Univerzita Komenského, Bratislava
Fakulta Matematiky, Fyziky a Informatiky

Konštrukcia grafov pomocou mobilných agentov

Diplomová práca

2014

Bc. Jakub Kováč

Univerzita Komenského, Bratislava
Fakulta Matematiky, Fyziky a Informatiky

Konštrukcia grafov pomocou mobilných agentov

Diplomová práca

Študijný program:	Informatika
Študijný odbor:	2508 Informatika
Školiace pracovisko:	Katedra Informatiky
Školiteľ:	prof. RNDr. Rastislav Kráľovič, PhD.

Bratislava, 2014

Bc. Jakub Kováč

Obsah

Zoznam obrázkov

Zoznam tabuliek

Kapitola 1

Úvod

Graf je vhodnou reprezentáciu údajov potrebných na riešenie mnohých problémov. V tomto smere našiel široké uplatnenie aj v informatike a oblastiach matematiky, ktoré sú s ňou úzko späté. My sa budeme zaoberať pojmom graf tak, ako je chápaný v Teórii grafov. V našom ponímaní je graf abstraktnou štruktúrou obsahujúcou vrcholy a hrany. Pričom každá hrana má práve dva konce - vrcholy, ktoré spája. Ak rozlišujeme počiatočný a koncový vrchol pri hranách grafu, hovoríme, že hrany sú orientované. Ak sú počiatočný a koncový vrchol hrany totožné, hovoríme, že táto hrana je slučka. Hrana vedie medzi počiatočným a koncovým vrcholom. Ka medzi niektorou dvojicou vrcholov vedie viacero hrán, hovoríme, že tieto hrany sú násobné. V rámci Teórie grafov sa skúmajú grafy z rôznych pohľadov a riešia sa na nich najrôznejšie úlohy. My sa budeme zaoberať grafmi bez slučiek, orientovaných a násobných hrán. Grafy boli na základe vybraných vlastností roztriedené do viacerých tried. Roztriedenie grafov medzi jednotlivé triedy umožnilo pozrieť sa na konkrétne prípady mnohých problémov pre ktoré nebolo známe všeobecné riešenie a dosiahnuť aspoň čiastočné výsledky. Taktiež to umožnilo optimalizovať všeobecné algoritmy a dosiahnuť na špecifickejších vstupoch rádovo lepšiu časovú náročnosť výpočtu.

Medzi významné problémy týkajúce sa grafov patria problémy ich konštrukcie a prehľadávania. V súčasntosti sa problému konštrukcie grafov venuje takmer výlučne teória grafových gramatík. Tento prístup používa podobné prostriedky a postupy ako známejšia teória formálnych jazykov a automatov s tým rozdielom, že grafové gramatiky sa týkajú grafov a sú teda o dosť zložitejšie. Napríklad pojem bezkontextovej grafovej gramatiky má viacero rozdielnych prirodzených interpretácií. V tejto teórii sa vyvinulo viacero prístupov a skúmaných modelov. Ďalším spomínaným problémom týkajúcim sa grafov je problém ich prehľadávania. V rámci teórie prehľadávania grafov sa skúmajú viaceré modely. Väčšina z nich má však spoločný základ. Týmto základom je entita nazývaná agent prípadne robot, automat atď., ktorá sa pohybuje po vrcholoch daného grafu a jeho úlohou je tento graf prehľať. Jednotlivé modely sa môžu líšiť vo viacerých vlastnostiach. Medzi najdôležitejšie z nich patrí vlastnosti agenta i spôsob jeho pohybu po grafe, vlastnosti grafu a definícia pojmu prehľadať. Prehľadanie grafu môže znamenať navštíviť každý vrchol grafu, ale tiež prejsť po každej hrane. Taktiež je dôležité, aké informácie získa agent o svojom okolí pri návšteve niektorého vrchola. Môže napríklad vidieť všetky vrcholy spojené hranou alebo len lokálne čísla portov. Naša práca skúma možnosti konštrukcie grafov pomocou mobilných agentov. Spája teda obe bližšie spomínané teórie. Teórie grafových gramatík sa týka v tom, že sa tiež zaoberá konštrukcio grafov. S prehľadávaním grafu pomocou agentov má spoločného viac. Ide totiž o akýsi duálny problém. Pri prehľadávaní ide často o získanie akejsi mapy neznámeho grafu a v našej práci sa naopak zaoberáme konštrukciou grafu pri znalosti mapy. V prvom prípade teda máme graf a chceme mapu, kdežto v druhom máme mapu a chceme graf. Keďže problém konštrukcie grafov pomocou mobilných agentov nebol dosiaľ skúmaný, zvolili sme si na skúmenie jednoduchý model. Na ňom sme skúmali možnosti konštrukcie grafov z jednotlivých tried. Zvlášť sme sa venovali skúmaniu konštrukcie hyperkocky a tried grafov $P_n \times P_m$ i $C_n \times C_m$. Výsledky na týchto triedach nájdete v hlavných kapitolách práce. Náš model sa skladá z agenta nachádzajúceho sa v počiatočnom vrchole grafu, ktorý ide konštruovať. Tento agent môže vždy vykonať jednu z troch možných operácií. Agent pozná lokálne značenie koncov hrán - portov vo vrchole, v ktorom sa nachádza. Toto značenie je dané poradím ich vzniku a agent nemá možnosť ho meniť. Operácie, ktoré môže agent vykonať, sú: otvorenie portu s danou značkou, prejdenie po hrane s lokálnou značkou portu k i vytvorenie novej hrany do dosiaľ neskonštruovaného vrchola; v ktorom sa agent vzápätí ocitne. K vlastnostiam modelu tiež patrí, že v momente, keď sú otvorené dva porty s rovnakou značkou, vznikne medzi nimi hrana. Krok agenta sme si definovali ako zmenu vrchola, v ktorom sa agent nachádza. Skúmali sme aký minimálny počet krokov agenta je potrebný na konštrukciu daného grafu pri rôzne veľkých množinách značiek, ktoré má agent k dispozícii.

Kapitola 2

Základné pojmy a definície

2.1 Graf

V teórii grafov sa používa viacero rozdielnych definícii grafu, preto v tejto kapitole uvádzame základné definície pojmov použitých v našej práci.

Graf je obrazová reprezentácia údajov obsahujúca body - vrcholy a čiary, ktoré body spájajú - hrany. Hrana je úsek spojitej krivky začínajúci a končiaci v bode. Hrana vedie medzi vrcholmi v ktorých má konce. O vrchole, do/z ktorého hrana vedie a tejto hrane, hovoríme, že sú navzájom incidentné. Niektoré definície povoľujú aby hrana začínala a končila v tom istom vrchole; takúto hranu nazývame slučka. Pokiaľ vedie medzi dvoma vrcholmi viac hrán, hovoríme o násobných hranách. Striktné definície grafu nepovoľujú násobné hrany a slučky a ani my sa nimi nebudeme zaoberať.

Na reprezentáciu vrcholov sa v teórii grafov používajú prvky množiny a hrana je v množine hrán reprezentovaná dvojicou vrcholov, ktoré spája, teda dvojprvkovou podmnožinou množiny vrcholov. Ak rozlišujeme začiatočný a koncový vrchol hrany, hovoríme, že hrana je orientovaná. Takéto hrany potom zvykneme reprezentovať usporiadanou dvojicou vrcholov.

Nasledujú formálne definície dôležitých pojmov z teórie grafov. Ak chceme rozlíšiť, ktorý graf máme na mysli, používame onačenie tohot grafu ako dolný index. Napríklad množina hrán E grafu G sa označuje E_G.

Definícia 1. *Graf G je usporiadaná dvojica (V, E), kde $E \subseteq \{u, v | u \in V, v \in V\}$.*

Množine V hovoríme množina vrcholov a množine E množina hrán.

Vrcholy označujeme malými písmenami prevažne z konca latinskej abecedy (okolo písmena v). Hrany označujeme malými písmenami zo začiatku latinskej abecedy (okolo písmena e) alebo ako dvojicu vrcholov, ktoré spája. Keďže sa zaoberáme neorientovanými grafmi, platí $(u, v) \equiv (v, u)$.

Definícia 2. *Hovoríme, že hrana $e \in E$ je incidentná s vrcholom $v \in V$, ak $\exists u \in V$: $e = (u, v)$*

Definícia 3. *Stupeň vrchola v je počet hrán s ktorými je incidentný; označujeme deg(v). Teda $deg(v) = |\{e|v \in e\}|$* [1]

Definícia 4. *Sled S v grafe G je postupnosť vrcholov a hrán končiaca a začínajúca vrcholom taká, že dva jej susedné prvky sú navzájom incidentné.*

Čiže $S = v_0, e_0, v_1, e_1, ..., e_{n-1}, v_n$; kde $\forall i : v_i \in V, e_i \in E$; $\{v_i, v_{i+1}\} = e_i$.

Definícia 5. *Ťah T v grafe G je taký sled S v tomto grafe, v ktorom sa neopakujú hrany. $T = S : \forall e_i, e_j \in S : e_i \equiv e_j \Rightarrow i = j$*

Definícia 6. *Hovoríme, že ťah alebo sled sú uzavreté, ak je ich posledný vrchol totožný s prvým vrcholom.*

Definícia 7. *Cesta P v grafe G je taký ťah T v tomto grafe, že sa v ňom neopakujú vrcholy.*

Definícia 8. *Cyklus C v grafe G je taký uzavretý ťah T v tomto grafe, že sa v ňom neopakujú vrcholy.*

Definícia 9. *Graf je súvislý, ak v ňom existuje sled obsahujúci všetky vrcholy.*

Definícia 10. *Podgraf H grafu G je taký graf, ktorého množina vrcholov V_H je podmnožinou vrcholov V_G grafu G. Množina hrán grafu H je podmnožinou tých hrán grafu G, ktoré sú incidentné len s vrcholmi z množiny V_H.*

Indukovaný podgraf je taký graf, ktorého množina hrán obsahuje všetky hrany, ktorých koncové vrcholy patria do jeho množiny vrcholov.

Definícia 11. *Most v grafe Gje taká hrana $e = \{u, v\}$ v grafe G, že ľubovoľný sled obsahujúci vrcholy u a v obsahuje hranu e.*

Definícia 12. *Hovoríme, že vrcholy v, u grafu G sú susedné vrcholy, ak $\exists e \in E_G$: $u, v, \in e$.*

Definícia 13. *Komponent K grafu G je taký jeho indukovaný súvislý podgraf, že v gafe G nevedie hrana medzi vrcholom v K a žiadnym vrcholom ležiacim vo zvyšku grafu G.*

Definícia 14. *Kostra súvislého grafu G je taký jeho súvislý podgraf, kde každá hrana je most.*

[1] Keďže sme si hranu definovali ako dvojprvkovú podmnožinu množiny vrcholov, môžeme si dovoliť tento zápis

Definícia 15. *Produkt kartézskeho súčinu grafov G a H označujeme $G \times H$, pričom platí:*

$V_{G\times H} = V_G \times V_H$

$E_{G\times H} = E_G \times V_H \cup E_H \times V_G$

Koncové vrcholy hrany $(d, v) \in E(G) \times V(H)$ sú vrcholy (x, v) a (y, v), kde $x, v \in d; d \in E(G)$, jedná sa teda o hranu $\{(x, v), (y, v)\}$. Pre hranu $(e, u) \in E(H) \times V(G)$ podobne.

2.2 Vybrané triedy grafov

Niektoré grafy majú spoločné vlastnosti na základe ktorých ich delíme do množín - tried grafov. Teraz si predstavíme niekoľko tried grafov potrebných v ďalších častiach diplomovky.

Definícia 16. *Cesta dĺžky n je graf, v ktorom existuje cesta na n vrcholoch a neobsahuje žiadne ďalšie hrany ani vrcholy. Takýto graf označujeme P_n.*

Podobne definujeme triedu grafov cykly.

Definícia 17. *Cyklus dĺžky n je graf s n vrcholmi, v ktorom existuje cyklus s n vrcholmi obsahujúci všetky hrany tohto grafu. Cyklus dĺžky n označujeme C_n.*

Definícia 18. *n-rozmerná kocka Q_n je graf, ktorý vznikol ako kartézsky súčin n činiteľov P_2.*

$Q_n \equiv \times_n^{i=1} (P_2)_i$

Keď označíme vrcholy grafu P_2 ako 0 a 1. Budú označenia vrcholov Q_n zodpovedať bitovým zápisom čísel od 0 po n-1 a hrany povedú medzi dvomi vrcholmi, pri ktorých sa zodpovedajúce čísla líšia v práve jednom bite.

Kapitola 3

Grafové gramatiky

3.1 Úvod

Naša práca sa zaoberá vytváraním grafov a preto je nutné spomenúť najpoužívanejší formalizmus vytvárania grafov - grafové gramatiky. Cieľom tejto kapitoly je uviesť čitateľa do problematiky grafových gramatík a poskytnúť mu stručný prehľad tejto oblasti.

Predpokladáme, že čitateľ má základné vedomosti z teórie formálnych jazykov a automatov.

3.2 Hlavné prístupy

Teória okolo grafových gramatík sa budovala podobne ako pri formálnych jazykoch a automatoch z viacerých prístupov pri ktorých boli neskôr dokázané určité ekvivalencie. Z toho vyplýva množstvo tried gramatík, prístupov a definícií. V konečnom dôsledku však možno rozlíšiť dva hlavné prístupy ku grafovým gramatikám - algoritmický (spájací) a algebraický (lepiaci). Pre ilustráciu uvedieme niekoľko tried a skúmaných problémov. Grafové gramatiky sa týkajú našej práce len okrajovo, v prípade hlbšieho záujmu oporúčame čitateľovi publikáciu Handbook of Graph Grammars [RE97] , z ktorej sme aj my čerpali.

3.3 Všeobecné pojmy

Za účelom uvedenia čitateľa do problematiky grafových gramatík sme si vybrali skupinu grafových gramatík založenú na prepisovaní vrcholov. Vo všeobecnosti má pravidlo odvodenia v grafovej gramatike tvar (M, D, E), kde M a D sú grafy a E je vkladací mechanizmus. Takéto odvodenie môže byť použité na ľubovoľný graf H, ktorý obsahuje podgraf M. Jeden výskyt podgrafu M je nahradený grafom D a spojený so zvyškom grafu H mechanizmom popísaným E. Zvyšok grafu H označujeme H^-. Na základe mechanizmu E sa rozlišujú spomínané dva prístupy. Pri lepiacom prístupe sa časti H^- stotožnia s časťami M a následne D. Graf D je skrz ne následne "vlepený"do grafu H^-. Pri spájacom prístupe vznikajú podľa E nové "spájacie"hrany medzi D a H^-. V spájacom prístupe sa v bezkontextových gramatikách skladá graf M len z jedného vrcholu. Teda vrchol je nahrádzaný grafom. Požaduje sa tiež, aby postupnosť nahrádzania vrcholov nemala vplyv na konečný výsledok. Pritom je viacero prirodzených spôsobov definovania mechanizmu E a obmedzení na "pravé strany"pravidiel - graf D.

3.4 NLC gramatiky

NLC (Node Label Controlled) grafové gramatiky sú jedným z najjednoduchších mechanizmov prepisovania grafov. Pravidlá NLC gramatiky sú tvaru $X \to D$, kde X je značka (label) prepisovaného vrchola a D je dcérsky graf. Pre použitie pravidie v NLC neexistujú žiadne obmedzenia, teda pravidlo sa môže použiť na ľubovoľný vrchol so značkou X. Vkladanie je v NLC lokálne, tj. vrcholy grafu D môžu byť spojené len s tými vrcholmi, s ktorými bol spojený prepísaný vrchol X. O NLC gramatikách, kde odvodenie je nezávislé na poradí použitia pravidiel odvodenia, hovoríme, že majú Church-Rosserovu vlastnosť - sú konfluentné (confluent). Vo všeobecnosti NLC túto vlastnosť nemajú a pri použití rovnakej sady pravidiel môžu vzniknúť rôzne grafy.

