UNIVERZITA KOMENSKÉHO V BRATISLAVE

Fakulta matematiky, fyziky a informatiky

Algoritmy kreslenia planárnych grafov

Bakalárska práca

Miroslav Kobliška

2018

ALGORITMY KRESLENIA PLANÁRNYCH GRAFOV

BAKALÁRSKA PRÁCA

Miroslav Kobliška

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY
A INFORMATIKY
KATEDRA ALGEBRY, GEOMETRIE A DIDAKTIKY
MATEMATIKY

Študijný odbor: **1113 Matematika**

Vedúci bakalárskej práce**:**

doc. RNDr. Andrej Ferko, PhD.

Bratislava 2018

41913365

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Miroslav Kobliška
Študijný program: matematika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: matematika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Algoritmy kreslenia planárnych grafov
Planar Graph Drawing Algorithms

Anotácia: Cieľovou skupinou sú vyučujúci a študujúci predmetu Webovská grafika. Vysvetlíme základné algoritmy kreslenia planárnych grafov z prednášky dr. Katreniakovej a ďalšie. Zvolíme niekoľko motivačných vypočítaných príkladov na dobre známych dátach. Navrhneme vizualizačnú a aktivizačnú funkčnosť (testy, vzdelávacia hra). Na implementáciu si zvolíme vhodný autorský nástroj, resp. jazyk.

Anotácia: Kreslenie grafov sa zaoberá geometrickým zobrazením grafov a sietí a je motivované takými aplikáciami, kde je nevyhnutné vizualizovať štrukturálne informácie ako grafy. Projekt by mal doplniť pre autorov web design základnú prednášku o vzdelávací portál.

Literatúra: Vybrané odkazy z portálu Graph Drawing. 2017. [online] http://www.graphdrawing.org/.
Portál Graph Visualization, WOLFRAM. 2017. [online] http://reference.wolfram.com/language/guide/GraphVisualization.html
KATRENIAKOVA, J. 2016. Vizualizácia a navigácia s použitím techník kreslenia grafov. [online]
[online] http://www.sccg.sk/~ferko/KreslenieGrafovDrKatreniakova.pdf.

Kľúčové slová: kreslenie grafov, vzdelávacia hra

Vedúci: doc. RNDr. Andrej Ferko, PhD.
Katedra: FMFI.KAGDM - Katedra algebry, geometrie a didaktiky matematiky
Vedúci katedry: doc. RNDr. Pavel Chalmovianský, PhD.

Dátum zadania: 09.11.2017

Dátum schválenia: 21.11.2017

prof. RNDr. Ján Filo, CSc.
garant študijného programu

.....
študent

.....
vedúci práce

Poďakovanie

Týmto by som chcel poďakovať svojej rodine za podporu počas celej doby štúdia a vedúcemu bakalárskej práce Doc. RNDr. Andrejovi Ferkovi, PhD. za odborný dohľad, cenné pripomienky a dobré rady pri písaní bakalárskej práce.

Abstrakt

Miroslav Kobliška. Algoritmy kreslenia planárnych grafov. Bakalárska práca. Univerzita Komenského, Bratislava. máj 2018. Fakulta matematiky, fyziky a informatiky, Katedra algebry, geometrie a didaktiky matematiky. Školiteľ práce: doc. RNDr. Andrej Ferko, CSc. Obhajoba: 2018, s.33.

V tejto bakalárskej práci sa venujeme kresleniu grafov so zameraním na prístup topológia – tvar – metrika a jeho implementácii vo forme aktivizácie používateľa. V prvej kapitole sa venujeme opisu problematiky. V druhej kapitole uvádzame základné štýly kreslenia grafov a opisujeme algoritmy použité pri tvorbe aplikácie na kreslenie grafov. V tretej kapitole popisujeme aplikáciu, ktorú sme vytvorili, jej funkcie a ovládanie. Výsledkom tejto práce je aplikácia, na ktorej si študenti môžu interaktívne vyskúšať jednotlivé fázy prístupu topológia – tvar – metrika.

Kľúčové slová: kreslenie grafov, aktivizácia, e- learning

Abstract

Miroslav Kobliška. Algoritms of Graph Drawing. Bachelor Thesis. Comenius University, Bratislava. May 2018. Faculty of Mathematics, Physics and Informatics; Department of Algebra, Geometry and Didactic Mathematics. Tutor: doc. RNDr. Andrej Ferko, CSc. Obhajoba: 2018, s.33.

In this bachelor thesis we deal with the characterisation of graph drawing focusing on topology - shape - metric approach and its implementation in the form of activation of the user. In the first chapter we introduce the problem formulation. In the second chapter, we present the basic styles of graph drawing and describe the algorithms used in creating the graph drawing application. In the third chapter, we describe the application, create its functions and control. The result of this work is an application where students can interact and test individual phases of approach topology - shape - metrics.

Keywords: graph drawing, activisation, e- learning

Obsah

Úvod 1

1 Prehľad problematiky 3

1.1 Základné definície 3

1.2 Vizualizácia informácii 4

1.3 Kreslenie grafov 6

1.4 História algoritmov planarity 7

1.5 Bežné algoritmické techniky a nástroje 7

1.6 Estetika grafov 8

1.7 Vlastnosti kresieb 9

1.8 Vzory kreslenia grafov 9

2 Špecifikácia projektu a návrh riešenia 15

2.1 Štýly kresieb 15

2.1.1 Planárna kresba 15

2.1.2 Kresby s lomenými hranami 15

2.1.3 Kresba s priamymi hranami 16

2.1.4 Konvexné kresby 16

2.1.5 Ortogonálne kresby 17

2.1.6 Krabicovo-ortogonálne kresby 17

2.1.7 Obdĺžnikové kresby 18

2.1.8 Krabicovo-obdĺžnikové kresby 18

2.1.9 Mriežkové kresby 18

2.1.10 Kresby viditeľnosti 19

2.2 Topológia – tvar – metrika 20

2.2.1 Planarizácia 20

2.2.2 Ortogonalizácia 22

2.2.3 Zhustenie 22
3 Implementácia riešenia 25
3.1 Dátový model 25
3.2 Pracovná plocha 26
3.3 Vstup 27
3.4 Ako funguje program 27
3.4.1 Planarizácia 28
3.4.2 Ortogonalizácia 28
3.4.3 Zhustenie 29
3.5 Výstup 30
Záver 31
Literatúra a internetové pramene 33

Úvod

Pri kreslení grafov sa venujeme problému vytvárania geometrických reprezentácii grafov. Kreslenie grafov má dôležité aplikácie v kľúčových počítačových technológiách, ako je vyvíjanie softvérov, databázových systémov a vizuálnych rozhraní [DiGi17]. Výskum kreslenia grafov bol vedený v rámci niekoľkých začínajúcich oblastí, zahŕňajúcich diskrétnu matematiku (topologická teória grafov, geometrická teória grafov), algoritmy (grafové algoritmy, štruktúra dát, výpočtová geometria) a interakciu ľudí s počítačom (vizualizačné jazyky, grafické používateľské rozhrania, vizualizácia softvéru).

Motiváciou tejto práce je podporiť výučbu kreslenia grafov v predmetoch, ako je Webovská grafika [Katr17] alebo iné predmety, kde sa vyučuje táto problematika.

Cieľom práce je poskytnúť základnú, pritom relatívne komplexnú charakteristiku o vybraných algoritmoch kreslenia planárnych grafov tak, aby bola zrozumiteľná pre študenta, či iného čitateľa, ktorého táto téma zaujíma. Preto v tejto práci opisujeme viacero vzorov kreslenia grafov a taktiež typy a najdôležitejšie vlastnosti kresieb. Hlavným cieľom práce je popísanie prístupu topológia – tvar – metrika, ktorý sme si vybrali spomedzi ostatných, pretože vhodne štruktúruje etapy spracovania.

V prvej kapitole definujeme základné pojmy z teórie grafov, ktoré sme použili v tejto práci. Ďalej sa venujeme otázke vizualizácie informácii a grafov a popisujeme základné vlastnosti kresieb a niektoré vzory kreslenia grafov. V druhej kapitole predstavujeme typy kresieb a popisujeme prístup topológia – tvar – metrika podrobnejšie. A v tretej kapitole sa venujeme implementácii prístupu topológia – tvar – metrika vo forme aktivizácie používateľa.

