Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE

Kryštof Hoder

Hra minolovka – výpočetní složitost a implementace hledání řešení

Katedra aplikované matematiky

Vedoucí bakalářské práce: RNDr. Jiří Fiala, Ph.D.
Studijní program: Informatika, obecná informatika

2007

Děkuji RNDr. Jiřímu Fialovi, Ph.D. za hodnotné rady, poskytnutá skripta a odborné vedení mé práce.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 22. 5. 2007 Kryštof Hoder

Obsah

Název práce: Hra minolovka – výpočetní složitost a implementace hledání řešení
Autor: Kryštof Hoder
Katedra (ústav): Katedra aplikované matematiky
Vedoucí bakalářské práce: RNDr. Jiří Fiala, Ph.D.
e-mail vedoucího: fiala@kam.mff.cuni.cz

Abstrakt: V předložené práci studujeme vytváření stromových rozkladů grafu se speciálním zřetelem na grafy užitečné při hraní hry Minolovka. Zároveň formalizujeme postup hry a zavádíme potřebnou terminologii. Na základě tohoto jsme našli širokou množinu konfigurací hry, o jejichž konzistentnosti lze rozhodovat v polynomiálním čase — že problém je v obecnosti NP-úplný bylo ukázáno již dříve v jiných pracech. Taktéž popisujeme algoritmy, které klasifikují konfigurace a případně v polynomiálním čase rozhodnou o jejich konzistentnosti.

Klíčová slova: Minolovka, stromový rozklad, rozklad grafu

Title: Minesweeper game – computational complexity and solver implemetation
Author: Kryštof Hoder
Department: Department of Applied Mathematics
Supervisor: RNDr. Jiří Fiala, Ph.D.
Supervisor's e-mail address: fiala@kam.mff.cuni.cz

Abstract: In the present work we study construction of tree decompositions with respect to graphs useful for playing the Minesweeper game. We also formalize rules of the game and present necessary terminology. We provide the set of game configurations whose consistency can be decided in polynomial time — problem of consistency-deciding of general game configuration has been proven NP-compete in other works. We also provide algorithms that classify game configurations and decide about consistency of those configurations, which we can decide about in polynomial time.

Keywords: Minesweeper, tree decomposition, graph decomposition

Kapitola 1

Úvod

1.1 Potřebné definice

Definice 1. *Mějme množinu symbolů* $S = \{\sharp, M, \diamond, 0, 1, 2, 3, 4, 5, 6, 7, 8\}$ *a nad ní matici* H *s* m *sloupci a* n *řádky, kterou budeme nazývat* hrací plochou typu n/m*. Prvek* H *v* i*-tém sloupci a* j*-tém řádku budeme značit* $H_{i,j}$ *a budeme mu říkat* pole.

K významu symbolů: $\sharp$ značí dosud neodkryté pole, M je pole na kterém je mina, 0, 1, …, 8 jsou odkrytá pole, na nichž mina není a která nám dávají informaci o počtu min v jejich okolí. A $\diamond$ je symbol pole, které je sice odkryté, mina na něm není, ale informaci o počtu okolních min nám neposkytuje. Tento typ pole se v klasické hře nevyužívá, ale nám se bude později hodit. Jeho využití našemu aparátu na síle nijak neubere — jsme-li schopni řešit zadání, v nichž se může vyskytnout, jsme schopni dozajista vyřešit i ta, v nichž se nevyskytuje.

Značení 1. *Fakt, že hodnotou pole* $H_{i,j}$ *je symbol* $s \in S$ *budeme značit* $H_{i,j} \equiv s$*, pokud hodnota pole* $H_{i,j}$ *patří do množiny* $P \subset S$*, budeme psát* $H_{i,j} \Lleftarrow P$*. Dále pro pole* $A_{i,j} \Lleftarrow \{0, \ldots, 8\}$ *zaveďme označení* $|A_{i,j}|$ *pro číselnou hodnotu tohoto pole. Pro pole s hodnotami* $\sharp$*,* M *a* $\diamond$ *není* $|A_{i,j}|$ *definováno.*

Definice 2. *Množinu všech hracích ploch s danými* m *a* n *nazveme* prostorem hracích ploch *a označíme* $\mathbf{F}_{m,n}$*.*

Nechť $A, B \in \mathbf{F}_{m,n}$*, potom řekneme, že* A je upřesněním B*, pokud* $\forall i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\} : A_{i,j} \equiv B_{i,j} \vee B_{i,j} \equiv \sharp$.

Dále řekneme, že je hrací plocha A typu n/m přípustná*, pokud má každé pole se symbolem $s \in \{0, \ldots, 8\}$ nejvýše s sousedů se symbolem M a alespoň s sousedů se symbolem buď M nebo $\sharp$.*

Pokud žádné pole hrací plochy nemá hodnotu $\sharp$, řekneme, že je tato hrací plocha dohraná*.*

Hrací plocha je konzistentní*, právě když existuje přípustná dohraná plocha, která je jejím upřesněním.*[1]

1.2 Cíl práce

Cílem této práce bude zabývat se složitostí rozhodovacího problému, který řekne, zda je daná hrací plocha konzistentní, nebo ne. Kaye [2] ukázal, že v obecnosti je tento problém NP-úplný a tak zde budeme hledat omezující podmínky pro hrací plochy, při jejichž splnění bude tento problém řešitelný polynomiálně.

Vlastní implementace hledání řešení se potom stává spíše okrajovou záležitostí. Jakmile jsme totiž schopni rozhodnout o konzistentnosti hrací plochy, dokážeme i hledat řešení. Stačí na konkrétní neodkryté pole hrací plochy vložit minu, resp. je označit jako pole bez miny, a řekne-li nám algoritmus o této ploše že je nekonzistentní, víme, že tam mina být nemůže, resp. musí.

[1] Tedy např. každá přípustná dohraná plocha je hned i konzistentní.

