

b19) Understand the difference between a problem without a solution (undecidable problem) and a problem that does have a solution

b20) Identify and solve a problem that is solvable by automata theory and recognize which is the simplest type of automata that solves the problem.

j7) Apply automaton theory for optimization and problem solving.

8. TOPICS

Unit 1: Basic Automata Computability and Complexity (20)	
Competences Expected: a	
Topics	**Learning Outcomes**
• Finite-state machines • Regular expressions • The halting problem • Context-free grammars • Introduction to the P and NP classes and the P vs. NP problem • Introduction to the NP-complete class and exemplary NP-complete problems (e.g., SAT, Knapsack) • Turing machines, or an equivalent formal model of universal computation • Nondeterministic Turing machines • Chomsky hierarchy • The Church-Turing thesis • Computability • Rice's Theorem • Examples of uncomputable functions • Implications of uncomputability	• Discuss the concept of finite state machines [Assessment] • Design a deterministic finite state machine to accept a specified language [Assessment] • Generate a regular expression to represent a specified language [Assessment] • Explain why the halting problem has no algorithmic solution [Assessment] • Design a context-free grammar to represent a specified language [Assessment] • Define the classes P and NP [Assessment] • Explain the significance of NP-completeness [Assessment] • Explain the Church-Turing thesis and its significance [Familiarity] • Explain Rice's Theorem and its significance [Familiarity] • Provide examples of uncomputable functions [Familiarity] • Prove that a problem is uncomputable by reducing a classic known uncomputable problem to it [Familiarity]
Readings : [Mar10], [Lin11], [Sip12]	

Unit 2: Advanced Computational Complexity (20)	
Competences Expected: a,b	
Topics	**Learning Outcomes**
• Review of the classes P and NP; introduce P-space and EXP • Polynomial hierarchy • NP-completeness (Cook's theorem) • Classic NP-complete problems • Reduction Techniques	• Define the classes P and NP (Also appears in AL/Basic Automata, Computability, and Complexity) [Assessment] • Define the P-space class and its relation to the EXP class [Assessment] • Explain the significance of NP-completeness (Also appears in AL/Basic Automata, Computability, and Complexity) [Assessment] • Provide examples of classic NP-complete problems [Assessment] • Prove that a problem is NP-complete by reducing a classic known NP-complete problem to it [Assessment]
Readings : [Mar10], [Lin11], [Sip12], [HU13]	

Unit 3: Advanced Automata Theory and Computability (20)	
Competences Expected: j	
Topics	**Learning Outcomes**

• Sets and languages
  – Regular languages
  – Review of deterministic finite automata (DFAs)
  – Nondeterministic finite automata (NFAs)
  – Equivalence of DFAs and NFAs
  – Review of regular expressions; their equivalence to finite automata
  – Closure properties
  – Proving languages non-regular, via the pumping lemma or alternative means
• Context-free languages
  – Push-down automata (PDAs)
  – Relationship of PDAs and context-free grammars
  – Properties of context-free languages
	• Determine a language's place in the Chomsky hierarchy (regular, context-free, recursively enumerable) [Assessment]
	• Convert among equivalently powerful notations for a language, including among DFAs, NFAs, and regular expressions, and between PDAs and CFGs [Assessment]

Readings : [HU13], [Bro93]	

9. WORKPLAN

9.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

9.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

9.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

10. EVALUATION SYSTEM

********* EVALUATION MISSING ********

11. BASIC BIBLIOGRAPHY

[Bro93] J. Glenn Brookshear. *Teoría de la Computación*. Addison Wesley Iberoamericana, 1993.

[HU13] John E. Hopcroft and Jeffrey D. Ullman. *Introducción a la Teoría de Autómatas, Lenguajes y Computación*. Pearson Education, 2013.

[Lin11] Peter Linz. *An Introduction to Formal Languages and Automata*. 5th. Jones and Bartlett Learning, 2011.

[Mar10] John Martin. *Introduction to Languages and the Theory of Computation*. 4th. McGraw-Hill, 2010.

[Sip12] Michael Sipser. *Introduction to the Theory of Computation*. 3rd. Cengage Learning, 2012.