1 Prehľad problematiky

V tejto časti práce uvedieme definície pojmov z teórie grafov, povieme si ako funguje vizualizácia informácií, ako kresliť grafy a pár informácií o histórii algoritmov planarity. A tiež uvedieme, ako by mal vyzerať správne esteticky graf, štýly kresieb a základné vzory kreslenia grafov.

1.1 Základné definície

Graf G=(V,E) je definovaný v učebnici Kvasnička-Pospíšil [KvPo08] pomocou množiny vrcholov V a množiny hrán E – neusporiadaných dvojíc e={u, v} vrcholov z V.

Orientovaný graf G=(V, E) je definovaný pomocou množiny vrcholov V a množiny E– usporiadaných dvojíc e=(u, v) vrcholov z V. *Neorientovaný* graf G=(V, E) je definovaný pomocou množiny vrcholov V a množiny E neorientovaných hrán, pričom každá hrana e ∈ E je reprezentovaná neusporiadanou dvojicou vrcholov z V

Dva vrcholy u a v v neorientovanom grafe G sa volajú *susedné* v G, keď {u, v} je hrana grafu G. Keď e={u, v}, o hrane e sa hovorí, že je *incidentná* s vrcholmi u a v alebo spája vrcholy u a v. *Stupeň* vrcholu v neorientovanom grafe je rovný počtu hrán s ním incidentných. Vrchol stupňa 0 sa volá *izolovaný*, ak nie je spojený zo žiadnym iným vrcholom.

Cesta je sekvencia odlišných vrcholov $v_1, v_2, \ldots, v_k$, s $k \geq 2$ spolu s hranami $(v_1, v_2), \ldots, (v_{k-1}, v_k)$. Dĺžka cesty je počet jej hrán. *Cyklus* je sekvencia odlišných vrcholov $v_1, v_2, \ldots, v_k$, s $k \geq 2$, spoločne s hranami $(v_1, v_2), \ldots, (v_{k-1}, v_k), (v_k, v_1)$. Dĺžka cyklu je počet jeho vrcholov alebo počet jeho hrán.

Graf je súvislý, ak každé dva vrcholy sú spojené cestou. *Nesúvislý graf* je zjednotením dvoch alebo viac súvislých grafov, z ktorých žiaden pár nemá spoločný vrchol. Tieto súvislé podgrafy sa volajú *komponenty* grafu. Niekedy odstránením hrany alebo vrcholu z grafu sa graf rozpadne na viac komponentov.

Strom je súvislý graf bez cyklov. *Koreňový strom* T je súvislý strom s význačným vrcholom, ktorý nazývame koreň r.

Kresba (drawing) Γ grafu G, ako sa píše v [Tam13], priradí každému vrcholu v odlišný bod Γ (v) roviny, a každej hrane (u, v) jednoduchú otvorenú Jordanovu krivku Γ (u, v), s koncovými bodmi Γ (u) a Γ (v). Kresba je *planárna*, ak sa nepretínajú dve

odlišné hrany, s výnimkou prípadných spoločných koncových bodov. Graf je *planárny*, ak pripúšťa planárnu kresbu. Planárna kresba rozdeľuje rovinu na spojité oblasti, ktoré nazývame *plochy*.

Podgraf grafu G=(V,E) je graf H=(W,F), kde W⊆ V a F ⊆E. Vzhľadom na dva grafy G1 (V1, E1) a G2 (V2, E2), ich *zjednotenie* G1 ∪ G2 je graf G (V1 ∪ V2, E1 ∪ E2). Analogicky ich *prienik* G1 ∩ G2 je graf G (V1 ∩ V2, E1 ∩ E2). Graf G2 je *podgraf* G1, ak G1 ∪ G2 = G1.

Vnorený graf je planárny graf s predšpecifikovanými typologickými vnoreniami (napr. množina stien), ktoré musia byť zakonzervované v grafe.

Neorientovaný graf G je súvislý, ak pre každú dvojicu uzlov u a v, G obsahuje cestu od u do v. Graf G s najmenej $k + 1$ vrcholmi je k-súvislý, ak po odstránení $k - 1$ vrcholov je G súvislý graf. Ekvivalentne, podľa [Tam13] je graf k-súvislý, „ak je k dispozícii k nezávislých ciest medzi každou dvojicou vrcholov“.

Graf G* = (V*, E*) nazývame *duálny* k planárnemu grafu G = (V, E), ak bol vytvorený nasledovným spôsobom: každá stena grafu G je reprezentovaná jedným vrcholom grafu G*. Cez každú hranu grafu G vedie práve jedna hrana grafu G* spájajúca vrcholy reprezentujúce steny po oboch stranách tejto hrany.

1.2 Vizualizácia informácii

Vizualizácia informácií je v dizertácii Frishman [Fri00] definovaná ako počítačom podporovaná, interaktívna reprezentácia abstraktných dát, ktorej cieľom je podporiť poznávanie. Použitie grafických reprezentácií dát umožňuje využívať ľudský vizuálny systém, ktorý je schopný rýchlo spracovať veľké množstvo dát a má dobré schopnosti rozpoznávania vzorov. [Chi00] rozdeľuje techniky vizualizácie informácií pomocou ôsmych vizualizačných dátových typov: časové, 1D, 2D, 3D, multi-D, stromové, sieťové a nakoniec pracovné plochy. Predtým sa navrhli aj vizualizačné techniky informácií založené nielen na dátových typoch, ale aj na operátoroch spracovania, ktoré sú obsiahnuté v každej vizualizačnej technike. Vizualizačné techniky informácií by mohli byť popísané pomocou modelu Data State Model (DSM). DSM rozdeľuje každú techniku na štyri dátové fázy, tri typy dátových transformácií a štyri typy operátorov v rámci fázy. Vizualizácia dátových reťazcov je rozdelená do štyroch odlišných dátových fáz: vstup, analytická abstrakcia, vizualizácia abstrakcie a výstup.

Na vstupe sú zadané základné údaje. Analytická abstrakcia obsahuje dáta o dátach alebo informácie. Vizualizačnú abstrakciu môžeme popísať ako informácie, ktoré je možné

zobraziť na obrazovke pomocou vizualizačnej techniky. A nakoniec výstup je konečný produkt vizualizačného mapovania, kde používateľ vidí a interpretuje obraz, ktorý je mu prezentovaný.

Transformácia údajov z jedného stupňa do druhého vyžaduje jeden z troch typov operátorov dátových transformácii: transformácia údajov (vytvára určitú formu analytickej abstrakcie od hodnoty najčastejšie extrakciou), transformácia vizualizácie (berie analytickú abstrakciu a ďalej ju redukuje na nejakú formu abstrakcie vizualizácie, ktorá je viditeľným obsahom) a vizuálne mapovanie transformácie (slúži na informácie, ktoré sú vo formáte, ktorý je vizualizovateľný a predstavuje grafické zobrazenie).

V rámci každej dátovej fázy sú aj operátory, ktoré nemenia základné dátové štruktúry. Jedná sa o operátory v rámci fázy, ktoré sú štyroch druhov a zodpovedajú týmto štyrom dátovým fázam: operátor v rámci vstupu (Within Value), operátor v rámci analytickej abstrakcie (Within Analytical Abstraction), operátor v rámci vizualizačnej abstrakcie (Within Visualization Abstraction) a operátor v rámci výstupu (Within View).

Taxonomizácia. Oddeľovaním závislostí môžeme jednoduchšie využiť rôzne časti reťazca spracovania údajov, aby sme vytvorili nové vizualizácie informácie. Preto sme použili tento model a využili sme ho na taxonomizáciu rôznych vizualizačných techník. Cieľom je analyzovať rôzne techniky, čím sa rozširujú naše znalosti o tom, ako každá technika môže byť vytvorená využitím rôznych operátorov. Vo svete, bohatom na komunikáciu a zobrazovacie technológie, je mnoho príležitostí pre inovatívne vizualizačné techniky. Niektoré aplikácie vizualizácie informácii zahŕňajú finančnú analýzu, vizualizáciu softvéru, vizualizáciu vo vede, ďalej vizualizáciu textu a dokumentov, grafov a sietí a nakoniec vizualizáciu sociálnych sietí. V nasledovnom je využitý tento model na analýzu vizualizačných techník používaných na sieti.

