
Specifikace & Dokumentace

pro

Vizualizace převoditelnosti grafových NP-Úplných problémů

Studium NP-Úplných problémů může být pro začínajícího studenta velmi obtížné. Cílem tohoto projektu je usnadnit studium pomocí vizualizace samotných problémů, převodů a řešení problémů.

1.0.0

David Kroupa

8. 1. 2024

Tabulka revizí

Jméno	Datum	Důvod změny	Verze
David Kroupa	2. 6. 2023	Počáteční text	0.1.0
David Kroupa	8. 1. 2024	Doplněn detailní popis funkcionality	1.0.0

1. Základní informace

1.1 Popis a zaměření softwarového díla

NP-úplné problémy jsou takové nedeterministicky polynomiální problémy, na které jsou polynomiálně převoditelné všechny ostatní problémy z třídy NP. Třída NP-Ú tedy obsahuje nejtěžší problémy z třídy NP. Na infromatickém oboru MFF UK se student může poprvé setkat s těmito složitými problémy na posledních přednáškách předmětu Algoritmy a datové struktury 2. Krátký úvod do problematiky většině studentů nevysvětlí důležitost a význam těžkých problémů a převoditelnosti.

Popisovaný projekt si klade za cíl vytvoření programu pro vizualizaci převoditelnosti těžkých grafových problémů. Tato ilustrace by měla aspoň trochu vysvětlit význam problémů, řešení a převodů. Jan Amos Komenský propagoval koncept výuky jako "školu hrou", což je jeden z cílů tohoto projektu.

Uživatel si bude moci jednoduše vytvářet neorientované grafy, převádět graf na instanci jiného grafového problému nebo hledat řešení problému na konkrétním grafu.

1.2 Použité technologie

- C# Windows Forms App
- Microsoft Automatic Graph Layout (grafická .NET knihovna pro vykreslování grafů)
- Microsoft Solver Foundation (.NET knihovna pro řešení CNF formulí – SAT Solver)

1.3 Odkazy (reference)

- [https://learn.microsoft.com/en-us/previous-versions/visualstudio/ff525148\(v=vs.93\)](https://learn.microsoft.com/en-us/previous-versions/visualstudio/ff525148(v=vs.93))
- <https://github.com/microsoft/automatic-graph-layout>
- <https://pruvodce.ucw.cz/static/pruvodce.pdf>
- <http://kti.ms.mff.cuni.cz/~kucera/NTIN090/NTIN090-poznamky.pdf>
- https://www.youtube.com/watch?v=u5W32YxmnL8&ab_channel=AlgorithmswithAttitude
- https://www.youtube.com/watch?v=e0nls0SLwjE&ab_channel=WraithofMath
- <https://cs.stackexchange.com/questions/112715/clique-leq-p-sat>
- <https://graphofnpsolverreductions.github.io/>

2. Stručný popis softwarového díla

2.1 Důvod vzniku SW díla a jeho základní části a cíle řešení

Důvodem vzniku je usnadnit pochopení složitosti NP-úplných problémů, jak bylo popsáno v části 1.1. Program bude navržen jako interaktivní formulářová aplikace s možností přidat vrcholy a hrany, čímž uživatel vytvoří neorientovaný graf. Dále bude existovat možnost výběru těžkého grafového problému, který chce uživatel vyřešit anebo převést na jiný problém, který obsahuje řešení právě tehdy, když původní instance problému obsahovala řešení. Všechny problémy z nabídky budou řešeny pomocí SAT solveru. Jakákoliv instance libovolného problému bude nejprve převedena na odpovídající CNF formuli, kterou ohodnotí SAT solver. Toto ohodnocení dále zajistí nalezení řešení původního problému.

V nabídce těžkých grafových problémů bude *Klika*, *Nezávislá množina*, *Vrcholové pokrytí*, *Dominantní množina*, *k-barvení grafu* a *Hamiltonovská kružnice*. Všechny uvedené problémy budou vzájemně převoditelné, ať už přímo anebo tranzitivně přes nějaký jiný grafový problém nebo *3SAT*.

