



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Programowanie ograniczeń

Przedmiot

Kierunek studiów

Sztuczna inteligencja

Studia w zakresie (specjalność)

Poziom studiów

drugiego stopnia

Forma studiów

stacjonarne

Rok/semestr

2/3

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

angielski

Wymagalność

obieralny

Liczba godzin

Wykład

15

Ćwiczenia

Laboratoria

15

Projekty/seminaria

Inne (np. online)

Liczba punktów ECTS

2

Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr inż. Adam Meissner

Odpowiedzialny za przedmiot/wykładowca:

email: Adam.Meissner@put.poznan.pl

tel. 61 665 37 24

Wydział Informatyki i Telekomunikacji

ul. Piotrowo 3 60-965 Poznań

Wymagania wstępne

Student ma podstawowe wiadomości z zakresu logiki obliczeniowej, programowania deklaratywnego, programowania imperatywnego oraz optymalizacji kombinatorycznej. Student posługuje się językiem angielskim w stopniu wystarczającym do porozumiewania się, a także do czytania ze zrozumieniem wskazanej literatury przedmiotowej. Student ma świadomość odpowiedzialności za pracę własną oraz gotowość do podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania.

Cel przedmiotu

Głównym celem przedmiotu jest zapoznanie studenta z podstawami paradygmatu programowania



ograniczeń (CP) oraz wypracowanie umiejętności stosowania tego ujęcia do modelowania i rozwiązywania problemów teoretycznych oraz praktycznych.

Przedmiotowe efekty uczenia się

Wiedza

[K2st_W3] Student ma zaawansowaną wiedzę szczegółową dotyczącą wybranych zagadnień z zakresu sztucznej inteligencji i dziedzin pokrewnych.

[K2st_W4] Student ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach informatyki i sztucznej inteligencji, oraz innych, wybranych, pokrewnych dyscyplin naukowych.

Umiejętności

[K2st_U1] Student potrafi pozyskiwać informacje z literatury, baz danych oraz innych źródeł (w języku polskim i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie.

[K2st_U3] Student potrafi planować i przeprowadzać eksperymenty, w tym pomiary i symulacje komputerowe, interpretować uzyskane wyniki i wyciągać wnioski oraz formułować i weryfikować hipotezy związane ze złożonymi problemami inżynierskimi i prostymi problemami badawczymi.

[K2st_U5] Student potrafi — przy formułowaniu i rozwiązywaniu zadań inżynierskich — integrować wiedzę z różnych obszarów informatyki i sztucznej inteligencji (a w razie potrzeby także wiedzę z innych dyscyplin naukowych) oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne.

[K2st_U16] Student potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia, w tym innych osób.

Kompetencje społeczne

[K2st_K2] Student rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu informatyki i sztucznej inteligencji w rozwiązywaniu problemów badawczych i praktycznych.

[K2st_K4] Student ma świadomość potrzeby rozwijania dorobku zawodowego oraz przestrzegania zasad etyki zawodowej.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

a) w zakresie wykładów

- na podstawie odpowiedzi na pytania dotyczące materiału prezentowanego podczas wykładów;

b) w zakresie laboratoriów

- na podstawie aktywności w zakresie realizacji bieżących ćwiczeń.



Ocena podsumowująca:

a) w zakresie wykładów

- na podstawie pisemnego sprawdzianu obejmującego ok. 5 pytań teoretycznych lub prostych zadań; warunkiem zaliczenia sprawdzianu jest uzyskanie co najmniej 50% możliwych do zdobycia punktów

b) w zakresie laboratoriów

- na podstawie sprawdzenia wiadomości polegającego na rozwiązaniu ok. 3 zadań programistycznych.

Treści programowe

Program wykładu obejmuje następujące zagadnienia. Problem spełniania ograniczeń (CSP) - definicja oraz charakterystyka. Podstawowe zasady formułowania problemów CSP. Ogólne metody rozwiązywania problemów CSP - propagacja i dystrybucja ograniczeń, drzewo poszukiwań i jego przeszukiwanie heurystyczne. Usprawnienia w rozwiązywaniu problemów CSP - eliminacja rozwiązań symetrycznych, wprowadzanie ograniczeń redundantnych. Optymalizacyjne problemy CSP i ich rozwiązywanie za pomocą strategii dystrybucji dwuwymiarowej. Ograniczenia reifikowane. Ograniczenia globalne. Przykładowe problemy CSP - łamigłówki i zagadki logiczne, kolorowanie grafów oraz inne zagadnienia kombinatoryczne. Zastosowanie paradygmatu CP do rozwiązywania praktycznych problemów dużej skali. Zalety i ograniczenia paradygmatu CP. Wskazówki dla programistów. Przegląd języków i środowisk do programowania ograniczeń.

Podczas ćwiczeń laboratoryjnych są rozpatrywane zadania dotyczące zagadnień prezentowanych na wykładzie.

Metody dydaktyczne

Wykład – prezentacje multimedialne uzupełniane przykładami na tablicy.

Laboratorium – zadania programistyczne rozwiązywane przez studentów indywidualnie lub w niewielkich zespołach; wspólna analiza proponowanych rozwiązań.

Literatura

Podstawowa

1. Rossi F., Beek van F., Walsh T. (eds), Handbook of Constraint Programming. 1st Edition, Elsevier, 2006.

Uzupełniająca

1. Apt K.R., Principles of Constraint Programming, Cambridge University Press, 2003.

2. Van Hentenryck P., Michel L., Constraint-Based Local Search, The MIT Press, 2005.

3. Stuckey P.J., Mariott K., Tack G., MiniZinc Handbook, (<https://www.minizinc.org/>).



Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	50	2,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	30	1,0
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych, przygotowanie do sprawdzenia wiadomości) ¹	20	1,0

¹ niepotrzebne skreślić lub dopisać inne czynności