Kapitola 2

Využití teorie grafů

2.1 Graf plochy

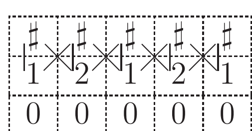
Definice 3. *Nechť $A \in \mathbf{F}_{m,n}$ je hrací plocha, $A_M = \{A_{i,j} \| A_{i,j} \equiv M\}$ je množina polí obsahujících minu, $A_U = \{A_{i,j} \| A_{i,j} \equiv \sharp\}$ je množina neznámých polí a $A_K = \{A_{i,j} \| A_{i,j} \Lleftarrow \{0, \ldots, 8\}\}$ je množina odkrytých polí.* Grafem plochy *A potom bude graf $G(V, E)$, pro který platí $V = A_M \cup A_U \cup A_K$ a zároveň*

$(a, b) \in E \Leftrightarrow a \in A_K \wedge b \in A_M \cup A_U \wedge$ *a,b se liší v každé souřadnici nejvíc o 1*

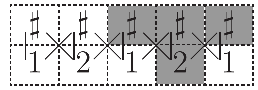
Na obr. 2.1 je ukázka hrací plochy s jejím grafem. Vrcholy grafu představují jednotlivá pole a hrany mezi nimi jsou znázorněny čárami.

Důsledek 1. *Přímo z definice grafů plochy plyne, že tyto grafy jsou bipartitní, přičemž jednou jejich partitou jsou vrcholy z $A_M \cup A_U$ a druhou ty z A_K.*

Značení 2. *Množinu vrcholů sousedních s vrcholem u v grafu G budeme značit $N_G(u)$, nehrozí-li nejasnost, o který graf se jedná, budeme psát pouze $N(v)$.*



Obrázek 2.1: Příklad plochy a jejího grafu



Obrázek 2.2: Příklad množiny, která má slupku

2.2 Ohodnocení

Definice 4. *Mějme graf plochy $G(V, E)$. Potom o množině $X \subset V$ řekneme, že* má slupku, *pokud platí $\forall p \in A_K \cap X : N(p) \subset X$.*

Množina se slupkou je tedy taková, v níž jsou ke každému odkrytému poli obsažena i všechna jeho sousední pole, např. množina vyznačená na obr. 2.2.

Tvrzení 1. *Množiny, které mají slupku jsou uzavřené na průnik a sjednocení.*

Důkaz. Ukážeme, že tvrzení platí pro průnik, pro sjednocení by byl důkaz analogický:
Mějme množiny E a F, které mají slupku, tedy pro ně platí $\forall p \in A_K \cap E : \mathrm{N}(p) \subset E$ a $\forall p \in A_K \cap F : \mathrm{N}(p) \subset F$. Chceme ukázat, že platí $\forall p \in A_K \cap (E \cap F) : \mathrm{N}(p) \subset (E \cap F)$. To je ale zřejmé, jelikož pokud je vrchol $p \in A_K$ v E i v F, jsou v každé z nich podle předpokladu i jeho sousedé, tedy jsou i v jejich průniku. □

Definice 5. Ohodnocením na množině X*, kde X má slupku, nazveme funkci $m : (A_U \cap X) \to \{0, 1\}$.*

Konzistentním ohodnocením na množině X *nazveme ohodnocení m na množině X, splňuje-li podmínku*

$$\forall p \in A_K \cap X : |p| = |A_M \cap N(p)| + \sum_{q \in A_U \cap N(p)} m(q)$$

(Konzistentním) ohodnocením plochy *nazveme (konzistentní) ohodnocení na množině X, je-li $X = A_M \cup A_U \cup A_K$ (neboli celá hrací plocha).*

Tvrzení 2. *Hrací plocha A je konzistentní, právě když pro ni existuje konzistentní ohodnocení plochy.*

Důkaz. Hrací plocha je konzistentní, právě když existuje přípustná dohraná plocha, která je jejím rozšířením.

Nejdříve směr zprava doleva:

Mějme konzistentní ohodnocení plochy m. Potom nechť tedy plocha B vznikne z A tak, že všem polím $p \in A_U$ (tedy těm se symbolem $\sharp$) dáme v závislosti na ohodnocení m hodnotu M (pokud $m(p) = 1$), nebo $\diamond$ (je-li $m(p) = 0$). Ostatní pole budou stejná jako v A.

Musíme ověřit následující:

B **je dohraná:** To je zřejmé, jelikož jsme do ní z A přenesli jen pole s hodnotou různou od $\sharp$ a na místa polí s $\sharp$ jsme dali buď M, nebo $\diamond$.

B **je rozšířením** A**:** Všechna pole různá od $\sharp$ byla do B přenesena beze změny a na polích $\sharp$ podle definice nezáleží.

B **je přípustná:** Jelikož je B dohraná, nejsou na ní žádná pole s hodnotou $\sharp$. Tedy aby byla splněna přípustnost, musí platit $\forall p \in B_K : |p| = |B_M \cap \mathrm{N}(p)|$. Toto ale zřejmě platí z podmínky kladené na funkci m: $|A_M \cap \mathrm{N}(p)|$ min už jsme přenesli přímo z A a zbytek jsme přidali z A_U, jelikož

$$\sum_{q \in A_U \cap \mathrm{N}(p)} m(q) = |p| - |A_M \cap \mathrm{N}(p)|$$

Směr zleva doprava dokážeme jednoduchým obrácením. □

Definice 6. *Řekneme, že se dvě funkce f_1 a f_2 shodují na množině S, je-li S podmnožinou definičního oboru každé z nich a $\forall x \in S : f_1(x) = f_2(x)$.*

Dále řekneme, že se dvě funkce maximálně shodují, pokud se shodují na průniku svých definičních oborů.

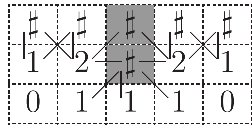
Pozorování 1. *Mějme konzistentní ohodnocení m na množině X a $Y \subset X$, která má slupku. Potom je ohodnocení m' na množině Y, které vzniklo omezením m na tuto množinu, také konzistentní.*

Důkaz. Plyne ihned z definice. □

Tvrzení 3. *Mějme konzistentní ohodnocení m_1 a m_2 na množinách X_1 a X_2, která se maximálně shodují. Potom ohodnocení m_0 na množině $X_1 \cup X_2$ definované následovně:*

$$m_0(v) = \begin{cases} m_1(v) & \textit{pokud } v \in X_1 \\ m_2(v) & \textit{jinak} \end{cases}$$

je také konzistentní.



