Univerzita Karlova v Praze

Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE

Bc. Jakub Melka

Výpočetní složitost v teorii grafů

Katedra aplikované matematiky

Vedoucí diplomové práce:	prof. RNDr. Jan Kratochvíl, CSc.
Studijní program:	Informatika
Studijní obor:	Diskrétní modely a algoritmy

Praha 2011

Děkuji svému vedoucímu prof. RNDr. J. Kratochvílovi, CSc., za vedení diplomové práce, cenné rady a konzultace.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V Praze dne 24. března 2011 Podpis autora

Název práce: Výpočetní složitost v teorii grafů
Autor: Bc. Jakub Melka
Katedra (ústav): Katedra aplikované matematiky
Vedoucí diplomové práce: prof. RNDr. Jan Kratochvíl, CSc.
e-mail vedoucího: honza@kam.mff.cuni.cz

V předložené práci studujeme problém rekonstrukce grafu ze seznamu uzavřených okolí vrcholů tohoto grafu. Tento problém, původně zformulovaný V. Sósovou, budeme zkoumat z hlediska teorie parametrizované složitosti a zobecníme jej na problém rekonstruovatelnosti vzhledem ke třídám grafů. V této práci dokážeme, že tento problém leží ve třídě složitosti FPT vzhledem k parametru omezené stromové šířky a omezeného maximálního stupně nebo ke 2-degenerovaných grafům s omezeným počtem jistých indukovaných podgrafů, kde parametr je počet těchto podgrafů. Dále dokážeme, že problém rekonstruovatelnosti vzhledem ke třídě grafů s omezeným vrcholovým pokrytím leží ve třídě složitosti XP. Na závěr dokážeme vzájemnou nezávislost získaných výsledků.

Klíčová slova: parametrizovaná složitost, rekonstrukce grafu, stromová šířka, hypergraf, hvězdné systémy

Title: Computational complexity in graph theory
Author: Bc. Jakub Melka
Department: Department of Applied Mathematics
Supervisor: prof. RNDr. Jan Kratochvíl, CSc.
Supervisor's e-mail address: honza@kam.mff.cuni.cz

In the present work we study the problem of reconstructing a graph from its closed neighbourhood list. We will explore this problem, formulated by V. Sós, from the point of view of the fixed parameter complexity. We study the graph reconstruction problem in a more general setting, when the reconstructed graph is required to belong to some special graph class. In the present work we prove that this general problem lies in the complexity class FPT, when parametrized by the treewidth and maximum degree of the reconstructed graph, or by the number of certain special induced subgraphs if the reconstructed graph is 2-degenerate. Also, we prove that the graph reconstruction problem lies in the complexity class XP when parametrized by the vertex cover number. Finally, we prove mutual independence of the results.

Keywords: Fixed parameter complexity, graph reconstruction, treewidth, hypergraph, star systems

Obsah

Kapitola 1

Úvod

Studium výpočetní složitosti algoritmů je pro praxi velmi důležité, neboť umožňuje řešit různé úlohy rychleji a efektivněji. Teorie grafů z hlediska teorie složitosti poskytuje celou řadu otevřených problémů. V této práci se budeme zabývat problémem rekonstruovatelnosti grafů ze seznamů jejich uzavřených okolí, definovaném původně V. Sósovou [8]. Tento problém je v obecnosti NP-úplný, což bylo dokázáno v publikaci [11]. Na zmíněné téma pak bylo napsáno několik dalších publikací, zejména [1], [4] a [6].

Hlavní téma této práce se zabývá otázkou, do jaké třídy složitosti náleží problém rekonstruovatelnosti vzhledem k nějaké třídě grafů $\mathcal{G}$, tj. zda existuje graf G ležící v $\mathcal{G}$, který má shodné seznamy uzavřených okolí vrcholů se zadaným systémem množin. Fedor V. Fomin a spoluautoři v publikaci [6] dokázali, že pro určité třídy grafů je problém rekonstruovatelnosti grafů polynomiálně řešitelný, tedy leží ve třídě P. V této práci rozšíříme výsledky o některé další třídy, nalezneme parametrizovanou třídu grafů $\mathcal{F}_k$ s parametrem k, pro kterou existuje FPT algoritmus s tímto parametrem, dále ukážeme, že pokud vezmeme jako parametr velikost vrcholového pokrytí rekonstruovaného grafu, tak problém rekonstruovatelnosti leží ve třídě XP, a budeme zkoumat, jak problém rekonstruovatelnosti souvisí se stromovým zdvihem grafu.

1.1 Terminologie

Definujeme si některé základní pojmy používané v této práci, aby nemohlo případně dojít k jejich špatné interpretaci, avšak nebudeme popisovat standardní pojmy používané v této práci, neboť je lze najít v základních učebnicích teorie grafů, například [12].

Definice 1. *Nechť $G(V,E)$ je neorientovaný graf a v je libovolný vrchol tohoto grafu. Pak množinu sousedů vrcholu v označme symbolem $N(v)$, tedy $N(v) = \{u|uv \in E\}$. Pokud nebude z kontextu jasné, k jakému grafu se označení vztahuje, označíme dolním indexem, o jaký graf se jedná, v tomto případě $N(v) = N_G(v)$.*

Definice 2. *Nechť $G(V,E)$ je neorientovaný graf a v je libovolný vrchol tohoto grafu. Uzavřené okolí $N[v]$ vrcholu v je množina $N(v) \cup \{v\}$. Analogicky označíme dolním indexem graf G, pokud nebude z kontextu jasné, o který graf se jedná.*

Protože například úplné grafy mají všechna uzavřená okolí vrcholů identická, objekt označovaný jako hypergraf v naší práci bude multihypergraf, tj. hypergraf s násobnými hranami.

V této práci budeme rovněž pracovat s postupným rozšiřováním zobrazení, proto zadefinujeme terminologii pro rozšíření funkce o nový definiční obor a obor hodnot.

Definice 3. *Nechť $\varphi : A \to B$ je libovolná funkce z množiny A do množiny B. Pak pro $U \subseteq A$ označme $\varphi|_U$ funkci definovanou restrikcí definičního oboru funkce φ na U, tj. jedná se o funkci $\xi : U \to \varphi(U)$, kde $\xi(x) = \varphi(x)$ pro $x \in U$.*

Definice 4. *Nechť A, A' a B, B' jsou disjunktní množiny a nechť $\xi : A \to B$, $\zeta : A' \to B'$ jsou dvě libovolné funkce. Pak zápisem*

$$\varphi = \zeta \cup \xi$$

myslíme takovou funkci $\varphi : A \cup A' \to B \cup B'$, pro kterou platí, že

$$\begin{aligned} \xi &= \varphi|_A, \\ \zeta &= \varphi|_{A'}, \end{aligned}$$

navíc pro každé $x \in A'$ označme zápisem $\xi \cup \{(x, \zeta(x))\}$ funkci $\varphi|_{A\cup\{x\}}$.

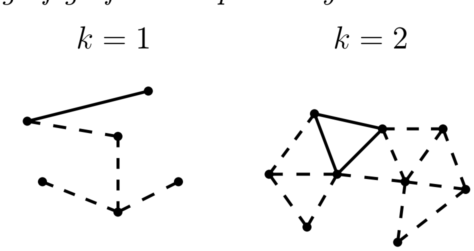
Pokud $G(V, E)$ je graf, budeme značit pro $A \subseteq V$ indukovaný podgraf grafu G na vrcholech z množiny A zápisem $G[A]$. Dále označme zápisem $\omega(G)$ velikost největší kliky v grafu G a $\chi(G)$ barevnost grafu G.

Definice 5. *Nechť $G(V, E)$ je graf. Pak G je chordální, pokud neobsahuje indukovanou kružnici délky větší než tři. Hrana, protínající nějakou kružnici délky alespoň čtyři, se nazývá chorda neboli tětiva.*

1.2 Stromová šířka a další parametry grafů

Stromová šířka, neboli také stromový zdvih, je jeden z důležitých parametrů grafů, neboť umožňuje řešit i některé NP-těžké úlohy v lineárním čase na grafech s omezenou stromovou šířkou, například ty, které jdou zapsat formulí v jazyce MSOL, což je monadická logika druhého řádu. Existuje několik ekvivalentních definic stromového zdvihu, uvedu je podle přednášky doc. RNDr. Daniela Král'e, Ph.D. [10] a z učebního textu doc. RNDr. Petra Hliněného, Ph.D. [9].

Definice 6. *G je k-strom, pokud G lze vytvořit z úplného grafu K_{k+1} přidáváním vrcholů s přesně k sousedy v původním grafu, takových, že tito sousedé tvoří kliku v grafu G. Libovolný podgraf grafu G se pak nazývá částečný k-strom.*



Obrázek 1.1: Příklady k-stromu

Na obrázku 1.1 vidíme příklady takovýchto k-stromů. Z k-stromů získáme první definici stromové šířky.

Definice 7. *Nechť G je graf. Pak stromová šířka grafu G, značeno $tw(G)$, je nejmenší k takové, že G je částečný k-strom.*

Alternativní definici získáme pomocí chordálních grafů.

Věta 1. *[10] Nechť G je graf. Pak stromová šířka grafu G je nejmenší k takové, že existuje chordální graf H s $\omega(H) = k + 1$ takový, že G je jeho podgrafem.*

Existuje ještě několik alternativních definic, nicméně teď si uvedeme tu hlavní.

Definice 8. *Nechť G je graf. Pak stromový rozklad grafu G je strom T a systém množin χ_t, $t \in V(T)$, kde χ_t se nazývají balíčky vrcholů, a tento strom má následující vlastnosti*

(1) pro každé $t \in V(T)$ platí $\chi_t \subseteq V(G)$, $\bigcup\limits_{t \in V(T)} \chi_t = V$,

(2) pro každou hranu $uv \in E(G)$ existuje $t \in V(T)$ takové, že $\{u, v\} \subseteq \chi_t$,

(3) pro každý vrchol $v \in V(G)$ platí, že $T[\{t|v \in \chi_t\}]$ je strom.

Velikost stromového rozkladu pak je $\max\limits_{t \in V(T)} |\chi_t|$.

Věta 2. *[10] Nechť G je graf. Pak stromová šířka grafu G je nejmenší k takové, že existuje stromový rozklad velikosti $k + 1$ grafu G.*

Zjistit stromovou šířku grafu je v obecnosti NP-těžké, avšak Hans L. Boadlender ukázal v publikaci [3], že pro konstantní k existuje lineární algoritmus, který rozpozná, zda graf na vstupu má stromovou šířku nejvýše k, a v takovém případě vydá jeho stromovou dekompozici velikosti nejvýše $k + 1$, v opačném případě dokáže, že graf má stromovou šířku větší než k.

Kapitola 2

Parametrizovaná složitost

Standardní teorie složitosti počítá s jednorozměrným zadáním rozhodovacího problému, tj. složitost se počítá pouze vzhledem k velikosti vstupu dané instance nějakého rozhodovacího problému. U parametrizovaného problému máme ještě jednu součást dané instance rozhodovacího problému, tzv. parametr. Složitost se pak počítá jako dvojrozměrná funkce závisící na velikosti instance rozhodovacího problému a na zadaném parametru. Parametr může nějak charakterizovat danou instanci, například v teorii grafů se může jednat o grafy omezené stromové šířky, nebo naopak o velikost výsledku, kterého chceme dosáhnout, třeba velikost hledaného vrcholového pokrytí.

Na úvod uvedeme některé standardní definice z teorie složitosti, nebudeme ale uvádět úplné základy, například co je to Turingův stroj a jak se měří časová složitost, jak se definují formální jazyky, neboť je lze nalézt v každé základní učebnici teorie složitosti, například [2] nebo [7]. O parametrizované složitosti byla napsána například publikace [5].

Definice 9. *(Třída P) Nechť L je formální jazyk nad konečnou abecedou Σ. Pak $L \in P$ právě tehdy, když existuje deterministický Turingův stroj M pracující nad abecedou Σ takový, že pro každé $x \in L$ se M zastaví a vrátí odpověď 1 a pro každé $x \notin L$ se M opět zastaví a vrátí odpověď 0. Navíc M provede vždy nejvýše $|x|^c$ kroků pro $x \in \Sigma^*$, kde c je kladná konstanta závisející pouze na jazyku L.*

Definice 10. *Nedeterministický Turingův stroj je pětice $(Q, \Sigma, \delta, q, F)$, kde Q je konečná množina stavů, Σ je konečná abeceda obsahující symbol pro prázdné políčko λ, $q \in Q$ je počáteční stav, $F \subseteq Q$ je množina přijímacích stavů a $\delta : Q \times \Sigma \to \mathcal{P}(Q \times \Sigma \times \{R, N, L\})$ je přechodová funkce. Stroj přijme slovo $w \in \Sigma^*$, pokud existuje cesta z počáteční konfigurace Turingova stroje (slovo w na pásce) do nějaké konfigurace se stavem v množině F.*

Definice 11. *(Třída NP) Nechť L je formální jazyk nad konečnou abecedou Σ. Pak $L \in NP$ právě tehdy, když existuje nedeterministický Turingův stroj M pracující nad abecedou Σ takový, že pro každé $x \in L$ se M zastaví a vrátí odpověď 1 a pro každé $x \notin L$ se M opět zastaví a vrátí odpověď 0. Navíc M provede vždy nejvýše $|x|^c$ kroků pro $x \in \Sigma^*$, kde c je kladná konstanta závisející pouze na jazyku L.*

Rozhodovací problém chápeme jako nějaký jazyk L, kde L je množina všech instancí daného problému, na které zní odpověď ano. Rozdíl mezi třídou P a NP je tedy v použití nedeterministického Turingova stroje při řešení dané instance problému. Obecně se soudí, že $P \neq NP$, ale triviálně platí, že $P \subseteq NP$, neboť deterministický Turingův stroj je jen speciální případ nedeterministického Turingova stroje.

Definice 12. *(Polynomiální převoditelnost) Jazyk L je polynomiálně převoditelný na jazyk L' právě tehdy, když existuje polynomiální algoritmus A, že $x \in L \Leftrightarrow A(x) \in L'$, kde výrazem $A(x)$ myslíme výstup tohoto algoritmu na vstup x.*

Definice 13. *Rozhodovací problém L je NP-úplný právě tehdy, když $L \in NP$ a všechny ostatní $L' \in NP$ jsou na L polynomiálně převoditelné.*

Rozhodovací problém L je NP-těžký právě tehdy, když všechny $L' \in NP$ jsou na L polynomiálně převoditelné.

NP-úplné problémy jsou nejtěžší mezi všemi problémy z NP, protože všechny ostatní na ně jdou redukovat. Mezi NP-úplné problémy například patří problém obchodního cestujícího, SAT a další.

2.1 Parametrizované problémy a třída FPT

V této části práce se budeme věnovat základům parametrizované složitosti, k sepsání této části práce, tedy zbytku druhé kapitoly, byly použity zdroje [13] a [14], některé převzaté části jsem místy doplnil pro lepší pochopení. Začneme základní definicí parametrizovaného rozhodovacího problému.

Definice 14. *Nechť Σ je konečná abeceda. Parametrizovaný rozhodovací problém je jazyk $L \subseteq \Sigma^* \times \Sigma^*$, tedy jazyk dvojic slov tvořených písmeny z Σ. Dvojice, která leží v L, se nazývá kladná instance problému, v opačném případě se nazývá záporná instance. Druhý člen dané dvojice se nazývá parametr.*

Definice 15. *Parametrizovaný rozhodovací problém je fixed parameter tractable, pokud jej lze rozhodnout na deterministickém Turingově stroji v čase $f(k)|x|^{O(1)}$, kde $(x, k) \in \Sigma^* \times \Sigma^*$ je instance tohoto problému a $f(k)$ je vyčíslitelná funkce zavisející pouze na k.*

Definice 16. *(Třída FPT) Nechť Σ je konečná abeceda a $L \subseteq \Sigma^* \times \Sigma^*$. Pak L leží ve třídě FPT, právě tehdy, když parametrizovaný rozhodovací problém L je fixed parameter tractable.*

Výhoda problémů ležících ve třídě FPT spočívá v tom, že složitost nezáleží na parametru, tj. jedná se o polynom závisející pouze na velikosti vstupu, nikoliv na parametru. Jiné třídy, například XP, tuto vlastnost nemají. V praxi je to užitečné proto, že se změnou parametru se nemění stupeň polynomu $|x|^{O(1)}$, proto obvykle se změnou parametru hrozí menší růst časové složitosti než u zmíněné třídy XP.

Nyní definujeme převoditelnost mezi parametrizovanými problémy analogicky tak, aby byla teorie konzistentní vzhledem k parametrům. Jako parametr zde budeme uvažovat přirozené číslo.

Definice 17. *(Parametrická redukce) Nechť $L, L' \subseteq \Sigma^* \times \mathbb{N}$ jsou dva parametrizované problémy. Pak L se parametricky redukuje na L' právě tehdy, když existují vyčíslitelné funkce $f(k), g(k) : \mathbb{N} \to \mathbb{N}$ a algoritmus A pracující na deterministickém Turingově stroji, který z instance $(x, k) \in L$ vyrobí $x' \in \Sigma^*$ za následujících podmínek:*

1. *Algoritmus A pracuje v čase $g(k)|(x, k)|^c$ kde c je kladná konstanta.*
2. $(x, k) \in L \Leftrightarrow (x', f(k)) \in L'$

Oproti polynomiální redukci, kde se využívá polynomiálního algoritmu k transformaci instance rozhodovacího problému, zde můžeme použít algoritmus pracující v FPT čase.

