Constraint Programming

Karta opisu przedmiotu

Informacje podstawowe

Kierunek studiów Informatyka i Systemy Inteligentne **Specjalność** - **Jednostka organizacyjna** Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej **Poziom kształcenia** Studia inżynierskie I stopnia **Forma studiów** Stacjonarne **Profil studiów** Ogólnoakademicki	**Cykl dydaktyczny** 2022/2023 **Kod przedmiotu** EISIS.li50O.6245fa382b339.22 **Języki wykładowe** angielski **Obligatoryjność** Do wyboru **Blok zajęciowy** Przedmioty ogólne **Przedmiot powiązany z badaniami naukowymi** Tak

Koordynator przedmiotu	Antoni Ligęza, Weronika T. Adrian
Prowadzący zajęcia	Antoni Ligęza

Okresy Semestr 5, Semestr 7	**Forma zaliczenia** Zaliczenie **Forma prowadzenia i godziny zajęć** Wykład: 14 Ćwiczenia laboratoryjne: 14	**Liczba punktów ECTS** 3

Cele kształcenia dla przedmiotu

C1	Aby studenci znali podstawy programowania z ograniczeniami i umieli wykorzystać te 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. A 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 EAIiIB 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.

Efekty uczenia się dla przedmiotu

Kod	Efekty w zakresie	Kierunkowe efekty uczenia się	Metody weryfikacji
Wiedzy – Student zna i rozumie:			
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 dobierać paradygmat do rozwiązywania konkretnych problemów decyzyjnych i obliczeniowych w różnych dziedzinach informatyki	ISI1A_W05, ISI1A_W06	Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego, Prezentacja, Zaliczenie laboratorium
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	Wykonanie ćwiczeń laboratoryjnych, Prezentacja, Zaliczenie laboratorium
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	Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego, Prezentacja, Zaliczenie laboratorium
Umiejętności – Student potrafi:			

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	Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego, Zaliczenie laboratorium
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	Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego, Zaliczenie laboratorium
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	Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego, Zaliczenie laboratorium
Kompetencji społecznych – Student jest gotów do:			
K1	potrafi myśleć samodzielnie i konstruktywnie; studiować w celu podnoszenia swoich kwalifikacji i rozwoju osobowego; potrafi komunikować się i przekazywać wiedzę innym	ISI1A_K05	Wykonanie projektu, Prezentacja, Zaliczenie laboratorium

Treści programowe zapewniające uzyskanie efektów uczenia się dla modułu zajęć

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

Nakład pracy studenta

Rodzaje zajęć studenta	**Średnia liczba godzin* przeznaczonych na zrealizowane aktywności**
Wykład	14
Ćwiczenia laboratoryjne	14
Przygotowanie projektu, prezentacji, pracy pisemnej, sprawozdania	30
Samodzielne studiowanie tematyki zajęć	32
Łączny nakład pracy studenta	**Liczba godzin** 90
Liczba godzin kontaktowych	**Liczba godzin** 28

* godzina (lekcyjna) oznacza 45 minut

Treści programowe

Lp.	Treści programowe	Efekty uczenia się dla przedmiotu	Formy prowadzenia zajęć
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	Wykład
2.	Typy i definiowanie ograniczeń; przegląd wybranych predefiniowanych ograniczeń globalnych.	W1, W2, W3, U2	Wykład, Ćwiczenia laboratoryjne
3.	Bazowy algorytm Backtracking Search i jego modyfikacje; inne podejścia.	W1, W2, W3, U1, K1	Wykład
4.	Metody i algorytmy propagacji ograniczeń.	W3, U2, U3	Wykład
5.	Zwiększanie efektywności obliczeń: dekompozycja, relaksacja, sterowanie wnioskowaniem, heurystyki.	W3, U1, U2, U3	Wykład, Ćwiczenia laboratoryjne
6.	Informacje o wybranych narzędziach i przykłady problemów.	W3, U1, U2, U3, K1	Wykład, Ćwiczenia laboratoryjne
7.	Język MiniZinc — modelowanie rzeczywistych problemów	U1, U2, K1	Ćwiczenia laboratoryjne

Informacje rozszerzone

Metody i techniki kształcenia:

Mini wykład, Kształcenie zdalne, Praca grupowa, Grywalizacja, gamifikacja

Rodzaj zajęć	Metody zaliczenia	Warunki zaliczenia przedmiotu
Wykład	Prezentacja, Zaliczenie laboratorium	Zgodnie z regulaminem studiów
Ćwiczenia laboratoryjne	Wykonanie projektu, Wykonanie ćwiczeń laboratoryjnych, Wynik testu zaliczeniowego	Zgodnie z regulaminem studiów

Warunki i sposób zaliczenia poszczególnych form zajęć, w tym zasady zaliczeń poprawkowych, a także warunki dopuszczenia do egzaminu

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.

Sposób obliczania oceny końcowej

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.

Sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności studenta na zajęciach

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

Wymagania wstępne i dodatkowe

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 wytężonej pracy. Otwartość, kreatywność i zdolność do myślenia niestandardowego.

Zasady udziału w poszczególnych zajęciach, ze wskazaniem, czy obecność studenta na zajęciach jest obowiązkowa

Wykład: Studenci uczestniczą w zajęciach poznając kolejne treści nauczania zgodnie z syllabusem 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.

Literatura

Obowiązkowa

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/

Dodatkowa

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

Badania i publikacje

Badania

1. Knowledge Representation and Reasoning, Constraint Programming

Publikacje

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 Sztyber. — 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-1springer-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 Kowalczuk. — 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

Kierunkowe efekty uczenia się

Kod	Treść
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, dobierać 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.