Obrázek 2.3: Příklad rozhraní

Důkaz. Chceme dokázat, že platí

$$\forall p \in A_K \cap (X_1 \cup X_2) : |p| = |A_M \cap \mathrm{N}(p)| + \sum_{q \in A_U \cap \mathrm{N}(p)} m(q)$$

Toto lze rozepsat do následujících dvou rovností:

$$\forall p \in A_K \cap X_1 : |p| = |A_M \cap \mathrm{N}(p)| + \sum_{q \in A_U \cap \mathrm{N}(p)} m_0(q)$$

$$\forall p \in A_K \cap X_2 : |p| = |A_M \cap \mathrm{N}(p)| + \sum_{q \in A_U \cap \mathrm{N}(p)} m_0(q)$$

První plyne ihned z konzistence m_1 a druhé z konzistence m_2, jakmile si uvědomíme, že m_1 a m_2 se na $X_1 \cap X_2$ shodují. □

2.3 Rozhraní na ploše

Definice 7. *Mějme hrací plochu A a její graf $G(V, E)$. Množinu $J \subset (A_U \cup A_M)$ budeme nazývat* rozhraním na ploše A*, je-li vrcholovým řezem grafu G.*

Příklad rozhraní je na obr. 2.3. Tato naše *rozhraní* se od separátorů v [1] liší tím, že separátor se může skládat z libovolných vrcholů grafu, zatímco rozhraní neobsahuje vrcholy z množiny A_K. Díky tomu nám pomáhá získávat množiny se slupkou:

Tvrzení 4. *Mějme hrací plochu A, její graf $G(V, E)$ a množinu J, která je rozhraním na ploše A. Označme X_1 a X_2 komponenty grafu $G(V \setminus J, E)$. Potom množiny $X_1 \cup J$ a $X_2 \cup J$ mají slupku.*

Důkaz. Sporem pro $X_1 \cup J$:
Mějme $v \in A_K \cap X_1$ (v J být vrchol z A_K nemůže), jehož některý soused není obsažen v $X_1 \cup J$. Potom by byl ovšem v X_2, ale to je ve sporu s tím, že J je řez. □

Kapitola 3

Obecný rozklad grafu, stromová šířka grafu

Definice 8. *Mějme graf $G(V, E)$, potom jeho* obecným rozkladem[1] *je graf $H(X, T)$ takový, že platí:*

- $\bigcup\limits_{X_i \in X} X_i = V$
- $\forall (u, v) \in E : \exists X_i \in X : u \in X_i \wedge v \in X_i$
- $\forall v \in V : \{X_i \in X | v \in X_i\}$ *tvoří souvislý podgraf H*

Vrcholům grafu H říkáme uzly*.*

Šířkou rozkladu *H nazveme číslo $W(H) = \max\limits_{X_i \in X} |X_i| - 1$.*[2]

Je-li obecný rozklad H stromem, říkáme mu stromový rozklad*. Stromovou šířkou grafu G je potom $tw(G) = \min\limits_{H \in D} W(H)$, kde D je množina všech stromových rozkladů grafu G.*

3.1 Rozklady rozkladů

V této části ukážeme, že máme-li obecný rozklad H grafu G a obecný rozklad M rozkladu H, existuje rozklad grafu G, který je s M izomorfní, a navíc má

[1] Přívlastek „obecný“ používáme, abychom tuto strukturu odlišili od rozkladů bez přívlastku, což jsou systémy navzájem disjunktních množin, a zároveň abychom naznačili podobnost se stromovými rozklady, které jsou podmnožinou obecných rozkladů.

[2] Jedničku odečítáme, aby měl nejmenší možný obecný rozklad grafu s alespoň jednou hranou šířku 1.

šířku omezenou součinem šířek M a H. Stručně řečeno tedy to, že rozklady lze skládat. Nejdříve ale pomocné tvrzení, které říká, že rozklady zachovávají spojitost podgrafu:

Tvrzení 5. *Mějme graf $G(V, E)$ a jeho obecný rozklad $H(X, T)$. Dále mějme $U \subset V$ souvislou podmnožinu vrcholů grafu G. Potom je podmnožina $W \subset X$ souvislá, platí-li následující:*

$$\forall X_i \in X : X_i \in W \leftrightarrow \exists v \in U : v \in X_i$$

Důkaz. Sporem: nechť je W nesouvislá. Potom existují uzly $X_0, X_n \in W$, mezi kterými ve W nevede cesta. Z předpokladu v každém z nich musí existovat alespoň jeden vrchol z U, nechť tedy $u_0 \in X_0$ a $u_n \in X_n$. Jelikož je ovšem U spojité, existuje v něm cesta z u_0 do u_n, označme vrcholy na ní postupně u_1 až u_{n-1}. u_k nechť je poslední vrchol, z jehož nadřazených vrcholů $K = \{X_i \in X | u_k \in X_i\}$ nevede cesta do X_n (nadřazené vrcholu u_n). u_k musí existovat, přinejhorším je to u_0.

Dále víme, že jelikož mezi u_k a u_{k+1} vede hrana, existuje $X_m \in W$ obsahující oba tyto vrcholy. Z X_m tedy cesta do X_n vede, ale zároveň víme, že z vrcholů v K nevede. To je spor, jelikož $u_k \in X_m$ a tedy $X_m \in K$. □

Značení 3. *V následujícím textu budeme psát zkráceně $\bigcup M$ namísto $\bigcup\limits_{x \in M} x$.*

A nyní vlastní tvrzení o skládání rozkladů:

Tvrzení 6. *Mějme graf $G(V, E)$ a jeho obecný rozklad $H(\mathcal{X}, T)$. Dále mějme obecný rozklad grafu $H(\mathcal{X}, T)$, který si označme $M(\{O_1, \dots O_n\}, P)$. Potom $Z(\mathcal{Y}, U)$ takové, že*

$$\forall i \in \{1, \dots, n\} : Y_i = \bigcup O_i$$

$$\mathcal{Y} = \{Y_1, \dots, Y_N\}$$

$$(Y_i, Y_j) \in U \Leftrightarrow (O_i, O_j) \in P$$

je obecným rozkladem grafu $G(V, E)$ izomorfním s rozkladem $M(O, P)$.