2.2 Třídy složitosti W[1], W[t], XP a další

Podobně jako v teorii standardní složitosti existuje polynomiální hierarchie, podobná hierarchie existuje i v teorii parametrizované složitosti. Zadefinujeme si proto nové třídy parametrizované složitosti a uvedeme o nich některé základní výsledky.

Definice 18. *(Vážený SAT) Nechť $\varphi(x_1, x_2, \ldots, x_q)$ je libovolná booleovská formule. Rozhodovací problém váženého SATu má na vstupu formuli φ a otázka zní: Existuje splňující ohodnocení s právě k proměnnými nastavenými na hodnotu 1?*

Problém v definici 18 budeme specializovat na různé druhy formulí, například vážený CNF-SAT, vážený 2-SAT apod.

Definice 19. *(Třída W[1]) W[1] je třída všech problémů, které lze redukovat parametrickou redukcí na vážený 2-CNF-SAT. W[1]-těžký problém je problém, na který lze redukovat parametrickou redukcí vážený 2-CNF-SAT. W[1]-úplný problém je W[1]-těžký problém, který leží ve třídě W[1].*

Definice 20. *Booleovská formule φ je t-normalizovaná, pokud*

$$\varphi = \underbrace{\bigwedge \bigvee \bigwedge \bigvee \cdots}_{t\times} \xi_\mu,$$

kde ξ_μ jsou literály.

Definice 21. *(Vážený obvodový SAT) Nechť $O(x_1, \ldots, x_q)$ je Booleovský obvod libovolné hloubky. Rozhodovací problém váženého obvodového SATu má na vstupu obvod O a otázka zní: Existuje splňující ohodnocení s právě k proměnnými nastavenými na hodnotu 1?*

Nyní přistoupíme k definici dalších tříd, pomocí nichž pak vybudujeme hierarchii.

Definice 22.

1. *W[t] pro $t \geq 1$ je třída všech parametrizovatelných problémů, které mohou být redukovány parametrickou redukcí na vážený t-normalizovaný SAT.*
2. *W[SAT] je třída všech parametrizovatelných problémů, které mohou být redukovány (pomocí parametrické redukce) na vážený SAT.*
3. *W[P] je třída všech parametrizovatelných problémů, které mohou být redukovány (pomocí parametrické redukce) na vážený obvodový SAT.*

Příklad problému, který leží ve třídě W[1] je nezávislá množina a klika. Tyto problémy jsou zároveň W[1]-úplné. Příklad problému, který leží ve třídě W[2] je dominující množina.

Poslední třídou složitosti a také největší z nich, je třída XP. U ní může stupeň polynomu, který počítá složitost vzhledem k velikosti vstupu, záviset na parametru.

Definice 23. *Parametrizovaný problém L leží ve třídě XP právě tehdy, když existují vyčíslitelné funkce $f(k), g(k)$ takové, že je možné jazyk L rozhodovat v čase $f(k)|x|^{g(k)}$.*

Následuje věta o hierarchii.

Věta 3. *(O časové hierarchii)[13]*

$$FPT \subseteq W[1] \subseteq W[2] \subseteq \ldots \subseteq W[SAT] \subseteq W[P] \subseteq XP \tag{2.1}$$

2.3 Datová redukce

Datová redukce je technika, jak z instance nějakého problému v polynomiálním čase vyrobit menší instanci daného problému tak, že obě instance jsou buď kladné nebo záporné. To je dobré, protože to snižuje čas potřebný k vyřešení problému, což je obzvláště patrné při použití exponenciálního algoritmu. Pokud se nám dokonce podaří redukovat instanci problému na velikost nejvýše $g(k)$, kde k je parametr a $g(k)$ je vyčíslitelná funkce, pak mluvíme o tzv. jádru problému.

Definice 24. *(Jádro problému) Nechť L je parametrizovaný problém a (I, k) je jeho instance. Pak redukcí na jádro problému rozumíme transformaci z (I, k) na (I', k'), takovou, že*

1. $k' \leq k$
2. $|I'| \leq g(k)$ *pro nějakou vyčíslitelnou funkci* $g(k)$
3. $(I, k) \in L \Leftrightarrow (I', k') \in L$
4. *danou redukci lze provést v deterministickém polynomiálním čase*

Instanci (I', k') nazýváme jádro problému.

Příklad: Na příkladě vrcholového pokrytí s parametrem velikosti vrcholového pokrytí ukážeme, jak se datová redukce provádí v praxi. Máme tedy instanci problému, graf $G(V, E)$, a chceme vrcholové pokrytí velikosti nejvýše k, což je parametr. Použijeme následující redukční pravidla:

P1. Vrcholy stupně nula smaž z grafu G.

P2. Pokud $v \in V$ má stupeň jedna, tak smaž z grafu G vrchol v $N_G[v]$ a nastav $k = k - 1$.

P3. Pokud $v \in V$ má stupeň větší než k, pak smaž z grafu vrchol v a nastav $k = k - 1$.

Pravidla postupně aplikujeme tak dlouho, dokud můžeme použít alespoň jedno z pravidel. Pokud výsledný graf G' má více než k^2 hran nebo $k^2 + k$ vrcholů, pak graf nemá vrcholové pokrytí velikosti nejvýše k. V opačném případě jsme získali jádro o velikosti $O(k^2)$, které lze vyřešit hrubou silou.

Kapitola 3

Hvězdné systémy

Problém rekonstruovatelnosti v této kapitole přeformulujeme tak, aby byla možná i záporná odpověď, tj. na vstupu nemusí být seznam uzavřených okolí vrcholů. Proto problém rekonstrukce grafu zformulujeme jako rekonstrukci z hypergrafu, jehož hrany jsou případně uzavřená okolí vrcholů, ale nemusí tomu tak být.

Otázka bude zformulována jako rozhodovací problém, zda daný hypergraf je hvězdným systémem nějakého grafu, avšak při jeho řešení dostaneme přímo algoritmy, které dají jako odpověď reprezentovaný graf G, nebo řeknou, že daný hypergraf není hvězdným systémem žádného grafu. Hvězdný systém pro daný graf G je hypergraf se stejnými vrcholy jako vrcholy původního grafu a hrany hypergrafu tvoří všechna uzavřená okolí vrcholů.

3.1 Úvod do rekonstrukce grafů z hvězdných systémů

Definice 25. *Nechť G je neorientovaný graf, pak hvězdný systém grafu $G(V, E)$, značený jako $\mathcal{H}(G)$ je hypergraf $H(V, E')$, kde $E' = \{N[v] | v \in V\}$.*

Obecně je otázka, zda daný hypergraf je hvězdným systémem nějakého grafu, NP-úplná, což bylo dokázáno v publikaci [11]. Pro konkrétní třídy grafů je situace různá, dokonce pro některé třídy grafů je tento problém ve třídě P, dokázáno v publikaci [6]. Pro snadnější terminologii si zavedeme několik alternativních definic pro hvězdné systémy:

Definice 26. *Nechť $H(V, E')$ je hypergraf a $G(V, E)$ neorientovaný graf, pak*

1. *Hypergraf H je rekonstruovatelný do grafu G, pokud H je hvězdným systémem grafu G.*

2. *Graf G je rekonstrukcí hypergrafu H, pokud H je hvězdným systémem grafu G.*

3. *Hypergraf H je rekonstruovatelný, pokud existuje neorientovaný graf G' takový, že $\mathcal{H}(G') = H$.*

Otázka, zda je hypergraf rekonstruovatelný, je tedy NP-úplná. Proto zavedeme obdobné definice pro třídy a rovnou zadefinujeme problém rekonstrukce grafu vzhledem k nějaké třídě grafů. Vzhledem k tomu, že pro konečné třídy grafů lze problém rekonstrukce grafu řešit v konstantním čase (vyrobíme si seznam všech hvězdných systémů), předpokládáme nekonečně mnoho grafů ve třídě.

Definice 27. *Nechť $H(V, E')$ je hypergraf, $\mathcal{F}$ je nekonečná třída grafů, pak hypergraf H je $\mathcal{F}$-rekonstruovatelný, pokud existuje neorientovaný graf G ve třídě $\mathcal{F}$ takový, že $\mathcal{H}(G) = H$.*

Problém rekonstruovatelnosti hypergrafu H se zabývá otázkou, jak přiřadit vrcholy hyperhranám hypergrafu H tak, aby z těchto hran vznikla správně definovaná uzavřená okolí vrcholů a byl tak jednoznačně určen rekonstruovaný graf. Proto si zavedeme ještě alternativní definici rekonstruovatelnosti:

Věta 4. *Nechť $H(V, S)$ je hypergraf. Pak existuje graf $G(V, E)$, pro který platí, že $\mathcal{H}(G) = H$, právě tehdy, když existuje bijekce $\varphi : S \to V$ s následujícími vlastnostmi:*

(1) $\forall o \in S : \varphi(o) \in o$,

(2) $\forall o \in S, \forall x \in o, x \neq \varphi(o) : \varphi(o) \in \varphi^{-1}(x)$,

funkci φ také nazveme rekonstrukcí hypergrafu H.

Důkaz. „$\Rightarrow$“: Mějme graf $G(V, E)$, pro který platí, že $\mathcal{H}(G) = H$, tedy $S = \{N[v] | v \in V\}$ a $|S| = |V|$. Definujme bijekci $\varphi : S \to V$ takto: $\varphi(N[v]) = v$. Zbývá ověřit vlastnosti (1) a (2). Vlastnost (1) je triviální, neboť vrchol v leží ve svém uzavřeném okolí, což vyplývá přímo z definice. Nyní vezměme nějaké $o = N[v], v \in V$ a $x \in o, x \neq v$. Protože $x \neq v \Rightarrow x \in N[v]$, v grafu G je x soused v a navíc $\varphi(o) = v$. Tedy uvažme $\varphi^{-1}(x) = N[x]$. Protože hrana $vx \in E$, $v \in N[x]$, a proto $v = \varphi(o) \in \varphi^{-1}(x)$, čímž je vlastnost (2) dokázána.

„$\Leftarrow$“: Mějme takovou funkci φ, zkonstruujeme graf $G = (V, E)$, pro který bude platit, že $\mathcal{H}(G) = H$ následujícím způsobem: vrcholy budou z definice stejné jako vrcholy hypergrafu. Položme $E = \{uv | \exists o \in S, u = \varphi(o), v \in o \setminus \{\varphi(o)\}\}$. Nejprve dokažme, že pro každou hranu $uv \in E$, platí, že $vu \in E$, a tedy graf můžeme považovat za neorientovaný, neboť vede hrana tam i zpět. Zvolme $uv \in E$, kde $u = \varphi(o)$ pro nějaké $o \in S$, a položme $o' = \varphi^{-1}(v)$. Protože $u \neq v$, z vlastnosti (2) plyne, že $u = \varphi(o) \in \varphi^{-1}(v) = o'$. Protože $v = \varphi(o')$ a $u \in o' \setminus \{v\}$, je splněna podmínka v definici E, a proto $vu \in E$ a G je neorientovaný graf.

Zbývá dokázat, že $\mathcal{H}(G) = H$. Vezměme $v \in V$ a položme $o = \varphi^{-1}(v)$. Chceme dokázat, že $o = N[v]$. Protože graf G je neorientovaný a byl definován tak, že hrany z vrcholu $\varphi(o) = v$ vedou do vrcholů $o \setminus \{v\}$, platí, že $N(v) = o \setminus \{v\}$. Dále z vlastnosti (1) víme, že $v = \varphi(o) \in o$. Proto $N[v] = (o \setminus \{v\}) \cup \{v\} = o$. Protože to platí pro každý vrchol, dostáváme, že $\mathcal{H}(G) = H$. □

Většina algoritmů uvedených zde v této práci bude fungovat tak, že budou určitým způsobem hledat takové funkce φ, které budou rekonstrukcí předloženého hypergrafu, v opačném případě prohlásí, že hypergraf není rekonstruovatelný, pokud žádná taková funkce neexistuje. Z tohoto důvodu si ještě zavedeme pojem částečné rekonstrukce hypergrafu.

Definice 28. *Nechť $H(V,S)$ je hypergraf, $V' \subset V, S' \subset S$ a funkce $\varphi : S' \to V'$ je bijekce a splňuje*

(1) $\forall o \in S' : \varphi(o) \in o,$

(2) $\forall o \in S', \forall x \in o, x \neq \varphi(o), x \in V' : \varphi(o) \in \varphi^{-1}(x)$

Takovou funkci φ nazveme částečnou rekonstrukcí hypergrafu H. V případě, že existuje rekonstrukce hypergrafu ψ a $\varphi = \psi|_{S'}$, pak nazveme funkci φ poctivou rekonstrukcí hypergrafu, v opačném případě nazveme funkci φ falešnou rekonstrukcí hypergrafu. V případě, že $V' = V$, je funkce φ přímo rekonstrukcí hypergrafu H.

Obecný algoritmus bude fungovat tak, že postupným rozšiřováním částečných rekonstrukcí dostane úplnou rekonstrukci hypergrafu, případně dokáže, že daný hypergraf není rekonstruovatelný.

Kontrolu, zda daná funkce splňuje podmínky (1) a (2), lze snadno provést v polynomiálním čase. Pokud reprezentujeme množiny jako například vyvážené binární vyhledávací stromy, pak ověření podmínky (1) trvá $O(|V| \log |V|)$, neboť v každé množině $o \in S$ je nejvýše $|V|$ prvků, tedy při použití binárního vyhledávacího stromu vyhledáme prvek $\varphi(o)$ v množině o v čase $O(\log |V|)$, a množin je právě $|S| = |V|$, jinak ihned zamítneme rekonstruovatelnost grafu. Ověření podmínky (2) je obdobné, v každém stromě postupně procházíme prvky $x \in o, x \neq \varphi(o), x \in V'$, a prvek $\varphi(o)$ vyhledáme v $\varphi^{-1}(x)$ opět v čase $O(\log |V|)$, celkem bude časová složitost $O(|V|^2 \log |V|)$. Funkci φ můžeme reprezentovat například jako dvojité pole ukazatelů na množiny a vrcholy. Pokud místo binárních vyhledávacích stromů použijeme hašovací tabulky, zmizí i faktor $\log |V|$.

Input: (V,V',S,S',φ)
begin
    if $V = V'$ *a* $S = S'$ **then**
        return ANO
    end
    $o \leftarrow$ libovolný prvek z $S \setminus S'$
    foreach $x \in o \setminus V'$ **do**
        $\psi \leftarrow \varphi \cup \{(o,x)\}$
        if *ψ je částečná rekonstrukce a zároveň OBE(V,$V' \cup \{x\}$,S,$S' \cup \{o\}$,ψ)* **then**
            return ANO
        end
    end
    return *NE*
end

Algoritmus 1: Obecný algoritmus OBE(V,V',S,S',φ)

Algoritmus rozhoduje rekonstruovatelnost hypergrafu $H(V,S)$, pokud se mu zadají tyto parametry na vstupu: (V,$\varnothing$,S,$\varnothing$,$\varnothing$).

Věta 5. *Korektnost a složitost algoritmu OBE:*

1. *(korektnost) Nechť $H(V,S)$ je rekonstruovatelný hypergraf. Pak algoritmus 1 spuštěný s parametry (V, $\varnothing$, S, $\varnothing$, $\varnothing$) vrátí odpověď ANO. Pokud $H(V,S)$ není rekonstruovatelný, pak algoritmus vrátí odpověď NE.*

2. *(složitost) Algoritmus pracuje v čase $O(n^n n^2 \log n)$, kde n je počet vrcholů.*

Důkaz. 1. Protože $H(V,S)$ je rekonstruovatelný hypergraf, existuje jeho rekonstrukce $\varphi : S \to V$. Chceme dokázat, že algoritmus vrátí odpověď ANO. Algoritmus OBE reprezentuje backtracking skrz všechny možné částečné rekonstrukce, dokud buď nezkonstruuje úplnou, nebo vrátí odpověď NE. Nechť při rekurentním volání algoritmus OBE vybírá posloupnost prvků $(o_1, o_2, \ldots, o_n)$, $o_i \in S, \forall i \in \{1, \ldots, n\}$. Ukážeme, že algoritmus může zkusit vybrat v rekurentním volání postupně posloupnost prvků

$$(\varphi(o_1), \varphi(o_2), \ldots, \varphi(o_n)),$$

tj. že daná cesta stromem backtrackingu existuje a algoritmus ji případně zkusí. Platí, že $\varphi(o_i) \in o_i \setminus \{\varphi(o_1), \varphi(o_2), \ldots, \varphi(o_{i-1})\}$, protože funkce φ je bijekce a zároveň rekonstrukce hypergrafu, z jejíž vlastnosti (1) dostaneme $\varphi(o_i) \in o_i$.

Zároveň, protože φ je rekonstrukce, je restrikce $\varphi|_{\{o_1, o_2, \ldots, o_i\}}$ částečná rekonstrukce, proto je vyhověno podmínce v cyklu a můžeme se zanořovat hlouběji do rekurze. Proto buď je algoritmem vyzkoušena tato větev rekurze a je vrácena odpověď ANO, nebo případně je vrácena odpověď ANO dříve, pokud se prochází nejdříve jiná úspěšná větev backtrackingového stromu.

Naopak, pokud $H(V,S)$ není rekonstruovatelný, algoritmus odpoví NE, neboť pokud by odpověděl opačně, našel by takovou bijekci φ, která je částečnou rekonstrukcí pro množinu $V' = V$, což by znamenalo. že našel úplnou rekonstrukci, která ale neexistuje, protože $H(V,S)$ není rekonstruovatelný.

