**Nazwa przedmiotu:**

Languages, Automata and Computation

**Koordynator przedmiotu:**

Ilona Bluemke

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Computer Science

**Grupa przedmiotów:**

Technical Courses

**Kod przedmiotu:**

ELAC

**Semestr nominalny:**

5 / rok ak. 2015/2016

**Liczba punktów ECTS:**

6

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

lecture attendance: 15 x 2 h = 30 h;
laboratory attendance: 15 x 2 h = 30 h;
implementing laboratory projects: 15 x 3 h = 45h;
preparation to lectures (reviewing lecture notes, reading of recommended literature): 15 h;
preparation to written class tests (including participation in consultations): 2 x 6 h. + 3 h = 15 h;
Total: 30 + 30 + 45 + 15 + 15 = 135 h.

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

2.5

**Język prowadzenia zajęć:**

angielski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

3

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład:  | 30h |
| Ćwiczenia:  | 15h |
| Laboratorium:  | 0h |
| Projekt:  | 15h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

C/C++ or Java programming

**Limit liczby studentów:**

30

**Cel przedmiotu:**

The course aims at giving basic understanding of concepts underlying linguistic approach to the computation; important classes of languages (regular, context-free, context sensitive, general) and their respective grammar- and automata-based models are presented. Linguistic approach is also applied to the description of computability and computational complexity.
The course is structured along classical topics of finite grammatical description of infinite languages, automata based abstract devices for generating and/or recognizing them, and assessing computational complexity of algorithmic problems in different models of computation. The style of the coverage will underline practical relevance of the theory and understanding of concepts.

**Treści kształcenia:**

1. Introduction (4h): Short overview of topics to be covered; mathematical refresher: notation, alphabets, sequences, tuples, functions and relations, graphs, strings, grammars and formal languages, criteria for classifying languages.
2. Regular Languages and Finite Automata (6h): Three equivalent methods for specifying regular languages: regular grammars, regular expressions, finite automata; algorithmic conversions of representations; from RE to automata and back to RE; nondeterminism in automata, trading succinctness for determinism, deterministic recognizers. Closure properties of regular languages, the pumping lemma, some decision problems of regular languages.
3. Context–Free Languages and Push-down Automata (8h): Conceptually simplest non-regular language definition using context-free grammar. Properties of context-free grammars, language preserving transformation of CFGs, canonical forms, Chomski Normal Form, Greibach Normal Form; grammar as a generating device, derivation trees, ambiguity in CF grammars and languages. Push-down automata (PDA) recognizing CF languages, nondeterministic and deterministic PDAs. Pumping lemma for CF languages, closure properties, some decision problems of CF languages; simplest non-context-free language. Practical aspects of using CF grammars in defining programming languages; meta-notations and styles implied by parser generating tools.
4. Beyond Context-Free (4h): context-sensitive and general grammars and languages; linear bounded automata and Turing machines. Kuroda Normal Form for context-sensitive grammars. Some intuitive examples of non-context-free languages and their recognition. Parallel rewriting systems (Lindenmayer Systems), deterministic context-free L-systems, geometric interpretation of L-systems, examples of fractals generated by L-systems.
5. Complexity of computation (8h). The Church-Turing thesis, undecidable, decidable, intractable, and efficiently computable problems; examples. Algorithmic decision problems as language recognition problems. Time and space complexity; the notion of reducibility, polynomial reducibility and complexity classes P and NP. The notion of completness; NP-Complete class, the SAT problem and Cook's theorem. Examples of problems in NPC, some notable reductions of problems. Practical complexity with real processing platforms and high level languages. Approximation heuristics for intractable problems.

Tutorials
 Tutorials will offer exercises focused on topics covered during lectures to enhance understanding and exhibit practical relevance of theoretical concepts. In particular, exercises and homework problems concerning transformations of automata and context free grammars will be proposed.
Two midterm assessment tests will be scheduled.

The project will consists of two individual (or tandem) assignments concerning:
• Regular language processing / recognition
• Context-free language manipulation / recognition
• Attempts to approximate solutions of some hard problems or a mixture thereof.

**Metody oceny:**

Two midterm assessment tests will be scheduled.

**Egzamin:**

nie

**Literatura:**

1. J.E. Hopcroft, R. Motwani, J.D. Ullman, Introduction to Automata Theory, Language and Computation, 3nd ed., Addison-Wesley.
2. M. Sipser, Introduction to the Theory of Computation, 2nd ed., Course Technology, 2004.
3. P. Linz, An Introduction to Formal Language and Automata (4th ed.), Jones and Bartlett, 2006.

**Witryna www przedmiotu:**

https://studia.elka.pw.edu.pl/priv/14Z/ELAC.A/

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt ELAC\_W01:**

Student, who passed this course is able to apply deterministic and nondeterministic automata to describe regular languages.

Weryfikacja:

test, exam

**Powiązane efekty kierunkowe:** K\_W01

**Powiązane efekty obszarowe:** T1A\_W01, T1A\_W02, T1A\_W03, T1A\_W07

**Efekt ELAC\_W02:**

Student, who passed this course knows categories of languages (regular,context free, etc.).

Weryfikacja:

test, exam

**Powiązane efekty kierunkowe:** K\_W01

**Powiązane efekty obszarowe:** T1A\_W01, T1A\_W02, T1A\_W03, T1A\_W07

**Efekt ELAC\_W03:**

Student, who passed this course knows pushdown automata and Turing machines.

Weryfikacja:

test, exam

**Powiązane efekty kierunkowe:** K\_W01

**Powiązane efekty obszarowe:** T1A\_W01, T1A\_W02, T1A\_W03, T1A\_W07

**Efekt ELAC\_W04:**

Student who passed this course can give examples of undecidable, decidable, intractable, and efficiently computable problems.

Weryfikacja:

test, exam

**Powiązane efekty kierunkowe:** K\_W01

**Powiązane efekty obszarowe:** T1A\_W01, T1A\_W02, T1A\_W03, T1A\_W07

### Profil ogólnoakademicki - umiejętności

**Efekt ELAC\_U01:**

Student, who passed this course is able to use deterministic and nondeterministic automata to model simple behavior or language.

Weryfikacja:

test, exam, project

**Powiązane efekty kierunkowe:** K\_U01, K\_U08

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U08, T1A\_U09

**Efekt ELAC\_U02:**

Student, who passed this course is able to to use regular expressions and transform them to NFA.

Weryfikacja:

test, exam , project

**Powiązane efekty kierunkowe:** K\_U01, K\_U02, K\_U08, K\_U09

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U02, T1A\_U08, T1A\_U09, T1A\_U08, T1A\_U09

**Efekt ELAC\_U03:**

Student, who passed this course is able to describe regular languages by different methods and transform them.

Weryfikacja:

test, project, exam

**Powiązane efekty kierunkowe:** K\_U01, K\_U02, K\_U03, K\_U08

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U02, T1A\_U03, T1A\_U08, T1A\_U09