2.2 Hlavní funkce

Program bude umožňovat:

- Intuitivní vytváření a editování grafů v grafovém editoru
- Nalezení řešení různých grafových NP-úplných problémů a jejich následnou vizualizaci v případě, že řešení existuje
- Převod mezi instancemi problémů a následná vizualizace v případě, že řešení existuje

2.3 Motivační příklad užití

- Uživatel vytvoří úplný graf na 4 vrcholech
- Z nabídky převodů mezi instancemi problémů vybere převod z problému *Klika* (úplný podgraf na n vrcholech) na problém *Nezávislá množina* (podmnožina vrcholů, mezi kterými nevede jediná hrana)
- Program instanci převede na výsledný graf
- Uživatel zvolí možnost kopírovat výsledný graf do svého editoru
- Z nabídky problémů vybere problém *Vrcholové pokrytí* (podmnožina vrcholů, každá hrana v grafu je incidentní s alespoň jedním vrcholem v množině) a zadá parametr 2
- Program převede graf na instanci CNF formule, kterou vyřeší
- Editor vykreslí řešení, případně oznámí, že takové řešení neexistuje

2.4 Omezení díla

Aplikace bude odladěna jen pro operační systém Windows 10, nicméně implementace nebude využívat nic, co by neumožňovalo snadnému využití na jiných operačních systémech z rodiny Windows.