S jasnejším pochopením interakcií medzi údajmi a operátormi, budú realizátori lepšie pripravení na vytvorenie nových interakcií alebo nových vizualizácií. V praxi sa tieto analytické metódy aplikovali v systéme nazvanom Vizualizačná tabuľka a umožnili opätovné použitie a rýchly vývoj.

V internetových sieťach používame viacero vizualizačných techník. Jednou z nich je GraphViz. Vstupné údaje sú napríklad: moduly kódu softvéru, súborový systém, všetky druhy grafov. Pri transformácii údajov sa extrahujú hrany a uzly do grafu. V rámci analytickej abstrakcie je abstrakciou graf. Vizuálno mapovacia transformácia je sofistikovaný schematický algoritmus, ktorý inteligentne umiestňuje uzly na 2-

rozmernej ploche. A nakoniec výstupom je graf s minimálnym počtom pretínajúcich sa hrán.

1.3 Kreslenie grafov

Grafy sú abstraktné matematické objekty, určené na opis vzťahov medzi objektmi, hovorí v dizertácii Frishman [Fri00]. Kreslenie grafov sa zaoberá problémom nájdenia najlepšieho spôsobu, ako nakresliť graf. Jednou z bežných metód, ktoré sa používajú na kreslenie grafov, je node-link diagram. Pri tejto vizualizácii sú uzly nakreslené pomocou bodiek, kruhov alebo iných geometrických tvarov a hrany sú nakreslené pomocou priamych alebo zakrivených čiar. Šípky sa používajú na zobrazenie orientácie orientovaných hrán. Informácie zodpovedajúce uzlom a hranám je možné vizualizovať pomocou textových štítkov na rôznych pozíciách, v alebo vedľa objektu na grafe, rôznych farieb (ako na mape metra), alebo iných vizuálnych prvkov, ako je hrúbka čiar, veľkosť boxov atď. Graf môže byť nakreslený v rovine alebo v priestore. Môžeme ho vykresliť úplne, čiastočne alebo hierarchicky, t. j. zhluky sa skracujú na jediný uzol, ktorý môže byť rozšírený, ak potrebujeme.

Rozdielne zobrazenia alebo rozloženia grafov môžu zodpovedať rovnakému grafu. V abstraktnom grafe je dôležité, ktoré vrcholy sú spojené s ktorými, pomocou koľkých hrán. Vo vizuálnom zobrazení grafu však usporiadanie týchto vrcholov a hrán ovplyvňuje zrozumiteľnosť, využiteľnosť, výrobné náklady a estetika. Vykreslenie grafov je preto hlavným problémom pri vizualizácii informácií.

Vykreslenie grafov bez priesečníkov je jedným z najzaujímavejších a najfascinujúcejších algoritmických problémov pri kreslení grafov a v teórii grafov. Napriek tomu, že úplná charakterizácia rovinných grafov je známa už od roku 1930, prvé lineárne riešenie problému per se bolo nájdené až v sedemdesiatych rokoch.

Kreslenie grafov je v posledných rokoch časťou počítačovej grafiky, ktorej sa venuje veľa odborníkov. Boli navrhnuté rôzne metódy kreslenia grafov, ako sú hierarchické, planárne, kruhové, ortogonálne, symetrické, spektrálne a obdĺžnikové.

Kreslenie grafov sa využíva v rôznych sférach. Medzi niekoľko príkladov využitia patria sociálne siete, bioinformatika, mapy vlakových sietí, genealógia, vizualizácia grafov funkcií, vizualizácia vývoja softvéru, vizualizácia siete, databázy, dátové štruktúry, počítačová bezpečnosť alebo softvérové inžinierstvo.

1.4 História algoritmov planarity

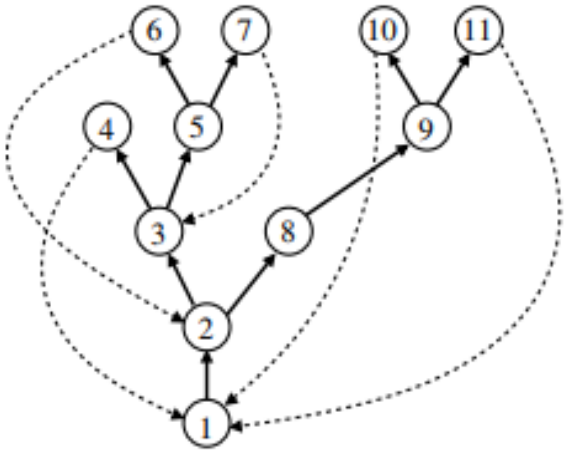
Priame použitie Kuratowského charakterizácie rovinných grafov, založených na subdivíziách, by prinieslo exponenciálny časový algoritmus, zatiaľ čo Wagnerova charakterizácia založená na minoroch by poskytla algoritmus vo faktoriálnom čase [Tam13]. Prvé polynomial-time algoritmy pre planaritu zaviedli Auslander a Parter, Goldstein a nezávisle od nich Bader.

V roku 1974 navrhli Hopcroft a Tarjan prvý algoritmus testujúci planaritu, a to algoritmus linearizácie. Tento algoritmus, tiež nazývaný "path-addition algorithm," začína od cyklu a pridáva k nemu vždy po jednej ceste.

Iný prístup má východiskový bod v algoritme, ktorý predstavujú Lempel, Even, a Cederbaum. Tento algoritmus tiež nazývaný "vertex addition algorithm" je založený na zvažovaní vrcholov jeden po druhom, nasledujúc st-numbering; bolo preukázané, že to je možné realizovať v lineárnom čase Bootherom a Luekerom. V tomto prípade boli potrebné ďalšie poznatky od Chiba, Nishizekiho, Abeho a Ozawu, aby sa ukázalo, ako vytvoriť implementáciu kreslenia kresby, ktorá je planárna.

1.5 Bežné algoritmické techniky a nástroje

V tejto časti uvádzame niektoré definície a bežné techniky, ktoré používajú algoritmy planarity [Tam13]. Najdôležitejšou technikou, ktorá je spoločná pre takmer všetky algoritmy, je Depth First Search (DFS). DFS je spôsob, ako prejsť cez všetky vrcholy grafu G.



Obrázok 1.1 DFS graf z [Tam13]. Hrubé čiary predstavujú hrany stromu, zatiaľ čo zadné hrany sú kreslené čiarkovanou čiarou. Každý vrchol je označený indexom DFS.

Táto technika začína od ľubovoľného vrcholu G a prechádza od aktuálneho vrcholu k susednému, a pokračuje dovtedy, kým sa nájdu nepreskúmaní susedia. Ak súčasný vrchol neobsahuje žiadnych nepreskúmaných susedov, prejde sa k prvému vrcholu s nepreskúmaným priľahlým vrcholom.

Hrany, ktorými DFS prechádza k ďalším vrcholom grafu G, tvoria vetviaci sa strom T grafu G nazývaného palmový strom. Koreňom T je vrchol, na ktorom začal traverz. Hrany T sa nazývajú hrany stromu, zatiaľ čo zvyšné hrany G sa nazývajú zadné hrany.

Po vykonaní algoritmu DFS môže byť každému vrcholu v G priradený DFS index, DFS(v). Pričom index v je určený poradím, v ktorom DFS prechádzalo cez graf. Koreň stromu má index jedna. Pre vrchol stromu (u, v) máme $DFS(u) < DFS(v)$. Naopak pre zadné hrany platí, že $DFS(v) < DFS(u)$. Ukážka DFS je na obrázku 1.1

Pre každý vrchol grafu G môžeme definovať dve množiny hrán nazývané B_{in}(v) a B_{out}(v). Tieto množiny obsahujú zadné hrany, ktoré vstupujú a opúšťajú v. Všimnime si, že každá zadná hrana v B_{in}(v) spája vrchol v s potomkom v DFS strome, zatiaľ čo zadná hrana v B_{out}(v) spája vrchol v s jeho predchodcom. Hranu stromu e = (u, v), dostaneme tak, že vrátené hrany sú tie zadné hrany, ktoré sú od potomka v (vrátane v samotného) idúce k predchodcovi u odlišného od u samotného. Napokon, dolný bod (lowpoint) vrcholu v, označovaný ako lowpt (v), je najnižší index DFS predchodcu od v, ktorý je dosiahnuteľný prostredníctvom zadnej hrany potomka od v. Analogicky, najvyšším bodom (highpoint) vrcholu v, ktorý sa označuje highpt(v), je najvyšší index DFS predchodcu v, ktorý je dosiahnuteľný prostredníctvom zadnej hrany potomka od v.