Dále platí, že $W(Z) \leq (W(H) + 1).(W(M) + 1) - 1$.

Důkaz. Dokážeme, že Z je obecný rozklad grafu $G(V, E)$. To, že je izomorfní s $M(O, P)$ vyplyne přímo ze způsobu jeho konstrukce. Potřebujeme dokázat tři věci:

- $\bigcup \mathcal{Y} = V$:
  Každý vrchol $v \in V$ leží v nějakém uzlu $X_i \in \mathcal{X}$ a každý uzel X_i leží v nějakém „naduzlu" $O_j \in O$. Jelikož Y_j vznikly jako sjednocení X_i z jednotlivých O_j, pro každé v alespoň jedna Y_j existuje.

- $\forall (u,v) \in E : \exists Y_i \in \mathcal{Y} : u \in Y_i \vee v \in Y_i$:
  $\forall (u,v) \in E : \exists X_i \in \mathcal{X} : u \in X_i \vee v \in X_i$. A každý uzel X_i je obsažena alespoň v jednom uzlu Y_i.

- $\forall v \in V : \{Y_i \in \mathcal{Y} | v \in Y_i\}$ tvoří souvislý podgraf H:
  Nechť $Q = \{X_i \in \mathcal{X} | v \in X_i\}$. Z definice obecného rozkladu je Q v H souvislá. Z tvrzení 5 je $R = \{O_j \in O | \exists X_i \in Q : X_i \in O_j\}$ souvislá v M. Ze způsobu konstrukce $\mathcal{Y}$ potom vyplývá, že v se nevyskytne jinde, než v Y_i izomorfních k prvkům R, které spolu také (díky izomorfismu) v Z tvoří souvislou množinu.

Pro horní odhad šířky obecného rozkladu Z chceme ukázat, že horní odhad velikosti Y_i je $(\mathrm{W}(H)+1).(\mathrm{W}(M)+1)$.
Y_i vznikají jako sjednocení uzlů X_i z jednotlivých „naduzlů" O_j. Horní odhad velikosti X_i je z definice $WH+1$ a horní odhad pro velikost O_j je $\mathrm{tw}(H)+1$, z čehož získáváme výše uvedené.

□

Značení 4. *$Z(Y,U)$ z předchozího tvrzení budeme nazývat složením rozkladů $H(X,T)$ a $M(O,P)$ a značit*

$$Z(Y,U) = M(O,P) \circ H(X,T)$$

Důsledek 2. *Z předchozího tvrzení také přímo vyplývá, že máme-li graf $G(V,E)$ a jeho obecný rozklad $H(X,T)$, platí*

$$tw(G) \leq (W(H)+1).(tw(H)+1)-1$$

3.2 Rozklady a řezy

V předchozí části jsme rozebírali skládání rozkladů, což se nám bude hodit k tomu, abychom ke grafu plochy získali jeho stromový rozklad. Následující tvrzení nám poslouží k tomu, abychom na základě stromového rozkladu byli schopni efektivně rozhodnout o konzistentnosti hrací plochy.

Tvrzení 7. *Mějme graf* $G(V, E)$ *a jeho obecný rozklad* $H(X, T)$*. Potom je-li* $D \subset X$ *vrcholový řez rozdělující* X *na dvě nesouvislé části* X_a *a* X_b *a platí-li* $\bigcup X_a \not\subset \bigcup D$ *a* $\bigcup X_b \not\subset \bigcup D$*, je* $C = \bigcup D$ *vrcholový řez v grafu* G*.*

Důkaz. Sporem — nechť C není vrcholový řez. Označme si $V_a = \bigcup X_a \setminus \bigcup D$ a $V_b = \bigcup X_b \setminus \bigcup D$. Tyto dvě množiny jsou z předpokladů neprázdné. Z předpokladu sporu existují vrcholy $a_0 \in V_a$ a $a_n \in V_b$, mezi nimiž existuje v grafu $G'(V \setminus C, E)$ cesta. Vrcholy na ní označme $a_1, \ldots, a_{n-1}$. Dále dozajista existují $X_1 \in X_a, X_2 \in X_b$ takové, že $a_0 \in X_1$ a $a_n \in X_2$. Podle tvrzení 5 potom ovšem mezi vrcholy X_1 a X_2 vede v $H'(X \setminus D, T)$ také cesta. To je ale ve sporu s tím, že D je vrcholový řez H. □

Kapitola 4

Graf plochy a jeho rozklady

V této kapitole využijeme obecných rozkladů grafu na grafech hracích ploch k tomu, abychom získali jejich stromový rozklad s omezenou šířkou, popř. dospěli k tomu, že graf plochy takovýto stromový rozklad nemá.

Nejdříve ukážeme postup, který nám dá pro bipartitní graf jeho obecný rozklad s jistými vlastnostmi:

Tvrzení 8. *Mějme bipartitní graf $G(V,E)$ bez izolovaných vrcholů, $V = V_1 \cup V_2$, kde V_1 a V_2 jsou jeho partity. Potom $H(X,T)$, pro které platí*

$$X = \{\{v\} \cup N(v) | v \in V_1\}$$

$$(X_i, X_j) \in T \Leftrightarrow X_i \cap X_j \neq \emptyset$$

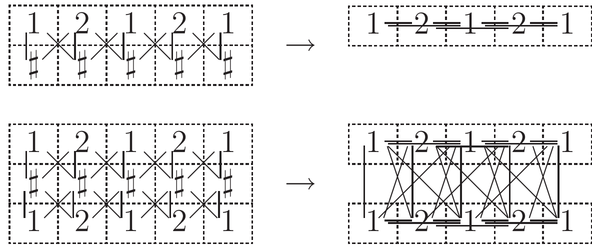
je obecným rozkladem grafu G a pro každý vrcholový řez rozkladu $C \subset X$ je $\bigcup C \cap V_2$ vrcholovým řezem grafu G.