3. Struktura a sémantika programu

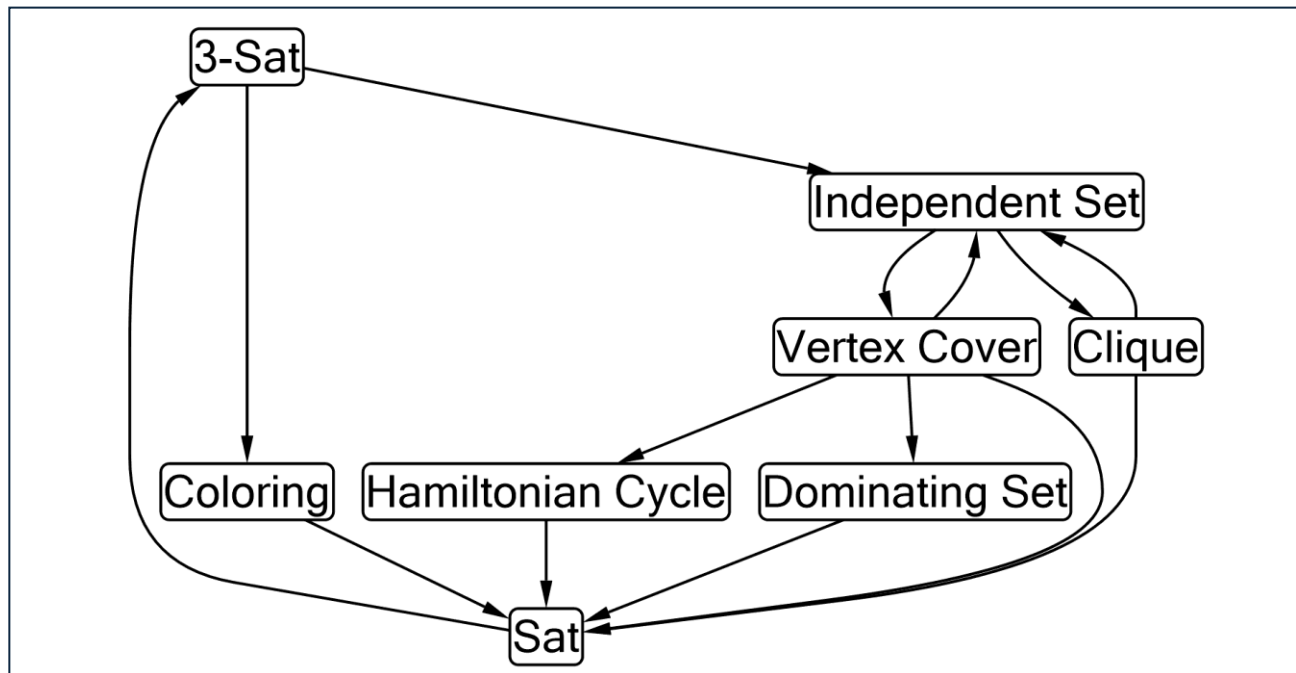
3.1 Hlavní části programu

- **interface IReducible** – Jedná se o klíčovou část programu, která zajišťuje potřebný kontrakt pro všechny třídy, které převádí instance nějakého problému na všechny ostatní problémy v programu, tento interface obsahuje metody:
 1. `Literal[][] ToSat();`
 2. `Clique ToClique();`
 3. `Colorability ToColorability();`
 4. `DominatingSet ToDominatingSet();`
 5. `HamilCycle ToHamilCycle();`
 6. `IndepSet ToIndepSet();`
 7. `VertexCover ToVertexCover();`
- **abstract class GraphProblem : IReducible** – Při případném rozšíření daného programu o další grafové NP-úplné problémy bude stačit vytvořit novou třídu s daným problémem a přidat funkci do interface *IReducible*. Tato třída bude dědit od abstraktní třídy *GraphProblem*, která naplňuje interfacovou tabulku *IReducible*. Všechny interfacové funkce v třídě *GraphProblem* jsou abstraktní. Samotná implementace se nachází až ve třídách, které vymezují konkrétní grafový problém a které právě dědí od této třídy. Tato třída obsahuje fieldy, které jsou společné pro všechny grafové problémy jako např.:
 - `public Graph G {get; protected set;}`
 - `protected int[] solution = null;`
 - `protected Literal[][] sat;`
 - `public int Param { get; protected set;}`
- **internal class _3Sat : IReducible** – Jediná třída v programu, která naplňuje kontrakt interface *IReducible*, ale není potomkem třídy *GraphProblem*. Taky je schopná převést svoji instanci CNF formule na libovolný grafový problém v tomto programu. Jedná se o klíčovou část programu, protože ne všechny grafové mají mezi sebou pěkný převod (např. *Klika* \rightarrow *Nezávislá množina*) a tudíž některé problémy je nejprve třeba převést na instanci CNF, kterou je třeba převést na *3SAT*, který lze převést na *Nezávislou množinou* nebo *k-barvení grafů*.
- **public partial class Form1 : Form** – Zde probíhá veškerá interakce mezi běžícím programem a uživatelem, všechny možné vstupy (všechna možná tlačítka a další řídicí prvky programu viz kapitoly 5 a 6) a výstupy jsou zpracovány uvnitř této třídy. Rovněž je zde kontrola, aby uživatel nepřekročil maximální velikost grafu (20 vrcholů)
- **public static class GraphUtilities** – Tato statická třída obsahuje některé důležité funkce, které využívají při svém řešení a při převodech všechny třídy, které dědí od abstraktní třídy *GraphProblem*. Jedná se o mezivrstvu mezi knihovnou pro kreslení grafů a mezi třídami reprezentující grafové problémy. V této třídě můžeme mj. nalézt funkce:
 1. `public static void ClearVertexColorAndEdgeStyle(Graph g)`
 2. `public static void RemoveNode(ref Graph g, string nodeId)`

3. `public static void AddEdge(Graph g, string source, string target)`
4. `public static Tuple<int, int>[] FindMissingEdges(Graph g)`
5. `public static Graph FlipEdges(Graph g)`
6. `public static int[] AdjacentNodes(Node node, Graph g)`

3.2 Sémantika programu a převodů

Jak již bylo v předchozí části řečeno, tak každý grafový problém, reprezentovaný třídou, dědí od abstraktní třídy *GraphProblem*, která naplňuje interface kontrakt *IReducible*. Každý problém lze převést na CNF formuli, kterou můžeme vyřešit pomocí SAT solveru. Pomocí tohoto řešení jsme schopni zpětně nalézt řešení původního grafového problému. V tomto programu není naimplementován převod pro každou uspořádanou dvojici problémů. Převoditelnost NP-úplných problémů je tranzitivní a této skutečnosti využíváme. Níže můžeme vidět orientovaný graf reprezentující převoditelnost. Graf a obrázek byl vytvořen právě grafickou knihovnou pro tvorbu grafů, kterou využíváme v programu. Některé převody jsou přímočaré, jasné a dokonce oboustranné. Jiné, na druhou stranu, jsou méně intuitivní a uživatel může nějakou dobu čekat, než se převod dokončí. Pokud bychom chtěli převést *Hamiltonovskou kružnici* na *Dominantní množinu*, tak bychom museli provést několik převodů (*HamilCycle* \rightarrow *SAT* \rightarrow *3SAT* \rightarrow *IndepSet* \rightarrow *VertexCover* \rightarrow *DominatingSet*). Taková posloupnost převodů způsobí, že výsledný graf bude mít velmi vysoký počet vrcholů a hran, kvůli čemuž bude hledání řešení velmi zdlouhavé a prostorově náročné.