2. Protože stupeň listu v backtrackingovém stromě je omezen na $n = |V|$, a hloubka rekurze je nejvýše $|S| = |V| = n$, algoritmus musí projít nejvýše n^n vrcholů v backtrackingovém stromě, to by se stalo v nejhorším případě. Při přechodu do nového vrcholu v backtrackingovém stromu musí algoritmus ověřit, zda daná funkce je částečnou rekonstrukcí, což lze v čase $O(n^2 \log n)$. Zbývající práce v uzlech stromu je konstantní. Celkem tedy algoritmus vykoná práci $O(n^n n^2 \log n)$.

$\square$

Pro třídu k-degenerovaných grafů existuje lepší algoritmus rozhodující rekonstruovatelnost hypergrafů (samozřejmě vzhledem k této třídě), založený na výběru množiny $o \in S \setminus S'$ takovém, že počet možností vyjádřený jako $|o \setminus V'|$ bude omezen konstantou $k+1$, což znamená, že backtrackingový strom bude mít pouze $(k+1)^n$ listů.

Input: (V, V', S, S', φ)

begin
    if $V = V'$ *a* $S = S'$ **then**
        return ANO
    end
    $o \leftarrow \arg\min_{o' \in S \setminus S'} |o' \setminus V'|$
    if $|o \setminus V'| \leq k + 1$ **then**
        foreach $x \in o \setminus V'$ **do**
            $\psi \leftarrow \varphi \cup \{(o, x)\}$
            if *ψ je částečná rekonstrukce a zároveň DEG(V,$V' \cup \{x\}$,S,$S' \cup \{o\}$,ψ)* **then**
                return ANO
            end
        end
    end
    return *NE*
end

Algoritmus 2: Algoritmus DEG pro třídu k-degenerovaných grafů

Věta 6. *Korektnost a složitost algoritmu DEG:*

1. *(korektnost) Nechť $H(V, S)$ je rekonstruovatelný hypergraf vzhledem ke třídě k-degenerovaných grafů. Pak algoritmus 2 spuštěný s parametry (V, $\varnothing$, S, $\varnothing$, $\varnothing$) vrátí odpověď ANO. Pokud $H(V, S)$ není rekonstruovatelný, pak algoritmus vrátí odpověď NE.*

2. *(složitost) Algoritmus pracuje v čase $O((k+1)^n n^2 \log n)$, kde n je počet vrcholů.*

Důkaz. 1. Důkaz toho, že algoritmus je korektní, je analogií důkazu korektnosti algoritmu OBE. Jediné, co musíme dokázat, je, že pokud je hypergraf $H(V, S)$ rekonstruovatelný vzhledem ke třídě k-degenerovaných grafů, pak můžeme vybrat v každém rekurentním volání $o \in S \setminus S'$ takové, že $|o \setminus V'| \leq k + 1$. Nechť φ je rekonstrukce hypergrafu H.

Obdobně jako v předchozím důkazu uvažme posloupnost vybrání množin o_i v rekurzivním volání a indukcí podle i dokážeme, že v každém kroku můžeme vybrat takové o_i, pro které platí, že $|o_i \setminus V'| \leq k+1$, zároveň dokážeme možnost, že algoritmus zkouší větev backtrackingového stromu $\varphi(o_1), \ldots, \varphi(o_{i-1})$.

Pro $i = 1$ tvrzení zjevně platí. Nechť $i > 1$. V tom případě platí, že $V' = \{\varphi(o_1), \ldots, \varphi(o_{i-1})\}$. Nechť tedy algoritmus vybere $o_i \in S \setminus S'$ takové, že $|o_i \setminus V'|$ je nejmenší a hypergraf lze rekonstruovat do grafu $G(V, E)$ reprezentovaného rekonstrukcí φ, kde $o_i = N[v]$ pro nějaké určité $v \in V, \varphi(o_i) = v$. Grafy G i $G' = G[V \setminus V']$ jsou oba k-degenerované z definice degenerovanosti. Graf $G[V \setminus V']$ tedy obsahuje vrchol stupně nejvýše k, pojmenujme ho x a dostaneme nerovnost, že $|N_{G'}(x)| \leq k$, tedy $|N_{G'}(x) \cup \{x\}| \leq k + 1$. Protože $x \notin V'$, dostáváme nějakou $o' = \varphi^{-1}(x) \in S \setminus S'$. V grafu G může mít vrchol x hrany s vrcholy ve V'. Protože platí, že $N_G[x] \setminus V' = N_{G'}[x]$, dostáváme, že $|o' \setminus V'| = |N_G(x) \cup \{x\} \setminus V'| = |N_{G'}(x) \cup \{x\}| \leq k + 1$. Protože o_i byla nejmenší taková, musí platit, že $|o_i \setminus V'| \leq k + 1$, jinak bychom vybrali o'.

To, že v daném kroku můžeme vybrat $\varphi(o_i)$, se dokáže analogicky, jako v předchozím důkazu. Algoritmus tedy buď vyzkouší jinou větev a uspěje, nebo se nakonec dostane k této větvi backtrackingového stromu.

2. Backtrackingový strom má hloubku nejvýše n a v každém uzlu se dělí na nejvýše $k + 1$ větví. Proto má tento strom nejvýše $(k + 1)^n$ uzlů. V každém uzlu udělá práci $O(n^2 \log n)$, proto je celková časová složitost $O((k + 1)^n n^2 \log n)$.

□

Pokud víme, že hypergraf $H(V, S)$ je rekonstruovatelný do grafu $G(V, E)$, existuje triviální způsob, jak zjistit stupeň jednotlivých vrcholů. Uvažme tuto rovnost:

$$\deg_G(x) = |\{o | x \in o, o \in S\}| - 1 \tag{3.1}$$

Tato rovnost platí, protože každý vrchol x je právě v $\deg_G(x) + 1$ množinách - ve svém uzavřeném okolí a v uzavřených okolích vrcholů, kterým je sousedem.

3.2 Současné teoretické výsledky

Na téma hvězdných systémů bylo napsáno několik publikací, zejména [1], [4] a [6]. Uvedeme výsledky zejména z posledního zdroje. Autoři v něm studují speciální případ problému $\mathcal{F}$-rekonstrukce. Problém rekonstrukce se zde uvažuje vzhledem ke třídě tzv. H-free grafů, což je třída grafů, které nemají indukovaný podgraf izomorfní s grafem H. V článku bylo dokázáno, že pro některé grafy je tento problém ve třídě P, zatímco pro jiné je ve třídě NP.

Následující věta je dokázána ve zmíněné publikaci [6] a uvádí některé grafy H, pro něž je problém ve třídě P.

Věta 7. *[6] Nechť $H \in \{P_k, C_k\}$, kde $k \leq 4$. Pak problém rekonstruovatelnosti vzhledem ke H-free grafům leží ve třídě složitosti P.*

P_k a C_k se zde myslí cesta a kružnice velikosti k. Autoři také dokázali částečně charakterizovat třídu grafů, pro které je problém rekonstruovatelnosti NP-úplný, k tomu si uvedeme několik definic, převzatých opět z publikace [6].

Definice 29. *Nechť $G = (V, E)$ je neorientovaný graf. Pak grafem $\overline{G}$ rozumíme graf $G' = (V, E')$, kde $uv \in E' \Leftrightarrow uv \notin E$ pro dva vrcholy $u \neq v$ z V. Graf $\overline{G}$ nazveme doplňkem grafu G.*

Definice 30. *Nechť $G = (V, E)$ je neorientovaný graf. Pak graf $B(G)$ je definovaný jako bipartitní graf, kde partity jsou zdvojené vrcholy z V, a hrana vede z vrcholu a první partity do vrcholu b druhé parity, právě tehdy, když $ab \in E$, analogicky z druhé partity do první.*

Definice 31. *Nechť H je neorientovaný graf. Definujeme funkci $f(H)$ z grafů do kladných čísel a nekonečna. Pokud graf $B(\overline{H})$ je acyklický a neobsahuje komponentu se dvěma vrcholy stupně větších než dvě, pak položíme $f(H) = \infty$. V opačném případě, $f(H)$ definujeme jako menší číslo z délky nejkratší indukované kružnice a z délky nejkratší cesty mezi vrcholy stupně větších než dvě v grafu $B(\overline{H})$.*

Nyní uvedeme hlavní výsledek publikace [6], který částečně charakterizuje, pro které zakázané indukované podgrafy je problém rekonstruovatelnosti NP-úplný.

Věta 8. *[6] Problém rekonstruovatelnosti pro třídu H-free grafů je NP-úplný, pokud $f(H) \neq \infty$. Navíc, pokud $\mathcal{F}$ je množina grafů, pro které existuje takové přirozené číslo n, že pro každé $H \in \mathcal{F}$ platí $f(H) \leq n$, pak problém rekonstruovatelnosti vzhledem ke třídě $\mathcal{F}$-free grafů je NP-úplný.*

Tato věta nám poskytuje jistý ukazatel, že pro některé zakázané indukované podgrafy je problém NP-úplný, jak je v publikaci [6] uvedeno, například pro cesty a kružnice délky větší než pět.

3.3 Stromy a vrcholy malých stupňů

Pokud uvažujeme třídu grafů, ve kterých lze vždy nalézt vrchol nějakého malého stupně, získáme tím poměrně značné zjednodušení problému rekonstruovatelnosti hypergrafů vzhledem k této třídě. Intuitivně, pokud dokážeme vždy najít množinu malého stupně, máme méně možností na výběr, tedy se můžeme méně splést a z charakteru hran hypergrafu dokážeme získat další zajímavé informace. Například pokud uvažujeme stromy, problém rekonstruovatelnosti hypergrafů vzhledem ke stromům leží ve třídě P, i když obecný algoritmus pro 1-degenerované grafy má v tomto případě složitost $O(2^n n^2 \log n)$.

Následující lemma nám říká, jak se vypořádat s množinami velikosti 1, jehož důkaz je značně triviální. Postupně s rostoucí velikostí množiny značně roste komplexnost důkazu, což je pochopitelné, neboť obecný problém rekonstruovatelnosti je NP-úplný.

Lemma 1. *Nechť $H(V, S)$ je rekonstruovatelný hypergraf, $V' \subseteq V, S' \subseteq S$ a $\varphi : S' \to V'$ je poctivá částečná rekonstrukce vzhledem k množině S', jinak řečeno, existuje rekonstrukce ξ, pro kterou platí funkcionální rovnice $\varphi = \xi|_{S'}$. Dále nechť $o \in S \setminus S'$, $o \setminus V' = \{v\}$. Pak také rozšíření φ, definované jako*

$$\varphi' = \varphi \cup \{(o, v)\}$$

je opět poctivá částečná rekonstrukce.

Důkaz. Je jasné, že φ' je bijekce, neboť φ je bijekce. Dokážeme, že $\xi(o) = v$, pak $\varphi' = \xi|_{S' \cup \{o\}}$ je poctivá částečná rekonstrukce. Z vlastnosti rekonstrukce platí, že $\xi(o) \in o$. Zároveň ξ je bijekce, proto $\xi(o) \notin \{\xi(o') | o' \in S'\} = V'$. Protože $\xi(o) \notin V'$ a zároveň $\xi(o) \in o$, nutně musí platit, že $\xi(o) \in o \setminus V'$. Protože $o \setminus V'$ je jednoprvková množina, nezbývá jinak, než $\xi(o) = v$. $\square$

Toto lemma nám říká, jak se zachovat k jednoprvkovým množinám při nalezení částečných rekonstrukcí. Pro představu, pokud je funkce φ rekonstrukce nějakého hypergrafu $H(V,S)$, kde φ rekonstruuje hypergraf H do grafu $G(V,E)$ a φ' je jeho částečná rekonstrukce vzhledem k S', V', pokud se v $S \setminus S'$ vyskytne takové o, že $o \setminus V' = \{v\}$, pak v grafu $G[V \setminus V']$ je izolovaný vrchol v, který jsme naším lemmatem detekovali a korektně jsme rozšířili částečnou rekonstrukci o jeden prvek. Následující lemma nám řekne, jak se zachovat k dvouprvkovým množinám.

Lemma 2. *Nechť $H(V,S)$ je rekonstruovatelný hypergraf, $V' \subseteq V, S' \subseteq S$ a $\varphi : S' \to V'$ je poctivá částečná rekonstrukce vzhledem k množině S', jinak řečeno, existuje rekonstrukce ψ, pro kterou platí funkcionální rovnice $\varphi = \psi|_{S'}$. Dále nechť $o \in S \setminus S'$, $o \setminus V' = \{u, v\}$. Pak lze v čase $O(n^2 \log n)$ vybrat $x \in o \setminus V'$ tak, že*

$$\varphi' = \varphi \cup \{(o, x)\}$$

je opět poctivá částečná rekonstrukce.

Důkaz. Uvažme graf $G' = G[V \setminus V']$, kde G je graf vzniklý z rekonstrukce ψ. Uvažme vrcholy u, v v tomto grafu. Protože $|o \setminus V'| = 2$, alespoň jeden z těchto vrcholů má stupeň jedna, tento vrchol má hranu pouze s druhým vrcholem. Najdeme takové $o' \in S \setminus (S' \cup \{o\})$, že $\{u, v\} \subseteq o'$. Takové o' existuje, neboť uzavřené okolí druhého vrcholu obsahuje $\{u, v\}$, což je zajištěno z existence hrany $uv \in E(G')$. Zároveň je o' určeno jednoznačně, neboť vrchol se stupněm jedna se vyskytuje pouze ve dvou uzavřených okolích - ve svém uzavřeném okolí a v uzavřeném okolí vrcholu, se kterým je spojen hranou.

Pokud $\deg_{G'}(u) = 1, \deg_{G'}(v) > 1$, pak nutně musí platit $\psi(o) = u$, protože $|\psi^{-1}(u) \setminus V'| = \deg_{G'}(u) + 1 = 2$, což by neplatilo, kdyby $\psi(o) = v$. Rozšíříme φ o $\{(o, u)\}$. Případ $\deg_{G'}(u) > 1, \deg_{G'}(v) = 1$ je symetrický.

Nechť tedy $\deg_{G'}(u) = 1, \deg_{G'}(v) = 1$ a dvojice vrcholů $\{u, v\}$ tvoří v grafu G' komponentu, konkrétně kliku K_2. Bez újmy na obecnosti předpokládejme, že $\psi(o) = u$ a $\psi(o') = v$. Pokud rozšíříme φ o $\{(o, u), (o', v)\}$, dostaneme novou částečnou rekonstrukci definovanou jako $\psi|_{S' \cup \{o, o'\}}$. Pokud φ rozšíříme o $\{(o, v), (o', u)\}$, může nastat chyba v případě, že v grafu G vedou různé hrany od vrcholů u, v, tj. $N_G[u] \neq N_G[v]$. V případě, že $o = o'$, můžeme tyto množiny zaměnit, a rozšíření φ bude pořád splňovat podmínky (1) a (2) pro částečné rekonstrukce. Pokud existuje takové $x \in V'$, že $xu \in E(G), xv \notin E(G)$, při záměně dojde k porušení podmínky (2) a selže test částečné rekonstrukce se zadáním $\varphi \cup \{(o, v), (o', u)\}$, tedy korektně určíme, že $\psi(o) = u$ a rozšíříme φ o $\{(o, u)\}$.

Předpokládejme tedy, že $o = o'$, nalezneme novou rekonstrukci. Definujme $\psi' : S \to V$ tak, že $\psi' = \psi$ na $S \setminus \{o, o'\}$ a $\psi'(o) = v, \psi'(o') = u$ a chceme dokázat, že ψ' je rekonstrukcí, rovnou vidíme, že ψ' je bijekce a splňuje vlastnost (1). Nyní musíme dokázat, že v případě $o = o'$ je splněna podmínka (2) pro funkci ψ'. Platí $\forall p \in S, \forall x \in p, x \neq \psi(p) : \psi(p) \in \psi^{-1}(x)$, protože ψ je rekonstrukce. Nechť $p \in S \setminus \{o, o'\}$ takové, že $u \in p$. Pak z rovností $\psi|_{S \setminus \{o,o'\}} = \psi'|_{S \setminus \{o,o'\}}$ a $\psi^{-1}(u) = \psi^{-1}(v)$ dostáváme, že

$$\forall p \in S \setminus \{o, o'\}, \forall x \in p, x \neq \psi'(p) : \psi'(p) \in \psi'^{-1}(x).$$

Probereme také případ, že $p = o$. Tedy $\forall x \in o, x \neq \psi(o) : \psi(o) \in \psi^{-1}(x)$, neboli

$$\forall x \in o, x \neq u : u \in \psi^{-1}(x)$$

a chceme ukázat, že $\forall x \in o, x \neq v : v \in \psi'^{-1}(x)$. Protože ale $N_G[u] = N_G[v]$, musí $N_G(x)$ buď obsahovat $\{u, v\}$, nebo $N_G(x) \cap \{u, v\} = \varnothing$. V případě, že $x \in \{u, v\}$ tak v triviálně leží v o i o', tedy $v \in \psi'^{-1}(x)$. Pokud $x \notin \{u, v\}$, tak $N_G(x) \cap \{u, v\} \neq \varnothing$, a proto $\{u, v\} \subseteq N_G(x)$. Z toho opět $v \in \psi'^{-1}(x)$ a tedy ψ' je rekonstrukce.