Důkaz. Nejdříve dokažme, že $H(X,T)$ je obecný rozklad grafu G:

- $\bigcup X = V$:
  Každý z vrcholů V_1 je z definice v některém z X_i a jelikož je graf bez izolovaných vrcholů, každý z vrcholů V_2 je sousedem alespoň jednoho vrcholu z V_1 a tedy patří do jeho X_i.

- $\forall (u,v) \in E : \exists X_i \in X : u \in X_i \vee v \in X_i$:
  BÚNO nechť $u \in V_1$, potom v je jeho sousedem a tak jsou spolu v X_i, v němž je u.

- $\forall v \in V : \{X_i \in X | v \in X_i\}$ tvoří souvislý podgraf H:
  Z druhé podmínky pro $H(X,T)$ spolu tvoří nejen souvislý, ale dokonce úplný podgraf.



Obrázek 4.1: Grafy ploch a jejich bipartitní rozklady

Nyní k tomu, že $\bigcup C \cap V_2$ je vrcholovým řezem grafu G:
Z tvrzení 7 víme, že vrcholovým řezem G je $\bigcup C$. Pokud z množiny $\bigcup C$ ovšem odebereme vrcholy V_1, řezem bude stále, jelikož neodebrané vrcholy z V_1 budou izolované. $\bigcup C \cap V_2$ totiž z definice použitého rozkladu obsahuje všechny jejich sousedy. □

Definice 9. *Rozklad $H(X,T)$ z předchozího tvrzení budeme nazývat* bipartitním rozkladem podle V_1. *V každém uzlu X_i budeme vrchol $v \in X_i \cap V_1$ nazývat* řídícím vrcholem *uzlu.*

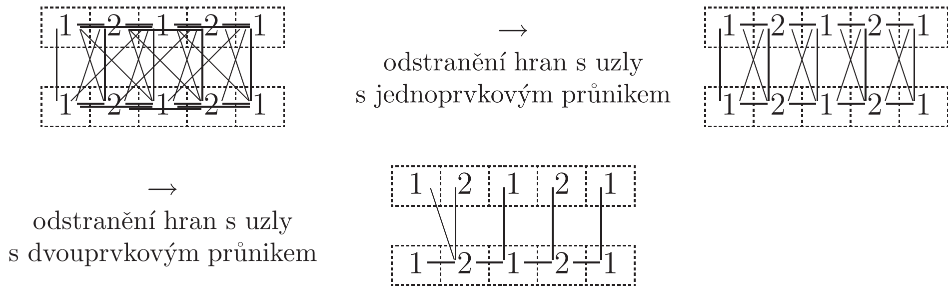
Důsledek 3. *Z definice množiny X v tvrzení 8 plyne, že šířka bipartitního rozkladu podle U je rovna nejvyššímu stupni ze všech vrcholů v U.*

Důsledek 4. *Dále z definice bipartitního rozkladu a její aplikace na graf plochy (tak, že získáme bipartitní rozklad podle A_K) je každý uzel bipartitního rozkladu množinou se slupkou. A jelikož jsou množiny se slupkou uzavřené na sjednocení, platí tato vlastnost i pro rozklady vzniklé skládáním bip. rozkladu s jinými.*

Při prvním pohledu na obr. 4.1 vidíme, že bipartitní rozklad má příliš mnoho hran. Nicméně hned při druhém je vidět, že mnohé z těchto hran je možné hned odstranit, aniž by náš rozklad přestal být obecným rozkladem. Na základě tohoto pozorování získáváme algoritmus, který nás přibližuje ke stromovému rozkladu:

4.1 Odstranění redundantních hran

Značení 5. Základem hrany *bipartitního rozkladu budeme nazývat průnik koncových uzlů této hrany.* Velikost hrany *bipartitního rozkladu bude potom počet prvků jejího základu.*



Obrázek 4.2: Odstranění redundantních hran

Máme-li obecný rozklad a odebereme-li z něj hranu, jediný důvod, proč by výsledek této operace neměl být nadále obecným rozkladem, může být porušení třetí podmínky pro obecné rozklady — tedy, že uzly obsahující společný vrchol musí indukovat souvislý podgraf. Nabízí se, že hrany s větší velikostí jsou pro nás ,,při uspokojování třetí podmínky“ cennější než ty s velikostí menší, jelikož pomáhají v souvislosti více podgrafům. Při implementaci se tento předpoklad ukázal jako správný.

Algoritmus na odstranění redundantních hran postupuje následovně:

function OdstraňRedundantníHrany(Bipartitní rozklad $T(X, W)$)
    hrany $\leftarrow$ seznam hran z W seřazený podle jejich velikosti. Hrany stejné velikosti jsou řazeny postupně podle jejich horní, levé, dolní a pravé souřadnice a směrnice.
    for all $(u, v) \in hrany$ **do**
        if $\exists$ cesta $u, w_1, \ldots, w_n, v$ v $T(X, W \setminus \{(u, v)\})$ taková, že $\forall i \in \{1, \ldots, n\} : u \cap v \subset w_i$ **then**
            Odstraň (u, v) z W a z *hrany*
        end if
    end for
    return rozklad T s modifikovaným W
end function

Detailní podmínky pro uspořádání hran (podle souřadnic a směrnic) nemají jiný význam, než aby byl výsledek běhu algoritmu určen jednoznačně.

Tvrzení 9. *Algoritmus na odstranění redundantních hran vrací obecný rozklad.*

Důkaz. Jediné, co algoritmus mění na rozkladu, který obdržel jako parametr,

je množina jeho hran. Ze tří podmínek pro rozklad tedy může porušit jen tu poslední:
Každá množina všech uzlů obsahujících společný vrchol indukuje souvislý podgraf.
Ukážeme tedy, že tuto podmínku neporušíme žádným odstraněním hrany a platí tedy i pro rozklad, který algoritmus vrací:

Souvislý podgraf S je takový, v němž mezi každými dvěma vrcholy vede v S cesta. Vedla-li v původním $T(X, W)$ taková cesta přes hranu (u, v), která byla odstraněna, existuje jiná cesta, v posloupnost vrcholů u, v nahradíme posloupností vrcholů $u, w_1, \ldots, w_n, v$. Jelikož platí $\forall i \in \{1, \ldots, n\} : u \cap v \subset w_i$, patří všechny vrcholy v této posloupnosti také do S. $\square$

Po proběhnutí algoritmu odstraňujícího redundantní hrany nám při reálné hře v obtížnosti Expert už velmi často zůstává jen stromový rozklad, nebo rozklad oproti původnímu bipartitnímu rozkladu výrazně zjednodušený.