1.6 Estetika grafov

Čo robí graf pekným? Ako je možné, že jeden graf môže byť objektívne lepší ako iný? Atribúty, ktoré definujú pekný graf v [Ger99], sa nazývajú estetika a boli stanovené pomocou štatistickej analýzy. Kvality, ktoré stanovili a definujú pekný graf, sú nasledujúce:

- Minimalizácia kríženia
- Minimalizácia oblasti
- Minimalizácia súčtu dĺžok hrán
- Vytvorenie jednotnej dĺžky hrán
- Minimalizácia ohybov

Ako vidíme zo zoznamu vyššie, niektoré požiadavky estetiky sú navzájom v rozpore. Jedna estetická požiadavka minimalizuje súčet dĺžok hrán a ďalšie vyzývajú

na minimalizáciu počtu ohybov. Tieto dve požiadavky sú navzájom v rozpore, pretože v niektorých prípadoch by bolo najlepšie zahrnúť ohyby, aby bola menšia suma dĺžok hrán.

1.7 Vlastnosti kresieb

Máme nekonečne veľa kresieb daného grafu [Nis04]. Keď kreslíme graf, radi by sme zvážili rôzne vlastnosti. V tejto časti predstavíme niektoré vlastnosti kreslenia grafov:

Oblasť. Kresba je zbytočná, ak je nečitateľná. Ak využitá oblasť kresby je veľká, musíme využiť veľa stránok, alebo znížiť rozlíšenie, čiže týmto spôsobom sa kresba stáva nečitateľnou. Preto je jedným z hlavných cieľov zabezpečiť malú oblasť. Malá oblasť kresby je tiež preferovaná v aplikačných doménach.

Pomer strán. Pomer strán je definovaný ako pomer dĺžky najdlhšej strany k dĺžke najkratšej strany najmenšieho obdĺžnika obklopujúceho kresbu.

Ohyby. Kresba s lomenými čiarami v ohybe niektorej z hrán mení smer, a preto ohyb na hrane zvyšuje zložitosť sledovania smeru hrany. Z tohto dôvodu, obe kritéria, celkový počet ohybov a počet ohybov na jednej hrane sa snažíme udržať čo najnižší.

Kríženia. Každé kríženie hrán nesie potenciál splynutia hrán, preto je snaha udržať počet krížení na minime.

Tvar plôch. Ak každá plocha kresby má pravidelný tvar, kresba vyzerá krajšie.

Symetria. Symetria je dôležité estetické kritérium pri kresbe grafov. Symetria dvoj-dimenzionálneho obrázku je izometria oblasti, ktorá fixuje obrázok. Poznáme dva druhy dvoj- dimenzionálnych symetrií a to rotačnú symetriu a reflexnú symetriu. Rotačná symetria je rotácia okolo bodu a reflexná symetria je reflexia na osi.

Uhlové rozlíšenie. Uhlové rozlíšenie sa mení vzhľadom na najmenší uhol medzi dvoma priľahlými hranami v kresbe. Vyššie uhlové rozlíšenie je vhodné na zobrazovanie kresieb na rastrovom zariadení.

1.8 Vzory kreslenia grafov

„Ako iste viete, neexistuje žiadny určitý spôsob, ako nakresliť graf a neexistuje žiaden algoritmus, ktorý by vždy nakreslil najkrajší graf“, hovorí sa v [Ger99]. Preto existuje niekoľko vzorov pre kreslenie grafov, ktoré opisujú základné kroky, ako nakresliť graf. Každý z týchto krokov potom môžeme nahradiť konkrétnymi algoritmami, ktoré majú odlišnú estetiku, alebo majú iné požadované vlastnosti.

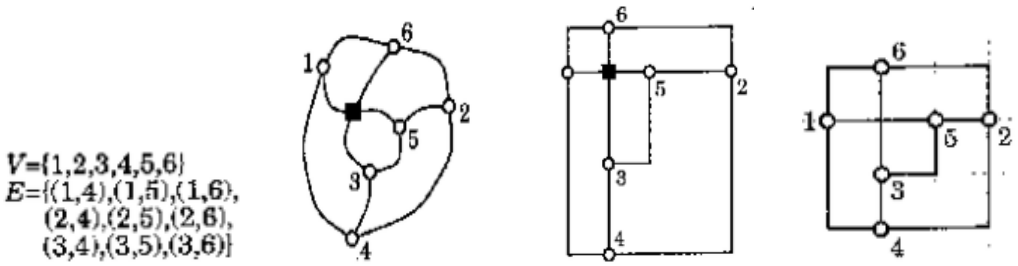
Topológia – tvar – metrika. Tento prístup kreslenia grafov je trojúrovňové zdokonalenie počiatočného grafu. Tento prístup tiež zdôrazňuje minimalizáciu priesečníkov pri kreslení, pretože prvým krokom v postupe je planarizácia.

Tromi formami, ktoré sa použijú v tomto prístupe, sú topológia, tvar a metrika. Dve ortogonálne kresby majú rovnakú topológiu, ak je možné vnoriť jeden do druhého pomocou deformácie, pričom nie je narušené poradie vrcholov a hrán. Dve ortogonálne vykreslenia grafov majú rovnaký tvar, ak je možné vnoriť jeden do druhého zmenou dĺžky hrán, nie však uhlov, ktoré zvierajú hrany. Posledným vylepšením tohto prístupu je metrika. Dve ortogonálne kresby majú rovnakú metriku, ak je možné vnoriť jeden do druhého, iba posunutím a rotáciou.

Každá z hore uvedených foriem poskytuje popis kresby, ktorá zdokonaľuje predchádzajúcu. Konkrétne dve kresby s rovnakou metrikou majú rovnaký tvar, a taktiež dve kresby s rovnakým tvarom majú rovnakú topológiu.

Všeobecný postup pre prístup topológia – tvar – metrika spočíva v tom, že graf najskôr prechádza procesom planarizácie. V tomto kroku sa graf prevedie z jeho matematickej reprezentácie (napríklad V = {1, 2, 3, 4, 5, 6} a E = {(1,4), (1,5) (1,6), (2,4), (2,5), (2,6), (3,4), (3,5), (3,6)}) do 2D výkresu, kde je počet priesečníkov obmedzený na minimum. Ak v splanarizovanom grafe existujú priesečníky, sú nahradené fiktívnymi vrcholmi. Výstupom tohto procesu je prvá forma, topológia.

Po tom, ako graf splanarizujeme, pokračujeme procesom ortogonalizácie. V tomto kroku sú oblúky prerobené do pravých uhlov a vrcholy sú zarovnané. Výsledkom tohto kroku je forma, tvar.



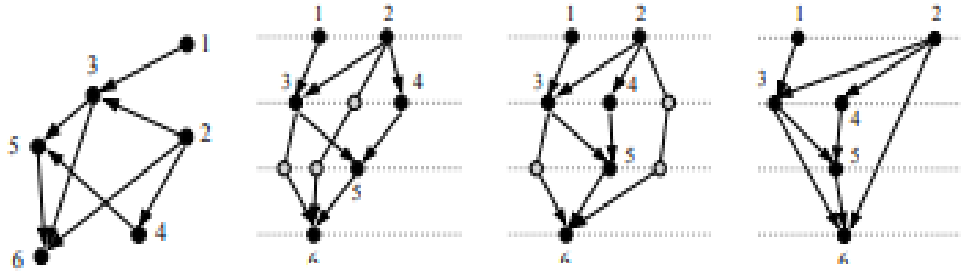
Obrázok 1.2. [Ger99] Príkladom troch krokov v prístupe topológia - tvar - metrika. Prvým krokom je planarizácia a kríženie hrán je nahradené fiktívnym vrcholom. Pri druhom kroku sa graf stáva ortogonálnym a nakoniec je zhutnený do jeho konečnej podoby.

Posledným krokom tohto prístupu je zhustenie. Pri tomto procese sa fiktívne vrcholy odstránia a graf je upravený tak, aby zaberal minimálne množstvo plochy. Po tomto kroku má posledný graf formu, metrika.