4. Detailní popis převoditelnosti

4.1 Převod instance grafového problému na SAT

Pro každý grafový problém je naimplementován převod na SAT, neboli na logickou formuli, která je ve tvaru *Konjunktivní normální formě*. SAT formule je reprezentována jako pole polí neboli *Literal*[[[]]. *Literal* je hodnotový datový typ, který je reprezentován číslem *int* *var* a boolovskou hodnotou *bool* *fSense*, která označuje jestli je logický literál kladný nebo záporný. Před převodem grafového problému na SAT vytvoříme pomocnou matici `protected int[,] indexToSatVar;`, která slouží při konstrukci CNF formule. V každém grafovém problému potřebujeme v nějaké podobě iterovat přes vrcholy, hrany (ať už existující nebo neexistující) a parametr, který je součástí problému. Právě při tomto iterování využíváme matici *indexToSatVar*. Další pomocná struktura je `protected int[] satVarToVertex;`, která slouží při nalezeném řešení CNF formule. Jednoduše zaindexujeme kladnými literály do tohoto pole a nalezneme odpovídající vrcholy, které jsou součástí hledaného řešení.

4.2 Převod instance grafového problému na jiný grafový problém

Orientovaný graf znázorňující vzájemnou převoditelnost můžeme nalézt v části **3.2 Sémantika programu a převodů**. Jeden z nejdůležitějších problémů je *Vertex Cover*, protože je to pomyslná křižovatka mezi grafovými NP-úplnými problémy. *Vertex Cover* lze přímo převést na každý jiný problém, kromě *k*-barvení grafů (teoreticky lze každý NP problém převést na každý NP-úplný problém, NP-úplnost znamená, že je problém prvkem množiny NP problémů a NP těžkých problémů, ovšem ne všechny kombinace převodů jsou „hezké“ a ne všechny jsou známe). Při převodu preferujeme co nejpřímější převod a snažíme se vyhnout převodu „přes SAT“, protože SAT se převede na *3SAT*, který v našem programu umíme převést na *k*-barvení grafů anebo *Nezávislou množinu*, a tím se masivně zvětší počet vrcholů a hran. Nejproblematictější problém je jednoznačně *Hamiltonovská kružnice*, protože každý převod z tohoto problému musí být proveden „přes SAT“ a protože se jedná o specifický problém oproti ostatním. 4 z 6 problémů v tomto programu hledají nějakou podmnožinu vrcholů, která splňuje nějaké podmínky. Naopak *Hamiltonovská kružnice* hledané uspořádání vrcholů tak, aby mezi sousedícími vrcholy existovala hrana a zároveň, aby první a poslední vrcholy byly spojené.

