



Constraint Programming

Course description sheet

Basic information

Field of study Computer Science and Intelligent Systems		Didactic cycle 2022/2023	
Major -		Course code EISIS.II500.6245fa382b339.22	
Organisational unit Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering		Lecture languages english	
Study level First-cycle (engineer) programme		Mandatoriness Elective	
Form of study Full-time studies		Block General Modules	
Profile General academic		Course related to scientific research Yes	
Course coordinator	Antoni Ligęza, Weronika T. Adrian		
Lecturer	Antoni Ligęza		
Periods Semester 5, Semester 7	Method of verification of the learning outcomes Completing the classes	Number of ECTS credits 3	
	Activities and hours Lectures: 14 Laboratory classes: 14		



Goals

C1	Aby studenci znali podstawy programowania z ograniczeniami i umieli wykorzystać tę wiedzę w praktyce.
C2	Aby studenci docenili piękno i siłę programowania deklaratywnego! Programowanie z ograniczeniami bazuje w dużej mierze na Constraint Programming in Logic (CLP) - w tym Prologu. Chcę pokazać Studentom, że poza programowaniem proceduralnym i obiektowym jest jeszcze inny, piękny Świat; to Świat programowania deklaratywnego - w tym programowania z ograniczeniami (po polsku nazwa ta brzmi trochę głupio/dwuznacznie; może lepiej pozostać przy ang. Constraint Programming?). Wejście do tego Świata to trochę jak przejście Alicji na "drugą stronę lustra". Na początku trudne; jednak wytrwali odkryją jak pięknym i prostym językiem programowania jest Prolog i jego narzędzia dla programowania z ograniczeniami (np. clp(fd)) - a zwłaszcza w zastosowaniu do Constraint Programming. Przewiduję także wprowadzenie do innych systemów, w tym głównie MiniZinc - nowoczesny i szybko rozwijany język wysokiego rzędu dla programowania stricte deklaratywnego z ograniczeniami i optymalizacji. Celem kształcenia jest zatem aby Studenci rozumieli podstawy programowania z ograniczeniami - ironizując: aby pozbyli się ograniczeń w programowaniu... I aby dostrzegli i celowość jego stosowania - na choćby elementarnym poziomie... Ale z pełnym zrozumieniem. I także by rozumieli rolę, zalety i ograniczenia programowania deklaratywnego w informatyce. I docenili jego piękno. I - na koniec - dwa cytaty ze zbioru moich "Golden Ideas": "W żadnym innym języku programowania myślenie nie wyprzedza tak kładzenia kodu jak w Prologu." Ta uwaga rozciąga się także na programowanie deklaratywne z ograniczeniami. "Prolog to jednak język dla Elit (do studentów, którzy na wykład przychodzą nielicznie)." Ta uwaga rozciąga się także na programowanie deklaratywne z ograniczeniami. Tak - Prolog - programowanie deklaratywne/programowanie z ograniczeniami - jest dla elit; ale czyż aktualni Studenci - a przyszli absolwenci Wydziału EAIiB AGH nie aspirują do tego grona? W programie będą też inne narzędzia: Język MiniZinc - aktualnie mocno rozwijane potężne narzędzie programowania i optymalizacji z ograniczeniami. A dla entuzjastów - dowolne inne narzędzia, w tym nowoczesny Picat., którego składnia bazuje jednak na Prologu.

Course's learning outcomes

Code	Outcomes in terms of	Learning outcomes prescribed to a field of study	Methods of verification
Knowledge - Student knows and understands:			
W1	ma podstawową wiedzę na temat różnych paradygmatów programowania w tym programowania deklaratywnego, w tym zwłaszcza programowania logicznego, oraz opartego na koncepcji relacji i proceduralnego; wie jak dobrać paradygmat do rozwiązywania konkretnych problemów decyzyjnych i obliczeniowych w różnych dziedzinach informatyki	ISI1A_W05, ISI1A_W06	Execution of laboratory classes, Test results, Presentation, Completion of laboratory classes
W2	ma podstawową wiedzę na temat logiki i jej zastosowań w programowaniu; zna składnię i semantykę logiki rachunku zdań i rachunku predykatów; ma wiedzę dotyczącą modelowania ograniczeń z wykorzystaniem elementów logiki i algebry	ISI1A_W01, ISI1A_W04	Execution of laboratory classes, Presentation, Completion of laboratory classes
W3	ma podstawową wiedzę na temat paradygmatu programowania deklaratywnego w logice; zna podstawy opisu problemów i programowania w logice z ograniczeniami; zna podstawowe techniki i algorytmy rozwiązywania problemów z ograniczeniami.	ISI1A_W04, ISI1A_W06	Execution of laboratory classes, Test results, Presentation, Completion of laboratory classes
Skills - Student can:			