Hierarchizácia. Hierarchický prístup je spôsob, ako vytvoriť hierarchie z orientovaných grafov. Prvý krok v hierarchickom prístupe je priradenie vrstiev. V tomto kroku sú vrcholom priradené vrstvy, ktoré začínajú v hornej časti a smerujú nadol. Každý vrchol je priradený vrstve Ln tak, že ak existuje hrana od u do v, a vrstva u je L_i a vrstva v je L_j, potom i<j. Ak existuje medzera medzi vrstvami, napríklad keď hrana prechádza z L_1 do L_3, umiestni sa fiktívny vrchol na L_2.

Po priradení vrstiev nasleduje proces redukcie križovania. V tomto kroku sa redukuje počet krížení hrán, aby graf vyzeral krajšie.

Posledný krok sa nazýva priradenie x-súradníc. V tomto štádiu sa všetkým bodom priradí pozícia, odstránia sa fiktívne vrcholy a všetky hrany sa vykreslia. V tomto kroku je možné zdôrazniť veľa rôznych estetických zmien, ako je minimalizácia ohybov alebo minimalizácia plochy.

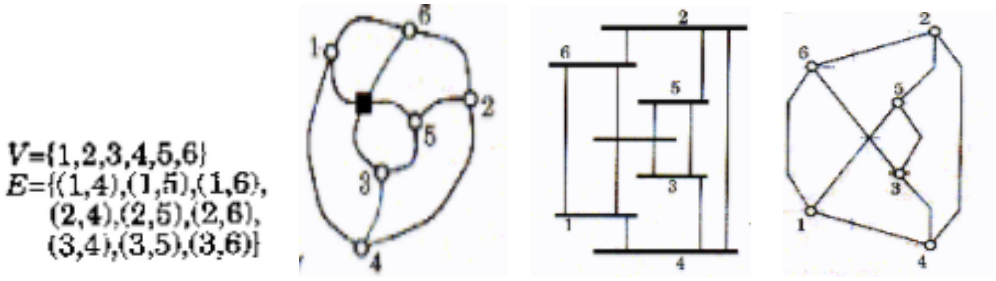


Obrázok 1.3. Vyššie uvedený obrázok z [Ger99] je príkladom hierarchického prístupu v procese. Priradia sa vrstvy a vložia sa fiktívne vrcholy. Ďalej sa zníži počet krížení čiary a potom sa konečná podoba je bez fiktívnych vrcholov.

Viditeľnosť. Prístup viditeľnosti pre kreslenie grafov je všestranná všeobecne používaná technika na kreslenie nadväzujúcich úsečiek. Rovnako ako v prístupe topológia – tvar – metrika, prvým krokom v tomto prístupe je planarizácia. Opäť, mať tento proces ako prvý krok znamená, že prioritou je zníženie počtu krížení.

Krok viditeľnosti je v tomto prípade ťažké vysvetliť bez pohľadu na obrázok 1.4. nižšie. V podstate sa každý vrchol mení na vodorovnú čiaru, vrátane fiktívneho vrcholu. Všetky hrany sa potom nahradia vertikálnymi čiarami, ktoré spájajú vodorovné čiary.

Konečný diagram sa potom vytvorí po dokončení výmenného kroku. V tomto kroku sa horizontálne čiary nahradia jednotlivými bodmi a hrany sa ťahajú tak, aby tesne nasledovali zvislé čiary.

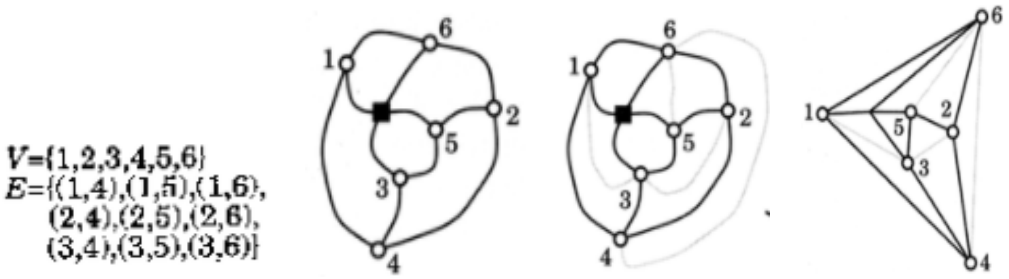


planarizácia viditeľnosť výmenný krok

Obrázok 1.4. Vyššie uvedený obrázok z [Ger99] je príkladom prístupu viditeľnosti. Prvým krokom je planarizácia. Estetika minimalizácie plochy je zdôraznená vo fáze viditeľnosti a v záverečnom kroku môže byť zdôraznené množstvo rôznych estetík v závislosti od preferencie používateľa.

Augmentácia. Prístup augmentácie je podobný prístupu viditeľnosti v tom, že ide o general purpose graph drawing paradigm a jeho prvým krokom je planarizácia.

Druhý krok sa nazýva krok augmentácie. V tomto procese sa nakreslia falošné hrany v poradí s cieľom získať graf, v ktorom má každá plocha tri strany (maximálne rovinné). V nižšie uvedenom príklade vidíme, že na prvom obrázku má plocha {3,5,2,4} štyri strany. Po dokončení augmentačného kroku sa nakreslí falošná hrana {3,2}, čím sa vytvoria dve 3-stranné plochy {3,5,2} a {3,2,4}.



planarizácia augmentácia triangulácia

Obrázok 1.5. Opäť, tento prístup z [Ger99] zdôrazňuje minimalizovanie kríženia hrán, ale okrem tejto estetiky sa tiež snaží minimalizovať ohyby v dôsledku kroku priamych čiar k triangulácii.

V triangulačnom kroku sa nakreslí konečný graf pomocou geometrických vlastností trojuholníkov, vytvorených v predchádzajúcom kroku. Fiktívne vrcholy sú v tomto kroku odstránené a všetky hrany sú zmenené na priame čiary.

2 Špecifikácia projektu a návrh riešenia

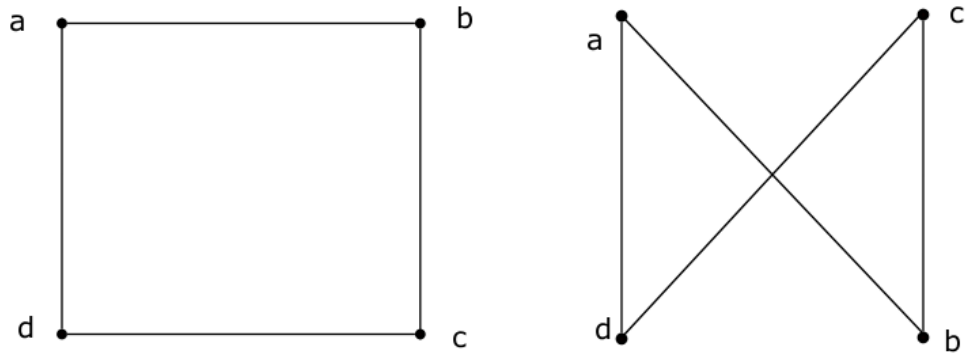
V tejto časti práce sa budeme venovať štýlom kresieb, a popíšeme podrobnejšie prístup topológia – tvar – metrika. Tým máme na mysli, že popíšeme jednotlivé algoritmy, v ktorých sa píše, ako vytvoriť jednotlivé kresby zo zadaného grafu a stručný opis k nim.

Pre príliš veľkú zložitosť sme sa rozhodli, že nebudeme robiť automatické kreslenie grafov, ale umožníme používateľovi upravovať graf v aplikácii interaktívne. Rozhodli sme sa použiť programovací jazyk C#, s ktorým máme skúsenosti z predchádzajúceho štúdia.

2.1 Štýly kresieb

2.1.1 Planárna kresba

Na obrázku 2.1 vidno planárnu kresbu a neplanárnu kresbu toho istého grafu. Je výhodnejšie nájsť planárnu kresbu grafu, ak má graf takúto vlastnosť. Samozrejme, nie každý graf má svoju planárnu kresbu.