Definícia 19. *NLC grafová gramatika je 5-tica $G = (\Sigma; \Delta; P; C; S)$ kde $\Sigma - \Delta$ je abeceda neterminálnych znaciek uzlov a $\Delta(\Delta \subseteq \Sigma)$ je abeceda terminálnych znaciek uzlov. P je konecná množina NLC pravidiel, C je binárna relácia spojení nad Σ a S je pociatocný graf.*

Definícia 20. *Grafový jazyk generovaný G je $L(G) = \{H \in GR_\Delta | S \Rightarrow^* H\}, kdeGR_\Delta$ je množina neorientovaných grafov so značkami uzlov v Δ, $\Rightarrow$ reprezentuje jeden prepisovací krok, a $\Rightarrow^*$ reprezentuje odvodenie, t.j. postupnosť prepisovacích krokov.*

3.5 NCE gramatiky

NCE (Neighborhood Controlled Embeding) grafové gramatiky sú rozšírením NLC grafových gramatík.V NCE mechanizme prepisovania je, na rozdiel od NLC, množina spájacích inštrukcií C definovaná pre každé pravidlo zvlášť a NCE gramatika môže k uzlom podgrafu D pristupovať priamo napr. ich očíslovaním; nemusí používať len značky týchto vrcholov. Toto už nie je možné pri vrcholoch hosťovského grafu (graf H), pretože v ňom môže byť počet susedov nahrádzaného vrchola neohraničený. Existujú rôzne rozšírenia NCE gramatík. edNCE gramatiky sú rozšírením NCE gramatík s dynamickým preznačovaním hrán pri prepisovaní. Písmenko e znamená, že sa budú značiť hrany a písmenko d označuje použitie orientovaných grafov.

Definícia 21. *edNCE grafová gramatika je definovaná ako 6-tica* $G = (\Sigma; \Delta; \Gamma; \Omega; P; S)$. *Kde* Σ *je abeceda značiek uzlov,* $\Delta \subseteq \Sigma$ *je abeceda terminálnych značiek uzlov,* Γ *je abeceda značiek hrán,* $\Omega \subseteq \Sigma$ *je abeceda terminálnych značiek hrán,* P *je konečná množina pravidiel a* $S \in \Sigma - \Delta$ *je počiatočný neterminál. Pravidlá sú tvaru* $X \to (D, C), kdeX \in \Sigma - \Delta$ *a* $(D, C) \in GRE_{\Sigma,\Gamma}$. *Relácia spojení je* $C \subseteq \Sigma \times \Gamma \times \Gamma V_H \times in, out$.

Vo všeobecnosti množinu grafov izomorfných s grafom H budeme značit $[H]$. Niekedy sa hovorí o H ako „konkrétnom“ grafe a o $[H]$ ako „abstraktnom“ grafe.

Definícia 22. *Množinu všetkých konkrétnych grafov nad* Σ *a* Γ *značíme* $GR_{\Sigma,\Gamma}$ *a množinu všetkých abstraktných grafov značíme* $[GR_{\Sigma,\Gamma}]$. $GRE_{\Sigma,\Gamma}$ *označuje množinu grafov s vnorením nad* Σ, Γ. *Obyčajný graf môžeme považovať za graf s prázdnym vnorením, teda* $GR_{\Sigma,\Gamma} \subseteq GRE_{\Sigma,\Gamma}$.

Definícia 23. *Grafový jazyk generovaný G je* $L(G) = \{[H] | H \in GR_{\Sigma,\Omega}$ *a* $sn(S, z) \Rightarrow^* Hprenejak\acute{e}z\}$ *(t.j. všetky grafy v odvodení G, ktoré majú iba terminálne značky hrán aj uzlov).*

Teraz si definujeme hlavnú triedu bezkontextových grafových gramatík so spájacím prístupom. Označujeme ju C-edNCE. Táto je najrozšírenejšia v spájacom prístupe má značne všeobecný mechanizmus vkladania a žiadne reštrikcie na pravé strany.

Definícia 24. *edNCE gramatika* $G = (\Sigma; \Delta; \Gamma; \Omega; P; S)$ *je konfluentná (confluent), alebo C-edNCE gramatika, ak pre všetky pravidlá* $X_1 \to (D1, C1)$ *a* $X_2 \to (D_2, C_2)$ *z P, všetky uzly* $x_1 \in V_{D_1} a x_2 \in V_{D_2}$ *, a všetky hrany so značkami* $\alpha, \delta \in \Gamma$*, platia nasledujúce ekvivalencie:* $\exists \beta \in \Gamma : (X_2, \alpha/\beta, x_1, out) \in C_1 a(\lambda_{D_1}(x_1), \beta/\delta, x_2, in) \in C_2 \Leftrightarrow (\exists \gamma \in \Gamma : (X_1, \alpha/\gamma, x_2, in) \in C_2 a(\lambda_{D_2}(x_2), \gamma/\delta, x_1, out) \in C_1$.

Poznámka 1. *Existuje charakterizácia C-edNCE nezávislá na gramatikách pomocou Monadic Second Order (MSO) logiky. Veľa výsledkov ju pre jednoduchosť využíva v dôkazoch.*

3.6 Rozličné známe výsledky

Hodno spomenúť, že vlastnosti, ktoré poznáme ako bezkontextové nemusia byť nutne konfluentné. [RE97] spomína pomerne jednoduchú lemu o konfluentnosti gramatiky, kde každý vrchol má "svoju"polovicu značky hrany a pri odvodení mení len tú. Následne nezáleží na poradí odvodenia dvoch vrcholov spojených hranou, lebo tieto menia len "svoju"časť značky hrany. Medzi zaujímavé veci v teórii o C-edNCE gramatikách patrí existencia takzvaných zakázaných (forbidden) hrán. Sú to hrany označené neterminálnym symbolom spájajúce dvojice terminálnych vrcholov. Z grafu obsahujúceho čo len jedinú takúto hranu sa už konečný graf nestane. Existujú mnohé modifikácie gramatík, ktoré sme si bližšie nespomínali. Skúmali sa aj problémy rozpoznávania, či graf patrí do jazyka; napríklad existuje NP-úplný LIN-A-edNCE jazyk. Taktiež sú preskúmané rôzne normálne formy. Veta o medzere hovorí, že existuje c pre ktoré sa nedá odvodiť v danom jazyku dosť grafov a tak ďalej.

3.7 Grafové gramatiky s nahrádzaním hyperhrán

Pre úplnosť venujeme krátky priestor aj vlepovaciemu prístupu v grafových gramatikách. Grafové gramatiky s nahrádzaním hyperhrán sú považované za grafovú ekvivalenciu k reťazcovým gramatikám. Ich začiatky siahajú do skorých 70. rokov. Základom je hyperhrana. Hyperhrana je atomický prvok s fixným počtom chápadiel - typov. Môže byť pripojená k ľubovoľnej štruktúre odvodenej z množiny vrcholov pripojením chápadla do vrcholu. Hyperhrana môže byť prepísaná štruktúrou R. R však musí mať, kvôli lepiacemu prístupu, rovnaký počet externých vrcholov ako nahrádzaná hrana typov. Gramatiky v tejto skupine majú konečný počet pravidiel a počiatočnú štruktúru. Jazyk tvoria množiny konečných štruktúr odvodených takouto gramatikou. Prepísanie hyperhrany je bezkontexotvé, pretože nezasahuje do okolia. Bolo dokázaných niekoľko výsledkov, ktoré majú blízko k podobným bezkontextovým gramatikám na reťazcoch. Ukázalo sa, že veľa zaujímavých problémov je pre tieto gramatiky rozhodnuteľných. Môžeme sa pýtať dva typy otázok: či hypergrafy generované príslušnou gramatikou spĺňajú vybrané vlastnosti alebo klasickú otázku príslušnosti do jazyka. Otázka príslušmosti je vo všeobecnosti NP-kompletná a len niektoré obmedzené triedy sú polynomiálne ťažké.

Kapitola 4

Prehľadávanie grafu

4.1 Úvod

Jednou z motivácií nášho prístupu ku konštrukcii grafu je možnosť využitia takejto konštrukcie ako duálneho problému k prehľadávaniu grafu. Náš model konštrukcie vychádza z modelov používaných pri prehľadávaní grafov s tým rozdielom, že kým v niektorých úlohách z prehľadávania je daný neznámy graf a úlohov agenta je skonštruovať jeho mapu, v našom modely je dané ako má graf vyzerať a úlohou agenta je skonštruovať ho pomocou operácií, ktoré má k dispozícii. V tejto kapitole uvádzame čitateľa do problematiky prehľadávania grafu, príslušných definícií a zaujímavých výsledkov z tejto oblasti. Nájde tu prehľad vybraných prístupov k tejto problematike a používaných modelov.

V problémoch prehľadávania má agent zostrojiť kompletnú mapu grafu (prostredia) bez akejkoľvek počiatočnej informácie o jeho topológii, prípadne len navštíviť každý vrchol alebo prejsť každou hranou. Tiež sú známe tri typy riešených úloh podľa toho, ako agent ukončí prehľadávanie. Prvým typom je prehľadávanie s návratom, keď agent po skončení musí zastať v počiatočnom vrchole. Druhým typom je prehľadávanie so zastavením, kedy agent musí v konečnom čase po prehľadaní celého grafu zastaviť. Posledným, tretím, typom úloh o prehľadávaní grafu je trvalé prehľadávanie. V tomto prípade agent pokračuje v prechádzaní grafu aj po jeho úplnom prehľadaní. Patria sem tiež úlohy, v ktorých má agent navštevovať každý daný prvok grafu (vrchol alebo hranu) s určitou periódou. Význam takejto úlohy je napríklad pri neustálej kontrole siete, ktorá sa môže kaziť počas prevádzky. Úspešnosť algoritmov prehľadávania grafov sa porovnáva s takzvanými offline problémami. Pri offline probléme agent graf pozná a jeho úlohou je navštíviť každý jeho vrchol, prípadne hranu podľa rovnakého zadania ako pri prehľadávaní. Napríklad ak má agent navštíviť každý vrchol v neznámom ohodnotenom grafe, online problémom je problém obchodného cestujúceho (TSP).

Prvý formálny model na prehľadávanie grafu uviedol [PY91]. Prvý raz bolo skúmané prehľadávanie celého grafu v [DP90], avšak uvažovali preskúmanie každej hrany v označenom orientovanom grafe. Teda, keď agent prišiel do vrcholu, dostal informáciu o počte neobjavených hrán odchádzajúcich z vrcholu ale už nie informáciu, kam tieto hrany vedú. Zodpovedajúcim offline problémom pre tento problém však nie je TSP ale polynomiálne riešiteľný Problém čínskeho poštára. Takéto zadanie problému prehľadávania má blízko k nášmu poňatiu problému konštrukcie grafu pomocou jedného mobilného agenta. V tejto oblasti prebieha rozsiahli výskum na orientovaných aj neorientovaných grafoch. Existuje aj iná trieda problémov, v literatúre označovaných ako online TSP, kde graf je známy dopredu a vrcholy, ktoré majú byť navštívené sa o bjavujú postupne. Zodpovedajúci offline problém pre túto triedu problémov je TSP s dátumami uvoľnenia (release dates).

Predpoklad, že konečný automat s konečným počtom kamienkov nedokáže prehľadať každý graf bol dlho otvoreným problémom až [Rol79] dokázal o niečo širšie tvrdenie. Rollik [Rol79] dokázal, že žiadna konečná množina konečných automatov nedokáže prehľadať všetky kubické planárne grafy.

Labyrint je šachovnica Z^2 so zakázanými políčkami. Labyrint je konečný. Prehľadávaním labyrintov sa vo veľkej miere zaoberal Budach. Prehľadať konečný labyrint znamená, že robot je schopný odísť ľubovoľne ďaleko zo svojej štartovnej pozície pre každú štartovnú pozíciu. Hrany labyrintu sú konzistentne označené svetovými stranami (východ, juh, západ, sever).

4.2 Prehľadávanie jedným agentom

[MMS11] sa zaoberá problémom prehľadávania na neorientovaných spojitých grafoch s hranami ováhovanými nezápornými reálnymi číslami a označenými (labeled) vrcholmi (agent ich vie rozlíšiť). Efektivita online algoritmov sa zisťuje ich porovnávaním s prislúchajúcimi offline riešeniami. V tomto prípade je prislúchajúcim offline problémom Problém obchodného cestujúceho (TSP).

Ďalši článok [FIP$^+$05] skúma prehľadávanie grafu konečným automatom (robotom). Graf je neoznačený s hranami lokálne označenými v každom uzle; rovnako ako v našom modely konštrukcie. Úlohou robota je prehľadať každú hranu v grafe bez akejkoľvek predošlej znalosti o veľkosti a topológii grafu. Medzi zaujímavé výsledky tohto článku patrí dôkaz existencie planárnych grafov, pre ktoré sa to nedá. V prípade unikátneho značenia hrán a vrcholov môže byť prehľadanie dosiahnuté ľahko. Existujú neznáme prostredia, v ktorých takéto značenie nemusí byť k dispozícii, alebo robot nemusí byť schopný rozlíšiť od seba dve podobné značky. Z tohto dôvodu sa skúma prehľadávanie anonymných grafov; tj. grafov bez unikátne označených hrán a vrcholov. Je len samozrejmé, že bez možnosti lokálne rozlíšiť konce hrán by nebolo možné prehľadať dokonca ani hviezdu s tromi listami. Predpokladá sa teda v každom vrchole označenie portov $1..d$, kde d je stupeň vrchola. Avšak nepožaduje sa žiadna konzistencia tohto značenia. Keďže v mnohých aplikáciách sa požaduje, aby bol robot malé lacné zariadenie, autori článku optimalizujú pamäť. Takže hľadajú robota schopného preskúmať graf danej neznámej veľkosti s tak malou pamäťou, ako je to len možné. Robot s k-bitovou konečnou pamäťou sa modeluje konečným automatom. Trap je graf, ktorý nie je možné celý prehľadať automatom bez kamienkov (pebbles). Výsledok článku hovorí, že pre každý K-stavový automat a $d \geq 3$ existuje K+1 vrcholový graf - trap s max. stupňom d, ktorý sa nedá prehľadať týmto automatom.

4.3 Distribuované prehľadávanie

Autori článku [DFK$^+$05a] používajú pre prehľadávanie neznámeho prostredia viacero totožných agentov. Neznáme prostredie je v tomto prípade opäť modelované grafom a úlohou agentov je skonštruovať totožnú mapu grafu pre každého agenta. Bolo dosiahnuté deterministické riešenie za čo najslabších možných nastavení, pričom agenty [1] poznali iba veľkosť grafu alebo počet agentov.

Autori skúmajú distribuovanú verziu klasického problému prehľadania garfu s návratom do počiatočného vrchola, pričom agent má prejsť všetky hrany neoznačeného (anonymného) grafu. Počas tohto procesu je úlohou agenta vytvoriť mapu grafu. Model pozostáva z neoznačeného grafu a k agentov roztrúsených po n vrcholoch grafu. Komunikácia medzi agentmi prebieha prostredníctvom zápisov na malé tabuľky nachádzajúce sa v každom vrchole grafu. Tento problém je zložitejší od prehľadania grafu jediným agentom, pretože vyžaduje kooperáciu medzi agentmi. Keďže je tento problém vo všeobecnosti deterministicky neriešiteľný, článok predstavuje deterministický algoritmus za podmienky, že n a k sú nesúdeliteľné.

Vo svojej práci dávajú autori svoj výskum do súvisu s ostatnými problémami distribuovaných výpočtov, ako je napríklad voľba šéfa.

4.4 Prehľadávanie s radou

Z praxe sú známe situácie, keď graf, ktorý má agent prehľadať, nie je úplne neznámy. Teda je o ňom dostupná nejaká informácia. V [DKM12] skúmajú autori predovšetkým zložitosť rady. Teda aký je vzťah medzi veľkosťou informácie, ktorú agent o grafe

[1] Na návrh staršieho kolegu skloňujeme slovo agent v našom kontexte ako neživotné.

dostane a kvalitou riešenia. Úlohou agenta je prehľadať s návratom do počiatočného vrchola neznámy neorientovaný graf. Čiže agent má navštíviť každý vrchol grafu a vrátiť sa do počiatočného vrchola s čo najnižším súčtom cien prechodov po hranách. V tomto modely agent pri návšteve vrchola vidí ceny všetkých s ním incidentných hrán a kam tieto hrany vedú.