Nyní, pokud $\varphi' = \varphi \cup \{(o, u), (o', v)\}$, tak $\varphi' = \psi|_{S' \cup \{o, o'\}}$, naopak když $\varphi' = \varphi \cup \{(o, v), (o', u)\}$, tak $\varphi' = \psi'|_{S' \cup \{o, o'\}}$, v obou případech se jedná o částečnou rekonstrukci a lemma je tím dokázáno z hlediska korektnosti.

Co se týče časové složitosti, kontrola stupňů má složitost $O(n)$, najít o' trvá $O(n \log n)$ a zkontrolovat částečné rekonstrukce trvá $O(n^2 \log n)$. □

Nyní máme dostatečné prostředky, abychom zkonstruovali polynomiální algoritmus, řešící problém rekonstruovatelnosti hypergrafů vzhledem ke třídě lesů, tj. 1-degenerovaných grafů.

Input: Hypergraf $H(V, S)$

$\varphi \leftarrow \varnothing, V' \leftarrow \varnothing, S' \leftarrow \varnothing$
while $S \setminus S' \neq \varnothing$ **do**
  $o \leftarrow \arg\min_{o' \in S \setminus S'} |o' \setminus V'|$
  switch $o \setminus V'$ **do**
    case $\{v\}$
      $V' \leftarrow V' \cup \{v\}, S' \leftarrow S' \cup \{o\}, \varphi \leftarrow \varphi \cup \{(o, v)\}$
    endsw
    case $\{u, v\}$
      Použij Lemma 2 a zjisti, pro které $x \in \{u, v\}$ je $\varphi \cup \{(o, x)\}$ částečná rekonstrukce
      $V' \leftarrow V' \cup \{x\}, S' \leftarrow S' \cup \{o\}, \varphi \leftarrow \varphi \cup \{(o, x)\}$
    endsw
    otherwise
      return *NE*
    endsw
  endsw
end
if φ *je rekonstrukce* **then**
  return *ANO*
else
  return *NE*
end

Algoritmus 3: Algoritmus TREE pro 1-degenerované grafy

Věta 9. *Korektnost a složitost algoritmu TREE:*

1. *(korektnost) Nechť $H(V,S)$ je rekonstruovatelný hypergraf vzhledem ke třídě 1-degenerovaných grafů. Pak algoritmus TREE vrátí odpověď ANO. Pokud $H(V,S)$ není rekonstruovatelný vzhledem k této třídě, pak algoritmus vrátí odpověď NE.*

2. *(složitost) Algoritmus pracuje v čase $O(n^3 \log n)$, kde n je počet vrcholů.*

Důkaz. Nechť hypergraf $H(V,S)$ je rekonstruovatelný do stromu T, jehož rekonstrukce je funkce ψ. Pak pro libovolné $V' \subseteq V$ platí, že $T[V \setminus V']$ je strom nebo les, a proto obsahuje vrchol stupně nejvýše jedna. Když $\varphi : S' \to V'$ je částečná rekonstrukce, existuje $o \in S \setminus S'$ takové, že $|o \setminus V'| \leq 2$. Indukcí podle počtu kroků cyklu dokážeme, že v každé iteraci algoritmus zkonstruuje korektně větší poctivou částečnou rekonstrukci. Na začátku je $\varphi = \emptyset = \psi|_\emptyset$ určitě poctivá částečná rekonstrukce. Předpokládejme, že v nějakém i-tém kroku cyklu máme poctivou částečnou rekonstrukci a chceme dokázat, že ji máme i v $i+1$ kroku. Protože $o \setminus V' \neq \emptyset$ a $|o \setminus V'| \leq 2$, máme buď $o \setminus V' = \{v\}$, nebo $o \setminus V' = \{u, v\}$ pro nějaké $u, v \in V \setminus V'$. Použitím lemmat 1 a 2 dostaneme novou větší poctivou částečnou rekonstrukci, algoritmus používá tato lemmata jako podprogramy. Pokud $S \setminus S' = \emptyset$, pak jsme dostali poctivou částečnou rekonstrukci, která je úplnou rekonstrukcí a algoritmus vrátí odpověď ANO.

Pokud hypergraf $H(V,S)$ rekonstruovatelný není, algoritmus vrátí odpověď NE, protože na konci ověřuje, zda je funkce φ rekonstrukcí tohoto hypergrafu. Konečnost je zajištěna z toho důvodu, že v každé iteraci cyklu se buď zvětší velikost S' o jedna (a S je konečná množina), nebo algoritmus vrátí odpověď NE, a to když velikost množiny $o \setminus V'$ nebude souhlasit ať už z toho důvodu, že je prázdná, nebo že má více prvků než dva.

Co se týče složitosti, výběr o trvá $O(\log n)$, pokud si držíme velikosti $o \setminus V'$ v haldě, kterou pak upravujeme. Počet iterací while cyklu je nejvýše n. Zjistit, zda je množina velikosti jedna, trvá konstantní čas, případná aplikace lemma 2 trvá čas $O(n^2 \log n)$. Ověřit na konci vlastnosti rekonstrukce trvá opět $O(n^2 \log n)$. Celkem získáme čas $O(n^3 \log n)$. □

3.4 Věta o záměně

Následující věta jistým způsobem charakterizuje vnitřní strukturu, případně počet rekonstrukcí daného hypergrafu $H(V,S)$. Pokud se vyskytuje v rekonstrukci $G(V,E)$ velká klika, která je jistým způsobem napojena na zbytek grafu, pak existuje značný počet různých rekonstrukcí hypergrafu $H(V,S)$, které jsou s grafem $G(V,E)$ izomorfní.

Věta 10. *(O záměně) Nechť $H(V,S)$ je rekonstruovatelný hypergraf a φ je jeho rekonstrukce do grafu $G(V,E)$. Nechť $A \subseteq V$ je množina vrcholů taková, že $G[A]$ je úplný podgraf grafu G takový, že pro každé dva vrcholy $u, v \in A$ platí, že $N_G[u] = N_G[v]$. Dále nechť $f : A \to A$ je bijekce.*

Pak $\varphi|_{V\setminus A} \cup f \circ \varphi|_A$ je rekonstrukce hypergrafu $H(V,S)$ taková, že graf definovaný touto rekonstrukcí je s G izomorfní.

Důkaz. Označme $\psi = \varphi|_{V\setminus A} \cup f \circ \varphi|_A$. Zobrazení ψ je určitě bijekce, která splňuje vlastnost (1) rekonstrukcí, protože všechny vrcholy z A mají stejná uzavřená okolí, tedy $f(v) \in N_G(u) \cup \{u\}$ pro $u, v \in A$. Pro spor předpokládejme, že ψ nesplňuje vlastnost (2), tedy že

$$\exists o \in S \exists x \in o, x \neq \psi(o) : \psi(o) \notin \psi^{-1}(x)$$

Rozborem případů dokážeme spor.

a) Nechť $\psi(o) \in A$, tedy o je okolí nějakého vrcholu $v \in A$. Proto $\psi(o) = f \circ \varphi(o)$. Jestliže $x \in A$, pak díky tomu, že $G[A]$ je úplný graf a $\psi^{-1}(x)$ je uzavřené okolí nějakého vrcholu $f^{-1}(x) = x' \in A$ v grafu G, pak platí, že $\psi(o) \in A \subseteq N_G[x'] \subseteq \varphi^{-1}(x') = \psi^{-1}(x)$, což je spor s tím, že $\psi(o) \notin \psi^{-1}(x)$.

   Jestliže $x \notin A$, pak $\psi^{-1}(x) = \varphi^{-1}(x)$. Kdyby $\psi(o) \notin \varphi^{-1}(x)$, znamenalo by to, že $f \circ \varphi(o) \notin \varphi^{-1}(x)$. Protože $N_G[\psi(o)] = N_G[f(\varphi(o))] = N_G[\varphi(o)]$, pak ani $\varphi(o) \notin \varphi^{-1}(x)$, což je spor s tím, že φ je rekonstrukce.

b) $\psi(o) \notin A$. Pak $\psi(o) = \varphi(o)$. Jestliže $x \notin A$, pak $\psi^{-1}(x) = \varphi^{-1}(x)$ a φ nemá vlastnost (2), což je spor.

   Kdyby $x \in A$, pak $\psi^{-1}(x) = \varphi^{-1}(f^{-1}(x)) = \varphi^{-1}(x)$, protože $G[A]$ je úplný graf a $N_G[x] = N_G[f(x)]$. Pak ale $\varphi(o) \notin \varphi^{-1}(x)$, což je spor.

Protože se jedná o permutaci vrcholů kliky a tyto vrcholy mají stejné sousedy, graf G' reprezentovaný rekonstrukcí ψ je s původním grafem izomorfní. $\square$

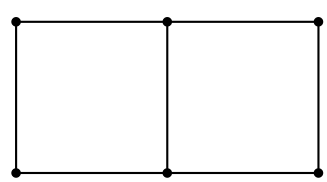
Kapitola 4

Třída 2-degenerovaných grafů

V této kapitole se budeme zabývat třídou 2-degenerovaných grafů. Naším záměrem bylo dokázat, že problém rekonstruovatelnosti vzhledem ke třídě 2-degenerovaných grafů lze řešit v polynomiálním čase. Tato otázka je však příliš obtížná, a proto jsme nebyli schopni ji zcela vyřešit. Identifikovali jsme graf J, který je na obrázku 4.1, pro nějž je problém rekonstruovatelnosti vzhledem ke třídě grafů bez tohoto indukovaného podgrafu NP-úplný. Pokud se omezíme na třídu 2-degenerovaných grafů bez tohoto indukovaného podgrafu, tak tento problém vzhledem k této nové třídě leží v P. Zároveň dokážeme, že pokud zvolíme parametrizovanou třídu grafů $\mathcal{F}_k$ jako třídu 2-degenerovaných grafů s nejvýše k indukovanými podgrafy izomorfními s J, pak problém rekonstruovatelnosti grafů vzhledem k $\mathcal{F}_k$ leží ve třídě FPT s parametrem k.

4.1 Problémový graf

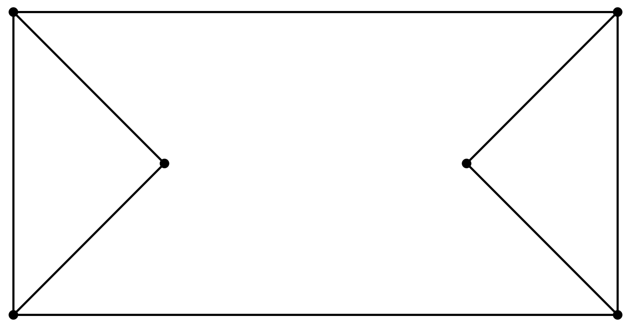
Nejprve ukážeme, že pro graf J je problém rekonstruovatelnosti vzhledem ke třídě J-free grafů NP-úplný, a pak budeme zkoumat 2-degenerované grafy bez indukovaného podgrafu izomorfního s grafem J.



Obrázek 4.1: „Problémový“ graf J

Věta 11. *Rekonstruovatelnost grafů vzhledem ke třídě J-free grafů, kde J je graf na obrázku 4.1, je NP-úplný problém.*

Důkaz. V důkazu využijeme Větu 8 dokázanou v článku [6], tím, že dokážeme, že $f(J) \neq \infty$. K tomu zkonstruujeme graf $\overline{J}$, který je na obrázku 4.2.



Obrázek 4.2: Graf $\overline{J}$

Jak je z obrázku patrné, graf $\overline{J}$ obsahuje cyklus, a proto i graf $B(\overline{J})$ obsahuje cyklus. Proto $f(J) \neq \infty$ a podle Věty 8 je problém NP-úplný. □

4.2 2-degenerované grafy bez J jako indukovaného podgrafu

Ukážeme, že pokud bereme v úvahu třídu 2-degenerovaných grafů bez indukovaného podgrafu izomorfního s grafem J, pak s vrcholy stupně 2 si už dokážeme snadno poradit, což tvrdí následující věta:

Věta 12. *Nechť $H(V,S)$ je rekonstruovatelný hypergraf, a to vzhledem ke třídě 2-degenerovaných grafů bez indukovaného podgrafu izomorfního s grafem J, $V' \subseteq V, S' \subseteq S$ a $\varphi : S' \to V'$ je poctivá částečná rekonstrukce vzhledem k množině S', jinak řečeno, existuje rekonstrukce ψ, pro kterou platí funkcionální rovnice $\varphi = \psi|_{S'}$. Dále nechť $o \in S \setminus S'$, $o \setminus V' = \{u, v, w\}$. Pak lze v čase $O(n^3 \log n)$ vybrat $o' \in S \setminus S'$ a $x \in o' \setminus V'$ tak, že*

$$\varphi' = \varphi \cup \{(o', x)\}$$

je opět poctivá částečná rekonstrukce.

K důkazu této věty ještě budeme potřebovat následující definici.

Definice 32. *Nechť $H(V,S)$ je (ne nutně rekonstruovatelný do grafu) hypergraf. Pak označme $\Omega_A = \{o | o \in S, A \subseteq o\}$, kde $A \subseteq V$. Dále označme $\Gamma_A = |\Omega_A|$. Pokud nebude z kontextu jasné, ke kterému hypergrafu se uvedené symboly vztahují, pak jako horní index za Γ a Ω použijeme označení daného hypergrafu, tj. Γ^H_A a Ω^H_A.*

Je jasné, že pokud hypergraf $H(V,S)$ je rekonstruovatelný a $x, y \in V$, $\Gamma_{\{x,y\}}$ označuje počet cest délky dvě plus dvakrát počet hran mezi vrcholy x, y v libovolné rekonstrukci $G(V,E)$.

Důkaz. Důkaz provedeme rozborem případů, o nichž pak dokážeme, že pro každý 2-degenerovaný graf bez indukovaného podgrafu izomorfního s grafem J nastane právě jeden z těchto případů. Předpokládejme tedy, že $G(V,E)$ je graf definovaný rekonstrukcí ψ a $G' = G[V \setminus V']$ je jeho indukovaný podgraf omezený na vrcholy mimo V'. Dále předpokládejme, že minimální stupeň v grafu G' je dvě, neboť na nižší stupně použijeme lemma 2 a lemma 1.

Dále označme restrikci hypergrafu H' původního hypergrafu H vzhledem k částečné rekonstrukci φ jako $H' = (V \setminus \varphi(S'), \{p \setminus \varphi(S') | p \in S \setminus S'\})$ a zároveň označme restrikci $q_A = q \setminus A$ pro libovolné $q \in S$, ale v důkazu nebudeme rozlišovat mezi q a q_A, tj. jde jen o technickou terminologii určenou k zkrácení důkazu, abychom všude nepsali $q \setminus A$, stejně to platí pro hrany hypergrafu H'.

a) Existují právě tři množiny, jejichž podmnožinou je $\{u, v, w\}$, označme je o, o', o'', kde o je naše původní množina z předpokladů věty. Jinak řečeno, $\Omega^{H'}_{o_{V'}} = \{o, o', o''\}$. Pak ale $G'[o_{V'}]$ je izomorfní s grafem K_3, tedy vrcholy z $o_{V'}$ tvoří trojúhelník jak v grafu G' tak i G, neboť zbývající dvě množiny tvoří okolí vrcholu $\psi(o)$, tj. $\psi(\{o, o', o''\}) = \{u, v, w\} \subseteq o, o', o''$, a proto

$$\{\psi(o)\psi(o'), \psi(o')\psi(o''), \psi(o)\psi(o'')\} \subseteq E' \subseteq E.$$

Vrchol $\psi(o)$ není napojen na žádný z jiných vrcholů, než na vrcholy z množiny $\{u, v, w\}$, proto

$$\psi(o) \in Q = \{u, v, w\} \setminus \bigcup_{p \in S \setminus (S' \cup \Omega^{H'}_o)} p_{V'},$$

protože hvězdy vrcholů z množiny $\{u, v, w\}$ jsou právě $\Omega^{H'}_o$. Pokud $|Q| = 1$, pak jsme byli úspěšní a $\psi(o)$ je právě ten jeden prvek v Q. V opačném případě nevedou hrany z vrcholů v Q do libovolného vrcholu $x \notin \{u, v, w\}$ v grafu G', avšak mohou vést různé hrany mezi vrcholy $\psi(o), \psi(o')$ a $\psi(o'')$ a vrcholy V' v grafu G. Proto pro každý vrchol $\alpha \in \{u, v, w\}$ otestujeme, zda $\varphi \cup \{(o, \alpha)\}$ splňuje podmínky (1) a (2) pro částečné rekonstrukce. V případě, že je nesplňuje, očividně platí $\psi(o) \neq \alpha$. V případě, že je splňuje, očividně musí platit, že $N_G[\alpha] = N_G[\psi(o)]$, a proto podle Věty 10 (Věta o záměně) platí, že $\varphi \cup \{(o, \alpha)\}$ je poctivá částečná rekonstrukce.

b) Bez újmy na obecnosti platí, že $\deg_{G'} u = 2, \deg_{G'} v > 2$ a $\deg_{G'} w > 2$. Zde je situace triviální a musí nutně být $\psi(o) = u$, protože $3 = |o_{V'}| = \deg_{G'} \psi(o) + 1$, což by neplatilo, kdyby $\psi(o) \in \{v, w\}$. Proto φ rozšíříme o (o, u), což je poctivá částečná rekonstrukce.

c) Bez újmy na obecnosti platí, že $\Gamma^{H'}_{\{u,v\}} = \Gamma^{H'}_{\{u,w\}} = 2$ a $\Gamma^{H'}_{\{v,w\}} = 1$. Pokud by existovala hrana $vw \in G'$, pak nutně $v \in N_{G'}[w]$ a $w \in N_{G'}[v]$, a proto by muselo platit, že $\Gamma^{H'}_{\{v,w\}} \geq 2$. Kdyby $\psi(o) \in \{v, w\}$, tak by tato hrana existovala, což je spor s předpokladem. Proto $\psi(o) = u$ a φ rozšířená o (o, u) je poctivá částečná rekonstrukce.