U1	potrafi ocenić przydatność różnych paradygmatów programowania i związanych z nimi środowisk programistycznych, w tym zwłaszcza w zakresie programowania deklaratywnego/programowania z ograniczeniami	ISI1A_U01, ISI1A_U06	Execution of a project, Execution of laboratory classes, Test results, Completion of laboratory classes
U2	potrafi czytać ze zrozumieniem, pisać, uruchamiać i weryfikować programy zapisane w języku programowania deklaratywnego; potrafi wykorzystać podstawowe pojęcia i konstrukcje języka programowania z ograniczeniami do zapisu programów	ISI1A_U06, ISI1A_U07	Execution of a project, Execution of laboratory classes, Test results, Completion of laboratory classes
U3	potrafi czytać ze zrozumieniem, pisać, uruchamiać i weryfikować programy zapisane w języku programowania logicznego, rozumie potrzebę, zasady i zastosowania programowania deklaratywnego, w tym zwłaszcza programowania logicznego	ISI1A_U06, ISI1A_U07	Execution of a project, Execution of laboratory classes, Test results, Completion of laboratory classes
Social competences - Student is ready to:			
K1	potrafi myśleć samodzielnie i konstruktywnie; studiować w celu podnoszenia swoich kwalifikacji i rozwoju osobowego; potrafi komunikować się i przekazywać wiedzę innym	ISI1A_K05	Execution of a project, Presentation, Completion of laboratory classes

Program content ensuring the achievement of the learning outcomes prescribed to the module

Przedmiotem nauczania jest programowanie z ograniczeniami (ang. Constraint Programming). Jest to programowanie deklaratywne polegające na definiowaniu ograniczeń, które powinno spełniać poszukiwane rozwiązanie.

Student workload

Activity form	Average amount of hours* needed to complete each activity form
Lectures	14
Laboratory classes	14
Preparation of project, presentation, essay, report	30
Realization of independently performed tasks	32
Student workload	Hours 90
Workload involving teacher	Hours 28

* hour means 45 minutes

Program content



No.	Program content	Course's learning outcomes	Activities
1.	Wstęp: wprowadzenie do programowania z ograniczeniami i programowania logicznego z ograniczeniami; elementarny przegląd problematyki i przykładów zastosowań programowania z ograniczeniami. Podstawy algebraiczne i logiczne definiowania ograniczeń. Strategia Backtracking Depth-First Search i jej modyfikacje.	W1, W2, W3, K1	Lectures
2.	Typy i definiowanie ograniczeń; przegląd wybranych predefiniowanych ograniczeń globalnych.	W1, W2, W3, U2	Lectures, Laboratory classes
3.	Bazowy algorytm Backtracking Search i jego modyfikacje; inne podejścia.	W1, W2, W3, U1, K1	Lectures
4.	Metody i algorytmy propagacji ograniczeń.	W3, U2, U3	Lectures
5.	Zwiększanie efektywności obliczeń: dekompozycja, relaksacja, sterowanie wnioskowaniem, heurystyki.	W3, U1, U2, U3	Lectures, Laboratory classes
6.	Informacje o wybranych narzędziach i przykłady problemów.	W3, U1, U2, U3, K1	Lectures, Laboratory classes
7.	Język MiniZinc — modelowanie rzeczywistych problemów	U1, U2, K1	Laboratory classes

Extended information/Additional elements

Teaching methods and techniques:

Lectures, E-learning, Group work, Gamification

Activities	Methods of verification	Credit conditions
Lectures	Presentation, Completion of laboratory classes	Zgodnie z regulaminem studiów
Lab. classes	Execution of a project, Execution of laboratory classes, Test results	Zgodnie z regulaminem studiów

Conditions and the manner of completing each form of classes, including the rules of making retakes, as well as the conditions for admission to the exam

Aby otrzymać zaliczenie, konieczne jest uzyskanie 50% z poniższych komponentów: — zadania laboratoryjne; — kolokwium; — projekt. W przypadku braku zaliczenia w pierwszym terminie, termin poprawkowy pozwoli na kolejne podejście do powyższych zadań.