Výsledky sa týkajú dolnej hranice, kde autori ukázali, že $\Omega(nlogn)$ bitov rady je potrebných na dosiahnutie pomeru porovnania 1 oproti algoritmu pozanjúcemu topológiu grafu. Taktiež dokázali, že s radou $O(n)$ bitov pri ohraničených váhach je možné dosiahnuť konštantný pomer.

Autori sa venovali aj klasickej otázke prehľadania grafu a posunuli dolnú hranicu prehľadania grafu deterministickým algoritmom bez rady.

Kapitola 5

Model a všeobecné výsledky o konštrukcii

5.1 Definícia

V tejto kapitole uvedieme nami používaný model a všeobecné výsledky o ňom. Obsahuje definície používaných pojmov a označení. Začneme predstavením agenta a vlastností grafu, v ktorom operuje.

Definícia 25. *Model sa skladá z grafu a agenta. Graf má nemeniteľné lokálne číslovanie koncov hrán vo vrchole, podľa poradia vzniku. Koniec hrany, ktorý vznikol ako prvý, má číslo 1. Prvý koniec vznikne spolu s vrcholom, zvyšné otvorením portu. Port je vlastne jeden z koncov ešte nevytvorenej hrany. Každá entita v modely môže byť nejako dodatočne označená. Napríklad aj vrcholy alebo hrany grafu. V našom modely však pracujeme len so značením portov. Značky portov slúžia na to, aby mohlo byť naraz otvorených viacero portov bez toho, aby medzi nimi vznikla hrana a aby sa dalo určiť, do ktorého vrchola má hrana ísť v prípade, že je otvorených viacero portov. Počiatočný graf obsahuje práve jeden izolovaný vrchol a nič viac. Akonáhle sú otvorené dva porty s rovnakými značkami, vznikne medzi nimi nová hrana a tieto porty tým zaniknú.*

Agent môže vo vrchole, v ktorom sa nachádza, vykonávať tieto operácie: pohnúť sa po hrane do susedného vrcholu, otvoriť port so značkou, vytvoriť hranu do nového vrchola v ktorom sa následne ocitne. Agent rozlišuje hrany podľa lokálneho číslovania ich koncov vo vrchole, kde sa práve nachádza. Agent vie ktorý koniec patrí hrane po ktorej prišiel do vrcholu. Agent začína vo vrchole počiatočného grafu.

Agent pri vykonávaní algoritmu konštrukcie grafu nemá obmedzenú výpočtovú silu ani pamäť. Množina značiek portov je konečná a vopred daná. Efektivita algoritmu konštrukcie daného grafu sa pri tomto modely meria počtom pohybov agenta (tj. sumou počtu návštev cez všetky vrcholy) pri danej množine značiek portov, tento počet sa *snažíme minimalizovať.*

Nasleduje uvedenie označení a použitých pojmov.

Označenie 1. *Použité skratky operácií agenta: - nová hrana = NH - otvoriť port (s číslom n) = OP(n) - prejsť sa po hrane s lokálnym číslom konca k = Sk*

Označenie 2. *Agent začína v grafe, ktorý sme už nazvali počiatočný graf - v našom prípade obsahuje jeden izolovaný vrchol. Graf, ktorý má agent skonštruovať, budeme volať cieľový, konečný alebo konštruovaný. Graf existujúci v danom kroku konštrukcie v modely budeme volať aktuálny graf alebo len graf.*

Poznámka 2. *Pri práci s neorientovanými grafmi používame pojem "hrana vedie z vrchola v do vrchola u"len na rozlíšenie týchto dvoch vrcholov od seba.*

Označenie 3. *Nad prvkami grafu, konštruovaného jedným agentom (sú to najmä vrcholy, hrany i porty), existuje prirodzené úplné usporiadanie podľa poradia ich vzniku, podľa ktorého môžeme o jednom prvku vyhlásiť, či je starší, mladší alebo rovnako starý ako iný prvok. Platí, že hrana je rovnako stará ako jej druhý port. Všeobecne prvky vzniknuté jednou operáciou sú rovnako staré. Pri modely s jediným agentom sú operácie vykonávané sekvenčne a podľa ich poradia vieme určiť aj "veköbjektov, ktoré ic vykonaním vzniknú.*

Označenie 4. *Krokom agenta budeme nazývať vykonanie operácie, pri ktorej sa zmení vrchol, v ktorom sa agent nachádza.*

Označenie 5. *Portová hrana je hrana, ktorá vznikla operáciou OPk, kde k je značka portu, ktorého otvorením hrana vznikla.*

Označenie 6. *Kostrová hrana je hrana, ktorá vznikla operáciou NH.*

Poznámka 3. *Ako neskôr ukážeme - "kostrové hrany" tvoria kostru grafu.*

Označenie 7. *Medzi otovrením prvého a druhého portu portovej hrany h, prejde agent po slede vrcholov. Najmenší cyklus z tohto uzavretého sledu obsahujúci hranu h budeme označovať tvorivý cyklus hrany h. Hrany tohto cyklu bez hrany h tvoria konštrukčnú cestu hrany h.*

Označenie 8. *Postupnosť návštev vrcholov a hrán v poradí v akom ich agent vykonal počas konštrukcie cieľového grafu budeme volať tvorivý sled grafu.*

5.2 Základné vlastnosti

V tejto časti kapitoly sú umiestnené výsledky platné, pre konštrukciu grafov pomocou mobilných agentov za použitia nášho modelu, všeobecne pre každý graf.

Najprv sme skúmali ako sa model správa vo všeobecnosti pri konštrukcii všetkých grafov. Tiež aké sú obmedzenia na pohyb agenta a čím sa líšia tri operácie, ktoré môže vykonávať - čo mu ktorá dovoľuje.

Keďže je podstatou našej práce optimalizovať model na počet krokov, ktoré musí agent prejsť po grafe, najprv nás zaujímalo ako sa po ňom môže pohyboovať a kedy sa musí do vrcholu ešte vrátiť.

Lema 1. *Agent počas pohybu v grafe musí ísť po hranách grafu. Tieto hrany po momente keď po nich prešiel, už v grafe trvale existujú.*

Dôkaz. Žiadna operácia agentovi nedovoľuje priamo prejsť medzi dvoma vrcholmi, ktoré nespája hrana. Neexistuje spôsob ako v našom modely odstrániť hranu z grafu. □

Lema 2. *Ak agent odíde z vrcholu v ktorom je súčet počtov incidentných hrán a otvorených portov nižší, ako je počet hrán, s ktorými má byť vrchol incidentný v cieľovom grafe, potom sa do tohto vrchola musí agent neskôr vrátiť.*

Dôkaz. Neexistuje postupnosť operácií agenta, pomocou ktorej môže agent vytvoriť hranu do existujúceho vrchola bez jeho návštevy, ak v ňom nie je otvorený port pre túto hranu. Teda ak sa v danom vrchole agent nenachádza a nemá v ňom otvorený port pre hranu, ktorú ide skonštruovať, nemôže hranu skonštruovať bez návštevy tohto vrchola. □

Jedinou operáciou, ktorá umožňuje agentovi vytvárať nové hrany je operácia NH. Agent pri nej vytvorí hranu z vrchola, v ktorom sa práve nachádza do nového vrchola a v tomto vrchole sa agent ocitne. Pomerne jednoducho sa dá ukázať, že takto vzniknuté hrany tvoria kostru grafu. Vyplýva z toho tiež to, že časť grafu, ktorú agent doteraz skonštruoval je súvislá i to, že ak má agent skonštruovať hranu do vrchola, ktorý už existuje, nemôže na to využiť operáciu NH.

Lema 3. *Hrany vytvorené operáciou "nová hrana"tvoria kostru grafu.*

Dôkaz. Okrem inicializačného vrcholu, všetky vrcholy vzniknú operáciou "nová hrana"(NH). Každý vrchol okrem inicializačného je spojený hranou, s lokálnym číslom konca hrany 1, s nejakým starším vrcholom (vrchol, ktorý vznikol skôr). Po týchto hranách sa dá z každého vrchola grafu dostať do inicializačného, teda podgraf, tvorený hranami pochádzajúcimi z oprácií NH, je súvislý. Počet týchto hrán je zároveň o jeden menší ako počet vrcholov, teda ide o kostru. □

Hrany, ktoré nevznikli operáciou NH, vytvoril agent otvorením ich portov. Za týmto účelom musel navštíviť oba konce každej portovej hrany. Skúmali sme, aký najnižší počet krokov musí agent vykonať pri konštrukcii konkrétnej hrany.

Lema 4. *Hrany, ktoré agent prejde od otvorenia prvého portu hrany po otvorenie druhého, tvoria v grafe s novovzniknutou hranou uzavretý sled.*

Dôkaz. Je to dané operáciami, ktoré môže agent vykonať. Agent totiž prechádza len po hranách v grafe. Z vrchola ide do incidentnej hrany a z hrany do incidentného vrchola. Pričom žiadna hrana nemôže zaniknúť, teda medzi otvorením dvoch portov prejde agent po nejakom slede v grafe. Novovzniknutá hrana spája koncový a začiatočný bod tohoto sledu; tvorí s ním teda uzavretý sled. □

Poznámka 4. *Zo sledu z lemy 4 vieme vybrať kružnicu obsahujúcu práve konštruovanú hranu. Ak totiž sled nie je kružnica, obsahuje viacnásobný výskyt niektorého vrchola okrem počiatočného a koncového, ktoré sú pri uzavretom slede totožné. Ak prvky sledu medzi dvomi takýmito výskytmi odstránime a tieto výskyty zlúčime do jedného, výsledkom bude tiež uzavretý sled.*

Lema 5. *Minimálny počet hrán, ktoré agent potrebuje prejsť pri konštrukcii hrany pomocou portov je rovný alebo väčší ako dĺžka najmenšej kružnice, na ktorej sa táto hrana v konštruovanom grafe nachádza.*

Dôkaz. Nájdeme menší takýto sled ako je najkratšia kružnica. Tento tvorí s novovzniknutou hranou uzavretý sled. Z neho vieme vybrať kružnicu, čo je spor s minimálnosťou najkratšej kružnice. Keďže neexistuje v našom modely možnosť, aby hrana zanikla, musí sa táto "nová"najkratšia kružnica nachádzať aj v konštruovanom grafe. □

Agent môže mať k dispozícii viac značiek a preto môže niektoré hrany konštruovať súčasne. Potom jeden krok agenta môže patriť ku konštrukcii viacerých hrán a celkový počet krokov tak môže byť menší ako v prípade, keď agent konštruuje hrany po jednom. Aký je teda minimálny počet krokov, ktoré agent musí vykonať?

Je isté, že musí skonštruovať všetky vrcholy grafu. Teda každý z nich musí navštíviť aspoň raz. Musí teda vykonať aspoň toľko krokov, koľko má kostra hrán. Tiež nemôže prejsť menej hrán, ako je dĺžka najkratšieho sledu obsahujúceho všetky vrcholy.

Lema 6. *Ak má agent k dispozícii dostatočný počet značiek portov, tak sa problém efektívnej konštrukcie grafu redukuje na hľadanie najkratšieho sledu v ktorom sú všetky vrcholy. Pri hamiltonovskom grafe ide o hamiltonovskú kružnicu.*

Dôkaz. Agent skonštruuje graf takto: prechádza vrcholmi, podľa najkratšieho sledu obsahujúceho všetky vrcholy, pričom mu stačí jedna návšteva každého vrcholu, v ktorom otvorí potrebný počet portov príslušných značiek. Ak má v slede prejsť do vrcholu, ktorý ešte neexistuje, použije operáciu NH. □

Aj v tomto prípade je možná ďalšia optimalizácia a to na počet použitých značiek pre porty. Celkovo to funguje takto: agent otvorí port so značkou k, vytvorí pár kostrových hrán a následne v nejakom vrchole otvorí druhý port s rovnakou značkou, čím sa mu značka uvoľní na ďalšie použitie. V každom vrchole, ktorý agent medzitým vytvorí, musí otvoriť porty, pre všetky jeho budúce hrany v konštruovanom grafe. Pri tom nemôže použiť značku k. Vo všeobecnosti teda platí, že čím kratšie sú porty otvorené, tým menší je celkový počet potrebných značiek, toto platí najmä v grafoch s vrcholmi rovnakého stupňa.

Agent však nemusí mať dostatok značiek na to, aby mu stačila jedna návšteva vrchola. Aký je teda minimálny počet návštev pri konštrukcii ľubovoľného grafu?

Lema 7. *Vrchol v grafe musí agent navštíviť minimálne* $\frac{deg(v)}{|Z|*2}$ *krát, kde Z je množina značiek pre porty, ktorú ma agent k dispozícii.*

Dôkaz. Pri každej návšteve môže agent otovrením portov so všetkými značkami vytvoriť nové hrany do vrcholov, v ktorých sú otovrené príslušné porty a následne otvoriť toľko portov s navzájom rozdielnymi značkami, koľko má k dispozícii značiek (dvojica portov rovnakej značky v jednom vrchole by vytvorila slučku a v našom modely uvažujeme hrany bez slučiek). □

Ako ukážeme neskôr, pri obmedzenom množstve značiek portov môže byť výhodnejšie neminúť všetky značky pre porty v jednom vrchole, aj za cenu opakovanej návštevy. Výhodnejšie v tomto prípade znamená, že agent pri danej konštrukcii vykoná menej krokov.

5.3 Jedna značka pre porty

V tejto časti sa budeme zaoberať všeobecnými výsledkami pre model, v ktorom má agent k dispozícii len jedinú značku pre porty a teda v grafe môže byť naraz otvorený len jediný port. Toto obmedzenie núti agenta konštruovať porotvé hrany po jednom. Nasledujúca lema hovorí, že ak sú dva vrcholy portovej hrany, ktorú ide agent konštruovať rôzne ďaleko, nikdy nie je horšie ísť do bližšieho z nich otvoriť prvý port.

Lema 8. *Najmenej krokov agent na konštrukciu konkrétnej hrany medzi už existujúcimi vrcholmi použije, ak pôjde do najbližšieho z nich, otvorí v ňom port a následne* *najkratšou cestou po už existujúcich hranách prejde do druhého vrcholu a otvorí port v ňom.*

Dôkaz. Agent má k dispozícii len jednu značku pre porty. Teda počas konštrukcie hrany medzi už existujúcimi vrcholmi, žiadne nové hrany v grafe, ktoré by ležali na ceste medzi koncami tejto hrany nevzniknú. Jediné nové hrany môžu vzniknúť operáciou NH, ale tieto by viedli do "slepej uličky"(komponentu grafu obsahujúceho len nové vrcholy a spojeného mostom so zvyškom grafu), z ktorej by sa musel agent vracať V grafe pri jednej značke pre port je najkratšia cesta od otvorenia prvého po otvorenie druhého portu hrany rovnaká bez ohľadu na to, v ktorom vrchole agent otvorí prvý port hrany. Jediné, čo sa mení, je počet krokov po otvorenie prvého z portov a poloha agenta po konštrukcii. Aby bol tento a teda aj celkový počet krokov čo najkratší, pôjde agent do najbližšieho z vrcholov konštruovanej portovej hrany. Ak v pôvodnej konštrukcii agent končil v druhom vrchole hrany, na jeden krok doň prejde. Táto konštrukcia teda nie je horšia. □

Najmenší počet krokov, ktoré agent vykoná pri konštrukcii jednej portovej hrany je dĺžka najkratšej kružnice, na ktorej leží. Pre všetky portové hrany je to potom suma týchto kružníc. Ktpré hrany sú portové závisí id toho, ktoré sú kostrové. Preto v nasledujúcej leme robíme dolný odhad počtu krokov cez všetky kostry.

Lema 9. *Agent, ktorý má k dispozícii len jednu značku pre porty, musí prejsť pri konštrukcii grafu aspoň k krokov, kde* $k = min_{i=1}^{y}\{\sum_{e \in G(E); e \notin G(K_i)} j(e) - 1\}$*, kde K je kostra grafu G; y je počet navzájom rôznych kostier garfu G; j(e) je najmenšia kružnica v grafe obsahujúca hranu e a* K_i *je i-ta kostra, pričom* $K_i \equiv K_j \iff i \equiv j$

Dôkaz. Podľa lemy 5 musí agent prejsť pri konštrukcii hrany aspoň dĺžku minimálnej kružnice bez jednej - bez hrany ktorú tým konštruuje. Keďže má k dispozícii len jednu značku pre porty, ďalšiu hranu môže začať konštruovať až keď dokončil predošlú. Pri každej hrane prejde teda aspoň dĺžku najmenšej kružnice, v ktorej sa táto hrana nachádza mínus jedna. Dokopy prejde agent ich súčet. □

V grafe, ktorý agent doteraz skonštruoval, môžu byť niektoré vrcholy ďalej ako v konečnom grafe. Kostrová hrana je v čase svojho vzniku mostom a medzi časťami grafu, ktoré spája môže agent prechádzať len po nej až dovtedy, kým medzi nimi nevytvorí nejakú portovú hranu.

