Úvod do umělé inteligence (NAIL120)

3. cvičení

Jirka Fink

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

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

Letní semestr 2025/26
Poslední změna 9. března 2026

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.

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

Perfektní párování

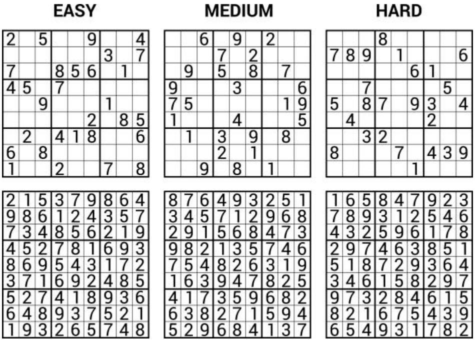
Příklad

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

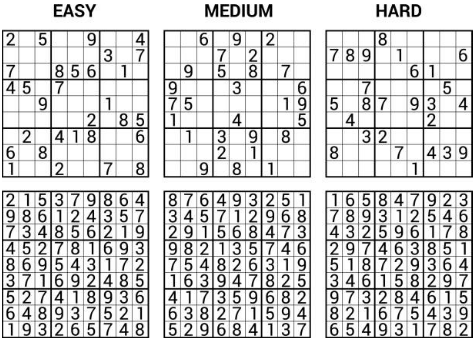
Řešení pro obecné grafy

- 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

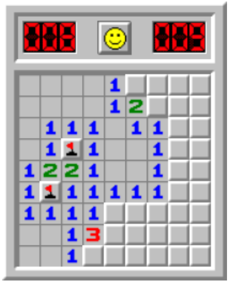


Popište sudoku jako úlohu SAT



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?
- Můžeme pomocí SAT nebo CSP zjistit, zda existuje políčko, na kterém určitě není mina?

SAT vs. CSP

- 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. $(x_1 \& \neg x_2 \& x_3) \vee (x_2 \& \neg x_3 \& \neg x_4) \vee \neg x_1$

CNF vs. DNF

Pojmy

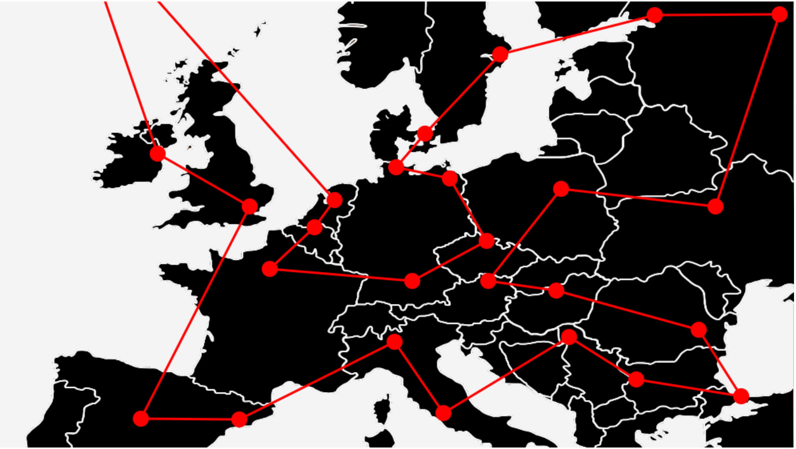
- Formule v konjunktivní normální formě (CNF) je konjunkce klauzulí
  např. $(x_1 \vee \neg x_2 \vee x_3) \& (x_2 \vee \neg x_3 \vee \neg x_4) \& \neg x_1$
- Formule v disjunktivní normální formě (DNF) je disjunkcí P-termů, kde P-term je konjunkce literálů
  např. $(x_1 \& \neg x_2 \& x_3) \vee (x_2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

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?

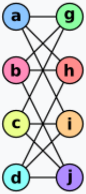
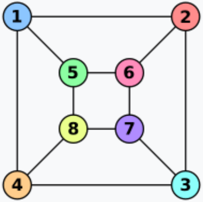
Je možné daný graf rozdělit do $n/3$ skupin po třech vrcholech tak, že každá skupina tvoří trojúhelník.

Zapište úlohu hledání Hamiltonovské kružnice pomocí CSP/SAT



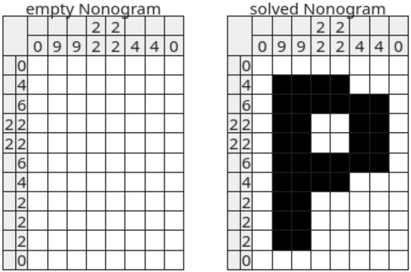
Izomorfismus grafů

- Grafy G a H jsou izomorfní, jestliže existuje bijekce mezi vrcholy $f : V(G) \to V(H)$ taková, že pro každou dvojici vrcholů $u, v \in V(G)$ graf G má hranu $\{u, v\}$ právě tehdy, když graf H má hranu $\{f(u), f(v)\}$.
- Zapište problém izomorfismu grafů pomocí SAT/CSP.

<table>
<tr><th>Graph G</th><th>Graph H</th><th>An isomorphism between G and H</th></tr>
<tr>
<td>
</td>
<td>
</td>
<td>$f(a) = 1$
$f(b) = 6$
$f(c) = 8$
$f(d) = 3$
$f(g) = 5$
$f(h) = 2$
$f(i) = 4$
$f(j) = 7$</td>
</tr>
</table>

Nonogram, Zakódovaný obrázek

- Prázdná dvourozměrná čtvercová mřížka.
- Cílem je obarvit některé čtverečky na černo.
- Okolo mřížky umístěná legenda s čísly.
- Každé číslo v legendě určuje počet za sebou následujících čtverečků, které mají být obarveny na černo.
- Podrobnosti na `https://en.wikipedia.org/wiki/Nonogram`.
- Popište hledání řešení pomocí SAT/CSP.



Další problémy řešitelné pomocí SAT/CSP

- Logik, Mastermind
  `https://en.wikipedia.org/wiki/Mastermind_(board_game)`
- Pandemic
  `https://en.wikipedia.org/wiki/Pandemic_(board_game)`
- Libovolný NP-úplný problém
  `https://en.wikipedia.org/wiki/List_of_NP-complete_problems`

Zadání (zkráceno)

Pomocí SAT najděte řešení problému 3-partition:
Rozdělit $3n$ čísel do trojic tak, aby součet každé skupiny byl stejný.
Pro zjednodušení úlohu jsou čísla po dvou různá.

3. domácí úkol: SAT

Zadání (zkráceno)

Pomocí SAT najděte řešení problému 3-partition:
Rozdělit $3n$ čísel do trojic tak, aby součet každé skupiny byl stejný.
Pro zjednodušení úlohu jsou čísla po dvou různá.

Zdroje

- 3-partition
  `https://en.wikipedia.org/wiki/3-partition_problem`
- python-sat: Řešič problému SAT
  `https://pypi.org/project/python-sat/`