Dostaneme-li stromový rozklad, můžeme jej přímo předat níže popsanému algoritmu pro rozhodnutí o konzistentnosti hrací plochy.

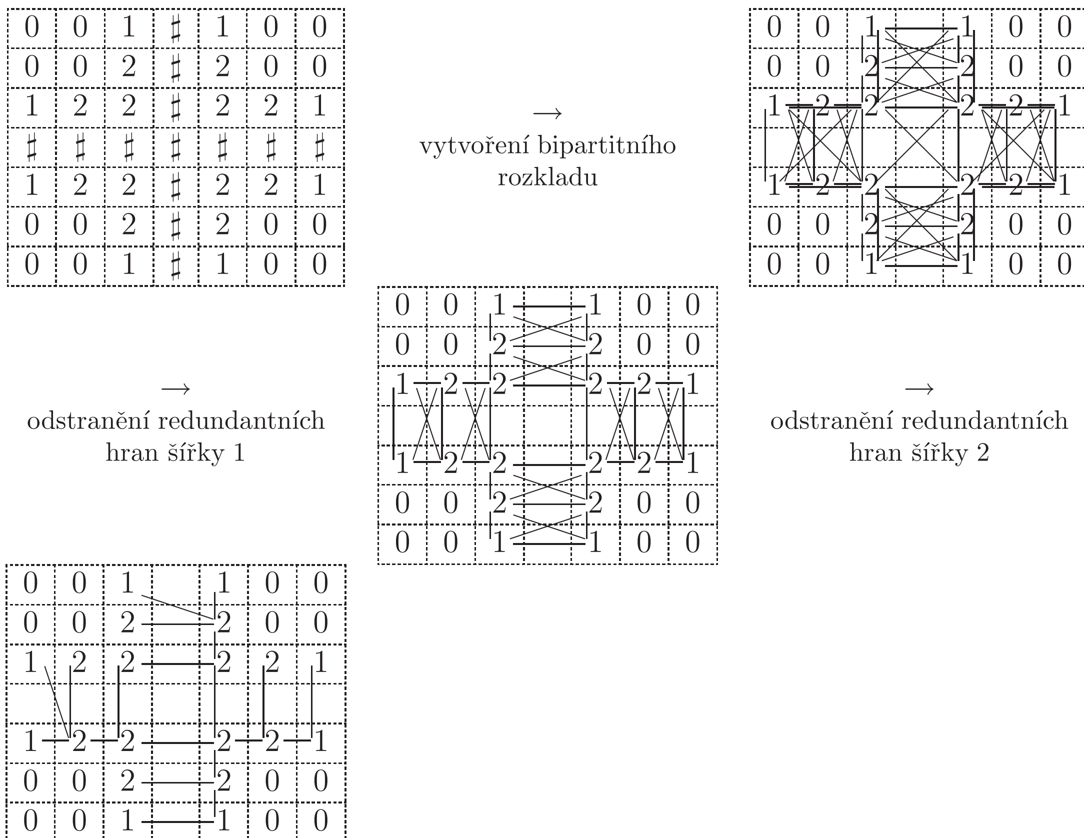
4.2 Sériově-paralelní grafy

Pokud jsme stromový rozklad nedostali odstraněním redundantních hran, stále ještě se můžeme pokusit jej získat. Rozdělíme získaný graf na 2-souvislé komponenty a budeme hledat jejich stromový rozklad s omezenou šířkou. Tříd grafů s omezenou stromovou šířkou je dozajista velmi mnoho, my se však omezíme na sériově-paralelní grafy, jejichž definice i důkaz o jejich omezené stromové šířce je v [1], str. 13–14. Praktické zkušenosti se hrou Minolovka, v níž je implementována vizualizace bipartitního rozkladu plochy s odstraněnými redundantními hranami, zároveň ukazují, že sériově-paralelní podoba rozkladu je v reálných situacích hry velmi častá.

V [3], str. 4–5 je prezentován algoritmus, který o orientovaném grafu rozhodne, zda je sériově-paralelní, a pokud ano, poskytne nám jeho *konstrukční strom*[1] V [1] je potom postup, jak z konstrukčního stromu sériově paralelního grafu získat jeho stromový rozklad s omezenou šířkou.

Nám tedy zbývá obdržené 2-souvislé komponenty zorientovat tak, abychom z neorientovaného sériově-paralelního grafu vždy získali orientovaný sériově-paralelní graf. To učiní algoritmus ZorientujSPGraf, který jako

[1] V originále construction tree.



Obrázek 4.3: Postup při zjednodušování složitější plochy

vstupní parametry požaduje mimo samotného grafu i zadání vstupního a výstupního vrcholu. Toto však není problém, jelikož i pokud zavoláme ZorientujSPGraf na všechny dvojice vrcholů, zůstává algoritmus stále polynomiálním. My potom vybereme tu dvojici, pro niž bylo zorientování grafu nalezeno a algoritmem z [3] potvrzeno. Pokud taková dvojice neexistuje, graf není sériově-paralelní.

```
procedure ZorientujSPGraf(var Graf G(V, E), Vrchol s, Vrchol t)
    if s = t then
        return
    end if
    c ← nejkratší cesta z s do t po dosud nezorientovaných hranách.
    Zorientuj hrany na c po směru cesty.
    for all U ∈ komponenty G(V, E′), kde E′ jsou dosud nezorientované hrany do
        s′ ← vrchol U nejbližší vrcholu s na cestě c
        t′ ← vrchol U nejbližší vrcholu t na cestě c
        ZorientujSPGraf(G(U, E),s′, t′)
    end for
end procedure
```

Tvrzení 10. *Pokud je graf $G(V, E)$ sériově-paralelním se zdrojem s a stokem t, algoritmus* ZorientujSPGraf *jej správně zorientuje.*

Myšlenka důkazu:

Nejkratší cesta od zdroje ke stoku je dozajista správně zorientována. Kdyby šla cesta proti směru hran orientovaného SP grafu, musela by se do tohoto opačného směru v některém vrcholu otočit, ale tímtéž vrcholem by potom cesta musela projít ještě jednou ve správném směru, takže by nebyla nejkratší.

