Úvod do umělé inteligence (NAIL120)

4. cvičení

Jirka Fink

https://ktiml.mff.cuni.cz/~fink/

Katedra teoretické informatiky a matematické logiky
Matematicko-fyzikální fakulta
Univerzita Karlova v Praze

Letní semestr 2023/24

Poslední změna 11. března 2024

1. domácí úkol: A* heuristiky

Zadání (zkráceno)

Implementujte **monotónní** heuristiky pro **A*** algoritmus spuštěný na **podgrafy** následujících nekonečných mřížek.

- Klasická dvourozměrná mřížka
- Klasická třírozměrná mřížka
- Dvourozměrná mřížka obsahující i úhlopříčky
- Třírozměrná mřížka obsahující stěnové i prostorové úhlopříčky
- Hrany odpovídají právě pohybům věže po šachovnici
- Král v sedmimílových botách, který může až o 8 políček horizontálně i vertikálně
- Třírozměrná mřížka obsahující stěnové úhlopříčky ale nikoliv prostorové
- Skokan se pohybuje o 3 políčka v jedné souřadnici a o 2 políčka v druhé souřadnici

Zadání: `https://gitlab.mff.cuni.cz/finkj1am/introai/-/blob/master/01-a_star_heuristic/task.md`

2. domácí úkol: CSP

Zadání (zkráceno)

Pomocí CSP najděte úplné obarvení grafu (total chromatic index)

- Máme obarvit vrcholy i hrany grafu pomocí minimálního počtu barev
- Každé dva sousední vrcholy musí mít různou barvu
- Každé dvě hrany sdílející společný vrchol musí mít různou barvu
- Incidentní vrchol a hrana musí mít různou barvu

Knihovny pro Python

- networkx: Knihovna pro práci s grafy
  `https://pypi.org/project/networkx/`
- python-constraint: Triviální řešič CSP
  `https://pypi.org/project/python-constraint/`

Přístupy k řešení algoritmických úloh

Hledání specializovaných algoritmů na konkrétní úlohy

- Dijkstrův algoritmus na hledání nejkratší cesty
- Borůvkův algoritmus na hledání minimální kostry
- Aho-Corasic algoritmus na vyhledávání v textu
- Dinicův algoritmus na hledání maximálního toku
- Strassenův algoritmus na násobení matic

Nebylo by lepší mít obecný algoritmus na řešení podobných úloh?

Je možné na každou úlohu najít algoritmus?

Příklad neexistuje algoritmus, který rozhodne, zda daný program zastaví

Obecnější postupy řešení algoritmických úloh

Jak zkonstruovat postup na řešení obecnější třídy úloh?

- Zvolit rozumnou třídu úloh
- Vymyslet zápis úloh této třídy
- Najít obecný algoritmus na řešení těchto úloh

Příklady tříd úloh

- Lineární programování
- Konvexní optimalizace
- Constraint satisfaction programming (splňování podmínek)
- SAT (splnitelnost logických formulí)
- Automatické plánování

Příklady obecných postupů k řešení algoritmických úloh

- Lineární programování
- Konvexní optimalizace
- Constraint satisfaction programming (splňování podmínek)
- SAT (splnitelnost logických formulí)
- Automatické plánování

Splnitelnost booleovské formule (SAT)

Popis úlohy pomocí SAT

- Konečná množina logických (binárním) proměnných, např. $x_1, x_2, \ldots$
- Literál je buď proměnná x_1 nebo negace proměnné $\neg x_1$
- Klauzule je disjunkce literálů, např. $x_1 \vee \neg x_2 \vee x_3$
- Formule v konjunktivní normální formě je konjunkce klauzulí, např. $(x_1 \vee \neg x_2 \vee x_3) \& (x_2 \vee \neg x_3 \vee \neg x_4) \& \neg x_1$
- Formule je splnitelná, pokud existuje ohodnocení všech proměnných logickými hodnotami takové, že celá formule je pravdivá, tj. v každé klauzuli je alespoň jeden literál pravdivý.

Příklad

Rozhodněte, zda jsou následující formule splnitelné

- $(x_1 \vee \neg x_2 \vee x_3) \& (x_1 \vee x_2 \vee \neg x_3) \& \neg x_1$
- $(x_1 \vee x_2) \& (\neg x_1 \vee x_3) \& \neg x_2 \& \neg x_3$

Perfektní párování

Příklad

- Perfektní párování grafu je podmnožina hran P taková, že každý vrchol má právě jednu incidentní hranu v P
- Popište hledání perfektního párování pomocí SAT

Řešení pro obecné grafy

- Proměnné x_e pro všechny hrany e
- Každý vrchol u je incidentní alespoň s jednou hranou párování: $x_{uv_1} \vee \cdots \vee x_{uv_k}$, kde $v_1, \ldots, v_k$ jsou všechny vrcholy sousední s u
- Pro každé dvě hrany e, f sdílející společný vrchol je nejvýše jedna v párování: $\neg x_e \vee \neg x_f$

