

ÚVOD DO UMĚLÉ INTELIGENCE (CVIČENÍ 3)

Simona Ondrčková

A* - DOMÁCÍ ÚKOL

Vaším úkolem je implementovat monotónní heuristiky pro A* na mřížkách:

- 1) Dvourozměrná mřížka
- 2) Třírozměrná mřížka
- 3) Dvourozměrná mřížka s uhlopříčkou
- 4) Třírozměrná mřížka se stěnovou i prostorovou uhlopříčkou
- 5) Třírozměrná mřížka se stěnovou ale ne prostorovou uhlopříčkou
- 6) Hrany odpovídající pohybům věže po šachovnici.
- 7) Skokan (3 políčka v jedné souřadnici a dvě v druhé) –(velký kůň)
- 8) Král jež se může pohnout o 8 políček v obou souřadnicích.

Zadání: https://gitlab.mff.cuni.cz/finkj1am/introai/-/tree/master/01-a_star_heuristic?ref_type=heads

JAK VYŘEŠIT ALGORITMICKÉ ÚLOHY?

Specializovaných programů:

Nejkratší cesta – Dijkstrův algoritmus

Hledání minimální kostry – Borůvkův algoritmus

Vyhledávání v textu – Aho-Corasick algoritmus

Co kdybychom našli jeden algoritmus na sadu podobných úloh?

Existuje jeden algoritmus pro všechny úlohy?

JAK NAJDEME ALGORITMUS PRO PODOBNOU SADU ÚLOH?

Jakou třídu problému zvolit?

Jak tyto úlohy zapsat/zakódovat?

Jak najít algoritmus jež je vyřeší?

Příklady tříd:

SAT (splnitelnost logických formulí)

Lineární programování

CSP (splňování podmínek)

Plánování

CONSTRAINT SATISFACTION PROBLEM

Konečná množinou proměnných

Každá proměnná má omezený počet možných hodnot (domain)

Množina podmínek na ohodnocení proměnných

CSP řešiče hledají ohodnocení, která splňují všechny podmínky (obvykle prohledáváním stromu ohodnocení).

V pythonu existuje knihovna `python-constraint`, jež řeší CSP.

Řešení je vždy v hloubce stromu n , kde n je počet proměnných (lze použít DFS), počáteční stav prázdné přiřazení, akce přiřazení hodnoty do proměnné, tak, že neporuší podmínku, cílem je úplné konzistentní přiřazení.

Sudoku
#17

1	4	NO	9	4
3	8	7	4	2
4	3		1	8
7	9	6	5	4
2				

Sudoku
#18

7		3		4
6			3	1
2	6			
		9	6	5
	7	2	5	



ZAPIŠTE SUDOKU JAKO CSP PROBLÉM

SUDOKU

Proměnné: $X = \{x_{i,j} \mid 1 \leq i, j \leq 9\}$

Doména: $\{1..9\}$

Pro každou předepsanou hodnotu v na pozici i,j : $x_{i,j} = v$

Různé hodnoty v rámci řádku: $\forall x, y, i, j, m: x_{i,j} \neq y_{i,m}$

Různé hodnoty v rámci sloupce: $\forall x, y, i, j, m: x_{i,j} \neq y_{m,j}$

Různé hodnoty ve čtverci: Nadefinujeme si $B_{1,1}, B_{1,2}, \dots, B_{3,3}$, kde každé B symbolizuje jeden čtverec. Potom platí, že $x_{i,j} \in B_{\frac{i-1}{3}+1, \frac{j-1}{3}+1}$ a vyžadujeme aby: $\forall i, j: all_diff(B_{i,j})$ Jak to zapsat bez $B_{i,j}$?

Ohodnocení proměnných splňující všechny podmínky odpovídá řešení sudoku a naopak.

