

NAIL062 Projekt: Aplikace SAT solveru

Následující pravidla, podmínky a termíny může Váš cvičící libovolně upravit a doplnit.

Vášim úkolem bude vytvořit systém pro řešení zadaného výpočetního problému za použití SAT solveru. Budeme řešit *rozhodovací problém* (odpověď ano/ne), jde-li o optimalizační problém, řešte jeho rozhodovací variantu: *Má daná instance řešení, jehož objektivní funkce má alespoň/nejvýše danou hodnotu?*

Termíny

- Vyjádření preferencí případně návrh vlastních problémů k řešení: **do 25. října**
- Odevzdání hotového projektu (neurčí-li Váš cvičící jinak): **do konce listopadu**

Etická pravidla

Při řešení musíte postupovat zcela samostatně. Není dovoleno řešení konzultovat s nikým kromě cvičícího ani používat nástroje generativní AI. Můžete vyhledávat doplňující informace o daném problému (např. zajímavé instance nebo upřesnění zadání problému), ale nesmí se týkat jeho zakódování do CNF nebo použití SAT solveru k jeho řešení. Nesmíte také číst žádný kód řešící daný (nebo příbuzný) problém. V případě nejasností jak postupovat se obraťte na cvičícího, který vám poskytne pomoc nebo nápovědu.

Zásady pro vypracování

Pro vývoj projektu používejte GitHub nebo GitLab repozitář, často a pravidelně commitujte, aby bylo možné sledovat postup řešení. K řešení použijte programovací jazyk Python 3 a SAT solver Glucose 4.2. Nepoužívejte žádné knihovny, které umí převádět formule do CNF apod. Naprogramujte skript, který:

- na vstupu přijme instanci problému (buď parametry, nebo cestu k souboru popisujícímu instanci, případně kombinaci obojího),
- zakóduje tuto instanci do SAT ve formátu DIMACS CNF,
- zavolá na CNF formuli SAT solver Glucose,
- dekoduje nalezené ohodnocení proměnných (existuje-li) zpět na řešení daného problému,
- toto řešení vypíše v nějakém rozumném, člověkem čitelném formátu.
- Skript navíc volitelně vypíše zkonstruovanou CNF formuli v DIMACS CNF formátu, a statistiky o průběhu řešení SAT solveru.

K řešení přiložte několik instancí, a to alespoň:

- malou, člověkem analyzovatelnou splnitelnou instanci,
- malou, člověkem analyzovatelnou nesplnitelnou instanci (existuje-li),
- splnitelnou instanci, která poběží netriviálně dlouho (alespoň 10s), pokud se Vám ji nepodaří nalézt, popište Vaše snažení.

Dále musí být součástí řešení soubor README.md se stručnou dokumentací obsahující:

- přesný, matematický popis problému, který řešíte (parametry, rozhodovací proměnné, constraints, aj.),
- popis Vámi zvoleného zakódování do CNF, případně jeho alternativ,
- uživatelská dokumentace Vašeho skriptu (vč. popisu formátu vstupu, a také výstupu není-li zřejmý),
- popis přiložených instancí,
- popis experimentů, které jste prováděli (např. jak velké instance dokáže váš skript vyřešit v rozumném čase?) případně kód k provádění těchto experimentů.

Ukázkové řešení

Pro inspiraci si prohlédněte ukázkové řešení (.zip) problému 15 Puzzle, které laskavě vypracoval Dr. Jirka Švancara (kterému tímto děkuji).

Zadání problému

Každý student v rámci jedné skupiny řeší jiný problém. Můžete si vybírat ze seznamu níže, nebo i navrhnout vlastní problém, který vás zajímá (podléhá schválení, musí být rozumně obtížný, proto raději uveďte preference problémů ze seznamu).

Neurčí-li Váš cvičící jinak, Vaše preference zadejte v SISu v modulu Studijní mezivýsledky jako čísla problémů oddělená čárkami, např. 17,3,21,38. Můžete napsat libovolné množství problémů, dle pořadí preference. Zároveň můžete zaslat emailem cvičícímu návrh vlastního problému, s odkazem na podrobnou definici a vysvětlením, proč vás zajímá.

Odevzdání a prezentace

Hotový projekt odevzdejte v SISu v modulu Studijní mezivýsledky (neurčí-li Váš cvičící jinak) jako link na GitHub nebo GitLab repozitář. Repozitář může být buď veřejný, nebo povolte svému cvičícímu přístup. Z repozitáře musí být zřejmá historie vývoje projektu, není přípustné nahrání pouze výsledné verze skriptu resp. dokumentace.

V rámci odevzdání může být požadována krátká prezentace projektu, ukázka funkčnosti kódu a popis Vašeho řešení a implementačních rozhodnutí, podrobnosti upřesní Váš cvičící.

Seznam problémů

1. Betweenness Problem
2. Bipartite Dimension
3. Cubic Subgraph (problém č. 26)
4. Degree-constrained Spanning Tree
5. Domatic Number
6. Feedback Arc Set
7. Futoshiki
8. Garden of Eden
9. Graph Isomorphism
10. Graph Partitioning (problém č. 25)

11. Hamiltonian Path
12. Hidato
13. Hitori
14. Induced Subgraph Isomorphism
15. Intersection Number
16. Killer Sudoku, zobecnění pro mřížku $n \times n$ (kde n je čtverec)
17. Largest Common Subgraph (problém č. 84)
18. Longest Circuit (problém č. 17)
19. Maximum Density Still Life
20. Multi-agent pathfinding, stačí pro mřížkový graf, makespan objective
21. Nonogram
22. Numberlink
23. Nurikabe, stačí omezení na čísla 1 a 2
24. Peaceably Co-existing Armies of Queens, zobecnění pro šachovnici $n \times n$
25. Rural Postman (problém č. 34)
26. Shortest Common Superstring (problém č. 16)
27. Slitherink
28. Social Golfers Problem
29. Sports Tournament Scheduling
30. Square-Tiling (problém č. 54)
31. Stable Roommates
32. Steiner Triple Systems
33. Steiner Tree, kde váhy hran uvažujte jednotkové
34. Subgraph Isomorphism
35. Tetris
36. Word Design for DNA Computing on Surfaces