Po odstranění této cesty nám zjevně zůstanou komponenty, které jsou samy o sobě také SP grafy a tak na ně algoritmus zavoláme rekurzivně. Při tomto volání musíme také správně zvolit zdroj a stok těchto podgrafů, ale ty jsou zřejmé — místa, kde odstraněná cesta do podgrafu vstoupila a z něj zase vystoupila.

Kapitola 5

Algoritmus pro rozhodnutí o konzistentnosti plochy

V předchozích kapitolách jsme směřovali k tomu, abychom získali stromový rozklad grafu plochy. Tento rozklad nyní využijeme v algoritmu rozhodujícím o konzistentnosti hrací plochy:

Vstupem algoritmu je hrací plocha a její stromový rozklad $H(X,T)$ šířky k, pro který platí $(X_1, X_2) \in T \Rightarrow X_1 \cap X_2 \subset A_U \cup A_M$[1].

K řešení využijeme pomocnou funkci OHODNOŤKOŘEN volající samu sebe rekurzivně na principu průchodu do hloubky. Ta pro zakořeněný podstrom stromového rozkladu $H(X,T)$ vrátí množinu ohodnocení na množině vrcholů v kořenovém uzlu. Níže ukážeme, že právě vrácená ohodnocení jsou konzistentní a lze je rozšířit i na vrcholy obsažené ve zbylém podstromu tak, že konzistentní zůstanou.

Algoritmus *rozhodnutí o konzistentnosti plochy* zavolá OHODNOŤKOŘEN na celý strom $H(X,T)$ s libovolným vrcholem jako kořenem. Pokud dostane prázdnou množinu ohodnocení, odpoví NE, jinak odpoví ANO.

Nyní vlastní funkce OHODNOŤKOŘEN:

[1] Tedy, že žádné dva uzly se neprotínají ve známých vrcholech.

```
function OhodnoťKořen(H(X' ⊂ X, T), X_0)
    res ← množina všech konzistentních ohodnocení na množině X_0
    for all X_1 ∈ množina potomků X_0 ve stromě H(X', T) do
        Sols_{X_1} ← OhodnoťKořen(podstrom s kořenem X_1, X_1)
        res ← KompatibilníOhodnocení(res, Sols_{X_1})
    end for
    return res
end function
```

Pomocná funkce KompatibilníOhodnocení vrátí svůj první parametr bez těch ohodnocení, která nejsou maximálně slučitelná s žádným ohodnocením z parametru druhého. Zápis v pseudokódu následuje:

```
function KompatibilníOhodnocení(množina ohodnocení T na
množině X_0, množina ohodnocení S na množině X_1)
    res ← ∅
    for all a ∈ T do
        if ∃b ∈ S : a, b se maximálně shodují then
            res ← res ∪ {a}
        end if
    end for
    return res
end function
```

Tvrzení 11. *Funkce* OhodnoťKořen *na stromovém rozkladu* $H(X', T)$ *s šířkou omezenou konstantou skončí v čase* $O(|X'|)$.

Důkaz. Jelikož má $H(X', T)$ šířku nejvýše k, je i velikost množiny všech ohodnocení na množině X_0 omezena konstantou. A jelikož návratová hodnota této funkce je její podmnožinou, je i její velikost omezena. Doba běhu funkce KompatibilníOhodnocení je tedy také omezena konstantou.

Ze způsobu rekurzivního volání je zřejmé, že pro každý uzel stromu $H(X', T)$ je funkce OhodnoťKořen volána právě jednou. Pro každý uzel je vygenerována množina všech ohodnocení (z nichž každé je otestováno na konzistenci) a pro každý vrchol kromě X_0 je tato množina sloučena funkcí KompatibilníOhodnocení s množinou ohodnocení rodiče. Jelikož jsou všechny tyto operace prováděny v konstantním čase, běží celá funkce v čase $O(|X'|)$. □

Důsledek 5. *Algoritmus* rozhodnutí o konzistentnosti plochy *běží na stromu* $H(X, T)$ *taktéž v čase* $O(|X|)$, *jelikož pouze zavolá funkci* OhodnoťKořen *a porovná vrácenou hodnotu s prázdnou množinou.*

Tvrzení 12. *Funkce* OHODNOŤKOŘEN *vrátí pro podstrom $H(X', T)$ stromového rozkladu $H(X, T)$ množinu ohodnocení na kořenu takovou, že právě ohodnocení v ní obsažená lze rozšířit i na vrcholy ve zbytku podstromu tak, aby byly konzistentní.*

Důkaz. Označme si kořen jako R a jeho potomky jako $S_1, \ldots, S_n$.

Všechna ohodnocení konzistentní na množině R můžeme vygenerovat snadno — probráním všech možností a vyřazením těch nepřípustných. Právě ta totiž nejsou konzistentní, jelikož nám na R nezbývají už žádná neznámá políčka a tak pojmy konzistence a přípustnost splývají.

Správnost celé funkce ukážeme nyní indukcí podle počtu vrcholů v podstromu, na nějž je funkce volána:
Má-li zadaný podstrom pouze jeden vrchol, jedná se o kořen R. Ve zbytku podstromu tedy žádné vrcholy nejsou a tvrzení platí.

Má-li podstrom $n > 1$ vrcholů, na potomky (i nepřímé) kořene jich připadá $n - 1$. Tedy zavoláme-li funkci OHODNOŤKOŘEN na některý podstrom, s kořenem $S_1, \ldots, S_{n-1}$, nebo S_n, dokazované tvrzení pro něj podle předpokladu indukce platí.