Lema 10. *Po použití operácie NH, ktorou vznikne hrana h, sa agent nedostane do už existujúceho vrchola bez následného použitia operácie OP, ktorá vytvorí hranu do vrcholov starších ako h, alebo bez opätovného prejdenia po hrane h.*

Dôkaz. Vrchol v, ktorý vznikne operáciou NH je so zvyškom grafu spojený len hranou h, ktorá tvorí most, do vzniku hrany operáciou OP, medzi komponentmi na oboch stranách tohto mostu. Teda jediný spôsob ako prejsť medzi týmito dvomi časťami grafu je po moste, ktorý ich spája - po hrane h. □

Lema 11. *Existuje graf, pri ktorého konštrukcii na najmenej krokov, agent skonštruuje kostrové hrany, ktoré s konštruovanou portovou hranou netvoria tvoriaci cyklus.*

Dôkaz. Napríklad cyklus, kde do každého vrchola pridáme hranu do vrchola stupňa jedna. Je zrejmé, že pri konšrukcii týchto vrcholov stupňa jedna, agent urébí operácie NH a S1, lebo nemôže ostať vo vrchole ak ešte nedokončil graf a iná hrana z neho nevedie. Tiež je zrejmé, že takýto graf má práve jednu portovú hranu a to ľubovoľnú hranu cyklu. Pri jej konštrukcii prejde agent cez všetky vrcholy v cykle a ak by neskonštruoval príslušný vrchol stupňa jedna incidentný s vrcholom cyklu, pri jeho prvej návšteve, musel by sa doň vrátiť a teda vykonať viac krokov. Agent treda otvorí port a skonštruuje vrcholy cyklu, pričom v každom skonštuuje aj príslušný vrchol stupňa jedna s ktorým je tento incidentný. □

5.3.1 Dve a viac značiek pre porty

Ušetriť (agent vykoná menej krokov) oproti situácii, keď agent môže použiť len jednu značku pre porty možno v prípade že má k dispozícii viaceron značiek pre porty vtedy, ak konštruovaný graf obsahuje cykly, kotré majú spoločné hrany. Vtedy, ak agent pri konštrukcii niektorej hrany, prechádza vrcholom s ktorým bude incidentná niektorá portová hrana, ktorú bude konštruovať po tejto, v ňom otvorí port. Následne sa do tohto vrchola už nemusí pri konštrukcii tejto nasledujúcej portovej hrany vracať otvoriť port, ak existuje kratšia cesta do druhého koncového vrchola tejto hrany. V metrike vzdialeností, kde sa počíta počet hrán najkratšej cesty medzi dvomi vrcholmi ako ich vzdialenosť platí trojuholníková nerovnosť. Z tohto dôvodu agent nevykoná viac krokov pri najlacnejšej (na najmenej krokov) konštrukcii portovej hrany ak jeden port tejto hrany otvorí skôr oproti konštrukcii kde ho otvorí neskôr, pričom sa kroky rátajú až po dokončení predošlej portovej hrany. Krok, vykonaný agentom v čase, keď má tento otvorených viacero porto, sa do celkového súčtu počíta len raz. Ako neskôr ukážeme, nie vždy spôsobí snaha skonštruovať jednu hranu na čo nejmenej krokov zlepšenie celkového počtu krokov agenta počas konštrukcie cieľového grafu.

Kapitola 6

Konštrukcia mriežky

Definícia 26. *Mriežka veľkosti* $n \times m$ *je graf* $P_n \times P_m$

Definícia 27. *Cyklická mriežka veľkosti* $n \times m$ *je graf* $C_n \times C_m$.

V tejto kapitole sa budeme zaoberať mriežkou a cyklickou mriežkou. Na to, aby sme sa v týchto dvoch grafoch dokázali dobre orientovať, zavedieme pojem dimenzie a jej súradnice. Pričom využujeme, že oba grafy sú produktmi kartézskeho súčinu. Výsledky v tejto kapitole môžu byť užitočné pri hľadaní konštrukcií grafov, ktoré sú výsledkom kartézskyh súčinov zložitejších grafov. Cesta a cyklus totiž patria medzi najjednoduchšie podgrafy.

Označenie 9. *Pojem dimenzie v produkte kartézskeho súčinu* G*, grafov* U *a* V *zavedieme prirodzene vzhľadom na označenie vrcholov produktu ako usporiadanej dvojice* (u, v)*, kde* $u \in U$ *a* $v \in V$*. i-tou dimenziou vrchola produktu tohto súčinu bude i-ty člen tejto dvojice. Súradnicou v dimenzii i je prvok na i-tej pozícii v usporiadanej dvojici. Pre zjednodušenie označíme vrcholy pôvodných grafov prirodzenými číslami od 1 po m respektíve od 1 po n. Vrcholy produktu kartézskeho súčinu týchto dvoch grafov budú pre dosť veľké m,n napríklad* $(2, 4)$*,* $(5, 17)$*,* $(9, 1)$*, atď.*

Poznámka 5. *Pri mriežke a cyklickej mriežke budeme číslovať vrcholy pôvodných grafov tak, aby sa označenia susedných vrcholov líšili o jedna, v prípade cyklickej mriežky budú v pôvodných grafoch - cykloch spojené aj vrcholy 1 a n, repektíve 1 a m.*

Nasledujúcu lemu budeme používať neskôr. Hovorí o tom, že ak mriežka má oba rozmery väčšie ako tri, agent pri konštrukcii každej hrany vykoná najmenej tri pohyby. Cyklická mriežka s rozmerom dva nemôže existovať, lebo by obsahovala násobné hrany. Pokiaľ má cyklická mriežka niektorý rozmer tri, tak vie agent niektoré jej hrany skonštruovať na dva kroky. Hrany, ktoré pri tom prejde však musia byť buď kostrové alebo ich skonštruuje na tri kroky. Toto je veľmi špeciálny prípad.

Lema 12. *Pri konštrukcii portovej hrany v mriežke aj v cyklickej mriežke, ktorá nemá žiaden rozmer menší ako štyri, urobí agent medzi otvorením prvého a druhého portu najmenej tri kroky.*

Dôkaz. Najmenší cyklus v oboch grafoch má dĺžku štyri. Teda medzi otvorením dvoch portov hrany vykoná agent najmenej tri pohyby po hranách. Buď prejde po už existujúcich alebo vytvorí nové operáciou NH. □

6.1 Jedna značka pre porty

Z výsledkov pre konštrukciu cyklickej mriežky v prípade, keď má agent k dispozícii len jednu značku pre port, je najdôležitejšie zistenie, že pri konštrukcii musí agent pri niektorej z portových hrán od otvorenia prvého po otvorenie jej druhého portu prejsť po všetkých súradniciach okrem tej, v ktorej táto hrana vedie. Toto platí pre obe dimenzie.

Pre konštrukciu mriežky s jednou značkou pre porty sme dokázali, že sa dá skonštruovať na tri kroky pre každý port plus agent vykonal navyše $m-1$ krokov, kde m je menší z rozmerov. Vyslovili sme hypotézu, že na menej krokov to nie je možné. Aj keď sa nám ju nepodarilo uspokojivo dokázať, urobili sme viacero zaujímavých pozorovaní.

Teraz nasledujú prípravné lemy k týmto výsledkom. Pripomenieme, že pojem tvoriaci cyklus portovej hrany označujenajmenší cyklus obsahujúci danú hranu vybraný zo sledu, ktorý agent prešiel medzi otvorením jej prvého a druhého portu.

Pri dokazovaní výsledku o cyklickej mriežke využijeme fakt, že nahradením hrany v tvoriacom cykle s párnym počtom výskytov ostatných súradníc sa nemení ich parita. Pokiaľ sa teda zmení, musela byť nahrádzaná hrana s nepárnym počtom výskytov ostatných súradníc v tvoriacom cykle a tento nepárny výskyt musel agent "odšľapať".

Lema 13. *Kostrové hrana je staršia, vznikla skôr, ako všetky portové hrany incidentné s jej mladším koncovým vrcholom.*

Dôkaz. Aby agent mohol otvoriť vo vrchole port, musel už tento vrchol existovať. Jediná operácia ktorou mohol vrchol vzniknúť je operácia NH. Mladší vrchol vznikol spolu s kostrovou hranou a preto doň predtým nemohli viesť žiadne hrany. □

Lema 14. *Tvoriaci cyklus hrany h obsahuje len hrany staršie ako h.*

Dôkaz. Tvoriaci cyklus obsahuje hrany, po ktorých agent prešiel, keď konštuoval hranu h. Museli teda v čase po prechode agenta už existovať a sú preto staršie ako hrana h, ktorú agent vytvoril až potom, ako po týchto hranách prešiel. □

Lema 15. *Každý tvoriaci cyklus je možné nahradiť tvoriacim cyklom, ktorý obsahuje len hrany kostry a príslušnú portovú hranu.*

Dôkaz. Majme tvoriaci cyklus hrany h. Každú portovú hranu, okrem h v ňom nahradíme jej tvoriacim cyklom, až kým nemáme uzavretý sled na kostrových hranách a hrane h. Z neho vieme vybrať tvoriaci cyklus hrany h. □

Teraz si zavedieme pojem dimenzie v kartézskom súčine, čo budeme potrebovať pri mriežke aj cyklickej mriežke.

Označenie 10. *Graf prislúchajúci dimenzii d grafu G je d-ty činiteľ kartézskeho súčinu, ktorého je G produktom.*

Označenie 11. *Hrana v mriežke a cyklickej mriežke vedie medzi dvomi vrcholmi, ktoré sa líšia v práve jednej dimenzii a v tejto majú súradnice vrcholov, ktoré spolu susedia v grafe, ktorý prislúcha danej dimenzii. Teda ak je to prvá dimenzia jedná sa o prvý člen kartézskeho súčinu a ak je to druhá dimenzia o druhý. Súradnica hrany budeme volať súradnicu k, ak k je najvyššia súradnica v danej dimenzii a hrana spája vrcholy s najnižšou a najvyššou súradnicou dimenzie, alebo nižšia zo súradníc v ktorých sa líšia vrcholy spojené touto hranou.*

Hranu so súradnocou k budeme volať niekedy hrana v súradnici k. Rovnako budeme používať pojem dimenzia hrany a hrana dimenzie.

Označenie 12. *Prechodom po súradnici budeme volať prechod agenta po hrane danej súradnice.*

Označenie 13. *Susedná súradnica k súradnici k v dimenzii d, je každá súradnica vrchola, ktorý spája nejaká hrana s vrcholom so súradnicou k v grafe prislúchajúcom dimenzii d.*

Lema 16. *Portová hrana v súradnici k v cyklickej mriežke má v tvoriacom cykle odlišnú paritu počtu prechodov po súradnici k ako majú počty prechodov po ostatných súradniciach V ostatnej dimenzii majú všetky počty prechodov po jednotlivých súradniciach rovnakú paritu.*

Dôkaz. Aby otvoril druhý port hrany, agent musí svoju pozíciu zmeniť tak, aby sa nachádzal na susednej súradnici v diemnzii v ktorej sa líši s vrcholom s otvoreným prvým portom v dimenzii kam má viesť konštruovaná hrana. Keď rozdelíme pohyby agenta medi jednotlivé dimenzie, môžeme jeho pohyb premietnuť do grafov týchto dimenzií. Budeme teda sledovať len to ako sa mení tá, ktorá súradnica agenta pri jeho pohybe po tvoriacom cykle. Keďže vrcholy v ktorých otvára porty sa líšia iba v dimenzii konštruovanej hrany, v jednej dimenzii majú rovnakú súradnicu. Do grafu tejto dimenzie sa pohyb agenta premietne ako uzavretý sled, kde skončí vo vrchole, v ktorom začal. Poďme spočítať paritu prechodov po jednotlivých súradniciach v slede do ktorého sa v tomto grafe premietne prechod agenta. Vezmime tie výskyty toho istého vrcholu, medzi ktorými sú výskyty len iných vrcholov a to každého najviac raz. Takže aké pohyby mohol urobiť agent medzi dvomi najbližšími návštevami nejakého vrcholu? Mohol sa vrátiť zo susedného vrchola - zo vzdialenejšieho už nie, lebo susedný, cez ktorý by išiel by sa vyskytol na tomto slede viackrát ako raz. Alebo mohol agent prejsť po ceste, kde cieľom aj východiskom je tento vrchol. Takáto cesta je buď prázdna alebo obsahuje všetky iné vrcholy y je to vlastne prejdenie cyklu dookola. V prvom prípade sa parita prechodov po súradniciach zachová, lebo agent prešiel tam a spať po rovnakej súradnici. V druhom prípade sa zmení parita prechodov pre všetky súradnoce. V oboch prípadoch sa teda zachová rovnosť parít prechodov jednotlivých súradníc, ak platila aj predtým. Vieme teda, že medzi dvomi najbližšími výskytmi vrchola sa nič nepokazí. Na začiatku je počet prechodov pre každú súradnicu rovnaký - nulový a teda majú rovnakú paritu.Pri ďalšom počítaní môžeme teda tieto výskyty preskočiť - skontrahovať. Na takto upravenom slede postup opakujeme, až kým nemáme každý vrchol najviac raz okrem koncového a začiatočného, čo je ten istý. Preň platia rovnaké pravidlá a teda pre túto dimenziu je veta dokázaná.

V dimenzii konštruovanej hrany sa bude pri otváraní druhého portu nachádzať vo vrchole susednej súradnice k súradnici vrchola s prvým portom. Ak pridáme do sledu aj konštruovanú portovú hranu, dostaneme uzavretý sled, pre ktorý platí predošlý dôkaz. Z toho je jasné, že iba prechody po konštruovanej hrane majú odlišnú paritu počtu svojich výskytov. □

Veta 1. *Pri ľubovoľnej konštrukcii cyklickej mriežky G rozmerov $n \times m$ s jednou značkou pre porty, existuje aspoň jedna hrana pri konštrukcii ktorej agent vykoná aspoň $n-1$ krokov v prvej dimenzii a súčasne aspoň jedna portová hrana, pri konštrukcii ktorej agent vykoná aspoň $m-1$ krokov v druhej dimenzii, môže ísť o tú istú hranu.*

Dôkaz. Ako sme už povedali kostrové hrany sú hrany, ktoré vznikli operáciou NH a tieto hrany tvoria kostru grafu; ako bolo spomenuté grafom nazývame každý medziprodukt konštrukcie počnúc počiatočným vrcholom až po konečný skonštruovaný graf.

Každej konštrukcii grafu G zodpovedá nejaká kostra z kostrových hrán. Počet krokov, ktoré agent vykonal pri konštrukcii hrany h je súčet počtu krokov do vrchola v, v ktorom otvoril prvý port hrany a počet krokov z vrchola v do vrchola u, v ktorom otvoril druhý port hrany, tento sled z u do v tvoria s portovou hranou uzavretý sled. Tento sled obsahuje tvoriaci cyklus danej portovej hrany.

Majme ľubovoľnú konštrukciu cyklickej mriežky G a k nej zodpovedajúcu kostru. Ukážeme, že pre ňu platí tvrdenie vety. Táto kostra určuje aj to, ktoré hrany grafu sú portové - tie, ktoré nie sú v kostre. Pripomeňme si, že kostra zodpovedajúca konštrukcii je kostra tvorená práve všetkými hranami, ktoré vznikli operáciami NH.