Method of determining the final grade

Ocena końcowa wyliczana będzie na podstawie zajęć laboratoryjnych i będzie liczona jako średnia ważona trzech komponentów: - zadania laboratoryjne: 40% oceny. - projekt: 30% oceny. - kolokwium: 30% oceny.

Aby otrzymać zaliczenie, konieczne jest uzyskanie 50% punktów za każdy komponent z osobna. Dodatkowo, ocena może zostać podwyższona ze względu na aktywność na zajęciach lub wykładzie.

Manner and mode of making up for the backlog caused by a student justified absence from classes

W przypadku nieobecności, należy skontaktować się z prowadzącym, aby otrzymać dodatkowe zadania.



Prerequisites and additional requirements

Znajomość podstawowych zagadnień z zakresu logiki formalnej.

Student będzie korzystał z systemu kontroli wersji git.

Generalnie:

Znajomość matematyki w zakresie wymaganym dla studentów studiów informatycznych.

Entuzjazm i zapał do STUDIOWANIA - w tym samodzielnego poszerzania i pogłębiania wiedzy.

Uważność na zajęciach, zdolność do samodzielnego myślenia i wyłożonej pracy. Otwartość, kreatywność i zdolność do myślenia niestandardowego.

Rules of participation in given classes, indicating whether student presence at the lecture is obligatory

Wykład: Studenci uczestniczą w zajęciach poznając kolejne treści nauczania zgodnie z sylabusem przedmiotu. Studenci winni na bieżąco zadawać pytania i wyjaśniać wątpliwości. Prowadzenie notatek z wykładu jest usilnie rekomendowane. Rejestracja audiowizualna wykładu wymaga zgody prowadzącego. Ćwiczenia laboratoryjne: Obecność jest obowiązkowa. Studenci wykonują ćwiczenia laboratoryjne zgodnie z materiałami udostępnionymi przez prowadzącego. Student jest zobowiązany do przygotowania się w przedmiocie wykonywanego ćwiczenia, co może zostać zweryfikowane kolokwium w formie ustnej lub pisemnej. Zaliczenie zajęć odbywa się na podstawie zaprezentowania rozwiązania postawionych problemów, w tym jednego projektu wymagającego samodzielnej pracy. Zaliczenie modułu jest możliwe po zaliczeniu: wszystkich zajęć laboratoryjnych, projektu, kolokwium.

Literature

Obligatory

1. F. Rossi, P. van Beek, T. Walsh (eds.): Handbook of Constraint Programming, Elsevier, 2006.
2. R. Dechter: Constraint Processing, Morgan Kaufmann, 2003.
3. Krzysztof R. Apt: Principles of Constraint Programming. Cambridge University Press, 2003, 2006.
4. SWI-Prolog: biblioteka clp(fd)
5. MiniZinc Handbook: <https://www.minizinc.org/doc-latest/en/index.html>
6. <http://kti.mff.cuni.cz/~bartak/constraints/>

Optional

1. Krzysztof R. Apt and Mark G. Wallace: Constraint Programming usinc ECLiPSe. Cambridge University Press, 2007.2003, 2006.
2. <http://www.anclp.pl/>