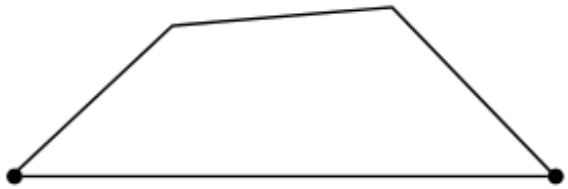
Obrázok 2.1 (a) planárna kresba, a (b) neplanárna kresba toho istého grafu

Ak chceme nájsť planárnu kresbu zadaného grafu, ako prvé potrebujeme otestovať, či je daný graf planárny, alebo nie, píše sa v [Nis04]. Ak je planárny, potrebujeme nájsť planárne vnorenie nášho grafu, čo je dátová štruktúra reprezentujúca adekvátne zoznamy: v každom zozname hrany závislé od vrcholu sú usporiadané, buď všetky v smere chodu hodinových ručičiek, alebo proti tomuto smeru, vzhľadom na planárne vnorenie. Planárny graf s fixným planárnym vnorením je rovinný graf.

2.1.2 Kresby s lomenými hranami

Kresba s lomenými hranami je kresba daného grafu, v ktorej každá hrana tohto grafu je reprezentovaná polygonálnym reťazcom, [Nis04]. Tento typ kresby je zobrazený

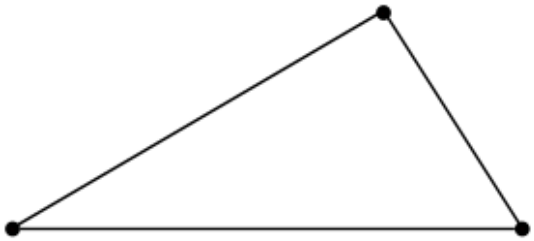
na obrázku 2.2. Bod, v ktorom hrana mení svoj smer v kresbe s lomenými čiarami, sa nazýva ohyb. Kresby s lomenými hranami sú užitočné na aproximáciu kresieb so zakrivenými hranami. Avšak nie je vhodné, aby na jednej hrane bolo viac ako dva alebo tri ohyby.



Obrázok 2.2 Kresba grafu s lomenými čiarami

2.1.3 Kresba s priamymi hranami

Kresba s priamymi hranami je kresba grafu, v ktorej každá hrana grafu je kreslená ako úsečka [Nis04], ako je ilustrované na obrázku 2.3. Kresba s priamymi čiarami je špeciálnym prípadom kresby s lomenými čiarami, kde hrany sú kreslené tak, aby neobsahovali žiadne ohyby. Wagner, Féry a Stein nezávisle na sebe dokázali, že každý planárny graf má svoju kresbu s priamymi hranami.



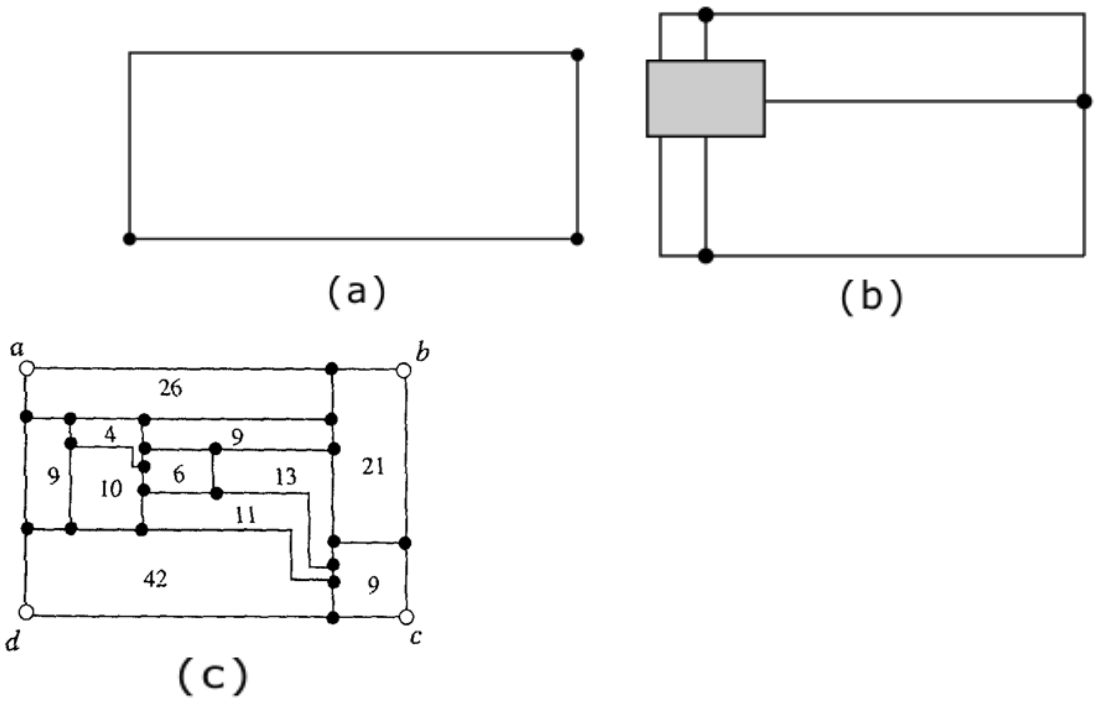
Obrázok 2.3 Kresba s priamymi hranami a aj konvexná kresba

2.1.4 Konvexné kresby

Kresba s rovnými čiarami rovinného grafu G sa nazýva konvexná kresba, ak hranice všetkých plôch G sú nakreslené ako konvexné polygóny (všetky vnútorné uhly sú menšie ako 180°) [Nis04], ako môžete vidieť na obrázku 2.3. Avšak nie každý rovinný graf sa dá upraviť na konvexnú kresbu, iba každý 3- súvislý rovinný graf má takýto typ kresby.

2.1.5 Ortogonálne kresby

Ortogonálna kresba je kresba rovinného grafu, v ktorej je každá hrana nakreslená ako reťazec vodorovných a zvislých úsečiek [Nis04], ako je ilustrované na obrázku 2.4(a). Ortogonálne kresby priťahujú veľa pozornosti vďaka ich početným aplikáciám napríklad v databázových diagramoch. Ortogonálna kresba sa nazýva oktagonálnou, ak každý vonkajší cyklus je vykreslený ako obdĺžnik a každá vnútorná plocha je vykreslená ako priamkový polygón s najviac ôsmymi rohmi.



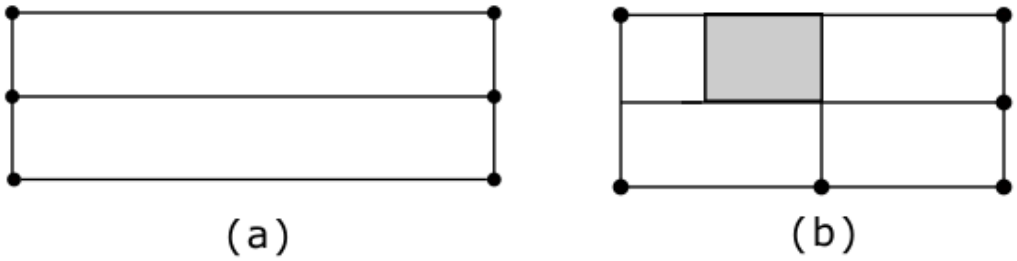
Obrázok 2.4 (a) ortogonálna kresba, (b) krabicovo-ortogonálna kresba, (c) oktagonálna kresba z [Nis04]

2.1.6 Krabicovo-ortogonálne kresby

Formálne každý vrchol v ortogonálnej kresbe je vykreslený ako bod [Nis04], ako na obrázku 2.4(a). Očividne graf, ktorého vrchol je stupňa päť a viac, nemá ortogonálnu kresbu. Krabicovo-ortogonálna kresba grafu má každý vrchol vykreslený ako obdĺžnik, ktorý nazývame krabica a každá hrana je vykreslená ako postupnosť striedajúcich sa vodorovných a zvislých úsečiek, ako je ilustrované na obrázku 2.4(b). Každý rovinný graf sa dá upraviť na krabicovo-ortogonálnu kresbu.

2.1.7 Obdĺžnikové kresby

Obdĺžniková kresba rovinného grafu G je kresba G, v ktorej všetky vrcholy sú vykreslené ako vrcholy, každá hrana je vykreslená ako vodorovná alebo zvislá úsečka bez kríženia hrán a každá plocha je vykreslená ako obdĺžnik [Nis04], ako na obrázku 2.5(a). Vieme, že nie každý rovinný graf má obdĺžnikovú kresbu. Thomassen a Rahman vyslovili nutné a postačujúce podmienky na to, aby mal rovinný graf obdĺžnikovú kresbu, ktoré hovoria, že graf musí byť maximálne stupňa tri.