Ukážeme, že existuje aspoň jedna hrana, ktorej tvorivý cyklus bez tejto hrany obsahuje jej súradnicu párny počet krát - teda ostatné súradnice v tejto dimenzii nepárny počet krát. Rovnako to dokážeme pre obe dimenzie. Možné sú dva prípdy, buď ide o dve rôzne hrany, alebo o jedinú hranu. Keďže agent musel prejsť do susednej súradnice vrchola s prvým portom hrany a nemohol tak urobiť v jej súradnici, musel prejsť po všetkých ostatných súradniciach a teda vykonať aspoň n-1 respektíve m-1 krokov v prípade, že platí alternatíva s dvomi rôznymi hranami. Alebo n+m - 1 ak platí alternatíva s jedinou hranou.

Pozrieme sa na ľubovoľnú kostru. V tejto kostre sa nachádzajú všetky vrcholy grafu. Aký je tvorivý cyklus jednotlivých portových hrán. V tvorivom cykle portovej hrany h je možné nahradiť niektorú portovú hranu jej tvorivou cestou. Takýmto nahrádzaním postupne vieme nahradiť všetky portové hrany. Keď z výsledného sledu vyberieme konštrukčnú cestu, tento už obsahuje iba kostrové hrany. Pozrime sa na paritu počtu prechodov jednotlivých súradníc v tejto ceste. Tvrdíme, že ak sme nenahrádzali hranu, ktorej tvorivá cesta obsahovala súradnicu tejto hrany párny počet krát, výsledná tvorivá cesta na kostrových hranách bude obsahovať nepárny počet prechodov po súradnici hrany h. Toto je zrejmé. Ak výskyt hrany nahradíme cestou, kde sa táto hrana nachádza nepárny počet krát, počet prechodov po súradnici hrany sa nezmení. Toto zároveň implikuje, že prechody po ostatných súradniciach sa na tejto ceste nachádzajú párny počet krát zvlášť pre každú súradnicu. Teda nezmení sa ani parita prechodov po ostatných súradniciach. Čo sa týka parít počtov prechodov v ostatnej dimenzii, tak tu platí to isté, ak nahrádzame hranu jej tvorivou cestou, kde je počet prechodov po súradnici tejto hrany párny, tak aj vo výslednom slede je počet prechodov párny pre každú súradnicu tejto dimenzie.

Keďže výsledný sled, krorý vznikol nahradením tvorivej cesty portovej hrany h vedie len po kostrových hranách a na tých neexistuje cyklus, tvorivá cesta hrany h vybraná z tohto sledu zachováva paritu prechodov po jednotlivých súradniciach takú, aká je v tomto slede. Kostra je totiž strom a ak sa na slede vyskytne dvakrát ten istý vrchol, tak medzi dvomi jeho susednými výskytmi je párny výskyt prechodov po jednotlivých súradniciach a teda ho môžeme zo sledu vybrať - skontrahovať na jeden výskyt tohoto vrchola. Jediná šanca ako pri pohybe po strome navštíviť opätovne niektorý vrchol je vrátiť sa doň.

Na kostre existuje pre každú portovú hranu jediná tvorivá cesta, inak by na nej bol cyklus, čo je spor. Vezmime si teraz dimenziu d a vrcholy, ktoré majú v ostatnej dimenzii rovnakú súradnicu, označme ju k. Indukovaný podgraf na týchto vrcholoch je cyklus. Niektoré jeho hrany sú kostrové, ostatné sú portové. Pokiaľ má každá portová hrana tvorivú cestu na kostre s nepárnym počtom prechodov po jej súradnici, existuje na kostre cyklus, čo je spor. Tento cyklus získame z uzavretého sledu vytvoreného z indukovaného grafu týchto vrcholov, kde portové hrany nahradíme ich tvorivými cestami na kostre. Vo výslednom slede sa každá hrana bude nachádzať nepárny počet krát a teda v ňom existuje cyklus, ktorý minimálne raz prejde po každej súradnici danej dimenzie.

Z toho vyplýva, že niektorá nahrádzaná hrana mala tvorivú cestu s párnym počtom prechodov po jej súradnici. Teda pre každú dimenziu existuje hrana, ktorej tvorivá cesta obsahuje nepárne počty prechodov po súradniciach tejto dimenzie. A agent pri jej konštrukcii musí vykonať aspoň $n-1$ krokov pre prvú dimenziu a $m-1$ iných krokov pre druhú dimenziu. □

Veta 2. *Mriežku $m \times n$ je možné s použitím jedinej značky pre port zostrojiť pomocou $3m(n-1)+m$ pohybov, kde $m <= n$.*

Dôkaz. Agent vykonáva: OP,NH,NH,NH,OP,(n-2)krát (OP,S1,NH,NH,OP),S1,NH,HN,OP čím vytvorí prvý riadok mriežky. Následne vykoná (m-1)krát (NH,OP,S1,S2,NH,OP,(n-1)krát (OP,S1,S2,NH,OP)) □

Domnievame sa, že ide o optimum. Teda že neexistuje konštrukcia, pri ktorej agent používa jedinú znakču pre porty a vykoná menej krokov.

Označenie 14. *Okraje mriežky sú vrcholy, ktoré majú jednu súradnicu buď minimálnu alebo maximálnu. Rohy mriežky sú vrcholy, ktoré majú maximálne alebo minimálne obe svoje súradnice. Rohy mriežky majú stupeň dva a vrcholy na okraji, ktoré nie sú rohmi, majú stupeň tri. Zvyšné vrcholy majú stupeň štyri a voláme ich vnútorné*

Keď prvý raz príde agent do vrchola, tak hrana po ktorej prišiel musí byť kostrová. Každé prerušenie ťahu tvoreného portovými hranami, znamená, že agent okrem krokov medzi otvorením prvého a druhého portu niektorej portovej hrany, musel vykonať aspoň jeden krok navyše, keď nebol otvorený žiadny port. Počet krokov, ktoré prešiel medzi konštrukciami dvoch hrán, nazvime cena prerušenia. Budeme za ňu počítať aj všetky kroky, ktoré vykoná medzi otvorením dvoch portov rovnakej hrany, ktoré presahujú počet tri kroky na portovú hranu. Toto nám umožní dať prednosť portovej hrane, konštruovanej na tri s následným jedným krokom pre otvorením portu nasledujúcej hrany, pred portovou hranou konštruovanou na päť krokov medzi konštrukciou jej dvoch portov.

Čím je menšia suma cien všetkých prerušení, na tým menej krokov agent skonštruoval mriežku. Vieme, že na každú portovú hranu musí agent vykonať aspoň tri kroky, tiež sme dokázali horný odhad potrebných krokov.

Prerušenie určite vznikne vo vrcholoch, ktoré majú nepárny počet portových hrán. Teda všetky vnútorné vrcholy, ktoré sú listami kostry tvoria prerušenia.

Pokiaľ agent skonštruoval hranu na tri kroky medzi otvorením jej portov, potom do vrchola s druhým portom príde agent po inej dimenzii, ako je tá, v ktorej vedie portová hrana. Ak agent pri konštruovaní portových hrán príde do okrajového vrchola má dve možnosti: buď preruší konštruovanie portových hrán alebo bude pokračovať po okraji. Každý súvislý ťah na okraji teda vytvára prerušenie a stojí aspoň jeden krok navyše.

Keďže vnútorné vrcholy sú stupňa štyri, ťah porotvých hrán sa nemôže v žiadnom vrchole pretínať, pretože každý vrchol má aspoň jednu hranu kostrovú. Taktiež v rohu je buď porotvý ťah prerušený, alebo doň porotvá hrana nevedie, v tom prípade však vedie do aspoň jedného s ním susedného vrchola a teda na okraj.

Poďme sa teraz pozrieť na okraje. Roh na okraji má stupeň tri, z čoho jedna hrana je nutne kostra. má teda tri možnosti: buď cez neho ťah porotvých hrán prechádza, porotvý ťah v ňom má koniec, portový ťah doň nevedie - má všetky tri incidentné hrany kostrové.

Pokiaľ sa nám podarí dokázať, že na veľa okrajových vrcholov pripadá prerušenie, dostaneme dolnú hranicu. Teda nemusí byť nutne v nich, ale že každému z nich viem priradiť iné. Aby sme dokázali, že nami predstavená horná hranica je aj dolnou, potrebujeme zarátať prerušenia aspoň $2m$ vrcholom, kde m je menší z rozmerov.

Portový ťah keď už vedie po okraji, tak z neho môže odísť len ak "zaplatí" aspoň dva kroky navyše. Na tri kroky totiž môže pokračovať len po okraji. Teda každému ťahu idúcemu po okraji môžeme zarátať jedno aspoň prerušenie.

Označenie 15. *Úsekom vrcholov ozančujeme množinu vrcholov, ktoré majú jednu dimenziu rovnakú a tvoria ťah.*

Úroveň je množina vrcholov s ronakou súradnicou v niektorej dimenzii.

Ak agent prišiel do vrchola na okraji, cez ktorý prechádza ťah, a neotvoril v ňom port, tak urobil dva kroky navyše, môžeme ich teda vynechať. Teda vrchol na okraji, cez ktorý prechádza ťah, bol skonštruovaný aj so svojou kostrovou hranou tesne pred otvorením portov pre ťahy v ňom. Z toho vyplýva, že vnútorný vrchol, s ktorým susedí, mal predtým stupeň tri alebo v ňom bolo prerušenie. Toto platí o všetkých vnútorných vrcholoch susediacich s ťahom na okraji; rátame susedov tých okrajových vrcholov, kde obe portové hrany idú po okraji. Teda máme úsek vnútorných vrcholov, ktoré boli stupňa tri alebo v nich je prerušenie.

Ak tieto vrcholy neexistovali pred konštrukciou ťahu po okraji, tak teraz sú stupňa tri a majú tri kostrové hrany. Teda štvrtá je buď kostrová alebo je v nich prerušenie. Všetky vrcholy, ktoré už existujú sú v kostre, lebo vznikli operáciou NH a touto operáciou sa môže agent dostať do vrchola raz. Ak teda existuje v grafe úsek vrcholov, ktoré sú stupňa tri potom do neho budú viesť kostrové hrany alebo portové. Každá porotvá je jedno prerušenie, pretože tieto vrcholy už sú stupňa tri. kostrové hrany môžu viesť iba do vrcholov, kotré ešte neexistujú.

Z úseku ešte neskonštruovaných vrcholov, ktorý susedí s úsekom vrcholov so stupňom tri v grafe, môže do tohto úseku viesť kostrová hrana alebo porotvá hrana. Pričom opäť každá portová hrana znamená prerušenie, lebo z vrcholov, ktoré už majú stupeň tri nebude môcť ťah pokračovať. Ak z niektorých dvoch budú viesť kostrové hrany, už nemôže viesť kostrová hrana medzi nimi. Pokiaľ sú to teda portové vrcholy a nemá v nich byť prerušenie, musí viesť kostrová hrana do druhého susedného úseku. Zvyšné dve hrany sú potom portové a tvoria ťah. Z tohto tiež vidno, ktorý úsek vznikol skôr a môžeme ich teda rozlíšiť podľa veku.

Poďme teraz hľadať prerušenia k vrcholom na okraji mriežky. Chceme ich nájsť $2m$ rôznych. Ak v okrajovom vrchole nie je prerušenie, má, ako sme ukázali vyššie, dve možnosti: buď ide portový ťah cez neho alebo cez je icidentný s tromi kostrovými hranami. Inak mu priradíme prerušenie ťahu v ňom. Medzi každým úsekom vrcholov s tromi kostrovými hranami je na okraji prerušenie. Na každý úsek ťahu na okraji pripadá jedno prerušenie.

Vrcholy na jendom okraji majú jednu súradnicu rovnakú a v druhej sa líšia. Každému z nich vieme priradiť stĺpec alebo riadok vrdholov, ktoré sa s ním zhodujú v tejto jednej súradnici. Ak vo vrchole nie je prerušenie, tak po tomto stĺpci z neho odchádza kostrová hrana, ak to nie je vrchol cez ktorý na okraj prišiel ťah z portových hrán. Takýto stĺpec musí byť niekde napojený na kostru, pokiaľ nezačína vo vrchole s tromi kostrovými hranami. Podľa kostry tiež vieme v akom poradí vrcholy vznikali. Každé vetvenie kostry znamená list na kostre.

Pokiaľ má úsek viac ako dve prerušenia, zvyšné môžeme zarátať iným úsekom. Tiež vieme, že všetky tieto portové hrany museli byť skonštruované na tri kroky. Konštrukciu hrany na päť krokov môžeme rátať ako štyri prerušenia, lebo medzi dvomi vrcholmi s prerušením rátame jedinú hranu.

Prvú portovú hranu do vrchola v mriežke vytvoril agent pri jeho prvej návšteve, alebo urobil najmenej jeden krok navyše nad tri kroky na portovú hranu.

Ak je v mriežke menej prerušení ako dve na každý vrchol kratšieho okraja, tak z dirichletovho princípu musí existovať úsek vrcholov dlhý ako väčší rozmer mriežky, ktorý obsahuje najviac jedno prerušenie. Vrcholy, ktoré nevznikli tesne pred portovými hranami, ktoré z nich vedú, znamenajú krok navyše a teda "prerušenie". Majme teda takýto úsek s najviac jedným prerušením a zvyšné úseky s najviac dvomi prerušeniami na úsek.

Spôsob akým sú pospájané vrcholy úseku čiastočne určuje, ako budú pospájané vrcholy v susedných úsekoch. Navyše požiadavka, aby bolo na úseku najviac jedno prerušenie a na zvyšných najviac po dve na úsek, značne limituje možnosti pospájania vrcholov. Ak sa v nierktorom úseku vyskytne viac ako dve prerušenia, musia existovať ďalšie úseky s menej ako dvomi prerušeniami. Teda môžeme predpokladať, že hľadáme také pospájanie úseku, že si nevynucuje zvýšený počet prerušení v iných úsekoch. Úsekov je totiž obmedzený počet a niektoré si skrátka už nebudú mať kde vynucovať viac prerušení.

Nech teda existuje úsek s najviac jedným prerušením. Hľadáme také vedenie hrán do neho, že z neho nevyplýva zvýšenie počtu prerušení v ostatných úsekoch. Keďže má dva okrajové vrcholy jeden musí byť incidentný s dvomi portovými hranami. Ak by sa na tomto úseku vyskytol vnútorný vrchol so štyrmi kostrovými hranami, museli by kostrové hrany viesť medzi všetkými susednými vrcholmi úseku, až po okraj minimálne na jednej jeho strane, lebo vnútorný vrchol ako list kostra je prerušenie.

Pozrime sa teraz na úsek idúci po dlhšom okraji mriežky. Aj naňho pripadajú najviac dve prerušenia. Ak ich obsahuje viac, tak musia za každé ďalšie existovať ďalšie úseky s jediným prerušením, alebo za každé dve prerušenia jeden úsek bez neho. Z konca vrchola na okraji tiež vieme určiť, z ktorého smeru doň prišiel portový ťah, keďže bez prerušenia sa opustiť okraj nedá.

Ak je niektorý vnútorný vrchol list kostry susedí len s vrcholmi, ktoré majú stupeň portových hrán dva, môžeme ho rátať za dve prerušenia. Buď je totiž najbližší koniec ťahu ďaleko alebo je niekde v susednom vrchole a potom je prerušenie aj v ňom.

Na dôkaz, že potrebujeme aspoň $2m$ prerušení nestačí statický pohľad. Teda cez ktoré vrcholy a ako prechádza ťah z portových hrán. Napríklad mriežka 3×3 má kostru pri ktorej tvoria portové hrany súvislý ťah. Napriek tomu ju agent nevie skonštruovať na 12 krokov a potrebuje trinásty.

Preto bude v dôkaze potrebné využiť aj algoritmickú stránku veci a pozrieť sa na to, ako sa správa ťah z porotvých hrán pri konštrukcii a ako jeho úseky "ukladať" vedľa seba tak, aby pokryli mriežku. Takiež na to, koľko ťahov na to treba a aká je vzdialenosť koncového bodu predošlého od začiatočného bodu nasledujúceho podľa postupnosti konštrukcie týchto ťahov.

TODO

Kapitola 7

N-rozmerná kocka

V tejto kapitole si v úvode pripomenieme niekoľko faktov o grafoch - hyperkockách dôležitých pre pochopenie zvyšku kapitoly. V ňom sa venujeme výsledkom výskumu konštrukcie grafov z triedy hyperkociek v našom modeli.

Vrcholy spojené hranou sa líšia práve v jednej súradnici. Každej hrane z vrcholu možno priradiť súradnicu v ktorej sa jej druhý vrchol od tohto odlišuje. Toto priradenie je vzhľadom na hrany incidentné s jedným vrcholom injekcia.