Popište sudoku jako úlohu SAT

EASY

2		5			9			4
						3		7
7			8	5	6		1	
4	5		7					
		9				1		
					2		8	5
	2		4	1	8			6
6		8						
1			2			7		8

2	1	5	3	7	9	8	6	4
9	8	6	1	2	4	3	5	7
7	3	4	8	5	6	2	1	9
4	5	2	7	8	1	6	9	3
8	6	9	5	4	3	1	7	2
3	7	1	6	9	2	4	8	5
5	2	7	4	1	8	9	3	6
6	4	8	9	3	7	5	2	1
1	9	3	2	6	5	7	4	8

MEDIUM

		6		9		2		
			7		2			
	9		5		8		7	
9				3				6
7	5						1	9
1				4				5
	1		3		9		8	
			2		1			
		9		8		1		

8	7	6	4	9	3	2	5	1
3	4	5	7	1	2	9	6	8
2	9	1	5	6	8	4	7	3
9	8	2	1	3	5	7	4	6
7	5	4	8	2	6	3	1	9
1	6	3	9	4	7	8	2	5
4	1	7	3	5	9	6	8	2
6	3	8	2	7	1	5	9	4
5	2	9	6	8	4	1	3	7

HARD

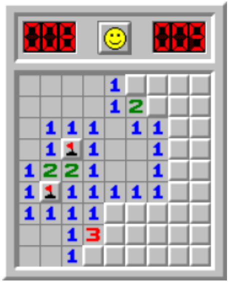
			8					
7	8	9		1				6
					6	1		
		7					5	
5		8	7		9	3		4
	4					2		
		3	2					
8				7		4	3	9
					1			

1	6	5	8	4	7	9	2	3
7	8	9	3	1	2	5	4	6
4	3	2	5	9	6	1	7	8
2	9	7	4	6	3	8	5	1
5	1	8	7	2	9	3	6	4
3	4	6	1	5	8	2	9	7
9	7	3	2	8	4	6	1	5
8	2	1	6	7	5	4	3	9
6	5	4	9	3	1	7	8	2

1. Proměnné x_{ijk} udávají, zda na pozici (i, j) má být číslo k
2. Na každé pozici (i, j) je alespoň jedno číslo: $x_{ij1} \vee \cdots \vee x_{ij9}$
3. Na každé pozici (i, j) nemůže být zároveň $k \neq k'$: $\neg x_{ijk} \vee \neg x_{ijk'}$
4. Pro každé dvě pozice (i, j) a (i', j') mající mít různá čísla a číslo k: $\neg x_{ijk} \vee \neg x_{i'j'k}$
5. Je-li na pozici ij předepsaná hodnota k, tak přidáme klauzuli x_{ijk}

Hledání min

Je možné na hru hledání min (minespeeper) použít SAT nebo CSP řešič?



- Je možné SAT převést na CSP?
  Formálně: Je možné každou instanci SAT převést na instanci CSP?
- Je možné CSP převést na SAT?
  Formálně: Je možné každou instanci CSP převést na instanci SAT?
- V obou případech chceme jednoduše získat instanci, ze které dokážeme jednoduše získat řešení původní instance.

CNF vs. DNF

Pojmy

- Formule v konjunktivní normální formě (CNF) je konjunkce klauzulí
  např. $(x_1 \vee \neg x_2 \vee x_3) \& (x_2 \vee \neg x_3 \vee \neg x_4) \& \neg x_1$
- Formule v disjunktivní normální formě (DNF) je disjunkcí P-termů, kde P-term je konjunkce literálů
  např. $(x_1 \& \neg x_2 \& x_3) \vee (x_2 \& \neg x_3 \& \neg x_4) \vee \neg x_1$

Algoritmy a složitost

- Jak najít splnitelné ohodnocení formulí v CNF a DNF?
- Jaká je algoritmická složitost rozhodovacích problémů, zda formule v CNF a DNF je splnitelná?

Je možné kryptogramy řešit pomocí SAT?

Příklad

Dosaďte za písmena S, E, N, D, M, O, R, Y cifry 0 až 9 tak, aby

- různým písmenům byla přiřazena různá čísla
- S i M byla různá od 0
- platila rovnost

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

Cílem není najít řešení, ale popsat úlohu pomocí SAT.

Jednoznačné řešení

9567 + 1085 = 10652

3. domácí úkol: SAT

Zadání (zkráceno)

Pomocí SAT najděte úplné obarvení grafu (total chromatic index)

- Máme obarvit vrcholy i hrany grafu pomocí minimálního počtu barev
- Každé dva sousední vrcholy musí mít různou barvu
- Každé dvě hrany sdílející společný vrchol musí mít různou barvu
- Incidentní vrchol a hrana musí mít různou barvu

Knihovny pro Python

- networkx: Knihovna pro práci s grafy
  `https://pypi.org/project/networkx/`
- python-sat: Řešič problému SAT
  `https://pypi.org/project/python-sat/`