Máme tedy množinu konzistentních ohodnocení O_R na množině R a $E_{S_1}, \ldots, E_{S_n}$ na množinách $S_1, \ldots, S_n$ (ta lze rozšířit na jim odpovídající podstromy).

Chceme, aby platilo, že ohodnocení z O_R lze konzistentně rozšířit na celé X', právě když v každé z množin $E_{S_1}, \ldots, E_{S_n}$ existuje ohodnocení, které se s ním maximálně shoduje.

Nejdříve směr zleva doprava:
Mějme ohodnocení $m \in O_R$ a jeho konzistentní rozšíření m'. Potom se ohodnocení k vzniklé omezením m' na S_1 dozajista nachází v E_{S_1}, jelikož má rozšíření na celý podstrom pod S_1 (příslušné omezení ohodnocení m'). Obdobně platí i pro $S_2, \ldots, S_n$.

A zprava doleva:
Máme ohodnocení $m \in O_R$ a $k_1 \in E_{S_1}, \ldots, k_n \in E_{S_n}$, která se s ním maximálně shodují. Označme $k'_1, \ldots, k'_n$ rozšíření $k_1, \ldots, k_n$ na celé příslušné podstromy. Z toho, že se jedná o stromový rozklad, plyne, že definiční obory každého z $k'_1, \ldots, k'_n$ se mohou protínat pouze v R.

Jelikož se ohodnocení $m, k'_1, \ldots, k'_n$ po dvou maximálně shodují, lze je podle tvrzení 3 sloučit a získáme tak ohodnocení na celém X'. □

Tvrzení 13. *Algoritmu* rozhodnutí o konzistentnosti plochy *vrátí ANO, je-li zadaná plocha konzistentní a NE, pokud tomu tak není.*

Důkaz. Funkce OHODNOŤKOŘEN nám pro zadaný uzel vrátí množinu všech jeho ohodnocení, která lze rozšířit na konzistentní ohodnocení celé plochy.

Je zřejmé, že pokud plocha konzistentní ohodnocení nemá, bude tato množina prázdná i to, že má-li nějaká, dostaneme jejich omezení na vrcholy v daném uzlu. □

Kapitola 6

Závěr

V této práci jsme ukázali, že ač je problém konzistentnosti hrací plochy ve hře Minolovka ve své obecnosti NP-úplný, omezíme-li se jen na určitou třídu konfigurací, lze tento problém řešit v polynomiálním čase.

Nejdříve jsme problém konzistentnosti hrací plochy zformalizovali a nalezli jsme problém z teorie grafů, na který jej umíme převést v polynomiálním čase. Tento problém lze, dostaneme-li stromový rozklad grafu s šířkou, jež je omezena konstantou, řešit v konstantním čase a algoritmus, který toto provádí je prezentován v kapitole ,,Algoritmus pro rozhodnutí o konzistentnosti plochy."

Hlavním úkolem práce je tedy určit, pro které situace hrací plochy jsme schopni zkonstruovat stromový rozklad jejího grafu, který má šířku omezenu konstantou.

K tomu jsme si nejdříve zadefinovali strukturu ,,obecný rozklad," která je zobecněním stromového rozkladu a kterou lze zkonstruovat snadno. Takto zkonstruovanou strukturu se následně snažíme heuristickým algoritmem co nejvíce zjednodušit. Toto zjednodušování může dopadnout dvěma způsoby:

- a) zjednodušování je úspěšné a získáme buď strom, nebo graf, jehož stromový rozklad omezené šířky umíme zkonstruovat.[1]
- b) ani po zjednodušení nejsme schopni z grafu získat stromový rozklad s omezenou šířkou.

[1] V našem případě toto implementujeme pro grafy sériově-paralelní, které mají stromovou šířku 2.

		1	1	2	1	
	2	2	0	0	1	
	1	0	0	0	1	
	2	1	1	2	2	
		1	1			

Obrázek 6.1: Příklad plochy s hladkou hranicí

Z [2], kde je ukázáno, že problém rozhodnutí o konzistentnosti plochy je NP-úplný, plyne, že případ (b) nastávat musí.[2]

V naší implementaci, nalezneme-li stromový rozklad grafu plochy, má tento šířku nejvýše 26. Bipartitní rozklad, který získáme z grafu plochy má šířku nejvýše 8,[3] a sériově-paralelní grafy mají šířku nejvýše dva. Podle tvrzení 6 potom získáváme omezení 26 pro šířku složení těchto dvou rozkladů. Při hře v obtížnosti Expert[4] se následně ukazuje, že je stromový rozklad s takto omezenou šířkou pro většinu pozic nalezen.

Dále, i když to není v práci formálně dokázáno, se dá předpokládat, že algoritmus nalezne takto omezený stromový rozklad pro všechny plochy, na níž mají oblasti odkrytých polí hladkou hranici a navzájem se neovlivňují, viz např. obr. 6.1. Hladkou hranicí rozumíme to, že lze odkrytá políčka seřadit do cest či cyklů tak, že odkrytí sousedé každého neodkrytého políčka tvoří souvislý podgraf.

Zmiňovaná implementace je založena na dříve řešeném ročníkovém projektu a je včetně zdrojových kódů v Microsoft Visual Studiu 2005 k dispozici na přiloženém CD a na adrese
`http://www.ms.mff.cuni.cz/~hodek4am/rp.zip`

[2]Za předpokladu, že NP$\neq$P.
[3]Každé políčko má nejvýše osm sousedů.
[4]30x16 polí, 99 min

Literatura

[1] Fiala, J. (2007): *Introduction to graph minors and treewidth.* `http://kam.mff.cuni.cz/~fiala/tw.pdf`

[2] Kaye, R. (2000): *Minesweeper is NP-complete*, Math. Intell. 22 (2) 9–15.

[3] Valdes, J., Tarjan, R. E., Lawler, E. L. (1979): *The recognition of Series Parallel digraphs*, Proceedings of the eleventh annual ACM symposium on Theory of computing 1–12.