Susedia vrchola sa navzájom líšia v práve dvoch súradniciach. Polohu agenta môžeme vyjadriť vrcholom, v ktorom sa práve nachádza. Jeho polohe teda zodpovedá binárny reťazec, ktorého hodnoty bitov budeme volať súradnice. Súradnica v hyperkocke môže mať hodnotu 0 alebo 1. Po dvoch zmenách je súradnica rovnaká, ako na začiatku. Jedným pohybom po hrane mení agent práve jednu súradnicu svojej pozície. Z tohto vyplývajú obmedzenia pre úsporu pohybov zväčšovaním množiny značiek pre porty. Napríklad pri použití dvoch značiek pre porty potrebuje agent na vytvorenie dvoch nových portových hrán aspoň 5 ťahov. Úspora nastáva, keď port pre druhú hranu otvorí už počas cesty za otvorením portu prvej hrany.

Graf indukovaný podmnožinou bodov hyperkocky má počet hrán závislý na ich vzájomnej polohe. Ak má agent dosť veľkú množinu značiek pre porty, môže tieto vrcholy prejsť v ľubovoľnom poradí a v každom otvoriť potrebný počet hrán. Agent nedokáže skonštruovať graf, ktorý by nebol súvislý v nejakej fáze svojej konštrukcie.

V tejto kapitole skúmame možnosti efektívnej konštrukcie hyperkociek rôznych rozmerov. Agent je pri konštrukcii limitovaný veľkosťou množiny značiek pre porty, ktorú môže použiť. Počet hrán, ktoré agent vytvorí jednotlivými operáciami je pre daný graf konštantný, ako sme ukázali vo všeobecnej kapitole o konštrukcii grafu. Efektivitu algoritmu meriame počtom krokov, ktoré agent musí pri konštrukcii vykonať. Za krok sa považuje zmena vrchola, v ktorom sa agent nachádza a teda zmena polohy agenta.

K hlavným výsledkom v tejto kapitole patrí postupnosť vrcholov hyperkocky, ktorá obsahuje najvyšším možný počet hrán v podgrafe indukovanom takto veľkou množinou vrcholov hyperkocky. Zároveň sme dokázali, že toto platí aj o každej podpostupnosti vrcholov hyperkocky, ktorá je jej prefixom.

Ak má agent k dispozícii dostatočne veľkú množinu značiek pre porty, dokáže skonštruovať hyperkocku tak, že každý vrchol navštívi práve raz - pri jeho vytvorení. Veľkosť množiny potrebných značiek pri takejto konštrukcii vieme určiť zo spomenutej postupnosti. Počet hrán, ktoré vychádzajú z vrchola hyperkocky je pre každý vrchol danej hyperkocky rovnaký a teda aj počet portov, ktoré agent musí otvoriť pri jeho návšteve, aby sa doň už nemusel vracať. Akonáhle vznikne portová hrana, značka jej portov sa uvoľní a agent ju môže znovu použiť. Počet portov, ktoré musia byť otvorené v danom momente určuje počet potrebných značiek, pretože žiadne dva porty nemôžu mať rovnakú značku, inak by medzi nimi vznikla hrana. Teda čím viac hrán vedie medzi doteraz prejdenými vrcholmi,tým menej je otvorených portov a tým menej je aj používaných značiek.

Z tejto optimálnej postupnosti vrcholov pre konštrukciu hyperkocky jedným ťahom je možné zistiť aj počet potrebných značiek pre porty. Je ním maximum rozdielu súčtu stupňov prejdených vrcholov a počtu hrán vedúcich medzi nimi vo vrcholoch postupnosti.

Na základe vlastností tejto postupnosti je tiež možné skúmať možnosti efektívnej konštrukcie hyperkocky v modeloch s nižším počtom značiek pre porty ako je potrebných na to, aby agent navštívil každý vrchol práve raz.

7.1 Všeobecné výsledky

Veta 3. *Pri konštrukcii n-rozmernej hyperkocky na čo najmenší počet krokov, kedy navštívi každý vrchol práve raz, potrebuje agent pre nepárne n* $\frac{2^{n+1}-1}{3}$ *a pre párne n* $\frac{2^{n+1}-2}{3}$ *značiek pre porty.*

Lema 17. *Dva vrcholy, ktoré sa navzájom líšia v práve dvoch súradniciach majú práve dvoch spoločných susedov.*

Dôkaz. Susedné vrcholy sa líšia práve v jednej súradnici. Spoločný sused dvoch vrcholov sa líši v práve jednej súradnici od každého z nich. Teda má všetky ostatné súradnice rovnaké ako jeden z nich a líši sa od neho v jednej z tých dvoch. Teda má túto súradnicu rovnakú ako druhý vrchol. Druhú z týchto súradníc má potom odlišnú od druhého vrchola. Čiže sa od každého z nich líši v práve jednej súradnici. Toto môžu spĺňať práve dva rôzne vrcholy, ktoré susedia z oboma. □

Otázkou ostáva, koľko hrán vie agent vytvoriť počas jednej návštevy vrcholu otváraním portov?

Lema 18. *Pri jednej návšteve vrcholu po prejdení sledu dĺžky n vrcholov (vrátane toho v ktorom sa práve nachádza) môže agent vytvoriť najviac* $(fracn - 12) > \#dimenzii$ *hrán otvorením portu.*

Poznámka 6. *Rátajú sa hrany, ktoré vzniknú pri tejto návšteve tj. pri ktorých agent otvorí druhý port.*

Dôkaz. Je zrejmé: že vo všetkých vrcholoch, kam majú tieto hrany viesť, už agent bol; medzi dvomi vrcholmi vedie najviac jedna hrana. Z tohto a predošlej lemy vyplýva, že po návšteve n vrcholov môže agent otvorením portu vytvoriť z vrcholu, kde sa nachádza, najviac $(\frac{n-1}{2}) > \#dimenzii$ hrán. Z týchto $(\frac{n-1}{2})$ vrcholov sa od neho každý líši v inej súradnici, lebo ak by sa dva líšili v rovnakej, museli by to byť totožné vrcholy. □

Teraz ukážeme, že taká postupnosť vrcholov existuje.

Lema 19. *Existuje taká postupnosť n vrcholov, že pri návšteve posledného agent vytvorí* $(\frac{n-1}{2}) > \#dimenzii$ *hrán.*

Dôkaz. Ak vezmeme takúto postupnosť odzadu, vyzerala by až na substitúciu dimenzií takto: 1, 2, 1, 3, 2, 4, 3, 5, 4, 6, 5, ... □

Avšak ľahko vidieť, že žiadne iné portové hrany by medzi vrcholmi v tejto postupnosti vzniknúť nemohli. Taktiež porty sú otvorené celý čas až kým agent nepríde do vrcholu, kam vedú všetky tieto hrany. Teda agent potrebuje vždy novú značku. My však hľadáme takú postupnosť konštrukcie vrcholov, kde sú porty otvorené čo najkratšie, aby agent mohol potom znovu použiť ich značky. Skúsme teda nájsť postupnosť (cestu v hyperkocke) danej dĺžky s najväčším možný počtom hrán medzi jej vrcholmi.

Označenie 16. *O hrane povieme, že vedie v dimenzii k, ak sa vrcholy, ktoré spája, líšia v práve tejto dimenzii.*

Označenie 17. *Poloha agenta je vyjadrená súradnicami v jednotlivých dimenziách kocky. v každej dimenzii môže nadobúdať jednu z dvoch súradníc. Označme ich* 0 *a* 1. *Pri prechode po hrane agent preklopí jednu z týchto súradníc v práve jednej dimenzii na druhú.*

Je jasné, že v akomkoľvek súvislom slede vrcholov povedie hrana medzi dvojicami nasledujúcich vrcholov, ide o hrany po ktorých agent prejde z vrcholu do vrcholu. Portová hrana môže vzniknúť medzi vrcholmi, ktoré sa líšia v práve jednej súradnici, lebo iné hrany v hyperkocke nie sú.

Lema 20. *V ľubovoľnej postupnosti hrán medzi vrcholmi hyperkocky spojenými hranou, sa nachádza párny počet hrán v každej dimenzii okrem dimenzie v ktorej sa tieto vrcholy líšia a teda v nej vedie hrana medzi nimi.*

Dôkaz. Takéto dva vrcholy sa líšia v práve jednej dimenzii. Ak agent zmenil pri pohybe svoju polohu v nejakej dimenzii nepárny počet krát, tak je v tejto dimenzii v inej polohe ako bol na začiatku. Ak to nie je dimenzia v ktorej sa líši koncový vrchol od začiatočného - agent sa nenachádza v koncovom vrchole, lebo polohy agenta a tohto vrcholu nie sú rovnaké. □

Označenie 18. *Ak viem súradnice vrchola, môžem postupnosť ťahov agenta zapísať ako postupnosť zmien jeho súradníc v jednotlivých dimenziách.*

Pripomeniem, že hľadáme najkratší sled v hyperkocke, prejdením ktorého mohol agent skonštruovať medzi jeho vrcholmi s danou množinou značiek daný počet portových hrán. Čím viac hrán agent skonštruuje pri jednej návšteve vrcholu, tým menej krát ho bude musieť navštíviť a tým menej krokov sa započíta do konštrukcie hyperkocky. Hľadáme najnižší počet značiek pre porty, ktoré agent potrebuje na konštrukciu hyperkocky, pri ktorej navštívi každý vrchol len raz. V každom vrchole, ktorý agent navštívi otvorí rovnako veľa portov. Preto hľadáme spôsob, ako čo najviac portov čím skôr pozatvárať, aby agent mohol ich značky opäť použiť.

Lema 21. *S každou hyperkockou v ktorej vrchol neleží ho spája najviac jedna hrana. Táto hrana vedie v inej dimenzii ako sú dimenzie tejto hyperkocky.*

Dôkaz. Každý vrchol hyperkocky má hrany vo všetkých jej dimenziách. Vo zvyšných dimenziách sú vrcholy hyperkocky totožné, lebo sa dva susedné líšia práve v jednej. Vrchol mimo hyperkocky spojený s vrcholom v nej sa od neho líši v inej dimenzii a v tejto sa líši teda aj od všetkých ostatných. Zároveň je vo všetkých dimenziách hyperkocky totožný s vrcholom z nej a teda sa líši v aspoň jednej dimenzii hyperkocky od jej ostatných vrcholov a má teda s nimi aspoň dve rozdielne dimenzie. Nemôže ho preto s nimi spájať hrana. □

Lema 22. *Bijektívna substitúcia dimenzií na všetkých vrcholoch podgrafu hyperkocky zachováva hrany medzi vrcholmi tohto podgrafu. Rovnako aj preklopenie súradníc v niektorej dimenzii.*

Dôkaz. Ak sa dva vrcholy líšia práve v jednej dimenzii, budú sa v práve jednej dimenzii líšiť aj po tejto substitúcii, lebo dimenzie v ktorých boli rovnaké sa zobrazia na rovnaké a tá v ktorej sa líšili sa u oboch zobrazí na rovnakú a budú sa v nej líšiť aj naďalej; teda medzi nimi povedie hrana. Pri preklopení súradníc sa medzi vrcholmi zachovajú rovnosti a rozdiely v danej dimenzii, teda aj hrany. □

Lema 23. *Vrcholy súvislého podgrafu hyperkocky s počtom vrcholov m sa navzájom líšia v najviac m-1 súradniciach.*

Dôkaz. Vezmime kostru tohto podgrafu. Táto má m-1 hrán. Každá vedie v niektorej dimenzii. Cesta v tejto kostre udáva v ktorých dimenziách sa daná dvojica jej koncových vrcholov líši. Všetky cesty v tejto kostre idú cez najviac m-1 rôznych dimenzií, čo je jej počet hrán. V ostatných dimenziách sa dané vrcholy nelíšia, majú v nich teda navzájom rovnaké súradnice. □

Nasledujúca lema hovorí, že množina vrcholov hyperkocky je súvislá. Z toho vyplýva, že agent je schopný skonštruovať indukovaný graf týchto vrcholov.

Lema 24. *Podmnožina m vrcholov hyperkocky, ktoré indukujú graf s najvyšším počtom hrán je súvislá.*

Dôkaz. Nech existujú dva komponenty. Na jednom z nich urobíme postupné popreklápanie súradníc tak, aby susedil niektorý jeho bod s bodom v druhom komponente a teda medzi nimi viedla hrana. Čiže vyberieme jednu dimenziu v ktorej sa líšia a vo všetkých ostatných dimenziách, kde majú rozdielne súradnice preklápame postupne súradnice jedného z vrcholov. V prípade, že počas tohto postupu vznikne hrana spájajúca vyhrané dva komponenty, skončíme skôr. Pôvodné hrany sa zachovajú a jedna nová pribudne, čo je spor s najvyšším počtom hrán. Tieto kroky môžeme urobiť bez toho, aby sa dva vrcholy zobrazili na seba. Ak budeme robiť preklápanie súradníc po jednotlivých dimenziách, v tom prípade, ak by sa nejaký vrchol transformovaného komponentu podgrafu zobrazil na niektorý vrchol druhého komponentu, musela by medzi nimi viesť predtým hrana, teda stav v ktorom by boli komponenty prepojené a preklápanie by teda už nepokračovalo. □

Potrebujeme zistiť ako vybrať m vrcholový indukovaný podgraf hyperkocky tak, aby počet hrán v ňom bol maximálny možný. Po konštrukcii tohto podgrafu, bude totiž otvorený najnižší možný počet portov pre dané m, ak sa agent už nebude vracať do jeho vrcholov.

Teraz ukážeme, že vieme z vrcholov hyperkocky vybrať pre každé m m-vrcholovú podmnožinu, ktorej indukovaný graf obsahuje pre dané m najvyšší možný počet hrán tak, aby obsahovala ľubovoľnej podmnožiny s touto vlastnosťou.

Ďalej ukážeme, že agent môže vrcholy tejto množiny prechádzať v takom poradí, aby prejdené vrcholy vytvárali v každom momente takúto množinu. Pri takomto prechádzaní vrcholov počas konštrukcie bude vždy použitý najmenší možný počet značiek pre porty. Počet portov, ktoré agent dovtedy otvoril je pre danú hyperkocku a počet už prejdených vrcholov vždy rovnaký a teda otvorené sú tie, ktoré sa ešte nezavreli, teda ktoré nepatrili hrane medzi doteraz prejdenými vrcholmi.

Lema 25. *Vrcholy podgrafu hyperkocky sa dajú rozdeliť na dve množiny A,B podľa súradnice vo vybranej dimenzie. Počet hrán vedúcich z vrcholov v jednej množine do vrcholov v druhej množine je rovný alebo menší ako mohutnosť menšej z nich. Čiže* $|\{e|e = (a, b); a \in A; b \in B\}| <= min\{|A|, |B|\}$

Dôkaz. Každý vrchol má v jednej dimenzii najviac jednu hranu a hrana vedie v danej dimenzii medzi dvomi vrcholmi vtedy ak je to jediná dimenzia v ktorej majú rozdielne súradnice. V inej dimenzii medzi A a B hrana viesť nemôže, lebo by musela viesť medzi vrcholmi líšiacimi sa v najmenej dvoch dimenziách - v dimenzii hrany a v dimenzii podľa súradníc ktorej sú A a B vytvorené. Teda hrany medzi A a B môžu viesť len v jednej dimenzii a do menšej z A,B môže viesť najviac toľko hrán v jednej dimenzii, koľko má vrcholov. □

Lema 26. *Keď vezmeme z podgrafu hyperkocky len vrcholy, ktoré majú hranu v jednej vybranej dimenzii a označíme nimi indukovaný podgraf G. Vieme vrcholy G rozdeliť na dva izomorfné indukované podgrafy hyperkocky* G_1 *a* G_2 *podľa súradnice vrchola v tejto dimenzii.*

Dôkaz. Vrcholy, medzi ktorými vedie hrana sa líšia v práve jednej dimenzii, v ostatných dimenziách sú rovnaké. Keď teda vezmeme hrany vedúce medzi G_1 a G_2 a vytvoríme podľa nich bijektívne zobrazenie medzi týmito podgrafmi tak, aby sa dva vrcholy medzi ktorými vedie hrana vo vybranej dimenzii zobrazili na seba, dostaneme izomorfizmus. Ak totiž viedla medzi dvomi vrcholmi hrana, bude hrana viesť aj medzi ich obrazmi, lebo tieto majú rovnaké súradnice ako ich vzory vo všetkých dimenziách okrem jednej, kde sa oba obrazy od svojich vzorov líšia a teda majú súradnicu v tejto dimenzii rovnakú; obaja sú v komponente, kde majú túto súradnicu rovnakú všetky vrcholy. Ak sa teda líšili v jednej súradnici vzory, budú sa aj obrazy a ak sa líšili vzory vo viacerých súradniciach, budú sa v rovnakých súradniciach líšiť aj obrazy. Zobrazenie je teda izomorfizmus. □

Lema 27. *Existuje indukovaný podgraf G hyperkocky na m vrcholoch, ktorý je možné postupne rozdeliť podľa dimenzií až po jednovrcholové indukované podgrafy tak, že v každom kroku rozdelíme vrcholy každého podgrafu s počtom vrcholov aspoň dva do dvoch podmnožín A a B. Takých, že* $|A| - |B| <= 1$.