   Zde je nutno dodat, že nemůže nastat případ $\Gamma^{H'}_{\{v,w\}} > 2$, protože by dříve nastal případ b).

d) Platí, že $\Gamma^{H'}_{\{u,v\}} = \Gamma^{H'}_{\{u,w\}} = 2$ a $\Gamma^{H'}_{\{v,w\}} = 2$. Dále nechť bez újmy na obecnosti $\deg_{G'} u = \deg_{G'} v = 2$. Protože vrcholy $\{u, v, w\}$ netvoří podgraf G' izomorfní K_3, nutně existuje vrchol x tak, že $\{u, v, w, x\}$ tvoří v G' indukovanou kružnici délky čtyři. Bez újmy na obecnosti existují množiny

$$\begin{aligned} o_{V'} &= \{u, v, w\}, \\ o'_{V'} &= \{u, v, x\}, \\ o''_{V'} &= \{v, x, w, \ldots\}, \\ o'''_{V'} &= \{u, x, w, \ldots\}, \end{aligned}$$

pro něž platí, že $\psi(\{o, o', o'', o'''\}) = \{u, v, w, x\}$. Takové čtyři množiny existují unikátně, neboť díky předpokladům lze snadno ověřit, že právě jedno $x \in V \setminus V'$ může splňovat tyto předpoklady, kdyby jich bylo víc, vrchol u nebo v by neměl stupeň 2 v G'. Nadále předpokládejme, že $\deg_{G'} x$ nebo $\deg_{G'} w$ není roven dvěma, a opět rozborem případů dokážeme rozšířit φ tak, aby daná funkce byla částečná rekonstrukce.

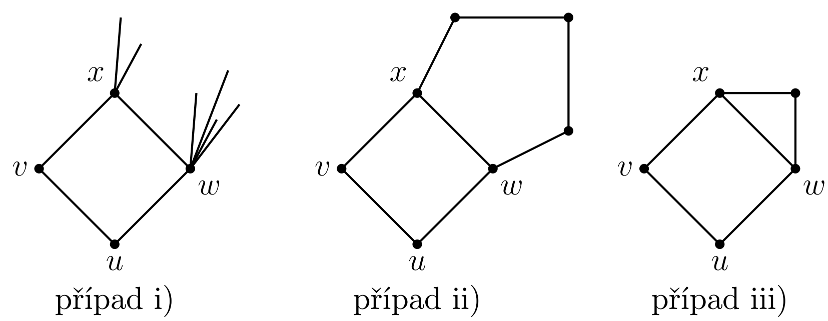
i) Jestliže $\deg_{G'} x \neq \deg_{G'} w$, pak podle velikosti $o''_{V'}$ a $o'''_{V'}$ dokážeme najít buď $\psi(o'')$, nebo $\psi(o''')$. Pokud $\deg_{G'} x < \deg_{G'} w$ a $|o''_{V'}| < |o'''_{V'}|$, pak nutně $\psi(o''') = w$, ostatní případy jsou analogické a rozšíříme φ o nalezenou množinu a vrchol.

ii) Nyní platí, že $\deg_{G'} x = \deg_{G'} w > 2$, jinak by nastal případ i). Proto $\psi(\{o'', o'''\}) = \{w, x\}$. Nechť existuje $s \in S \setminus (S' \cup \{o, o', o'', o'''\})$ takové, že $x \in s, w \notin s$ a $s_{V'} \cap o'''_{V'} = \{x\}$. Nevíme sice, který vrchol je $\psi(s)$, ale víme, že je sousedem vrcholu x. Pokud by platilo, že $\psi(o''') = x$, pak nutně $\psi(s) \in o'''$, protože hrana $x\psi(s) \in E(G') \subseteq E(G)$. Potom by muselo platit, že $|s_{V'} \cap o'''_{V'}| \geq 2$, což je spor. Proto platí $\psi(o''') = w$ a rozšíříme φ o (o''', w). Analogicky otestujeme všechny další možnosti, tj. množinu o'' a prohození x a w. Tento bod pokrývá případ, kdy do vrcholu w nebo x vede nějaká hrana z cizí části grafu nenapojená cestou na druhý vrchol, nebo pokrývá případ, kdy mezi w a x vede indukovaná cesta délky nejméně pět, případně indukovaná cesta délky nejméně čtyři nekončící ve druhém vrcholu.

iii) Probereme případ, že v grafu G' jsou vrcholy w, x napojené na jeden společný vrchol $a \in V \setminus V'$. Pak existuje $\alpha \in S \setminus (S' \cup \{o, o', o'', o'''\})$ takové, že $\{w, x, a\} \subseteq \alpha$. Ukážeme, že $\psi(\alpha) = a$. Protože $\psi(\{o'', o'''\}) = \{w, x\}$, musí $\psi(\alpha) \neq w, x$, jinak by ψ nebyla bijekce a tedy rekonstrukce. Sporem předpokládejme $\psi(\alpha) \neq a$. Protože oba vrcholy w, x mají společného souseda a, je $\{wx, wa, xa\}$ podmnožinou hran grafu G'. Protože navíc $\psi(\alpha) \neq a$, má vrchol $\psi(\alpha)$ společné sousedy w, x, a. Proto graf G' obsahuje všechny hrany z množiny $\{\psi(\alpha)w, \psi(\alpha)x, \psi(\alpha)a\}$. Proto $G'[w, x, a, \psi(\alpha)]$ je úplný graf K_4, který není 2-degenerovaný, což je spor s předpokladem, že G i G' jsou 2-degenerované grafy. Proto musí být $\psi(\alpha) = a$ a φ rozšíříme o (α, a), což bude opět poctivá částečná rekonstrukce.

Zbývá případ, kdy stupně vrcholů u, v, w, x jsou všechny rovny dvěma a graf $G'[u, v, w, x]$ je kružnice C_4. Tato kružnice je sice komponenta grafu G', avšak nemusí být komponentou grafu G. Proto podmínka (2) může být porušena jen vzhledem k množinám S' a o, o', o'', o''', což lze snadno otestovat vyzkoušením všech 24 kombinací. Pokud φ rozšířená o některou z těchto kombinací splňuje podmínky (1) a (2) a je bijekce, pak už je to nutně poctivá částečná rekonstrukce, protože nesprávné přirazení množin této kružnice vrcholům se projeví pouze při restrikci na graf $G[V' \cup \{u, v, w, x\}]$.

Dokážeme, že jsme probrali skutečně všechny možnosti, jak daný graf může vypadat. Nechť tedy máme graf G' a u je vrchol stupně dvě a o je jeho uzavřené okolí, $N_{G'}[u] = \{v, w\}$. Pokud oba sousedi vrcholu u mají stupeň větší než dvě, nastane případ b). Pokud $G[o_{V'}]$ je úplný graf na třech vrcholech, pak nastane případ a). Pokud $vw \notin E(G')$ a mezi vrcholy v a w nevede cesta délky dvě neprocházející přes vrchol u, pak nastává případ c), kdy $\Gamma^{H'}_{\{v,w\}} = 1$. Nyní zbývá poslední možnost, kdy spolu s dalším vrcholem $x \in V \setminus V'$ tvoří $G'[u, v, w, x]$ cyklus délky čtyři. Pokud je tento cyklus komponentou grafu G', pak nastane poslední část případu d). Jinak nechť například w má větší stupeň než dvě, případ v je symetrický a oba zároveň nastat nemohou, na obrázku 4.3 jsou uvedeny různé případy pokryté bodem d).



Obrázek 4.3: Rozbory případů k důkazu Věty 12

Co se týče složitosti, všechny body jdou provést nejvýše v $O(n^3 \log n)$. □

Nyní již máme potřebné prostředky k tomu, abychom dokázali zkonstruovat algoritmus, který řeší problém rekonstruovatelnosti hypergrafů vzhledem ke třídě 2-degenerovaných grafů bez indukovaného podgrafu izomorfního s grafem J.

Input: Hypergraf $F(V,S)$

$\varphi \leftarrow \emptyset, V' \leftarrow \emptyset, S' \leftarrow \emptyset$

while $S \setminus S' \neq \emptyset$ **do**

    $o \leftarrow \arg\min_{o' \in S \setminus S'} |o \setminus V'|$

    switch $o \setminus V'$ **do**

        case $\{v\}$

            $V' \leftarrow V' \cup \{v\}, S' \leftarrow S' \cup \{o\}, \varphi \leftarrow \varphi \cup \{(o,v)\}$

        endsw

        case $\{u,v\}$

            Použij lemma 2 a zjisti, pro které $x \in \{u,v\}$ je $\varphi \cup \{(o,x)\}$ částečná rekonstrukce

            $V' \leftarrow V' \cup \{x\}, S' \leftarrow S' \cup \{o\}, \varphi \leftarrow \varphi \cup \{(o,x)\}$

        endsw

        case $\{u,v,w\}$

            Použij Větu 12 a zjisti, pro které $o \in S \setminus S'$ a $x \in o$ je $\varphi \cup \{(o,x)\}$ částečná rekonstrukce

            $V' \leftarrow V' \cup \{x\}, S' \leftarrow S' \cup \{o\}, \varphi \leftarrow \varphi \cup \{(o,x)\}$

        endsw

        otherwise

            return *NE*

        endsw

    endsw

end

if φ *je rekonstrukce* **then**

    return *ANO*

else

    return *NE*

end

Algoritmus 4: Algoritmus 2DEGBH pro 2-degenerované grafy bez J jako indukovaného podgrafu

Věta 13. *Korektnost a složitost algoritmu 2DEGBH:*

1. *(korektnost) Nechť $H(V,S)$ je rekonstruovatelný hypergraf vzhledem ke třídě 2-degenerovaných grafů bez J jako indukovaného podgrafu. Pak algoritmus 2DEGBH vrátí odpověď ANO. Pokud $H(V,S)$ není rekonstruovatelný vzhledem k této třídě, pak algoritmus vrátí odpověď NE.*

2. *(složitost) Algoritmus pracuje v čase $O(n^4 \log n)$, kde n je počet vrcholů.*

Důkaz. Důkaz bude rozšířením důkazu Věty 9 o jeden bod, neboť jsme algoritmus TREE (viz. str. 23) rozšířili pouze o jeden případ - a to, když jsme narazili na vrchol stupně dvě v grafu G'.

Nechť V', S' jsou aktuální množiny, se kterými v algoritmu pracujeme. Protože je graf $G' = G[V \setminus V']$ 2-degenerovaný, je v něm vrchol stupně nula, jedna nebo dvě, jejichž vyřešení je pokryto lemmaty 1 a 2 a Větou 12, to vše se zvládne vyřešit v čase $O(n^3 \log n)$, počet iterací cyklu je nejvýše n, tedy celková časová složitost je $O(n^4 \log n)$. $\square$

4.3 FPT algoritmus pro 2-degenerované grafy s nejvýše k indukovanými podgrafy J

Vzhledem k Větě 11 je přirozené si klást otázku, v jaké třídě složitosti leží problém rekonstruovatelnosti vzhledem ke třídě grafů s omezeným počtem indukovaných podgrafů izomorfních s grafem J. Zatímco v minulé kapitole jsme ukázali, že v případě 2-degenerovaných grafů bez podgrafu izomorfního s grafem J leží problém rekonstruovatelnosti ve třídě složitosti P, nyní ukážeme, že pokud uvážíme jako parametr počet indukovaných podgrafů izomorfních s grafem J, tak problém rekonstruovatelnosti leží ve třídě FPT vzhledem k tomuto parametru. Ukážeme to modifikací Věty 12 a algoritmu 4 tím, že dokážeme rozpoznat graf J jako indukovaný podgraf v rekonstrukci hypergrafu, a tím získáme omezený rozhodovací strom výšky nejvýše 3^k.

Věta 14. *Nechť $H(V, S)$ je rekonstruovatelný hypergraf a to vzhledem ke třídě 2-degenerovaných grafů, $V' \subseteq V, S' \subseteq S$ a $\varphi : S' \to V'$ je poctivá částečná rekonstrukce vzhledem k množině S', jinak řečeno, existuje rekonstrukce ψ, pro kterou platí funkcionální rovnice $\varphi = \psi|_{S'}$. Dále nechť $o \in S \setminus S'$, $o \setminus V' = \{u, v, w\}$ a graf G je definovaný rekonstrukcí ψ. Pak lze v čase $O(n^3 \log n)$ rozhodnout, zda graf $G' = G[V \setminus V']$ obsahuje podgraf izomorfní grafu J takový, že $\psi(o)$ je vrchol tohoto podgrafu NEBO lze vybrat $o' \in S \setminus S'$ a $x \in o' \setminus V'$ tak, že*

$$\varphi' = \varphi \cup \{(o', x)\}$$

je poctivá částečná rekonstrukce.

Důkaz. Důkaz provedeme jednoduchým rozšířením důkazu Věty 12. Při testování grafu provedeme test v bodech a), b) a c). Pokud žádný z těchto bodů nelze aplikovat na náš případ, začneme s testováním bodu d). Pokud mají v tomto bodu vrcholy x a w oba stupeň dva, pak v grafu G' je izolovaná komponenta - cyklus C_4 a zachováme se podle důkazu věty 12. Pokud nemůžeme aplikovat žádný z podbodů i), ii) nebo iii) bodu d), pak jediná možnost je, že mezi w a x existuje indukovaná cesta délky čtyři, neprocházející vrcholy z o. V tomto případě odpovíme, že graf G' obsahuje podgraf izomorfní grafu J, v opačném případě se zachováme podle důkazu Věty 12 a zkonstruujeme novou větší částečnou rekonstrukci.

Co se týče časové složitosti, je stejná, jako u Věty 12, neboť kód programu ověřující tuto větu je identický až na detekování případu nalezení podgrafu izomorfního grafu J, což zvládneme v konstantním čase. $\square$

Dokážeme korektnost a složitost algoritmu 5.

Věta 15. *Korektnost a složitost algoritmu 2DEGSJ:*

1. *(korektnost) Nechť $H(V, S)$ je rekonstruovatelný hypergraf vzhledem ke třídě 2-degenerovaných grafů, které obsahují nejvýše k indukovaných podgrafů izomorfních grafu J. Pak algoritmus 2DEGSJ vrátí odpověď ANO. Pokud ale $H(V, S)$ není rekonstruovatelný vzhledem k této třídě, pak algoritmus vrátí odpověď NE.*

2. *(složitost) Algoritmus pracuje v čase $O(3^k n^4 \log n)$, kde n je počet vrcholů.*

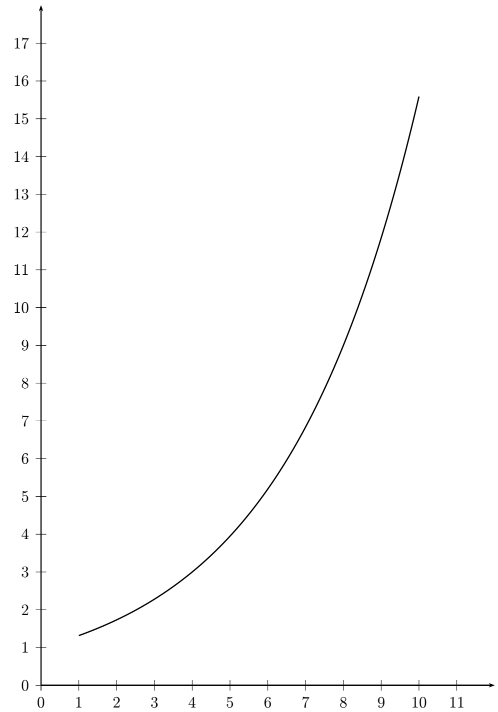
Důkaz. Důkaz bude rozšířením důkazu Věty 13 o případ, když nalezneme J jako indukovaný podgraf grafu G', protože pokud ho nenalezneme, algoritmus je korektní podle předchozí věty. Nechť tedy nalezneme J jako indukovaný podgraf grafu G' a o je aktuální množina, se kterou pracujeme. Neznáme sice $\psi(o)$, ale víme, že $|o \backslash V'| = 3$. Algoritmus proto zkusí všechny tři možnosti, které mohou nastat, a vrátí odpověď NE, pokud neuspěje, v opačném případě vrátí odpověď ANO. Pokud existuje rekonstrukce hypergrafu H, pak alespoň jedna z možností musí být správná, tj. vyzkoušíme možnost $\varphi \cup \{(o, \psi(o))\}$.

K důkazu složitosti si stačí uvědomit, že při zkoušení možností umažeme jeden indukovaný podgraf z grafu G' izomorfní grafu J, neboť vrcholy z množiny $o \backslash V'$ jsou všechny součástí tohoto podgrafu. Proto hloubka rekurze je omezena k a větvení má řád tři, proto má strom nejvýše 3^k uzlů. Pokud hrubě odhadneme práci v jednom uzlu shora jako práci původního algoritmu, tak dostaneme složitost $O(3^k n^4 \log n)$. □

Tento algoritmus je sice v obecnosti exponenciální - tedy nepoužitelný pro velké hypergrafy, protože obecně 2-degenerovaný graf může mít velmi mnoho indukovaných podgrafů izomorfních s grafem J, avšak jednoduchou modifikací dokážeme zajistit, že základ mocniny bude celkem malý, takže obecný algoritmus poběží pořád rychle pro malou velikost grafu, což si dokážeme v následující větě.

