

**AGH**AGH UNIVERSITY OF SCIENCE
AND TECHNOLOGY

Module name:	Logic Programming and Constraint Programming				
Academic year:	2019/2020	Code:	ZSDA-3-0166-s	ECTS credits:	3
Faculty of:	Szkola Doktorska AGH				
Field of study:	Szkola Doktorska AGH	Specialty:	—		
Study level:	Third-cycle studies	Form and type of study:	Full-time studies		
Lecture language:	Polski i Angielski	Profile of education:	Academic (A)	Semester:	0
Course homepage:	https://ai.ia.agh.edu.pl/wiki/en:dydaktyka:krr:start				
Responsible teacher:	prof. dr hab. inż. Ligęza Antoni (ligeza@agh.edu.pl)				

Module summary

Przedmiotem nauczania jest programowanie logiczne (ang. Logic Programming) i programowanie z ograniczeniami (ang. Constraint Programming).

Kurs obejmuje:

- Wprowadzenia do programowania logicznego i programowania z ograniczeniami,
- Zmienne, termy, listy, fakty i klauzule,
- Typy i definiowanie ograniczeń,
- Bazowy algorytm Backtracking Search i jego modyfikacje,
- Metody i algorytmy propagacji ograniczeń,
- Wybrane narzędzia i przykłady problemów.

Description of learning outcomes for module

MLO code	Student after module completion has the knowledge/ knows how to/is able to	Connections with FLO	Method of learning outcomes verification (form of completion)
Social competence: is able to			
M_K001	potrafi myśleć samodzielnie i konstruktywnie; studiować w celu podnoszenia swoich kwalifikacji i rozwoju osobowego; potrafi komunikować się i przekazywać wiedzę innym	SDA3A_K01, SDA3A_K03, SDA3A_K02	Presentation
Skills: he can			
M_U001	potrafi ocenić przydatność różnych paradygmatów programowania i związanych z nimi środowisk programistycznych, w tym zwłaszcza w zakresie programowania deklaratywnego/programowania z ograniczeniami	SDA3A_U02, SDA3A_U01	Test

M_U002	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	SDA3A_U06, SDA3A_U02, SDA3A_U01	Test
M_U003	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 z ograniczeniami	SDA3A_U06, SDA3A_U02, SDA3A_U01	Test
Knowledge: he knows and understands			
M_W001	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	SDA3A_W03, SDA3A_W01	Examination
M_W002	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	SDA3A_W03, SDA3A_W01	Examination
M_W003	ma podstawową wiedzę na temat paradygmatu programowania deklaratywnego w logice; zna podstawy opisu problemów i programowania w logice; zna podstawowe techniki i algorytmy rozwiązywania problemów z ograniczeniami.	SDA3A_W03, SDA3A_W01	Examination

Number of hours for each form of classes

Suma	Form of classes										
	Lectures	Auditorium classes	Laboratory classes	Project classes	Conversation seminar	Seminar classes	Practical classes	Fieldwork classes	Workshops	Prace kontrolne i przejściowe	Lektorat
30	0	0	0	0	0	0	0	0	30	0	0

FLO matrix in relation to forms of classes

MLO code	Student after module completion has the knowledge/ knows how to/is able to	Form of classes

		Lectures	Auditorium classes	Laboratory classes	Project classes	Conversation seminar	Seminar classes	Practical classes	Fieldwork classes	Workshops	Prace kontrolne i przejściowe	Lektorat
Social competence: is able to												
M_K001	potrafi myśleć samodzielnie i konstruktywnie; studiować w celu podnoszenia swoich kwalifikacji i rozwoju osobowego; potrafi komunikować się i przekazywać wiedzę innym	-	-	-	-	-	-	-	-	+	-	-
Skills: he can												
M_U001	potrafi ocenić przydatność różnych paradygmatów programowania i związanych z nimi środowisk programistycznych, w tym zwłaszcza w zakresie programowania deklaratywnego/programowania z ograniczeniami	-	-	-	-	-	-	-	-	+	-	-
M_U002	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	-	-	-	-	-	-	-	-	+	-	-
M_U003	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 z ograniczeniami	-	-	-	-	-	-	-	-	+	-	-
Knowledge: he knows and understands												
M_W001	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	-	-	-	-	-	-	-	-	+	-	-