Dôkaz. Dôkaz bude konštrukčný. Budeme postupne deliť m vrcholovú množinu i ďalšie z nej vzniknuté množiny na menšie, až kým nedostaneme m jednoprvkových množín vrcholov. Pritom budeme postupne určovať súradnice jednotlivých vrcholov.

Budeme postupovať postupne po dimenziách, až kým budú všetky množiny jednoprvkové. Vezmeme množinu M, rozdelíme ju na dve časti M_1, M_0 také že $|M_1| - |M_0| <=$

1. Vrcholy v M_1 budú mať v tejto dimenzii súradnicu 1 a vrcholy v M_0 budú mať v tejto dimenzii súradnicu 0. Ak je množina jednoprvková a delenie niektorých iných ešte pokračuje, dostane v každej ďalšej dimenzii súradnicu 1. Ak má kocka viac dimenzií ako je potrebných, aby obsiahla všetkých m vrcholov podgrafu, tak môžu dostať tieto vrcholy v týchto zvyšných dimenziách ľubovoľné súradnice, ale pre všetky vrcholy musí byť súradnica v každej z týchto dimenzií rovnaká. Tým sme zistili súradnice vrcholov vo všetkých dimenziách, podľa ktorých podgraf delíme. Pre účely tejto vety na súradniciach ostatných dimenzií nezáleží a môžu byť ľubovoľné. □

Lema 28. *Pri delení z lemy 27 vedie medzi množinami vrcholov M_1 a M_0, ktoré vznikli rozdelením jednej množiny M podľa tohto delenia, práve toľko hrán, ako je mohutnosť menšej z týchto množín. Pokiaľ, ak má kocka viac dimenzií ako je potrebných, aby obsiahla všetkých m vrcholov podgrafu, tak môžu dostať tieto vrcholy v týchto zvyšných dimenziách ľubovoľné súradnice, ale pre všetky vrcholy musí byť, na rozdiel od predošlej vety, súradnica v každej z týchto dimenzií rovnaká.*

Dôkaz. Stačí ukázať, že menší z indukovaných podgrafov má vo väčšom izomorfný obraz podľa vety 26. Vtedy vedie z každého vrchola v menšom podgrafe hrana do nejakého vrchola vo väčšom podgrafe a týchto hrán je presne toľko, koľko je vrcholov menšieho podgrafu. Teda potrebujeme dokázať, že vo väčšom podgrafe existuje skupina vrcholov, ktorej budú súradnice v dimenziách pridelené rovnako ako vrcholom v menšom podgrafe. Prideľovanie súradníc je deterministické, teda rovnako veľkým množinám vrcholov v rovnakej fáze budú poprideľované súradnice vo zvyšných dimenziách rovnako. Na menší a väčší podgraf sa rozdelia grafy s nepárnym počtom vrcholov. Použijeme úplnú indukciu. Báza indukcie bude množina veľkosti tri. Menšie podgrafy sú už len veľkosti dva a jedna. Jednovrcholový sa už ďalej nedelí a dvojvrcholový sa rozdelí na dva jednovrcholové. Predpokladáme, že pre delenie množín s mohutnosťou menšou ako n platí veta. Dokážeme, že potom musí platiť aj pre množinu M mohutnosti n: Ak je n párne, dá sa množina rozdeliť na dve množiny rovnakej mohutnosti, ktorých vrcholom sa zvyšné dimenzie poprideľujú rovnako. Teda každý vrchol v jednej bude mať iný vrchol v druhej množine s ktorým sa bude líšiť v práve jednej dimenzii a to v tej podľa ktorej sme M rozdelili. Ak n je nepárne, vzniknú dve množiny M_1, M_2. Obe z nich už podľa IP spĺňajú tvrdenie vety. Rozdelia sa každá na dve množiny, pričom z množiny s párnou mohutnosťou vzniknú dve rovnako veľké množiny M_{11},M_{12} a množina M_{21} tejto veľkosti vznikne aj z množiny s nepárnou mohutnosťou. Druhá množina M_{22} z tejto množiny bude mať o prvok viac alebo menej. Prvé tri množiny sa už budú deliť rovnako, teda viem povyberať trojice vrcholov, každý z inej množiny, ktoré budú mať vo zvyšných dimenziách rovnaké súradnice. Medzi dvomi vrcholmi z M_{11},M_{12}, označme ich m_{11},m_{12}, povedie hrana, lebo sa líšia práve v jednej dimenzii. Tretí vrchol, označme ho m_{21} bude spojený hranou s niektorý z tých dvoch, pretože pri druhom delení jeden z nich dostal rovnakú súradnicu ako on v dimenzii druhého delenia a teda sa líšia iba v dimenzii prvého delenia množiny M. Ak je prvok m_{21} spojený s prvkom z M_{22}, označme ho m_{22}, potom je m_{22} spojený aj s tým prvkom z dvojice m_{11},m_{12}, s ktorým nie je spojený m_{21}. Prvok m_{22} neexistuje iba vtedy, ak $|M_{22}| < |M_{21}|$. Ak $|M_{22}| > |M_{21}|$ potom v M_{22} majú všetky vrcholy z M_{21} izomorfný obraz a teda aj príslušná časť z párnej množiny. Z horeuvedeného vyplýva, že menšia množina z prvého delenia má obraz, izomorfný a spĺňajúci podmienky vety 26, vo väčšej množine vzniknutej z tohto delenia a keďže medzi vzorom a obrazom vedie hrana, veta platí. □

Teraz dokážeme, že neexistuje podmnožina vrcholov hyperkocky, vnútri ktorej by viedlo viac hrán, ako v nami predstavenej množine. Snahou je nájsť pre danú mohutnosť takú množinu vrcholov hyperkocky, z ktorej odchádza najmenší počet hrán. Vtedy totiž vedie najväčší možný počet hrán vnútri nej a agent má potom k dispozícii viac nepoužívaných značiek.

Veta 4. *Maximálny počet hrán, ktoré môžu viesť v podgrafe hyperkocky indukovanom m bodmi je daný postupnosťou a_m, kde $a_1 = 0; a_2 = 1;$ a pre $\forall i < 2; a_i = \lfloor \frac{i}{2} \rfloor + a_{\lfloor \frac{i}{2} \rfloor} + a_{\lceil \frac{i}{2} \rceil}$*

Dôkaz. V dôkaze použijeme úplnú indukciu. Báza indukcie je očividná: hyperkocka nemá slučky a teda v jej jednovrcholovom indukovanom grafe nevedie hrana a hyperkocka nemá násobné hrany, teda v jej indukovanom dvojvrcholovom podgrafe môže viesť najviac jedna hrana. Predpokladáme, že pre všetky členy s indexom menším ako m platí, že udávajú najväčší možný počet hrán v indukovanom podgrafe hyperkocky zo všetkých množín jej bodov, ktorých mohutnosť je rovná indexu daného člena postupnosti. ďalej predpokladáme, že pre každého člena postupnosti s menším indexom platí tvrdenie vety. Ukážeme, že existuje indukovaný podgraf hyperkocky s m bodmi s najvyšším možným počtom hrán G taký, ktorý niektorá dimenzia delí na polovice G_1 a G_2, teda na množiny vrcholov, ktorých mohutnosti sa líšia najviac o jedna. To vyplýva z toho, že neexistuje taký podgraf hyperkocky, ktorý by delila inak a mal by zároveň vyšší počet hrán. Nech taký podgraf G' existuje. Potom ho vybraná dimenzia delí na podgrafy s počtom bodov a_i, a_j. Keďže sú obe neprázdne, sú menšie ako m a teda pre ne platí indukčný predpoklad. $a_i = \lfloor \frac{i}{2} \rfloor + a_{\lfloor \frac{i}{2} \rfloor} + a_{\lceil \frac{i}{2} \rceil}$ a $a_j = \lfloor \frac{j}{2} \rfloor + a_{\lfloor \frac{j}{2} \rfloor} + a_{\lceil \frac{j}{2} \rceil}$ Bez ujmy na všeobecnosti, nech i < j. Aby sme zjednodušili indexovanie označme $a_{\lfloor \frac{i}{2} \rfloor} = a_{i1}; a_{\lceil \frac{i}{2} \rceil} = a_{i2}; a_{\lfloor \frac{j}{2} \rfloor} = a_{j1}$ a $a_{\lceil \frac{j}{2} \rceil} = a_{j2}$. Je zrejmé, že $a_{i2} - a_{i1} <= 1$ a $a_{j2} - a_{j1} <= 1$. Ak teda rozdelíme G podľa dimenzie, ktorá ho delí na G_1 a G_2, ich mohutnosti budú bez ujmy na všeobecnosti $a_{i2} + a_{j1}; a_{j2} + a_{i1}$. Z IP zároveň platí, že G_1 a G_2 majú aspoň tak dobré rozdelenie ako na množiny veľkostí a_{i2}, a_{j1} respektíve a_{j2}, a_{i1}. Počet hrán, ktoré vedú vo podgrafe G' je teda $a_{m'} = i + \lfloor \frac{i}{2} \rfloor + \lfloor \frac{j}{2} \rfloor + a_{i1} + a_{i2} + a_{j1} + a_{j2}$.

Je zrejmé, že menšia z množín na ktoré sa rozdelí G bude mať veľkosť $\lfloor \frac{m}{2} \rfloor$ a v týchto podmnožinách vedie aspoň $a_{i1} + a_{j2} + \lfloor \frac{i}{2} \rfloor$ respektíve $a_{j1} + a_{i2} + \lceil \frac{i}{2} \rceil$ hrán. Čiže $a_m >= \lfloor \frac{m}{2} \rfloor + a_{i1} + a_{j2} + \lfloor \frac{i}{2} \rfloor + a_{j1} + a_{i2} + \lceil \frac{i}{2} \rceil$ Z toho si ľahko ukážeme, že $a_m >= a_{m'}$. $a_m - a_{m'} >= \lfloor \frac{m}{2} \rfloor + \lceil \frac{i}{2} \rceil - i - \lfloor \frac{j}{2} \rfloor$ a keďže $\lfloor \frac{m}{2} \rfloor = \lfloor \frac{i+j}{2} \rfloor >= \lfloor \frac{i}{2} \rfloor + \lfloor \frac{j}{2} \rfloor$ potom platí, že $a_m - a_{m'} >= 0$. Tým sme dokázali, že neexistuje lepšie rozdelenie podgrafu hyperkocky indukovaného m vrcholmi po dimenziách ako keď sa rozdelí na polovice. Lepšie sa berie s ohľadom na počet hrán, ktoré vedú medzi jednotlivými množinami vrcholov v každom uzle delenia. Z predošlých viet vyplýva, že takéto rozdelenie možné je. □

Lema 29. *Zoberme jednu dimenziu a rozdeľme body na dve množiny, podľa toho, či v nej majú 1 alebo 0. Platia dve veci: medzi týmito dvomi množinami vedie najviac toľko hrán, koľko má menšia z nich vrcholov; v iných dimenziách už medzi týmito vrcholmi hrany nevedú.*

Dôkaz. Prvá vec vyplýva z toho, že každý vrchol má v jednej dimenzii najviac jednu hranu (pri nekompletnej hyperkocke, pri kompletnej práve jednu). Druhá vyplýva z toho, že hrana spája vrcholy, ktoré sa líšia práve v jednej súradnici a vedie v dimenzii tejto súradnice. Ak by mala viesť medzi týmito dvomi množinami hrana v nejakej inej súradnici, musela by spájať vrcholy, ktoré sa líšia v najmenej dvoch súradniciach (v tej ktorá rozdeľuje vrcholy na dve množiny a v tej v ktorej vedie táto hrana), čo nemôže. □

Už sme našli spôsob, ako zostrojiť množinu vrcholov hyperkocky, vnútri ktorej povedie najvyšší možný počet hrán. Teraz ešte ostáva nájsť spôsob, ako ju má agent skonštruovať tak, aby potreboval čo najnižší počet značiek. Na to potrebuje, aby vznikali portové hrany čo najskôr od otvorenia ich prvého portu. Teraz si ukážeme ako vyzerá postupnosť vrcholov medzi ktorými vedie najvyšší možný počet hrán a zároveň to platí aj o každom jej prefixe, teda podpostupnosti z jej prvých k prvkov.

Lema 30. *Pre* $\forall n, k : n < k$ *platí, že existuje taká postupnosť n vrcholov hyperkocky, medzi ktorými vedie najvyšší možný počet hrán, že medzi prvými k vrcholmi tejto postupnosti vedie najvyšší možný počet hrán.*

Dôkaz. Veta 4 a predošlé vety hovoria, ako vyzerajú množiny vrcholov, v ktorých vedie najvyšší možný počet hrán. Veta 28 hovorí o počte hrán, ktoré vedú medzi dvomi množinami vrcholov rozdelenými podľa nejakej dimenzie. Veta 26 zas hovorí, že menšia množina má vo väčšej izomorfný obraz, v ktorom má každý vrchol menšej dvojice vo väčšej susedný vrchol, s ktorým má všetky ostatné dimenzie rovnaké.

Potrebujeme nájsť takú postupnosť prechádzania tejto množiny vrcholov agentom, že v každom momente bude medzi dovtedy prejdenou množina vrcholov najvyšší možný počet hrán pre danú množinu vrcholov. Teda potrebujeme, aby prejdená množina spĺňala tvrdenie vety 27 a vety 28, teda aby sa dala rozdeliť po dimenziách na polovice +/- 1 vrchol a pritom, aby medzi týmito polovicami viedol počet hrán rovný mohutnosti menšej z nich.

Majme množinu n vrcholov z vety 4, teda v jej indukovanom grafe vedie najvyšší možný počet hrán. Keď ju delíme podľa vety 28, tak získame strom delenia. Pri každom vetvení bola od vrchola v ňom "odrezaná" najviac jedna hrana; ak bol v menšej množine, tak práve jedna. Ak je jedna množina väčšia, tak v nej existuje vrchol, ktorý nebol spojený s vrcholom v menšej. My potrebujeme nájsť vrchol, po ktorého odobratí získame opäť množinu s najvyšším počtom hrán pre danú mohutnosť.

Súradnice vrchola sú presne určené jeho polohou v strome delenia podľa dimenzií, keďže vo vete 28 sme určili všetky súradnice, na ktorých záleží (tie ktoré nemajú všetky vrcholy rovnaké). Budeme tento vrchol hľadať prechádzaním stromu delenia a cesty k nemu určíme jeho súradnice v relevantných dimenziách. Začneme na najvyššej úrovni a pôjdeme vždy po niektorej hrane nižšie. Vyberieme si vždy vetvu s väčšou množinou, ak budú obe množiny rovnaké, tak tú ktorá má v danej dimenzii súradnicu 0. Týmto sme jednoznačne určili výber vrchola.

Tvrdíme, že po jeho odobratí bude mať množina vrcholov tento strom delenia izomorfný so stromom delenia množiny danej mohutnosti s najvyšším možným počtom hrán v indukovanom podgrafe.