Věta 16. *Problém rekonstruovatelnosti vzhledem ke třídě 2-degenerovaných grafů je řešitelný v čase $O(3^{\frac{n}{4}} n^4 \log n)$.*

Důkaz. Uvažme $G' = G[V \setminus V']$, kde G je rekonstrukce daného hypergrafu. Stačí si uvědomit, že pokud najdeme indukovaný podgraf J grafu G' a když rozšíříme φ o (o, p), $p \in o \setminus V'$, pak jednoznačně také určíme rozšíření na o', o'', o''', tj. množiny z důkazu Věty 12, protože jinak by rozšířená φ byla v konfliktu s podmínkou (2) kvůli jinak určeným hranám. Proto nerozšíříme φ pouze o jeden vrchol, ale rovnou o čtyři vrcholy. Proto v každém uzlu větvení smažeme čtyři vrcholy, hloubka rekurze je nejvýše $n/4$, z čehož už plyne znění Věty. □



Obrázek 4.4: Růst funkce $3^{x/4}$ pro malé x

Na obrázku 4.4 vidíme, že pro malá n je tento algoritmus použitelný. Základ mocniny je totiž $3^{1/4} \approx 1.32$, tedy funkce roste přibližně jako 1.32^x.

```
Input: Hypergraf F(V,S), popř. (F(V,S), V', S', φ)
φ ← ∅, V' ← ∅, S' ← ∅ (pouze pokud nebyly zadány V', S', φ)
while S \ S' ≠ ∅ do
    o ← arg min_{o' ∈ S \ S'} |o \ V'|
    switch o \ V' do
        case {v}
            V' ← V' ∪ {v}, S' ← S' ∪ {o}, φ ← φ ∪ {(o, v)}
        endsw
        case {u, v}
            Použij lemma 2 a zjisti, pro které x ∈ {u, v} je φ ∪ {(o, x)} částečná
            rekonstrukce
            V' ← V' ∪ {x}, S' ← S' ∪ {o}, φ ← φ ∪ {(o, x)}
        endsw
        case {u, v, w}
            Použij Větu 14.
            if G' obsahuje indukovaný podgraf izomorfní s J obsahující ψ(o)
            then
                foreach x ∈ {u, v, w} do
                    if 2DEGSJ(F(V,S), V' ∪ {x}, S' ∪ {o}, φ ∪ {(o, x)}) then
                        return ANO
                    end
                end
            else
                Zjisti, pro které o ∈ S \ S' a x ∈ o je φ ∪ {(o, x)} částečná
                rekonstrukce
                V' ← V' ∪ {x}, S' ← S' ∪ {o}, φ ← φ ∪ {(o, x)}
            end
        endsw
        otherwise
            return NE
        endsw
    endsw
end
if φ je rekonstrukce then
    return ANO
else
    return NE
end
```

Algoritmus 5: Algoritmus 2DEGSJ pro 2-degenerované grafy s nejvýše k indukovanými podgrafy J

Kapitola 5

Parametrizace problému různými parametry

5.1 Parametrizace problému rekonstrukce velikostí vrcholového pokrytí

V této části práce se budeme zabývat problémem rekonstruovatelnosti grafů vzhledem ke třídě grafů, které mají vrcholové pokrytí určité velikosti nejvýše k. Ukážeme, že tento problém leží ve třídě XP, vzhledem k parametru k, tím, že nalezneme algoritmus, který problém řeší v čase $O(n^{f(k)})$, pro určitou vyčíslitelnou funkci $f(k)$.

Input: Hypergraf $H(V,S)$
foreach $A \in \binom{V}{k}$ **do**
  foreach $S' \in \binom{S}{k}$ **do**
    foreach $f : S' \to A,\ f$ *je bijekce* **do**
      Zkontroluj, zda pro každé $o \in S \setminus S'$ je $o \setminus A$ jednoprvková množina, pokud to neplatí, pokračuj další iterací cyklu
      Pro $o \in S \setminus S'$ přiřaď $\varphi(o) = v$, kde $o \setminus A = \{v\}$
      Pro $o \in S'$ přiřaď $\varphi(o) = f(o)$
      if φ *je rekonstrukce* **then**
        return *ANO*
      end
    end
  end
end
return *NE*

Algoritmus 6: Algoritmus VERT

Dokážeme korektnost a složitost algoritmu VERT uvedeného výše, který řeší problém rekonstruovatelnosti vzhledem k parametru velikosti vrcholového pokrytí.

Věta 17. *(Korektnost a složitost algoritmu 6)*

a) *(Korektnost) Nechť $H(V,S)$ je rekonstruovatelný hypergraf do grafu s vrcholovým pokrytím velkým nejvýše k. Pak algoritmus $VERT$ vrátí odpověď ANO.*

Pokud hypergraf $H(V,S)$ není rekonstruovatelný vzhledem ke třídě grafů s vrcholovým pokrytím velkým nejvýše k, algoritmus vrátí odpověď NE.

b) *(Složitost) Algoritmus pracuje v čase $O(\binom{n}{k}^2 k! n^2 \log n)$.*

Důkaz. a) Nechť φ je rekonstrukce hypergrafu $H(V,S)$ do grafu $G(V,E)$ s vrcholovým pokrytím A velikosti právě k. Dále nechť $S' \subseteq S$ jsou právě uzavřená okolí vrcholů z A a funkce $f : S' \to A$ přiřazuje uzavřeným okolím vrcholů tyto vrcholy, tedy $f = \varphi|_{S'}$.

Protože graf $G[V \setminus A]$ nemá žádné hrany, platí, že pro $o \in S \setminus S'$, tj. okolí vrcholů z $V \setminus A$, je $o \setminus A$ jednoprvková množina, neboť hrany vedou pouze do A, tedy $V \setminus A$ je nezávislá množina. Protože $\varphi(S') = A$, nelze jinak, než že pro každé $o \in S \setminus S' : \varphi(o) \in o \setminus A = \{v\}$, pro vrchol $v \in V \setminus A$, kde o je jeho uzavřené okolí. Proto $\psi : S \setminus S' \to V \setminus A$ definované jako $\psi(o) = v$, kde $\{v\} = o \setminus A$, je částečná rekonstrukce, neboť $\psi = \varphi|_{S \setminus S'}$.

Dále, protože $f = \varphi|_{S'}$, volbou nové funkce definované jako $\psi' = \psi \cup f$ dostaneme rekonstrukci grafu, která je shodná s naší rekonstrukcí φ, tj. $\psi' = \varphi$. Pokud algoritmus zvolí v cyklu postupně $A, S' = \varphi^{-1}(A)$ a $f = \varphi|_{S'}$, z výše uvedeného plyne, že korektně ověří, že daná algoritmem zkonstruovaná funkce je rekonstrukce a algoritmus vrátí odpověď ANO.

Naopak, pokud $H(V,S)$ není rekonstruovatelný, pak se nemůže stát, že by algoritmus nevrátil odpověď NE, neboť si pokaždé ověřuje, že daná funkce je rekonstrukce.

b) Počet iterací cyklu je nejvýše $\binom{n}{k}^2 k!$, protože $|\binom{V}{k}| = |\binom{S}{k}| = \binom{n}{k}$ a bijekcí je přesně $k!$. Zkontrolovat, zda jsou dané množiny jednoprvkové, trvá nejvýše $O(n^2)$, přiřazení funkce φ závisí na implementaci, ale bude při dobré implementaci trvat nejvýše $O(n)$ a zkontrolovat, zda φ je rekonstrukce, trvá čas $O(n^2 \log n)$. Celkově tedy jedna iterace cyklu trvá čas $O(n^2 \log n)$. Proto je celkový čas $O(\binom{n}{k}^2 k! n^2 \log n)$. □

Při implementaci algoritmu VERT je nutné dbát také na to, že testovaná funkce φ nemusí být vůbec bijekce. Proto je nutné nejen ověřit, že φ splňuje podmínky (1) a (2), tj. je rekonstrukce, ale také, že zároveň je to i bijekce.

Zatím jsme ale neukázali, že problém rekonstruovatelnosti hypergrafů vzhledem ke třídě grafů s vrcholovým pokrytím velikosti nejvýše k leží ve třídě XP, proto musíme ukázat, že algoritmus běží v čase $O(n^{f(k)})$, pro nějakou vyčíslitelnou funkci f, což uděláme v následující Větě.

Věta 18. *Problém rekonstruovatelnosti hypergrafů parametrizovaný velikostí vrcholového pokrytí rekonstrukce leží ve třídě XP.*

Důkaz. Použitím Věty 17 dostáváme algoritmus řešící tento problém s asymptotickou složitostí $O(\binom{n}{k}^2 k! n^2 \log n)$. Musíme ukázat, že existuje vyčíslitelná funkce f taková, že $O(\binom{n}{k}^2 k! n^2 \log n) \subseteq O(n^{f(k)})$. Pomocí známých odhadů dostáváme, že

$$\begin{aligned}
\binom{n}{k}^2 k! n^2 \log n &\leq \binom{n}{k}^2 k! n^3 \\
&\leq n^{2k} k! n^3 \\
&\leq n^{2k+3} k^k \\
&\leq n^{2k+3} e^{k \log k} \\
&\leq n^{2k+3} n^{k \log k} \\
&= n^{k \log k + 2k + 3}
\end{aligned}$$

Snadno se ověří, že funkce $f(k) = k \log k + 2k + 3$ je vyčíslitelná a je závislá pouze na k. Proto máme algoritmus řešící zmíněný problém v čase $O(n^{f(k)})$, a tak problém leží ve třídě XP vzhledem k parametru velikosti vrcholového pokrytí rekonstrukce. □

5.2 Grafy omezené stromové šířky a jejich rekonstruovatelnost

Mnoho obecně NP-těžkých problémů je snadno řešitelných nad grafy omezené stromové šířky, často i v lineárním čase. V této části práce ukážeme, že za určitých dalších předpokladů je i problém rekonstruovatelnosti hypergrafů řešitelný v polynomiálním čase nad třídou grafů omezené stromové šířky. Hlavní problém spočívá v tom, že pokud máme graf G s omezenou stromovou šířkou, tak druhá vzdálenostní mocnina G^2 už nemusí mít omezenou stromovou šířku, kterou bychom mohli využít v algoritmech.

Na chvíli předpokládejme, že pracujeme s grafy, jejichž druhá vzdálenostní mocnina má omezenou stromovou šířku, pro ně nalezneme obecný algoritmus, který řeší problém rekonstruovatelnosti v polynomiálním čase. Nejprve však dokážeme, že pokud máme zadaný hvězdný systém, pak dokážeme odvodit G^2 v polynomiálním čase.

Lemma 3. *Nechť $H(V, S)$ je hvězdný systém grafu G, pak $G^2 = (V, E)$, kde*

$$E = \{(u, v) | \exists o \in S : \{u, v\} \subseteq o\}$$

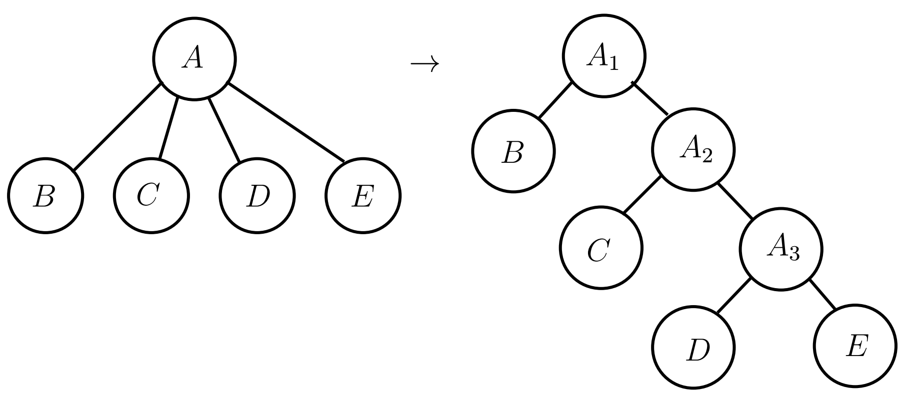
a G^2 lze zkonstruovat v čase $O(n(d+1)^2)$, pokud $\Delta(G) = d$.

Důkaz. Nechť φ je rekonstrukce grafu G a nechť $G' = (V, E')$ je druhá vzdálenostní mocnina grafu G. Chceme dokázat, že $E' = E$. Pokud $e \in E'$, pak buď je e hrana v původním grafu - a potom uzavřené okolí libovolného konce hrany obsahuje i druhý vrchol, a tedy $e \in E$, nebo pokud e reprezentuje cestu v původním grafu délky dvě, pak uzavřené okolí prostředního vrcholu té cesty obsahuje oba dva koncové vrcholy hrany (resp. cesty), a proto $e \in E$. Naopak, pokud $uv \in E$, pak existuje takové $o \in S$, že $u \in o, v \in o$. Pokud bez újmy na obecnosti $\varphi(o) = u$, pak $uv \in E(G)$ a $uv \in E(G^2) = E'$. Naopak, pokud $\varphi(o) \notin \{u, v\}$, pak $u\varphi(o) \in E(G)$ i $v\varphi(o) \in E(G)$, a proto existuje mezi vrcholy u, v cesta délky dvě a tak $uv \in E'$.

Pokud reprezentujeme hrany mezi vrcholy hašovacími tabulkami, tj. každý vrchol má hašovací tabulku se svými sousedy, pak začneme s prázdnými hašovacími tabulkami a projdeme n množin v S, každou $o \in S$ zpracujeme v čase $O((d+1)^2)$ tak, že pro dvojici $u, v \in o$ vložíme do hašovací tabulky vrcholu u vrchol v a naopak. Protože maximální stupeň v grafu G je d, platí, že $|o| \leq d + 1$, neboť větší množina by nemohla být uzavřeným okolím nějakého vrcholu. □

Níže si uvedeme algoritmus pracující s hypergrafem $H(V, S)$, který ze stromového rozkladu grafu G^2 (který je zároveň i stromovým rozkladem grafu G), dokáže získat rekonstrukci grafu G, případně zjistit, že hypergraf není rekonstruovatelný. V tomto případě G^2 nemusí označovat druhou vzdálenostní mocninu nějakého grafu G, ale konstrukci provedenou v lemmatu 3. Z tohoto důvodu ho budeme značit písmenem Q.

Idea algoritmu je taková, že zkonstruujeme stromový rozklad Q a nad tímto stromovým rozkladem budeme konstruovat poctivé částečné rekonstrukce (případně falešné, to v případě, že daný hypergraf není rekonstruovatelný). Ukážeme, jak spojovat částečné rekonstrukce dohromady tak, abychom později dostali rekonstrukci grafu, tj. byly splněny podmínky 1) a 2) pro rekonstrukci z předešlých kapitol. Bez újmy na obecnosti můžeme předpokládat, že stromový rozklad je binární strom, protože každý stromový rozklad lze lehce upravit na binární strom tak, že vrcholy vyššího stupně duplikujeme v řetěz vrcholů stupně tři tak, že levý potomek vrcholu v řetězu nových vrcholů je potomek původního vrcholu a pravý je duplikovaný původní vrchol. Takto získáme korektní stromový rozklad, který je zároveň binárním stromem. Schematické znázornění operace uvedené výše je na Obrázku 5.1.



Obrázek 5.1: Příklad binarizace stromového rozkladu nějakého grafu

Definice 33. *Nechť t je vrchol stromového rozkladu T grafu Q a S je množinový systém. Označení B_t znamená balíček vrcholů grafu Q přiřazených vrcholu $t \in V(T)$. Označení $R(t)$ znamená množinu jistých záznamů pro daný vrchol při práci algoritmu. Dále definujme*

$$
\begin{aligned}
C(S, B_t) &= \{o | o \in S, o \cap B_t \neq \varnothing\}, \\
D(S, B_t, k) &= \{S' | S' \subseteq C(S, B_t), |S'| = k\}.
\end{aligned}
$$

Input: $H(V, S)$

begin

  Zkonstruuj graf $Q(V, E)$ z hypergrafu H konstrukcí z lemma 3, tj. polož $E = \{(u, v) | \exists o \in S : \{u, v\} \subseteq o\}$

  Zkonstruuj stromový rozklad grafu Q, zakořeň ho a postupně ho procházej od listů ke kořeni následujícím způsobem:

  switch *aktuální pozice ve stromovém rozkladu* **do**

    case *list t*

      $R(t) = \{(S', \varphi) | S' \in D(S, B_t, k), \varphi : S' \to B_t,$ splňující a), b), c) $\}$

        a) φ je bijekce,

        b) $\forall o \in S' : \varphi(o) \in o$,

        c) $\forall o \in S', \forall v \in o \cap B_t : v \neq \varphi(o) \Rightarrow \varphi(o) \in \varphi^{-1}(v)$.

    endsw

    case *vrchol t stupně 2 s dětmi t_1 a t_2*

      $R(t) = \{(S', \varphi) | S' \in D(S, B, k), \varphi : S' \to B_t,$ splňující a),b),c),d),e)$\}$

        a) φ je bijekce,

        b) $\forall o \in S' : \varphi(o) \in o$,

        c) $\forall o \in S', \forall v \in o \cap B_t : v \neq \varphi(o) \Rightarrow \varphi(o) \in \varphi^{-1}(v)$,

        d) $\exists (S_1, \varphi_1) \in R(t_1) : \forall v \in B_t \cap B_{t_1} : \varphi^{-1}(v) = \varphi_1^{-1}(v)$,

        e) $\exists (S_2, \varphi_2) \in R(t_2) : \forall v \in B_t \cap B_{t_2} : \varphi^{-1}(v) = \varphi_2^{-1}(v)$.

    endsw

  endsw

  return $R(\text{kořen}) \neq \varnothing$

end

Algoritmus 7: Algoritmus SR řešící rekonstruovatelnost nad stromovými rozklady

Pro důkaz korektnosti budeme potřebovat několik dalších definic. Důkaz složitosti zde v této kapitole uvádět nebudeme, ani se nebudeme složitostí zabývat, protože tu vyřešíme později, s ohledem na otázku omezenosti velikosti stromového rozkladu druhých mocnin grafů.