5. Obrazovky

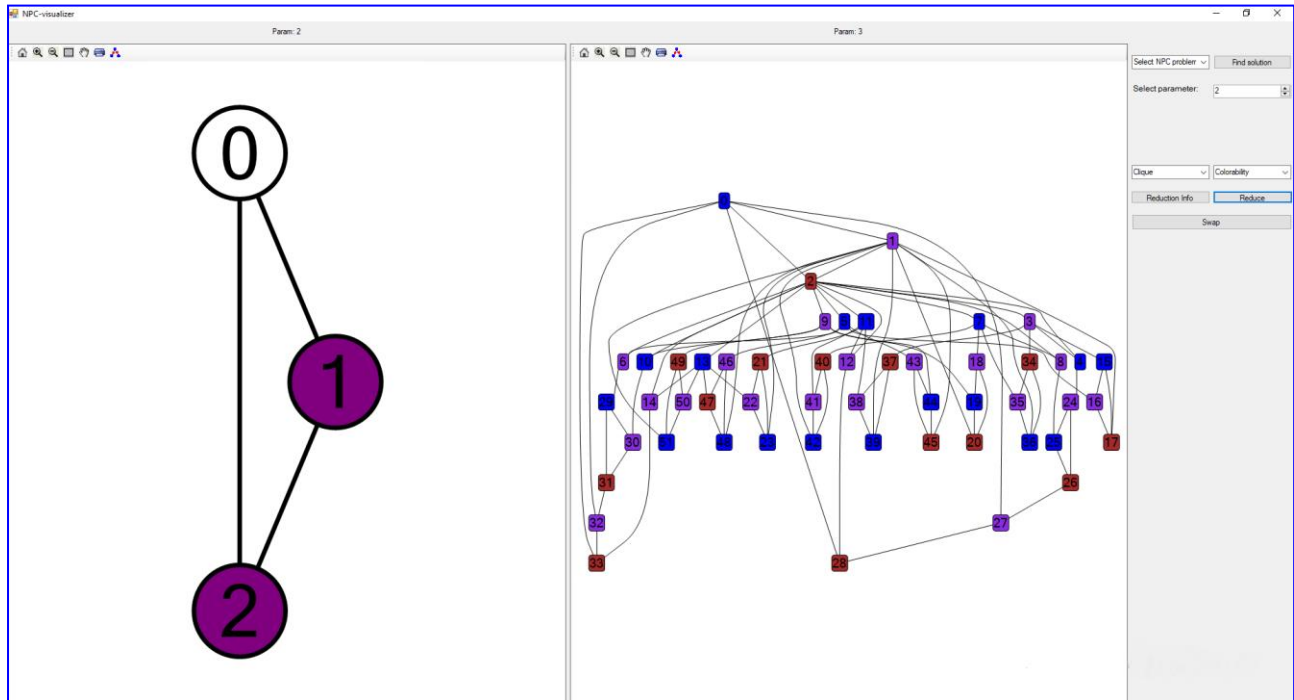
5.1 Editor grafů

Základní struktura okna se skládá z grafového editor a nabídky tlačítek pro převádění a hledání řešení problémů. Při převodu je uživatel informován o původním parametru a parametru výsledného grafu po převodu. Tuto informaci můžeme vidět nad editorem grafu.

Obrazovka ukazuje strukturu aplikace:

- Levý panel – uživatelský editor grafů, zde lze libovolně přidávat hrany a vrcholy, rovněž lze manipulovat s vrcholy pro lepší viditelnost

- Pravý panel – zde se bude objevovat graf po provedení převodu z grafu, který se nachází v levé části, pokud si to uživatel přeje, tak zde uvidíme i případné řešení problému
- Horní lišta – informace o parametru, který je asociován s danou instancí grafu a grafového problému
- Menu – zde si můžeme vybrat problém, který chceme řešit, zadat parametr problému anebo si vybrat konkrétní převod z jednoho problému na jiný, detailnější popis viz níže



5.2 Uživatelské Menu

Nyní se podíváme na tlačítka a jiné řídicí prvky, které jsou dostupné uživateli.

- Select NPC problem comboBox – zde si můžeme vybrat jeden z šesti možných problémů pro samostatné řešení
- numericUpDown – parametr, který je asociován s grafem, který si uživatel vytvoří v editoru, tento parametr se využívá jak při samostatném řešení problémů, tak i při převodu
- Find solution button – program nalezne řešení pro hledaný problém a daný parametr a ilustruje řešení v editoru grafů
- From/To comboBox – uživatel si vybere dvojici problémů, kterou chce převést na vytvořeném grafu a

vybraném parametru

- Reduce button – program převede instanci problému na instanci jiného problému a v případě, že je *Reduce and Solve checkBox* povolen, tak i znázorní obě řešení, pokud existují, při převodu se bere v potaz hodnota parametru (numericUpDown)
- Swap button – pokud má graf v pravém panelu méně jak 21 vrcholu, tak program přemístí graf do grafového editoru
- Reduction Info button – otevře se nové okno, ve kterém bude znázorněn graf převodů v tomto programu

6. Editace grafu

- Přidání vrcholu – zmáčknutí tlačítka *Enter*
- Přidání hrany – nejprve je třeba kliknout levým tlačítkem myši na vrchol, ze kterého chceme, aby vedla hrana, následně pravým tlačítkem myši vybereme druhý vrchol, pokud hrana neexistuje, tak se přidá do grafu, což způsobí překreslení grafu
- Přidání úplného grafu na n vrcholech – pro to slouží tlačítka 1 až 9 na numerické klávesnici
- Odstranění vrcholu/hrany – nejprve je třeba kliknutím levým tlačítkem myši vybrat hranu nebo vrchol, který chceme odstranit, následně zmáčkne tlačítko *Delete*, což vymaže prvek z grafu, což způsobí, že se graf překreslí

7. Poznámky

Tato specifikace je více než inspirována šablonami:

- <https://d3s.mff.cuni.cz/files/teaching/nprg045/SablonaSpecifikace.pdf>