A puzzle of a 100-dollar bill is laid out on a dark wooden surface. The puzzle pieces are arranged to form the shape of the bill, with some pieces missing and scattered around. The bill features the portrait of Benjamin Franklin, the number '100', and the text 'ONE HUNDRED DOLLARS'. The puzzle pieces are interlocking, and the overall scene is dimly lit, emphasizing the texture of the wood and the details of the bill.

**ZAPIŠTE KRYPTOGRAM SEND MORE
MONEY JAKO CSP**

KRYPTOGRAM

S a M nesmí být 0.

Každé písmeno je jedna cifra.

Písmena se musí navzájem lišit.

Stejná písmena mají stejné číslo.

$$\begin{array}{r} \text{S E N D} \\ + \text{M O R E} \\ \hline \text{M O N E Y} \end{array}$$

KRYPTOGRAM

Proměnné: Písmena P

Doména: $\{0..9\}$

Podmínky: $S \neq 0, M \neq 0 \forall x, y \in P: x \neq y,$

Jako rovnici: $1000 * S + 100 * E + N * 10 + D + M * 1000 + O * 100 + R * 10 + E = 10000 * M + O * 1000 + N * 100 + E * 10 + Y$

Alternativně přes příznak přenosu (carry).

$D+E=10*c0+Y$



ZAPIŠTE PROBLÉM PŘÍPRAVY JEVIŠTĚ JAKO CSP

ROZDĚLENÍ PRÁCE

Máte tým 5 lidí, 60 minut a musíte připravit jeviště pro koncert.

Přinesení hudebních nástrojů musí být před testem zvuku.

	Čas	Počet lidí
Úklid jeviště	20	1
Přinesení hudebních nástrojů	30	1
Test zvuku	10	3
Test světla	10	3

ROZDĚLENÍ PRÁCE

Proměnné jsou práce: U, H, Z, S

Doména: $\{0 \dots 60\}$ čas začátku aktivity

Podmínky: $U + 20 \leq 60, Z + 10 \leq 60, S + 10 \leq 60, (H + 30 \leq 60)$

$$H + 30 < Z$$

Nemáme dost lidí na testování zvuku a světel najednou: $|s - z| > 10$

PLANNING
TIME

**ZAPIŠTE PROBLÉM PLÁNOVÁNÍ SCHŮZEK
JAKO CSP**



PLÁNOVÁNÍ SCHŮZEK

Každé schůzky se musí zúčastnit daná množina lidí.

Všechny schůzky trvají jeden časový blok.

Každý člověk může být na maximálně jedné schůzce najednou.

Chceme najít rozvrh s **nejmenším** počtem bloků.

CSP řeší rozhodovací problémy, jak vyřešíme minimalizační problém?

PLÁNOVÁNÍ SCHŮZEK

Máme CSP co umí vyřešit problém pro daný počet časových bloků (n). Minimální n najdeme opakovaným řešením problémů pro různá n (například pomocí půlení intervalů).

Proměnné: S_i =schůzka i ,

Doména: $\{1..n\}$ (časový blok)

Množiny: L_{S_i} = lidé, kteří se mají účastnit schůzky S_i .

$$\exists l, l \in L_{S_i}, l \in L_{S_j} : S_i \neq S_j$$

2. DOMÁCÍ ÚKOL

Pomocí CSP najděte úplné obarvení grafu (total chromatic index):

Obarvěte hrany i vrcholy pomocí minimálního počtu barev.

Každé dva sousedního vrcholy musí mít jinou barvu.

Každé dvě hrany se společným vrcholem musí mít jinou barvu.

Každá hrana musí mít jinou barvu oba vrcholy s ní společné.

Úplné zadání na: https://gitlab.mff.cuni.cz/finkj1am/introai/-/blob/master/02-csp_total_coloring/task.md

Použije knihovny networkx (pro práci s grafy) a python-constraint (řešič CSP):
<https://pypi.org/project/networkx/> , <https://pypi.org/project/python-constraint/>