Definice 34. *Nechť G je graf, (T, B_t) je jeho stromový rozklad a T je zakořeněný strom s kořenem r. Pak zavedeme částečné uspořádání $\leq_T$ nad $V(T)$ následujícím způsobem:*

$$t_1 \leq_T t_2 \Leftrightarrow t_1 \text{ je potomek } t_2, \text{ nebo } t_1 = t_2 \tag{5.1}$$

Lze snadno nahlédnout, že $\leq_T$ je opravdu částečné uspořádání. K němu si uvedeme další definice, které budeme potřebovat.

Definice 35. *Nechť G je graf, (T, B_t) je jeho stromový rozklad a T je zakořeněný strom s kořenem r a nechť $\leq$ je částečné uspořádání $\leq_T$ dle definice 34. Pak zavedeme množinu A_t pro vrchol $t \in V(T)$ následujícím způsobem:*

$$A_t = \bigcup_{t' \leq t} B_{t'} \tag{5.2}$$

Zavedeme si velice důležitý pojem rozšíření záznamu funkce na celý podstrom, o němž pak dokážeme, že je částečnou rekonstrukcí.

Definice 36. *Nechť G je graf, (T, B_t) je jeho stromový rozklad a T je zakořeněný strom s kořenem r. Nechť $t \in V(T)$ je vrchol a t_1, t_2 jsou jeho potomci. Nechť proběhne algoritmus 7 a přiřadí záznamy vrcholům T. Řekneme, že záznamy $(S', \varphi_t) \in R(t)$ a $(S'', \varphi_{t_1}) \in R(t_1)$ jsou kompatibilní, pokud*

$$\forall v \in B_t \cap B_{t_1} : \varphi_t^{-1}(v) = \varphi_{t_1}^{-1}(v) \tag{5.3}$$

Je snadné vidět, že funkce φ_t^{-1} i $\varphi_{t_1}^{-1}$ jsou definovány ve vrcholu v a tak je definice korektní, neboť φ_t i φ_{t_1} jsou bijekce a dívali jsme se na průnik oboru hodnot obou funkcí. Opět lze snadno nahlédnout, že kompatibilní záznamy jsou právě ty záznamy, které splňují body d) a e) v definici algoritmu.

Definice 37. *Nechť G je graf, (T, B_t) je jeho stromový rozklad a T je zakořeněný strom s kořenem r. Nechť $t \in V(T)$ je vrchol, který byl už zpracovaný algoritmem 7. Dále nechť $(S', \varphi_t) \in R(t)$. Pak rozšíření φ ve vrcholu t záznamu (S', φ_t) je relace definovaná rekurzivně takto:*

(a) *t je list, pak $\varphi = \varphi_t$.*

(b) *t je vrchol s potomky t_1, t_2. Pak vezmeme φ_1 rozšíření ve vrcholu t_1 kompatibilního záznamu $(S'', \varphi'_1) \in R(t_1)$ a φ_2 rozšíření ve vrcholu t_2 kompatibilního záznamu $(S''', \varphi'_2) \in R(t_2)$ takové, že*

$$\varphi = \varphi_t \cup \varphi_1 \cup \varphi_2 \tag{5.4}$$

V následujícím lemmatu dokážeme, že tato relace je funkce, a nadále budeme chápat rozšíření jako funkci, nikoliv jako nějakou relaci.

Lemma 4. *Nechť G je graf, (T, B_t) je jeho stromový rozklad a T je zakořeněný strom s kořenem r. Nechť $t \in V(T)$ je vrchol, který byl už zpracovaný algoritmem 7. Dále nechť $(S', \varphi_t) \in R(t)$ a φ je rozšíření pro tento vrchol a tento záznam. Pak φ je funkce.*

Důkaz. Daná relace je funkce, pokud neexistuje takové $v_1, v_2 \in A_t$ a $o \in S$, že $\{(o, v_1), (o, v_2)\} \subseteq \varphi$, tedy by neplatilo, že každému vzoru přiřadíme právě jeden obraz. Dokážeme indukcí, že se tak nestane. Pokud t je list, pak $\varphi = \varphi_t$, kde φ_t je bijekce, a proto φ je funkce. Nechť tedy t je algoritmem zpracovaný vrchol s dětmi t_1, t_2 a $\varphi = \varphi_t \cup \varphi_1 \cup \varphi_2$. Nutně musí platit, že $\{v_1, v_2\} \not\subseteq A_{t_1}$ ani $\{v_1, v_2\} \not\subseteq A_{t_2}$, protože by to byl spor s indukčním předpokladem, protože pak by $\{(o, v_1), (o, v_2)\} \subseteq \varphi_1$ a naopak, což neplatí z indukčního předpokladu. Zároveň, $\{(o, v_1), (o, v_2)\} \not\subseteq \varphi_t$, protože φ_t je bijekce. Dále musí platit, že $\{v_1, v_2\} \subseteq o$. Proto $v_1 v_2 \in E(G)$ a proto existuje takový vrchol stromu $t' \in V(T)$, že $\{v_1, v_2\} \subseteq B_{t'}$. Pokud $t \leq_T t'$, pak $\{v_1, v_2\} \subseteq B_t$ z interpolační vlastnosti stromového rozkladu, což je ve sporu s tím, že φ_t je bijekce. Pokud $t' \leq_T t_i, i = 1, 2$ nebo $t' \leq_T t$, pak opět nastává z interpolační vlastnosti dekompozice situace, že $\{v_1, v_2\} \subseteq A_i$, což je spor. Poslední situace nastává v případě, že t' nelze porovnat s t, t_1, t_2, což nastává v případě, že je t' v jiné části stromu, tj. je za kořenem. Pak kořen r obsahuje oba vrcholy a $t \leq_T r$, což je případ zmíněný výše. □

Nutno podotknout, že rozšíření není určeno jednoznačně, neboť může existovat více kompatibilních záznamů. Dále si uvedeme důležité lemma, které bude tvrdit, že rozšíření má vlastnosti poctivé, resp. falešné částečné rekonstrukce.

Lemma 5. *Nechť G je graf, (T, B_t) je jeho stromový rozklad a T je zakořeněný strom s kořenem r. Nechť $t \in V(T)$ je vrchol, který byl už zpracovaný algoritmem 7. Dále nechť φ je libovolné rozšíření ve vrcholu t libovolného záznamu v $R(t)$. Pak φ má následující vlastnosti:*

1. *φ je bijekce mezi $\varphi^{-1}(A_t)$ a A_t*
2. $\forall o \in \varphi^{-1}(A_t) : \varphi(o) \in o$
3. $\forall o \in \varphi^{-1}(A_t), \forall v \in o \cap A_t : v \neq \varphi(o) \Rightarrow \varphi(o) \in \varphi^{-1}(v)$

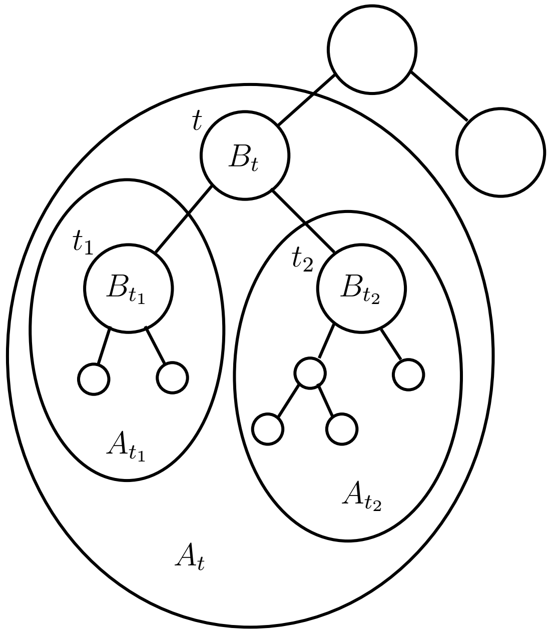
Důkaz. To, že je funkce φ na plyne přímo z konstrukce této funkce - protože A_t je sjednocení všech balíčků pod vrcholem t stromového rozkladu, nutně z toho plyne, že je to funkce na, protože vznikla ze sjednocování bijekcí.

Pokud je vrchol $t \in V(T)$ s potomky t_1 a t_2, pak označme

$$\varphi_{B_t} = \varphi|_{\varphi^{-1}(B_t)} \tag{5.5}$$
$$\varphi_{A_{t_1}} = \varphi|_{\varphi^{-1}(A_{t_1})} \tag{5.6}$$
$$\varphi_{A_{t_2}} = \varphi|_{\varphi^{-1}(A_{t_2})} \tag{5.7}$$
$$S' = \varphi^{-1}(A_t) \tag{5.8}$$
$$S_{t_i} = \varphi^{-1}(A_{t_i}), i = 1, 2 \tag{5.9}$$
$$S_t = \varphi^{-1}(B_t) \tag{5.10}$$

Je jasné, že $\varphi = \varphi_{B_t} \cup \varphi_{A_{t_1}} \cup \varphi_{A_{t_2}}$ a φ_{B_t} je daná bijekce přiřazená příslušnému záznamu v $R(t)$. Další část důkazu provedeme indukcí podle velikosti zpracovávaného podstromu a budeme používat označení výše. Pokud je $t \in V(T)$ list, pak $\varphi = \varphi_{B_t}$

a splňuje dané vlastnosti, protože φ_{B_t} je bijekce a splňuje body a), b) i c) v definici algoritmu. Nechť tedy t není list, ale je to vrchol se dvěma potomky t_1 a t_2. Situaci a označení vidíme na obrázku 5.2.



Obrázek 5.2: Ilustrace k lemma 5.

Pro spor předpokládejme, že φ není prostá, tedy existuje $o_1, o_2 \in S'$ takové, že $\varphi(o_1) = \varphi(o_2) = v \in A$. Z indukčního předpokladu musí platit, že $\{o_1, o_2\} \nsubseteq S_1$ ani $\{o_1, o_2\} \nsubseteq S_2$, protože by pak neplatil indukční předpoklad. Pokud $\{o_1, o_2\} \subseteq S_t$, pak φ_{B_t} není bijekce, což je spor. Nechť tedy bez újmy na obecnosti například $o_1 \in S_{t_1}$ a $o_2 \in S_t$. Pak $v \in A_{t_1}$ i $v \in B_t$, ale z interpolační vlastnosti stromového rozkladu musí platit, že $v \in B_{t_1}$, proto platí $v \in B_{t_1} \cap B_t$, a

$$o_1 = \varphi^{-1}_{A_{t_1}}(v) = \varphi^{-1}_{B_t}(v) = o_2, \tag{5.11}$$

což je spor. Obdobně pokud $o_i \in S_{t_i}, i = 1, 2$, pak $v \in A_{t_i}, i = 1, 2$ a z interpolační vlastnosti plyne, že $v \in B_t, B_{t_1}$ i B_{t_2}, a tedy podle vlastnosti d), e) algoritmu musí platit, že

$$o_1 = \varphi^{-1}_{A_{t_1}}(v) = \varphi^{-1}_{B_t}(v) = \varphi^{-1}_{A_{t_2}}(v) = o_2, \tag{5.12}$$

což je opět spor, protože jsmě předpokládali, že $o_1 \neq o_2$.

Druhá vlastnost plyne triviálně, neboť pro S_t to platí přímo z definice a pro $S_{t_i}, i = 1, 2$ to platí z indukčního předpokladu.

Třetí vlastnost dokážeme opět sporem. Nechť neplatí, to znamená, že existuje $o \in S'$ a $\varphi(o) \neq v \in o \cap A_t$ takové, že $\varphi(o) \notin \varphi^{-1}(v)$. Pak $\{\varphi(o), v\} \subseteq A_t$, ale zároveň $\{\varphi(o), v\} \nsubseteq A_{t_i}, i = 1, 2$, protože jinak by byl porušen indukční předpoklad. Zároveň existuje $t' \in V(T)$ takové, že $\{\varphi(o), v\} \subseteq B_{t'}$, protože hrana $\varphi(o)v \in E(G)$. Z interpolační vlastnosti stromového rozkladu pak plyne, že $\{\varphi(o), v\} \subseteq B_t$. Pak ale nesplňuje φ_{B_t} vlastnost b) algoritmu, což je spor.

□

Nyní už můžeme dokázat korektnost algoritmu 7, což tvrdí následující věta:

Věta 19. *Nechť $H(V, S)$ je hypergraf. Pokud je tento hypergraf rekonstruovatelný, pak algoritmus vrátí kladnou odpověď, v opačném případě vrátí zápornou odpověď.*

Důkaz. To, že je algoritmus konečný, je vidět snadno z jeho definice. Nechť je dán hypergraf a proběhl algoritmus, (T, B_t) je zakořeněný stromový rozklad grafu Q s kořenem $r \in V(T)$. Pokud $R(r) \neq \emptyset$, pak libovolné rozšíření ve vrcholu r libovolného záznamu v $R(r)$ je rekonstrukce daného hypergrafu, což ukážeme: použijme lemma 5 na kořen r a libovolný záznam v něm. Pak platí, že $A_r = V$ a platí:

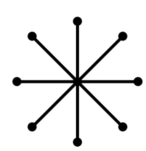
1. φ je bijekce mezi $\varphi^{-1}(V)$ a $A_t = V$
2. $\forall o \in \varphi^{-1}(V) : \varphi(o) \in o$
3. $\forall o \in \varphi^{-1}(V), \forall v \in o \cap A_t : v \neq \varphi(o) \Rightarrow \varphi(o) \in \varphi^{-1}(v)$

což jsou přesně vlastnosti rekonstrukce, jak jsme si ji definovali v dřívějších kapitolách, tedy daný hypergraf je rekonstruovatelný. Pokud hypergraf rekonstruovatelný není, pak daný záznam v $R(r)$ existovat nemůže. Naopak, pokud je hypergraf rekonstruovatelný a φ je jeho rekonstrukce, pak snadno z indukce dostaneme, že $(\varphi^{-1}(B_t), \varphi|_{\varphi^{-1}(B_t)})$ leží v $R(t)$ každého vrcholu t stromového rozkladu (T, B_t). □

Obecně, algoritmus může mít značnou složitost, protože druhá mocnina grafu má obvykle velký stromový zdvih. Obecně nebude platit, že pokud graf G má omezený stromový zdvih, tak G^2 má omezený stromový zdvih, protože například pro graf hvězdy to neplatí, neboť druhá vzdálenostní mocnina hvězdy je úplný graf. Neplatí to tedy ani pro stromy, takže nalézt vhodné třídy grafů bude poměrně těžké.

5.3 Omezená stromová šířka a časová složitost

Ukážeme, že pokud má graf G omezenou stromovou šířku a má omezený stupeň, pak G^2 má také omezenou stromovou šířku. Naopak to samozřejmě neplatí, protože například graf na obrázku 5.3 má stromovou šířku dvě, avšak jeho druhá vzdálenostní mocnina je úplný graf. Proto je omezenost stupňů nutnou podmínkou pro omezenost stromové šířky druhé mocniny grafu.



Obrázek 5.3: Příklad grafu G s omezenou stromovou šířkou

Věta 20. *Nechť graf G má stromovou šířku menší nebo rovnou k a $\Delta(G) \leq d$. Pak stromová šířka grafu G^2 je nejvýše $kd + k + d$.*

Důkaz. Uvažme libovolný stromový rozklad (T, B_t) grafu $G(V, E)$ velikosti nejvýše $k+1$. Tento rozklad upravíme tak, že do každého balíčku vrcholů B přidáme všechny sousedy vrcholů z B a ukážeme, že je to stromový rozklad grafu $G^2(V, E')$. Protože $|B| \leq k + 1$, pak nové balíčky mají velikost nejvýše

$$(k + 1) + d \cdot (k + 1) = (k + 1) \cdot (d + 1), \tag{5.13}$$

takže zbývá dokázat, že se jedná skutečně o stromový rozklad G^2. Protože jsme do balíčků vrcholy pouze přidávali, pak jejich sjednocení je opět V a pro každou hranu z E existuje balíček obsahující oba konce dané hrany. Totéž platí i pro hrany mocniny grafu, protože pokud mezi dvěma vrcholy vedla cesta délky dvě, pak balíček obsahující prostřední vrchol obsahuje oba dva koncové vrcholy dané cesty. Zbývá dokázat interpolační vlastnost. Musí pro libovolný vrchol platit, že množina těch balíčků, které obsahují daný vrchol, je podstrom stromového rozkladu. My ale přidáváme pouze sousedy vrcholů, tedy nový podstrom pro daný vrchol bude sjednocení podstromů pro jeho uzavřené okolí. □

Nyní už se můžeme zabývat složitostí algoritmu 7. Ukážeme, že problém rekonstruovatelnosti leží ve třídě FPT vzhledem k dvojici parametrů stromová šířka a maximální stupeň grafu.