M_W002	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	-	-	-	-	-	-	-	-	-	+	-	-
M_W003	ma podstawową wiedzę na temat paradygmatu programowania deklaratywnego w logice; zna podstawy opisu problemów i programowania w logice; zna podstawowe techniki i algorytmy rozwiązywania problemów z ograniczeniami.	-	-	-	-	-	-	-	-	-	+	-	-

Student workload (ECTS credits balance)

Student activity form	Student workload
Udział w zajęciach dydaktycznych/praktyka	30 h
Preparation for classes	20 h
przygotowanie projektu, prezentacji, pracy pisemnej, sprawozdania	17 h
Realization of independently performed tasks	20 h
Contact hours	5 h
Summary student workload	92 h
Module ECTS credits	3 ECTS

Additional information

Module content

Workshops

Programowanie logiczne i programowanie z ograniczeniami

Wstęp: wprowadzenie do programowania z ograniczeniami i programowania logicznego z ograniczeniami; elementarny przegląd problematyki i przykładów zastosowań programowania logicznego i z ograniczeniami. Podstawy algebraiczne i logiczne definiowania ograniczeń. Strategia Backtracking Depth-First Search w Prologu i jej modyfikacje.

Wybrane elementy programowania deklaratywnego, modele obliczeń, elementy lingwistyki formalnej, elementarny przegląd paradygmatów programowania. Krótka charakterystyka języków w konkretnych paradygmatach: proceduralnym, obiektowym, programowania w logice, funkcyjnym.

Programowanie w logice na przykładzie języka Prolog:

- Intuicyjne wprowadzenie do Prologu,
- Składnia, semantyka i podstawy logiczne Prologu,
- Przetwarzanie struktur listowych,
- Reprezentacja i przetwarzanie wiedzy w Prologu

- **Metaprogramowanie.**

Wybrane zastosowania i kierunki rozwojowe programowania w logice.

Typy i definiowanie ograniczeń; przegląd wybranych predefiniowanych ograniczeń globalnych.

Bazowy algorytm Backtracking Search w programowaniu z ograniczeniami i jego modyfikacje; inne podejścia.

Metody i algorytmy propagacji ograniczeń.

Zwiększanie efektywności obliczeń: dekompozycja, relaksacja, sterowanie wnioskowaniem, heurystyki.

6. Informacje o wybranych narzędziach i przykłady problemów.

Tematyka zajęć warsztatowych (zajęcia praktyczne przy komputerze)

1. Programowanie w wybranym języku (standardowo SWI Prolog i MiniZinc) (20g)

2. Samodzielne przygotowanie i prezentacja modułu (10g)

Teaching methods and techniques:

Workshops: Treści prezentowane na wykładzie są przekazywane w formie prezentacji multimedialnej w połączeniu z klasycznym wykładem tablicowym wzbogaconymi o pytania i problemy do rozwiązania oraz demonstrację działania prostych programów ilustrujących prezentowane zagadnia.

Pytania, dyskusja, propozycje i prezentacja własnych rozwiązań oraz prezentacje problemowe są pożądane i mile widziane.

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

Ocena końcowa jest wystawiana na podstawie oceny z laboratorium oraz ewentualnie zaliczenia wykładu/wyniku egzaminu.

Wiedza i umiejętności każdego studenta oceniane są w skali 0-100 pkt.

Przy zaliczeniach zajęć i egzaminach oraz wystawianiu oceny końcowej stosuje się aktualny Regulamin Studiów, np. następującą skalę ocen:

a) od 90% bardzo dobry (5.0);

b) od 80% plus dobry (4.5);

c) od 70% dobry (4.0);

d) od 60% plus dostateczny (3.5);

e) od 50% dostateczny (3.0);

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

Workshops:

- Attendance is mandatory: Yes

- Participation rules in classes: 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.