Dve rovnako veľké množiny majú rovnaký podstrom delenia podľa dimenzií, čo sme už ukázali. Keď sa pozrieme na strom odvodenia po odobratí vybraného vrchola, zistíme, že nebola porušená veta 27 o rozdiele veľkostí dvoch množín. Keďže sú v každom vetvení množiny buď rovnaké alebo je menšia tá s vrcholmi so súradnicou 0 v danej dimenzii, zostáva zachované priradenie súradníc z vety 28.

Našli sme teda spôsob, ako určiť odzadu poradie prechádzania vrcholov agentom tak, aby prejdená množina vrcholov obsahovala najvyšší možný počet hrán v indukovanom podgrafe. Pre úplnosť treba dokázať, že agent môže vrcholy v tomto poradí navštíviť. Teda, že odobratý vrchol susedí s vrcholom, ktorý bude odobratý najbližšie.

V prípade, že sú obe množiny rovnako veľké bude najprv odobratý prvok z množiny so súradnicou 0, čím sa táto množina zmenší a následne bude odobratý prvok z množiny so súradnicou 1 v prvej dimenzii delenia. Keďže obe množiny majú rovnakú veľkosť výber súradníc vo zvyšných dimenziách dopadne zhodne pre oba vrcholy. Líšia sa teda práve len v súradnici prvej dimenzie a teda ich spája hrana. Vyplýva to aj z toho, že menšia množina má izomorfný obraz vo väčšej.

V prípade, že jedna množina vrcholov je väčšia, tak to nemusí nutne platiť. Preto urobíme úpravu v postupe výberu vrchola. Ako sme ukázali vo vete **??** substitúcia súradníc v niektorej dimenzii nemá vplyv na počet hrán vedúcich medzi vrcholmi danej množiny, ak sa vymenia všetky súradnice v danej dimenzii na opačné. Ak preklopíme súradnice vo všetkých dimenziách, v ktorých má odoberaný vrchol súradnicu 1, tak nasledujúcim vybraným v takto substituovaných súradniciach bude jeho sused na najnižšej úrovni stromu.

Na konci môžeme spätne vypočítať skutočné súradnice vrcholov postupnosti. □

Dôkaz. hlavnej vety 3.

Otázkou ostáva ako optimálna postupnosť konštrukcie hyperkocky vyzerá. Keď sa na strom delenia pozrieme odspodu, uvidíme, že agent vlastne prechádza vrcholy vždy väčšej a väčšej hyperkocky. Po skonštruovaní každej hyperkocky zrkadlovo opakuje svoje predošlé pohyby. Najprv vytvorí jednorozmernú kocku podľa dimenzie, podľa ktorej sa prvý raz delí množina vrcholov. Následne sa pohne v druhej dimenzii a tam zopakuje konštrukciu jednorozmernej kocky. Vzniknú štyri vrcholy, ktoré majú hrany len v dvoch dimenziách a súradnice v ostatných dimenziách majú rovnaké - 1. Teda tvoria dvojrozmernú kocku. Keď sa pozrieme na strom delenia tak v každom vetvení pridá agent vrchol najprv do pravého a potom do ľavého podstromu pre každý podstrom - najprv so súradnicou 1 v danej dimenzii a potom so súradnicou 0. Odzadu potom odoberáme najprv vrchol so súradnicou 0 (pri rovnosti mohutností množín na oboch stranách) a potom so súradnicou 1.

Ukázali sme, že vrchol s hyperkockou, do ktorej nepatrí, spája najviac jedna hrana. Ideme zistiť, kde v postupnosti konštrukcie n-rozmernej hyperkocky, optimálnej na počet krokov a následne optimálnej na potrebu značiek pre porty, potrebuje agent najviac portov otvorených súčasne. Teda zistiť kde postupnosť vrcholov z optimálnej konštrukcie n-rozmernej hyperkocky rozdeliť na dve časti tak, aby medzi nimi viedol najväčší počet hrán.

Každá i-rozmerná hyperkocka sa dá rozdeliť na dve (i-1)-rozmerné hyperkocky. Medzi nimi vedie 2^{i-1} hrán v dimenzii, ktorú označíme ako prvá. Pokiaľ toto nie je maximum, nachádzajú sa maximá symetricky ďaleko odtiaľto v začiatku aj konci postupnosti, pretože postupnosť je symetrická. Ak odhliadneme od hrán v prvej dimenzii a rozdelíme prvú (n-1)-rozmernú hyperkocku opäť v polovici, dostaneme rovnaký prípad symetrie. Keď opäť zarátame hrany aj v prvej dimenzii, zistíme, že v prvej polovici postupnosti leží maximum v jej druhej polovici - teda druhej štvrtine celej postupnosti. Vrcholy tejto štvrtiny spája po jednej hrane v druhej dimenzii s prvou štvrtinou a po jednej hrane v prvej dimenzii s treťou štvrtinou postupnosti. Pri konštrukcii štvrtiny postupnosti sa teda súčet hrán v prvých doch dimenziách odchádzajúcich z prefixu postupnosti a teda aj počet im zodpovedajúcich portov nemení. Na tieto môžeme teda na čas zabudnúť a zúžiť hľadanie jedného z maxím na vrcholy druhej štvrtiny postupnosti, čo je postupnosť konštrukcie (n-2)-rozmernej hyperkocky, čím sme získali rekurzívny spôsob hľadania maxima. Hrán, na ktoré sme "zabudli"v prvej rekurzii je 2^{n-1}.

Ak je n párne skončíme pri štyroch 0-rozmerných hyperkockách, kde maximum otvorených portov sa bude dosahovať v prvom alebo druhom vrchole. Pri nepárnom n skončíme pri 1-rozmernej hyperkocke, kde maximum bude v jej prvom vrchole, lebo vtedy je otvorený ešte aj port do druhého z jej vrcholov. Teda pre nepárne n potrebuje agent $\frac{2^{n+1}-1}{3}$ a pre párne n potrebuje $\frac{2^{n+1}-2}{3}$ značiek pre porty. □

7.2 2-rozmerná kocka

Jednorozmerná kocka má len jednu hranu, ktorá je zároveň súčasťou každej jej kostry. Jej konštrukcia je v našom modely triviálna a stojí jednu operáciu - vytvorenie novej hrany s vrcholom na konci z počiatočného vrchola.

Dvojrozmerná kocka je najmenšia kocka, pri konštrukcii ktorej agent vytvorí portovú hranu. Je to elementárny prípad. Agentovi stačí jedna portová značka a vykonať tri kroky pohyby - operácie NH. Port otvorí v počiatočnom a poslednom vrchole, čím vznikne posledná potrebná hrana.

7.3 3-rozmerná kocka

7.3.1 1 značka

Pri použití jednej značky pre porty vie agent trojrozmernú kocku zostrojiť na 15 krokov: OP, NH, NH, NH, OP, OP, S1, NH, NH, OP, OP, S2, S2, NH, OP, OP, S1, S1, NH, OP, OP, S1, S2, S3, OP. Na menej to nie je možné. Agent potrebuje skonštruovať 5 mimokostrových hrán a na každú potrebuje aspoň 3 kroky - pohyby medzi otvorením jej prvého a druhého portu.

7.3.2 2 značky

Veta 5. *Trojrozmernú kocku pri použití dvoch značiek na porty dokáže agent zostrojiť na 11 pohybov a menej sa nedá.*

Dôkaz. Operácie agenta: OP1, NH, NH, OP2, NH, OP1, NH, OP1, NH, OP2, NH, OP2, NH, OP1, OP1, S1, S1, S2, OP1, S1, OP2 Na menej sa nedá, pretože porty treba otvoriť medzi dvojicou vrcholov líšiacich sa v práve jednej súradnici. Teda medzi otvorením portov jednej hrany potrebuje agent aspoň dve preklopenia na rovnakej súradnici a jedno na inej. Vie využiť najviac jedno preklopenie, ktoré nastalo pri konštrukcii predošlej hrany (pretože na dvojici súradníc ktoré preklápal už existuje pre vrchol v ktorom sa nachádza dvojrozmerná kocka - práve ju vytvoril), čiže na konštrukciu hrany potrebuje najmenej dva ďalšie pohyby. Na konštrukciu prvej hrany potrebuje tri pohyby. $3 + 2 * 8 = 11$ □

7.3.3 3 značky

Veta 6. *Trojrozmernú kocku agent za použitia troch portových značiek zostrojí na 9 pohybov. Na menej sa nedá.*

Dôkaz. Operácie agenta: OP1, OP3, NH, NH, OP2, NH, OP1, NH, OP1, NH, OP2, NH, OP2, NH, OP1, OP3, S3(?), S1, OP2 S použitím menšieho množstva pohybov to agent urobiť nemôže. Kým vznikne prvá portová hrana - značka sa uvoľní na ďalšie použitie,vytvorí agent tri nové vrcholy. Teda je vo štvrtom vrchole. Súčet stupňov predošlých troch je deväť, z čoho päť zaberú nové hrany. To znamená, že do jedného z týchto vrcholov sa agent bude musieť vrátiť otvoriť port alebo z neho vytvoriť novú hranu. Keby tento vrchol bol tretí podľa vzniku, tak sa doň agent môže vrátiť na jeden ťah. Tu môžu nastať dve možnosti: pôjde tam otvoriť port alebo vytvoriť novú hranu. Ak otvoriť port, bude mu chýbať značka na štvrtom vrchole podľa vzniku a bude sa musieť vrátiť aj do neho. Ak otvorí port s novouvoľnenou značkou na štvrtom vrchole a z tretieho urobí NH. Ocitne sa vo vrchole v ktorom nemôže otvoriť port, lebo sa nelíši v práve jednej súradnici od žiadneho z vrcholov v ktorých sú porty pootvárané. Teda v oboch prípadoch potrebujem ďalšie dva pohyby, ktoré nie sú operáciou NH. Ak sa agent nevráti hneď do vrchola, v ktorom neotvoril port, bude musieť prejsť do niektorého z jeho dvoch susedov (lebo z nich už odišiel a z iného vrchola sa tam nedostane). To sú dva pohyby okrem operácií NH. Počet operácií NH je konštantný pre daný graf a rovná sa počtu hrán kostry. V tomto prípade je to sedem. Teda agent vybavený tromi rôznymi značkami na porty potrebuje aspoň deväť krokov na konštrukciu trojrozmernej kocky. □

7.3.4 4 značky

Veta 7. *4 rôzne značky pre porty agentovi na konštrukciu trojrozmernej kocky stačia.*

Dôkaz. OP1, OP2, NH, OP3, NH, OP4, NH, OP1, NH, OP1, NH, OP4, NH, OP3, NH, OP2, OP1 Agent vykoná sedem pohybov pri konštrukcii kostry, čo je nevyhnutné. Pridanie ďalších značiek už teda nič nevylepší. □

7.4 4-rozmerná kocka

1 značka Ak má agent skonštruovať portové hrany použitím troch pohybov na hranu a žiadne ďalšie pohyby, tak musí vo vrchole, kam otvorením portu vznikne nová portová hrana, otvoriť port znovu, inak by musel vykonať krok navyše. Portové hrany teda musia tvoriť súvislý ťah.

Všetkých hrán v štvorrozmernej kocke je 32, pričom kostrových je 15. Portových hrán je teda 17. Sled portových hrán sa nemôže pretínať, pretože nemôže existovať vrchol do ktorého nevedie kostrová hrana a každý vrchol má stupeň 4. Keďže stupeň vrcholov je párny, kostra z hrán vzniknutých operáciou NH musí byť taká, aby dva z nich mali nepárny stupeň a zvyšné párny; teda aby tvorila cestu po všetkých vrcholoch v grafe. Čo je nutná podmienka zostrojenia jediného ťahu zo všetkých zvyšných hrán - portových hrán. Kostra má aspoň dva listy a stupeň listov je nepárny. Ak teda kostra nemá mať väčší počet listov, musia mať zvyšné vrcholy stupeň dva. Kostra je teda tiež cesta. Ukážeme, že je možné zostrojiť štvorrozmernú kocku na $3 * 17 = 51$ pohybov s jedinou značkou pre porty.

Veta 8. *Existuje konštrukcia štvorrozmernej kocky, pri ktorej agent vykoná 51 krokov a použije tri rôzne značky pre porty.*

Dôkaz. Aby sme skrátili zápis, označíme dimenzie - rozmery prirodzenými číslami od 1 po 4 a uvedieme len v ktorej dimenzii agent preklopil súradnicu svojej polohy pri pohybe. Pre lepšiu prehľadnosť zoskupíme zmeny dimenzií týkajúce sa konštrukcie rovnakej hrany do zátvoriek po troch: (121), (343), (121), (242), (434), (323), (212), (131), (141), (414), (313), (323), (212), (232), (141), (131), (343).

Čiže postupnosť operácií by vyzerala takto: OP, 3 × NH, OP, OP, 3 × NH, OP, OP, 3 × NH, OP, OP, S2, S2, S2, OP, OP, S2, NH, NH, OP, OP, ... □

7.5 Konštrukcie kociek vyšších rozmerov

Ako sme už spomínali, z postupnosti konštrukcie hyperkocky na jednu návštevu v každom vrchole - označme ju P, vieme odvodiť pravidlá aj pre efektívnu konštrukciu hyperkocky v prípadoch, keď treba niektorý vrchol navštíviť viackrát ako raz. Táto postupnosť totiž určuje aj to, koľko krokov musí agent vykonať na to, aby vytvoril daný počet portových hrán. Jedna z možností pre agenta je opakovane prechádzať túto postupnosť a otvoriť port v prípade, že má voľnú značku až kým nebudú všetky portové hrany skonštruované. Ako však ukazujú konštrukcie hyperkociek nižších dimenzií, nie je to vždy optimálne riešenie.

Limitujúcim faktorom je pre agenta mohutnosť množiny značiek, ktorú má k dispozícii. Pre ňu následne hľadá čo najkratšiu postupnosť R prechádzania vrcholov a otvárania portov pri konštrukcii danej hyperkocky. Postupnosť P je optimálna, teda nie je možné na k krokov zostrojiť viac portových hrán ako je medzi prvými k vrcholmi postupnosti P. Žiadny súvislý úsek postupnosti R preto nemôže obsahovať vrcholy s väčším počtom hrán medzi nimi, ako obsahuje prefix postupnosti P danej dĺžky. Mohutnosť množiny značiek tiež určuje, koľko môže vzniknúť hrán v postupnosti R na danom úseku. Ona limituje počet portov, ktoré môžu byť otvorené naraz, teda aj to ako dlhý prefix postupnosti P sa môže v postupnosti R opakovať. Ten je daný maximom naraz otvorených portov. Toto maximum môže byť nanajvýš rovné počtu značiek, ktoré má agent k dispozícii.

Literatúra

[DFK+05a] Shantanu Das, Paola Flocchini, Shay Kutten, Amiya Nayak, and Nicola Santoro. Distributed exploration of anonymous graphs by multiple agents. 2005.

[DFK+05b] Shantanu Das, Paola Flocchini, Shay Kutten, Amiya Nayak, and Nicola Santoro. Distributed exploration of unlabelled graphs by multiple agents. 2005.

[DKM12] Stefan Dobrev, Rastislav Královič, and Euripides Markou. Online graph exploration with advice. In *Structural Information and Communication Complexity*, pages 267–278. Springer, 2012.

[DP90] Xiaotie Deng and Christos H Papadimitriou. Exploring an unknown graph. In *Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on*, pages 355–361. IEEE, 1990.

[FIP+05] Pierre Fraigniaud, David Ilcinkas, Guy Peer, Andrzej Pelc, and David Peleg. Graph exploration by a finite automaton. *Theoretical Computer Science*, 345(2–3):331 – 344, 2005. Mathematical Foundations of Computer Science 2004 Mathematical Foundations of Computer Science 2004.

[MMS11] Nicole Megow, Kurt Mehlhorn, and Pascal Schweitzer. Online graph exploration: new results on old and new algorithms. In *Automata, Languages and Programming*, pages 478–489. Springer, 2011.

[PY91] Christos H Papadimitriou and Mihalis Yannakakis. Shortest paths without a map. *Theoretical Computer Science*, 84(1):127–150, 1991.

[RE97] Grzegorz Rozenberg and Hartmut Ehrig. *Handbook of graph grammars and computing by graph transformation*, volume 1. World Scientific Singapore, 1997.

[Rol79] Hans-Anton Rollik. Automaten in planaren graphen. In *Theoretical Computer Science 4th GI Conference*, pages 266–275. Springer, 1979.