Věta 21. *Nechť $k, d \in \mathbb{N}$. Nechť $\mathcal{G}$ je třída grafů stromové šířky nejvýše k a maximálního stupně d. Pak platí, že problém rekonstruovatelnosti hypergrafů vzhledem ke třídě grafů $\mathcal{G}$ leží ve třídě složitosti FPT vzhledem k parametru (k, d) a lze řešit v čase*

$$O\left((d+1)^{6\,(d+1)\,(k+1)+1}\,(k+1)^{2(d+1)\,(k+1)+1} n + f\left((k+1)(d+1)\right) n\right), \tag{5.14}$$

kde $f(x)$ parametrizuje konstantu z algoritmu pro nalezení stromového rozkladu v lineárním čase [3].

Důkaz. Nechť máme na vstupu hypergraf $H(V, S)$ a aplikujeme algoritmus 7. Vyrobit z něj graf Q zabere čas nejvýše $O((d+1)^2 n)$, protože každá množina má velikost nejvýše $d + 1$, neboť maximální stupeň vrcholu v rekonstrukci hypergrafu je menší, nebo roven d a máme n vrcholů. Vyrobit stromový rozklad (T, B_t) grafu Q trvá $O(f((k + 1)(d + 1))n)$ dle výsledku [3], kde $f(x)$ je funkce závisející pouze na x a schovává se do ní konstanta algoritmu v závislosti na stromové šířce grafu.

Podle Věty 20 platí pro každé $t \in V(T)$, že $|B_t| \leq (k + 1)(d + 1)$, navíc, každý vrchol $v \in V$ se nachází v nejvýše $d+1$ množinách, protože maximální stupeň vrcholu v rekonstrukci hypergrafu je d. Proto platí, že

$$|\{o | o \in S, o \cap B_t \neq \varnothing\}| \leq (k + 1)(d + 1)^2, \forall t \in V(T) \tag{5.15}$$

Nyní spočítáme časovou náročnost práce v uzlech stromového rozkladu. Pro daný uzel $t \in V(T)$ lze vybrat tyto množiny v čase $O(n(d + 1))$, pokud je B_t reprezentováno hašovací tabulkou. Počet způsobů, jak vybrat φ z $|B_t| \le (k+1)(d+1)$ množin, je nejvýše

$$(d+1)^{(d+1)(k+1)}, \tag{5.16}$$

a počet způsobů, jak vybrat tyto množiny je nejvýše

$$(d+1)^{2\,(d+1)\,(k+1)}\,(k+1)^{(d+1)\,(k+1)}, \tag{5.17}$$

celkem tedy máme

$$(d+1)^{3\,(d+1)\,(k+1)}\,(k+1)^{(d+1)\,(k+1)} \tag{5.18}$$

možností, jak vybrat dvojici (S', φ) jako kandidáta do $R(t)$. Spočítáme, jakou časovou složitost má ověření vlastnosti a), b), c) popř. ještě d) a e) v případě, že t není list. Ověřit, že φ je bijekce, lze snadno v čase $O(|B_t|^2) = O((d+1)^2(k+1)^2)$. Ověřit, zda $\varphi(o) \in o$ pro $o \in S'$, lze v čase $O(|B_t|) = O((d+1)(k+1))$ s využitím hašovacích tabulek. Verifikace vlastnosti c), tj. zda každé $v \in o \cap B_t, v \neq \varphi(o) \Rightarrow \varphi(o) \in \varphi^{-1}(v)$ lze v čase $O((d+1)|B_t|) = O((d+1)^2(k+1))$. Poslední zbývá složitost ověření vlastností d) a e). Nechť t_1, t_2 jsou děti vrcholu t. Spočítat průnik $B_{t_i} \cap B_t$ lze v čase $O((d+1)(k+1))$. Najít kompatibilní záznam v $R(t_i)$ lze hrubou silou v čase

$$O\left((d+1)^{3\,(d+1)\,(k+1)}\,(k+1)^{(d+1)\,(k+1)}(k+1)(d+1)\right). \tag{5.19}$$

V symbolu $O(\dots)$ je schován fakt, že toto provádíme dvakrát pro t_1 i t_2. Zároveň je výraz výše asymptoticky větší než asymptotická složitost všech předchozích ověřovacích operací. Proto je celková práce v uzlu rovna

$$O\left((d+1)^{6\,(d+1)\,(k+1)+1}\,(k+1)^{2(d+1)\,(k+1)+1}\right). \tag{5.20}$$

Protože je celkem uzlů ve stromovém rozkladu nejvýše $O(n)$, je celkový čas algoritmu

$$O\left((d+1)^{6\,(d+1)\,(k+1)+1}\,(k+1)^{2(d+1)\,(k+1)+1} n + f((k+1)(d+1))n\right), \tag{5.21}$$

což jsme chtěli dokázat. □

Lehkou modifikací algoritmu 7 dokážeme získat nejenom rekonstrukci, ale počet všech rekonstrukcí či rovnou všechny rekonstrukce průchodem do hloubky stromového rozkladu (T, B_t) podle záznamů $R(t)$. Počet rekonstrukcí lze získat v polynomiálním čase, avšak ne všechny rekonstrukce, neboť jich obecně může být exponenciálně mnoho.

Kapitola 6

Závěr

Pro některé třídy grafů jsme dokázali zkonstruovat algoritmus, řešící problém rekonstruovatelnosti hypergrafů vzhledem k této třídě poměrně efektivně, tj. tento problém leží ve třídách P, FPT, popř. XP. Tabulka níže shrnuje výsledky této práce.

	Třída grafů	Parametr	Třída složitosti
1.	2-degenerované grafy bez J jako ind. podgrafu	-	P
2.	2-degenerované grafy s k J ind. podgrafy	k	FPT
3.	grafy s vrcholovým pokrytím nejvýše k	k	XP
4.	grafy G s $tw(G) \leq k$ a $\Delta(G) \leq d$	(k, d)	FPT

Tabulka 6.1: Přehled hlavních výsledků práce

Zatímco algoritmus pro 2-degenerované grafy je prakticky použitelný, domnívám se, že to neplatí pro zbývající dva algoritmy řešící problém rekonstruovatelnosti vzhledem k třídě grafů s omezeným vrcholovým pokrytím resp. omezeného stromového zdvihu a stupně vrcholů, neboť konstanta roste velmi rychle s velikostí parametru, zejména to platí pro poslední algoritmus, jak je patrné ze vzorce složitosti.

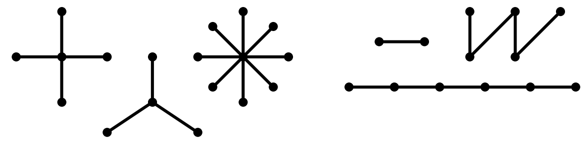
6.1 Neporovnatelnost výsledků

V této poslední části práce ukážeme, že výsledky v jednotlivých kapitolách nejsou vzájemně porovnatelné, tj. žádný výsledek není silnější, než druhý.

Následující věta říká, že výsledek 3. a 4. není silnější, než výsledek 1. a 2.

Věta 22. *Existuje třída $\mathcal{F}$ 2-degenerovaných grafů bez J jako indukovaného podgrafu, který nemá omezený stupeň a ani omezené vrcholové pokrytí.*

Důkaz. Důkaz provedeme konstruktivně. Uvažme $\mathcal{F} = \{S_n, P_n | n \in \mathbb{N}\}$, kde S_n je hvězda s n vrcholy a P_n je cesta s n vrcholy. Příklady grafů v této třídě jsou na obrázku 6.1.



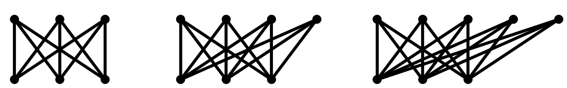
Obrázek 6.1: Některé grafy ve třídě $\mathcal{F}$

Je snadné nahlédnout, že hvězda nemá omezený stupeň a cesta nemá omezené vrcholové pokrytí. Přitom je jasné, že jsou tyto grafy 2-degenerované (dokonce jsou to stromy) a neobsahují graf J jako indukovaný podgraf, protože graf J obsahuje kružnici. □

V další větě ukážeme, že výsledky 1., 2. a 4. nejsou silnější, než výsledek 3.

Věta 23. *Existuje třída $\mathcal{G}$ grafů s omezeným vrcholovým pokrytím, které nejsou 2-degenerované a mají neomezené stupně.*

Důkaz. Důkaz provedeme opět konstruktivně. Položme $\mathcal{G} = \{K_{m,3} | m \geq 3\}$, jedná se tedy o podtřídu bipartitních grafů. Platí, že minimální stupeň těchto grafů je tři, proto grafy z této třídy nejsou 2-degenerované. Pokud $G \in \mathcal{G}$, pak $\Delta(G) \geq |V(G)| - 3$, proto tyto grafy nemají omezené stupně. Příklady grafů ve třídě $\mathcal{G}$ jsou na obrázku 6.2.



Obrázek 6.2: Některé grafy ve třídě $\mathcal{G}$

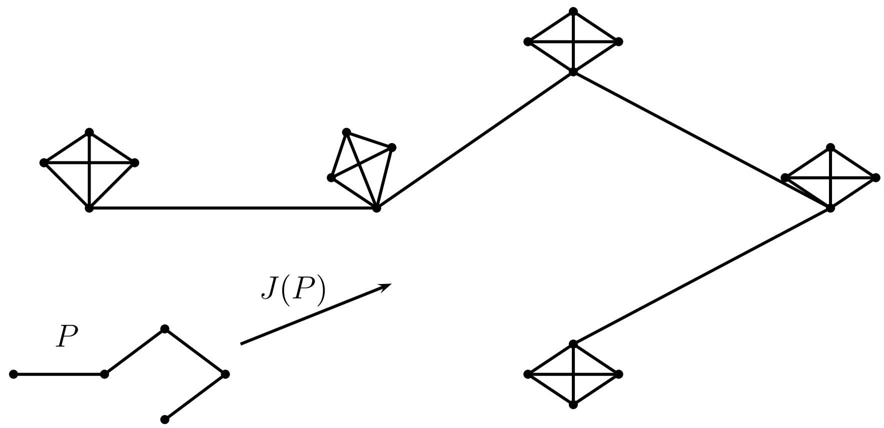
Přitom je snadné vidět, že každý graf ve třídě $\mathcal{G}$ má vrcholové pokrytí velikosti tři, neboť stačí odebrat z příslušného bipartitního grafu partitu velikosti tři a získáme nezávislou množinu.

□

V poslední větě ukážeme, že výsledky 1., 2. a 3. nejsou silnější, než výsledek 4.

Věta 24. *Existuje třída $\mathcal{H}$ grafů s omezeným stromovým zdvihem a omezeným maximálním stupněm, avšak s neomezeným vrcholovým pokrytím a grafy v této třídě nejsou 2-degenerované.*

Důkaz. Důkaz opět provedeme konstruktivně, a konstrukce bude složitější než v případě předchozích dvou vět. Nechť $\mathcal{P}$ je třída všech konečných cest. Definujeme operaci $J(P) : \mathcal{P} \to \textit{grafy}$ takto: Pokud $P \in \mathcal{P}$, ke každému vrcholu $v \in V(P)$ připojíme graf K_3 tak, že vznikne graf K_4, tj. vrchol v spojíme hranami se všemi vrcholy grafu K_3. Příklad této operace je na obrázku 6.3.



Obrázek 6.3: Příklad operace $J(P)$

Položme nyní $\mathcal{H} = \{J(P) : P \text{ je cesta }\}$. Pokud $G \in \mathcal{H}$, pak triviálně platí, že $\Delta(G) \leq 5$, je to patrné z obrázku 6.3. Zároveň, uvažme určitý stromový rozklad (velikosti dvě) cesty, ze které graf G vznikl, a to ten stromový rozklad, kde balíčky jsou ekvivalentní hranám, a stromový rozklad je opět cestou. Do každého balíčku přidáme všechny sousedy vrcholů z $V(G)$ v tom balíčku, a těch je nejvýše $3 + 3$ za grafy K_3 a 2 za sousedy po cestě, získáme tak stromový rozklad velikosti nejvýše 10 a tedy stromová šířka je nejvýše devět.

Naopak, již z obrázku 6.3 je vidět, že graf G má minimální stupeň tři a tedy není 2-degenerovaný. Protože K_4 má vrcholové pokrytí velikosti tři a těchto grafů je $\Omega(|V(G)|)$ pro $G \in \mathcal{H}$, mají grafy ze třídy $\mathcal{H}$ vrcholové pokrytí velikosti $\Omega(n)$, kde n je počet vrcholů daného grafu.

$\square$

V těchto třech větách jsme dokázali, že výsledky v jednotlivých kapitolách jsou vzájemně neporovnatelné, tzn. žádný výsledek není silnější než jiný.

6.2 Otevřené problémy

Přestože některé otevřené problémy spojené s rekonstruovatelností hypergrafů se podařilo vyřešit, pořád zde zůstává řada nezodpovězených otázek. V publikaci [6] byla položena otázka, zda je problém rekonstruovatelnosti vzhledem ke třídě K_4-free grafů, popř. K_4^-[1]-free grafů NP-úplný, nebo leží ve třídě P. Tato otázka zůstává stále nezodpovězená. Zároveň tato diplomová práce pokládá následující otázky:

1. Leží problém rekonstruovatelnosti vzhledem ke 2-degenerovaným grafům ve třídě P? Moje hypotéza je, že ano, avšak nepodařilo se mi vyřešit případ, kdy se nalezne podgraf izomorfní grafu J, definovanému na obrázku 4.1.

2. Pokud zní odpověď na první otázku záporně, domnívám se, že problém rekonstruovatelnosti vzhledem k 2-degenerovaným grafům lze řešit algoritmem,

[1] graf K_4 bez hrany

jehož průměrná časová složitost (přes všechny 2-degenerované grafy) je polynomiální.

3. V jaké třídě složitosti leží problém rekonstruovatelnosti vzhledem ke grafům s omezeným stromovým zdvihem a omezeným maximálním stupněm, kde zdvih a maximální stupeň jsou na vstupu algoritmu? Domnívám se, že jde o NP-úplný problém.

4. V jaké třídě složitosti leží problém rekonstruovatelnosti parametrizovaný různými dalšími parametry, například dominující množinou nebo barevností grafu?

Seznam obrázků

Seznam tabulek

Seznam algoritmů

Literatura

[1] Aigner, M.; Triesch, E.: *Reconstructing a graph from its neighborhood lists*, Combin. Probab. Comput., 2 (1993), pp. 103-113. 7, 20

[2] Arora, Sanjeev; Barak, Boaz (2009): *Computational Complexity: A Modern Approach*, Cambridge, ISBN 978-0-521-42426-4 10

[3] Bodlaender, Hans L.: *A Linear-Time Algorithm for Finding Tree-Decompositions of Small Treewidth*, SIAM J. Comput. 25, 1305 (1996), DOI:10.1137/S0097539793251219 9, 45

[4] Boros, E.; Gurvich, V.; Zverovich, I.: *Neighborhood hypergraphs of bipartite graphs*, J. Graph Theory 58 (2008), no. 1, pp. 69-95. 7, 20

[5] Downey, Rod; Fellows, M. (1999), *Parameterized complexity*, Berlin, New York: Springer-Verlag 10

[6] Fomin, Fedor V.; Kratochvíl, Jan ; Lokshtanov, Daniel; Mancini, Federico; Telle, Jan Arne: *On the Complexity of Reconstructing H-free Graphs from their Star Systems*, LATIN 2008: Theoretical Informatics, Lecture Notes in Computer Science, 2008, Volume 4957/2008, pp. 194-205. 7, 15, 20, 21, 26, 49

[7] Goldreich, Oded (2008): *Computational Complexity: A Conceptual Perspective*, Cambridge University Press 10

[8] Hajnal A.; Sós, V., eds.: *Combinatorics. Vol II*, vol. 18 of Colloquia Mathematica Societatis János Bolyai, North-Holland, 1978. 7

[9] Hliněný, Petr: *Základy teorie grafů pro (nejen) informatiky*, učební text, Masarykova Univerzita, Fakulta informatiky, `http://www.fi.muni.cz/~hlineny/` 8

[10] Král', Daniel: Přednáška *NDMI073 Kombinatorika a grafy III* - zimní semestr 2009/10, Univerzita Karlova v Praze, Matematicko-fyzikální fakulta 8, 9

[11] Lalonde, F.: *Le problème d'étoiles pour graphes est NP-complet.*, Discrete Math., 33 (1981), pp. 271-280. 7, 15

[12] Matoušek, Jiří; Nešetřil, Jaroslav: *Kapitoly z diskrétní matematiky*, Nakladatelství Karolinum, Praha 2002, ISBN 978-80-246-1411-3 7

[13] Niedermeier, Rolf: *Invitation to Fixed-Parameter Algorithms*, Oxford Lecture Series in Mathematics and Its Applications, Oxford University Press 312 pages, February 2006 11, 13

[14] Niedermeier, Rolf; Guo, Jiong: *Fixed-Parameter Algorithms*, PDF materiály k přednášce v zimním semestru 2005/2006, Friedrich-Schiller-Universität Jena, http://theinf1.informatik.uni-jena.de/ 11