Method of calculating the final grade

Przy zaliczeniach zajęć i egzaminach oraz wystawianiu oceny końcowej stosuje się aktualny Regulamin Studiów, np. następującą skalę ocen:

a) od 90% bardzo dobry (5.0);

b) od 80% plus dobry (4.5);

c) od 70% dobry (4.0);

d) od 60% plus dostateczny (3.5);

e) od 50% dostateczny (3.0);

f) poniżej 50% niedostateczny (2.0).

Sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności studenta na zajęciach:

W trybie konsultacji indywidualnych; jedynie w uzasadnionych i dopuszczanych regulaminem przypadkach, w ograniczonych ramach czasowych.

Prerequisites and additional requirements

Znajomość podstawowych zagadnień z zakresu:

1. Znajomość podstaw informatyki.
2. Podstawowa znajomość logiki i teorii relacji oraz funkcji.
3. Narzędzia: Linux/Unix, system plików, edytor ASCII.

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

Recommended literature and teaching resources

Prolog:

1. GEIST Lab z Prologu, http://ai.ia.agh.edu.pl/wiki/pl:prolog:prolog_lab
2. Jan Wielemaker et. al., SWI-Prolog
3. Universidade de Porto, YAP Prolog
4. I. Bratko, Prolog Programming for Artificial Intelligence, 4th ed, Addison-Wesley, 2011.
5. Dave Stuart Robertson, Quick Prolog
6. Dennis Merritt, Adventure In Prolog, Amzi!, 2010.
7. Dennis Merritt, Building Expert Systems in Prolog, Amzi!, 2010.
8. Ulf Nilsson and Jan Małuszyński, Logic, Programming and Prolog (2ed)., Wiley, 2006
9. Michael Covington et. al., Prolog Programming in Depth., 1995.

Programowanie z ograniczeniami:

F. Rossi, P. van Beek, T. Walsh (eds.): Handbook of Constraint Programming, Elsevier, 2006.

R. Dechter: Constraint Processing, Morgan Kaufmann, 2003.

Krzysztof R. Apt: Principles of Constraint Programming. Cambridge University Press, 2003, 2006.

Krzysztof R. Apt and Mark G. Wallace: Constraint Programming using ECLiPSe. Cambridge University Press, 2007.2003, 2006.

<http://kti.mff.cuni.cz/~bartak/constraints/>

<http://www.anclp.pl/>

Scientific publications of module course instructors related to the topic of the module

Antoni Ligęza: Logical Foundations for Rule-Based Systems. Springer, 2006.

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.

A study of methodological issues in design and development of rule-based systems: proposal of a new approach / Antoni LIGĘZA, Grzegorz J. NALEPA // Data Mining and Knowledge Discovery ; ISSN 1384-5810. — 2011 vol. 1 iss. 2, s. 117-137. — Bibliogr. s. 135-137

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

The HeKatE methodology : hybrid engineering of intelligent systems / Grzegorz J. NALEPA, Antoni LIGEZA // International Journal of Applied Mathematics and Computer Science ; ISSN 1641-876X. — 2010 vol. 20 no. 1: Computational intelligence in modern control systems, s. 35-53. — Bibliogr. s. 51-53

Additional information

Wybrane źródła internetowe:

Implementacja Prologu i dokumentacja: <http://www.swi-prolog.org/>
Materiały do Prologu: <http://ai.ia.agh.edu.pl/wiki/pl:dydaktyka:pp:start>
<http://home.agh.edu.pl/~ligeza/wiki/prolog:course>

https://en.wikipedia.org/wiki/Constraint_programming
<http://ktiml.mff.cuni.cz/~bartak/podminky/>
<http://www.hakank.org/>
<http://www.pathwayslms.com/swipltuts/clpfd/clpfd.html>
<https://www.minizinc.org/>

http://home.agh.edu.pl/~ligeza/wiki/ke:essential_thinking