Obrázok 2.5 (a) obdĺžniková kresba a (b) krabicovo-obdĺžniková kresba

2.1.8 Krabicovo-obdĺžnikové kresby

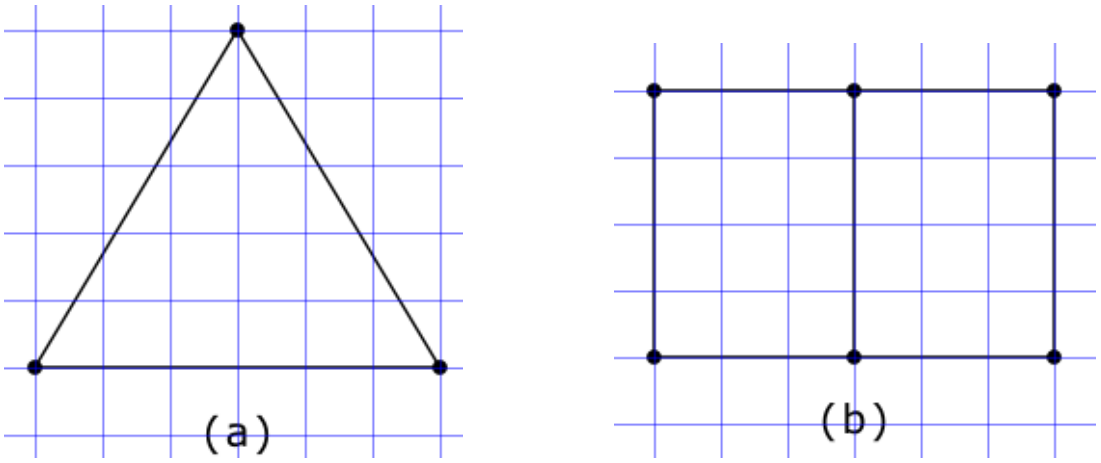
Krabicovo-obdĺžniková kresba rovinného grafu je taká kresba G v rovine, že každý vrchol je vykreslený ako obdĺžnik, ktorý voláme krabica a obrys každej plochy je vykreslený ako obdĺžnik [Nis04], ako je ilustrované na obrázku 2.5(b). Ak G má viacero hrán alebo vrcholov stupňa päť a viac, potom G nemá obdĺžnikovú kresbu, ale môže mať krabicovo-obdĺžnikovú kresbu. Avšak nie každá rovinná kresba má krabicovo-obdĺžnikovú kresbu.

2.1.9 Mriežkové kresby

Kresba grafu, v ktorej sú vrcholy a ohyby umiestnené na bodoch v mriežke [Nis04], ako je ilustrované na obrázku 2.6, sa nazýva mriežková kresba. Prístup mriežkových kresieb prekoná nasledujúce problémy pri kreslení grafov, ktoré vznikajú v dôsledku práce s reálnymi číslami.

a) Keď vnorenie má byť vykreslené na rastrovom zariadení, reálne súradnice vrcholu treba zmeniť na celočíselné mriežkové body, pričom nie je zaručené, že po zaokrúhľovaní sa dosiahne správne vnorenie.
b) Veľa vrcholov môže byť koncentrovaných na malom priestore našej kresby. Teda vnorenie môže byť chaotické a priesečníky čiar nemusia byť zachytené.
c) Nie je možné porovnať požadovanú oblasť pre dve alebo viac rôznych výkresov

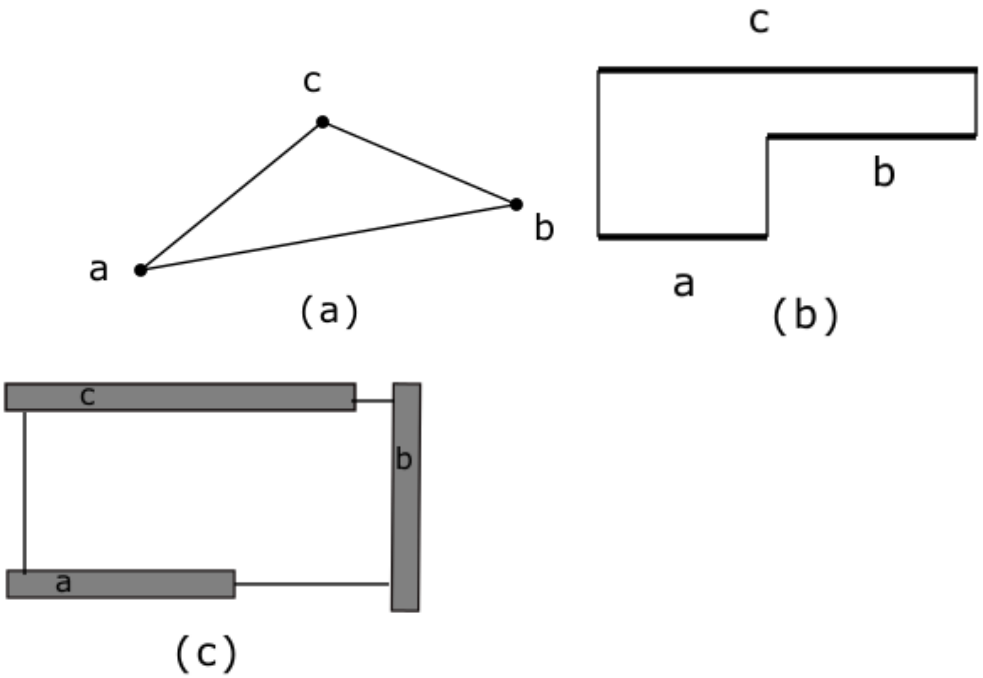
pomocou reálnej aritmetiky, pretože akákoľvek kresba sa nemusí hodiť na malú oblasť použitím zväčšenia.



Obrázok 2.6 (a) mriežková kresba s rovnými čiarami, (b) obdĺžniková mriežková kresba

2.1.10 Kresby viditeľnosti

Kresba viditeľnosti (visibility drawing) rovinného grafu G je kresba, kde každý vrchol je vykreslený ako vodorovná úsečka a každá hrana je vykreslená ako zvislá úsečka [Nis04].



Obrázok 2.7 (a) rovinný graf G, (b) kresba viditeľnosti G, (c) 2-viditeľná kresba G

Zvislá úsečka reprezentujúca hranu musí spájať body na vodorovných útvaroch, ktoré reprezentujú koncové vrcholy. Obrázok 2.7(b) znázorňuje kresbu viditeľnosti rovinného grafu G na obrázku 2.7(a).

2-viditeľná kresba je zovšeobecnením viditeľnej kresby, kde vrcholy sú vykreslené ako krabice a hrany sú vykreslené buď ako vodorovné alebo zvislé úsečky. Na obrázku 2.7(c) môžeme vidieť 2-viditeľnú kresbu rovinného grafu na obrázku 2.7(a).

2.2 Topológia – tvar – metrika

Tento vzor kreslenia grafov je trojúrovňový, ako sme uviedli už vyššie. Pozostáva z týchto troch krokov: planarizácia, ortogonalizácia a zhustenie. Každý z týchto troch krokov popíšeme podrobnejšie.

2.2.1 Planarizácia

Planarizačné techniky sú motivované dostupnosťou mnohých efektívnych a dobre analyzovaných kresliacich algoritmov pre planárne grafy [DETT99]. Ak graf nie je planárny, tak ho transformujeme na planárny prostredníctvom planarizačného kroku, v ktorom nahradíme každé kríženie fiktívnym vrcholom, a potom aplikujeme kresliacu metódu pre planárne grafy.

2.2.1.1 Relatívna planarizácia

Algoritmus planarizácie je jednoduchý algoritmus pre nájdenie planarizácie grafu. Tento postup obsahuje hľadanie planárneho podgrafu.