Scientific research and publications

Research

1. Knowledge Representation and Reasoning, Constraint Programming

Publications

1. Constraint programming for constructive abduction : a case study in diagnostic model-based reasoning / Antoni LIGĘZA // W: Advanced solutions in diagnostics and fault tolerant control / eds. Jan M. Kościelny, Michał Syfert, Anna Szytber. — Cham : Springer, cop. 2018. — (Advances in Intelligent Systems and Computing ; ISSN 2194-5357 ; vol. 635). — Zawiera materiały z: DPS'2017 : 13th international conference on Diagnostics of Processes and Systems : Sandomierz, Poland, September 10-13, 2017. — ISBN: 978-3-319-64473-8 ; e-ISBN: 978-3-319-64474-5. — S. 94-105. — Bibliogr. s. 105, Abstr.
2. An experiment in causal structure discovery : a constraint programming approach / Antoni LIGĘZA // W: Foundations of Intelligent system : 23rd international symposium, ISMIS 2017 : Warsaw, Poland, June 26-29, 2017 : proceedings / eds. Marzena Kryszkiewicz [et al.]. — Switzerland : Springer International Publishing AG, cop. 2017. — (Lecture Notes in Artificial Intelligence ; ISSN 0302-9743 ; LNAI 10352). — ISBN: 978-3-319-60437-4 ; e-ISBN: 978-3-319-60438-1. — S. 261-268. — Bibliogr. s. 267-268, Abstr.. — Publikacja dostępna online od: 2017-06-14. — tekst: https://link-springer-1com-10000483e00f0.wbg2.bg.agh.edu.pl/content/pdf/10.1007%2F978-3-319-60438-1_26.pdf



3. Towards knowledge compilation for automated diagnosis: a qualitative, model-based approach with constraint programming / Antoni LIGĘZA // W: Advanced and intelligent computations in diagnosis and control : [12th international conference on Diagnostics of Processes and Systems (DPS) : Ustka, Poland 6-9 September 2015] / ed. Zdzisław Kowalczyk. — Switzerland : Springer International Publishing, cop. 2016. — (Advances in Intelligent Systems and Computing ; ISSN 2194-5357 ; vol. 386). — ISBN: 978-3-319-23179-2 ; e-ISBN: 978-3-319-23180-8. — S. 355-367. — Bibliogr. s. 366-367, Abstr.
4. Improving efficiency in constraint logic programming through constraint modeling with rules and hypergraphs / Antoni LIGĘZA // W: FedCSIS [Dokument elektroniczny] : proceedings of the Federated Conference on Computer Science and Information Systems 2012 : September 9-12, 2012 Wrocław, Poland / eds. M. Ganzha, L. Maciaszek, M. Paprzycki. — Dane tekstowe. — Warsaw : Polskie Towarzystwo Informatyczne ; Los Alamitos : IEEE Computer Society Press, 2012. — Dane na dysku Flash. — W bazie Web of Science ISBN 978-83-60810-48-4. — ISBN: 978-83-60810-51-4. — S. 101-107
5. Models and tools for improving efficiency in constraint logic programming / Antoni LIGĘZA // Decision Making in Manufacturing and Services ; ISSN 1896-8325. — 2011 vol. 5 no. 1-2, s. 69-78. — Bibliogr. s. 78, Abstr.. — tekst: http://journals.bg.agh.edu.pl/DECISION/2011-01-02/DM_2011_1_2_06.pdf



Learning outcomes prescribed to a field of study

Code	Content
ISI1A_K05	Dostrzega i rozumie konieczność nieustannego doskonalenia swojej wiedzy, umiejętności i kompetencji społecznych.
ISI1A_U01	Potrafi wykorzystać nabytą wiedzę matematyczną do opisu procesów, tworzenia modeli, analizy algorytmów oraz innych działań w obszarze informatyki.
ISI1A_U06	Potrafi algorytmizować wybrane problemy, ocenić ich złożoność obliczeniową, estymować czas wykonania, dobrać właściwe algorytmy do zadanego problemu, stosować metody i techniki Sztucznej Inteligencji.
ISI1A_U07	Potrafi projektować i rozwijać aplikacje z wykorzystaniem poznanych technologii oraz języków programowania. Potrafi doskonalić umiejętności nabyte w trakcie studiów.
ISI1A_W01	Zna i rozumie zagadnienia matematyczne obejmujące analizę matematyczną, algebrę, matematykę dyskretną, logikę, metody probabilistyczne, statystykę i metody numeryczne - przydatne do formułowania i rozwiązywania prostych zadań związanych z informatyką.
ISI1A_W04	Ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie algorytmów, struktur danych i ich złożoności obliczeniowej.
ISI1A_W05	Ma uporządkowaną, podbudowaną teoretycznie wiedzę w zakresie języków formalnych, kompilatorów oraz języków programowania.
ISI1A_W06	Ma uporządkowaną, podbudowaną teoretycznie wiedzę z inżynierii oprogramowania, modelowania oprogramowania, zarządzania projektem informatycznym, wdrażania i komercjalizacja rozwiązań informatycznych.