

Theoretical Foundations of Computer Science - course description

General information	
Course name	Theoretical Foundations of Computer Science
Course ID	11.3-WI-INFP-TPI
Faculty	Faculty of Computer Science, Electrical Engineering and Automatics
Field of study	Computer Science / Industrial Information Systems
Education profile	academic
Level of studies	First-cycle studies leading to Engineer's degree
Beginning semester	winter term 2016/2017

Course information	
Semester	3
ECTS credits to win	6
Course type	obligatory
Teaching language	polish
Author of syllabus	- dr hab. inż. Marek Sawerwain, prof. UZ - prof. dr hab. inż. Andrzej Obuchowicz

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
Class	30	2	18	1,2	Credit with grade
Lecture	30	2	18	1,2	Exam

Aim of the course

- zapoznanie studentów z informacjami o podstawowych teoretycznych pojęciach dotyczących informatyki, jak algorytm, języki formalne, gramatyka bezkontekstowa, a także pojęciem automatów deterministycznych oraz niedeterministycznych,
- nauka podstawowych umiejętności w zakresie wyznaczania klasy złożoności algorytmów, sprawdzania własności stopu, dowodzenia częściowej poprawności,
- zapoznanie studentów z tezą Churcha-Turinga, z definicją problemu łatwo i trudno rozwiązywalnego oraz pojęciem NP-trudności,
- zapoznanie studentów z pojęciami dotyczącymi teorii algorytmów równoległych.

Prerequisites

Algorytmy i struktury danych, Logika dla informatyków, Podstawy systemów dyskretnych, Analiza matematyczna.

Scope

Wiadomości wstępne: algorytm i jego własności, notacja asymptotyczna. Poprawność algorytmów: algorytm poprawny, poprawność częściowa, własność określoności obliczeń, własność stopu; dowodzenie poprawności częściowej, dowodzenie własności stopu.

Podstawy teorii automatów i języków: automaty skończone i wyrażenia regularne, gramatyki bezkontekstowe, automaty ze stosem i języki bezkontekstowe, własności języków bezkontekstowych.

Prymitywne modele algorytmiczne. Teza Churcha--Turinga. Maszyna Turinga i jej warianty. Maszyna o dostępie swobodnym. Programy licznikowe.

Sprawność algorytmów. Miary efektywności algorytmów. Złożoność przestrzenna i czasowa. Złożoność pesymistyczna i średnia. Dolne i górne ograniczenie złożoności. Złożoność naturalna. Problemy algorytmicznie zamknięte i otwarte, luka algorytmiczna.

Klasyfikacja problemów algorytmicznych. Problemy łatwo-rozwiązywalne i trudno-rozwiązywalne. Klasy problemów algorytmicznych: logarytmiczne, wielomianowe, NP, NP-zupełne i wykładnicze. Otwarte problemy związane z klasyfikacją problemów algorytmicznych. Dowodzenie NP-zupełności. nierozwiązywalność i nierozstrzygalność.

Algorytmy współbieżne i probabilistyczne. Stała i rozszerzająca się współbieżność. Złożoność iloczynowa. Sieci - współbieżność o stałych połączeniach. Teza o obliczeniach równoległych. Klasa Nicka. Algorytmy RNC. Współbieżność rozproszona. Bezpieczeństwo i żywotność systemów współbieżnych.

Algorytmy probabilistyczne niektórych konwencjonalnych problemów algorytmicznych. Probabilistyczne klasy złożoności.

Teaching methods

Wykład: wykład konwencjonalny/tradycyjny.

Ćwiczenia: ćwiczenia rachunkowe.

Learning outcomes and methods of theirs verification

Outcome description	Outcome symbols	Methods of verification	The class form
---------------------	-----------------	-------------------------	----------------

Outcome description	Outcome symbols	Methods of verification	The class form
Potrafi oszacować złożoność czasową zadanego algorytmu.	K_W06 K_U09	a test with score scale	Lecture
Jest kreatywny w rozwiązywaniu problemów uwzględniając dostępne techniki i środki.	K_U10 K_K10	- a test with score scale - an observation and evaluation of activities during the classes	Class
Student zna pojęcie złożoności obliczeniowej dotyczące algorytmów równoległych.	K_W06	a test with score scale	Lecture
Potrafi definiować i charakteryzować klasy złożoności problemów algorytmicznych. Zna podstawowe pojęcia teorii poprawności algorytmicznej, automatów, gramatyk i języków.	K_W06	a test with score scale	Lecture
Jest w stanie przeprowadzić analizę poprawności algorytmu sekwencyjnego zapisanego w postaci blokowej i w języku programowania strukturalnego.	K_W06 K_U09	- a test with score scale - an observation and evaluation of activities during the classes	Class

Assignment conditions

Wykład - warunkiem zaliczenia jest uzyskanie pozytywnej oceny z egzaminu przeprowadzonego w formie pisemnej.

Ćwiczenia - warunkiem zaliczenia jest uzyskanie pozytywnych ocen z kolokwii pisemnych lub ustnych przeprowadzonych co najmniej raz w semestrze.

Składowe oceny końcowej = wykład: 50% + ćwiczenia: 50%

Recommended reading

- Banachowski L., Kreczmar A.: Elementy analizy algorytmów, WNT, Warszawa, 1982
- Bilski T., Chmiel K., Stokłosa J.: Zbiór zadań ze złożoności obliczeniowej algorytmów, Wydawnictwo Politechniki Poznańskiej, Poznań, 1992
- Cormen T. H., Leiserson C. E., Rivest R. L.: Wprowadzenie do algorytmów, WNT, Warszawa, 1997
- Harel D.: Rzecz o istocie informatyki, WNT, Warszawa, 2000
- Hopcroft J. E., Ullmann J.D.: Wprowadzenie do teorii automatów, języków i obliczeń, PWN, Warszawa, 2003
- Aho A.V., Ullmann J.D.: Wykłady z informatyki z przykładami w języku C, Helion, Gliwice, 2003

Further reading

- Ben Ari M.: Logika matematyczna w informatyce, WNT, Warszawa, 2005
- Graham R.L., Knuth D.E., Patashnik O.: Matematyka konkretna, PWN, Warszawa, 2002
- Papadimitriou C. H.: Złożoność obliczeniowa, Helion, Gliwice, 2012
- Ross K. A., Wright C. R. B.: Matematyka dyskretna, PWN, Warszawa, 2000
- Wróblewski P.: Algorytmy, struktury danych i języki programowania, Helion, Gliwice, 1997

Notes

Modified by dr hab. inż. Marek Sawerwain, prof. UZ (last modification: 06-09-2016 10:05)

Generated automatically from SylabUZ computer system