V prvom kroku planarizácie vytvoríme maximálny planárny podgraf zo vstupného grafu. Táto metóda má solídne teoretické základy v polyhedrálnej kombinatorike a dosahuje dobré výsledky v praxi. Druhý krok môžeme vykonať použitím niektorého z množstva algoritmov testujúcich planaritu. Pri treťom kroku je cieľom vytvorenie algoritmu najkratšej cesty. Niekoľko obmedzení typologického charakteru môže byť podporovaných v rámci planarizácie, vrátane:

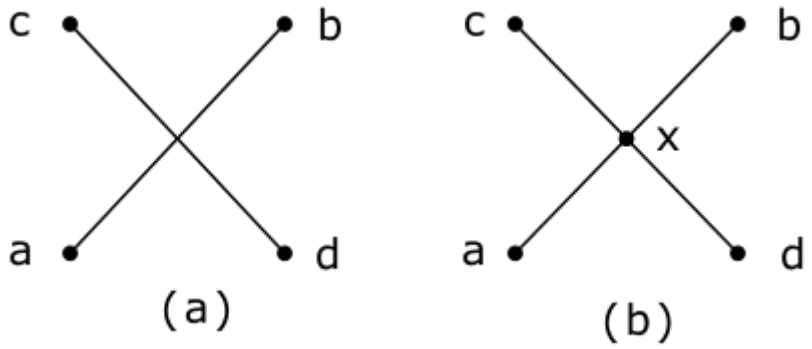
- Predchádzania kríženia v grafe
- Umiestnenie vrcholov na vonkajšiu hranicu

Keďže sme sa rozhodli, že kresliacu metódu budeme programovať tak, že vytvoríme planárnu kresbu interaktívne, modifikovali sme kroky 2, 3 algoritmu planarizácie nasledovne:

- V druhom kroku posúvame neplanárne hrany grafu G tak, aby sme minimalizovali počet krížení s planárnymi hranami.

- V treťom kroku pridávame fiktívne vrcholy tam, kde sa priesečníky nedajú odstrániť posunutím hrany.

Planarizačná operácia pre graf na obrázku 2.8. pozostáva z toho, že pridáme nový vrchol x a nahradíme dvojicu (a ,b), (c ,d) pretínajúcich sa hrán, štyrmi hranami (a ,x), (b ,x), (c ,x) a (d ,x). Fiktívny vrchol x pridávame na miesto, v ktorom sme vypočítali, že sa hrany (a ,b) a (c ,d) pretínajú. Planarizáciu G´ grafu G získame z planárneho grafu G tak, že urobíme túto planarizačnú operáciu. V [DETT99] ukázali, že každý graf má planarizáciu.



Obrázok 2.8: (a) neplanárna kresba, (b) splanarizovaná kresba

Algoritmus Planarizácia

Vstup: graf G

Výstup: planarizácia G´ z G

1. Vytvoríme maximálne planárny podgraf S zo vstupného grafu G a rozdelíme hrany do dvoch skupín „planárne“ a „neplanárne“ nasledovne:
    a. Začnime s podgrafom G´ pozostávajúcim iba z vrcholov G, ale bez hrán.
    b. Pre každú hranu e z G, ak graf získaný z pridávania e do G´ je planárny, potom pridajme e do G´ a zaraďme ju do skupiny „planárne“, inak vráťme e a klasifikujme ju ako „neplanárnu.“
2. Vytvorme planárne vnorenie planárneho podgrafu G´ a duálneho grafu S.
3. Pridajme k G´ neplanárne hrany jednu po druhej a zakaždým minimalizujme počet prekrížení. Toto spravíme pre neplanárnu hranu (u ,v) nasledovne:
    a. Nájdime najkratšiu (najmenší počet hrán) cestu v duálnom grafe z aktuálneho vnorenia G´ z plochy závislých na u do plochy závislých na v.

b. Pridajme neplanárnu hranu a aktualizujme G´ a S.

2.2.2 Ortogonalizácia

V tejto časti sa budeme zaoberať problémom ortogonalizácie vnoreného planárneho grafu, inak povedané vytvorením mriežkovo – ortogonálnej kresby, ktorá bola vytvorená algoritmom planarizácie [DETT99]. Dôležitou estetickou podmienkou pre planárnu ortogonálnu kresbu je minimalizácia počtu ohybov. Kresba musí spĺňať:

1. Uhol medzi každými dvoma hranami by mal byť násobkom 90 stupňov.
2. Každý vrchol musí byť maximálne stupňa štyri.

Ak nie sú splnené tieto dve podmienky, daná kresba nie je ortogonálna.

V tejto práci sa snažíme ortogonalizáciu dosiahnuť takto: V prvom kroku posúvame vrcholy spolu s k nim incidentnými hranami do bodov, kde sa čiary mriežky pretínajú. V druhom kroku rozdelíme hrany, ktoré sa nedali inak zortogonalizovať na dve a vytvoríme neviditeľný vrchol, ktorý sa dá prichytiť iba na priesečník čiar mriežky.

2.2.3 Zhustenie

V tejto časti riešime problém zhustenia ortogonálnej reprezentácie, teda vytvoríme ortogonálnu mriežkovú kresbu z danej mriežkovo – ortogonálnej reprezentácie. Chceme priradiť dĺžky segmentom hrán ortogonálnej reprezentácie tak, že v kresbe nebudú žiadne kríženia ani presahy medzi hranami a vrcholmi. Chceme, aby obaľujúci obdĺžnik kresby bol čo najmenší.

Algoritmus pre zhustenie rieši zvlášť rovnobežné a zvislé úsečky. Autori v [DETT99] sa zaoberajú opísaním tohto problému pre špeciálny prípad, že všetky plochy ortogonálnej reprezentácie majú tvar obdĺžnika a aj všeobecným prípadom. My sa budeme zaoberať všeobecným prípadom, teda jednotlivé plochy nemusia mať obdĺžnikový tvar. Finálna kresba by mala mať minimálnu výšku, šírku, oblasť a dĺžku hrán.

Pri zhustení všeobecnej ortogonálnej reprezentácie budeme postupovať tak, že najskôr zrealizujeme zhustenie vo vodorovnom smere a následne vo zvislom smere nasledovne:

1. Nájdeme najmenšiu x-ovú súradnicu vrcholu spomedzi všetkých vrcholov.
2. Nájdeme najmenšiu x-ovú vzdialenosť dvoch vrcholov kresby.
3. Túto vzdialenosť priradíme každým dvom po sebe idúcim bodom, pričom postupujeme zľava doprava.
4. Nájdeme najväčšiu y-ovú súradnicu vrcholu spomedzi všetkých vrcholov.

5. Nájdeme najmenšiu y-ovú vzdialenosť dvoch vrcholov kresby.
6. Túto vzdialenosť priradíme každým dvom po sebe idúcim bodom, pričom postupujeme zdola nahor.

3 Implementácia riešenia

V tejto časti práce sa venujeme implementácii všetkých troch algoritmov prístupu topológia – tvar – metrika. Popíšeme formát vstupu, výstupu, ako fungujú jednotlivé funkcie v programe a ako funguje program, pričom program sme spravili tak, že platí nasledovné:

- Používateľ nemôže začať vykonávať kroky nasledujúceho algoritmu bez toho, aby boli pre danú kresbu splnené kroky, ktoré je potrebné splniť pre daný algoritmus.

3.1 Dátový model

Program obsahuje 3 triedy *Graph, Edge* a *Vertex*.

Vertex. Trieda Vertex obsahuje súradnice vrcholu a niekoľko ďalších vlastností. Počas behu programu graf niekoľkokrát upravujeme a k vrcholom grafu pridávame ďalšie vrcholy v podobe ohybov (vlastnosť visible je nastavená na false) a imaginárnych vrcholov (vlastnosť dummy je nastavená na true). Každý vrchol obsahuje metódu *AddNeigbour* a zoznam *Neighbours*. Sú to vrcholy, ktoré sú s ním susedné. Zoznam *Neighbours* využívame pri zisťovaní, či vytvorená planárna kresba môže byť ortogonálna, teda zisťujeme, či stupeň každého vrcholu je menší ako päť, alebo väčší ako jedna.

Edge. Trieda *Edge* obsahuje vlastnosti hrany grafu – súradnice vrcholov. Pomocou logickej funkcie *PointOnEdge* zisťujeme, či sme klikli na danú hranu.

Graph. Všetky vrcholy a hrany sa pridávajú do objektu typu *Graph.* Pričom *Graph* obsahuje zoznamy *Vertices,* v ktorom si pamätáme vrcholy a *Edges,* v ktorom si pamätáme všetky hrany. Trieda Graph obsahuje metódy *AddVertex* na pridávanie vrcholov do grafu a *AddEdge* na pridávanie hrán do grafu.

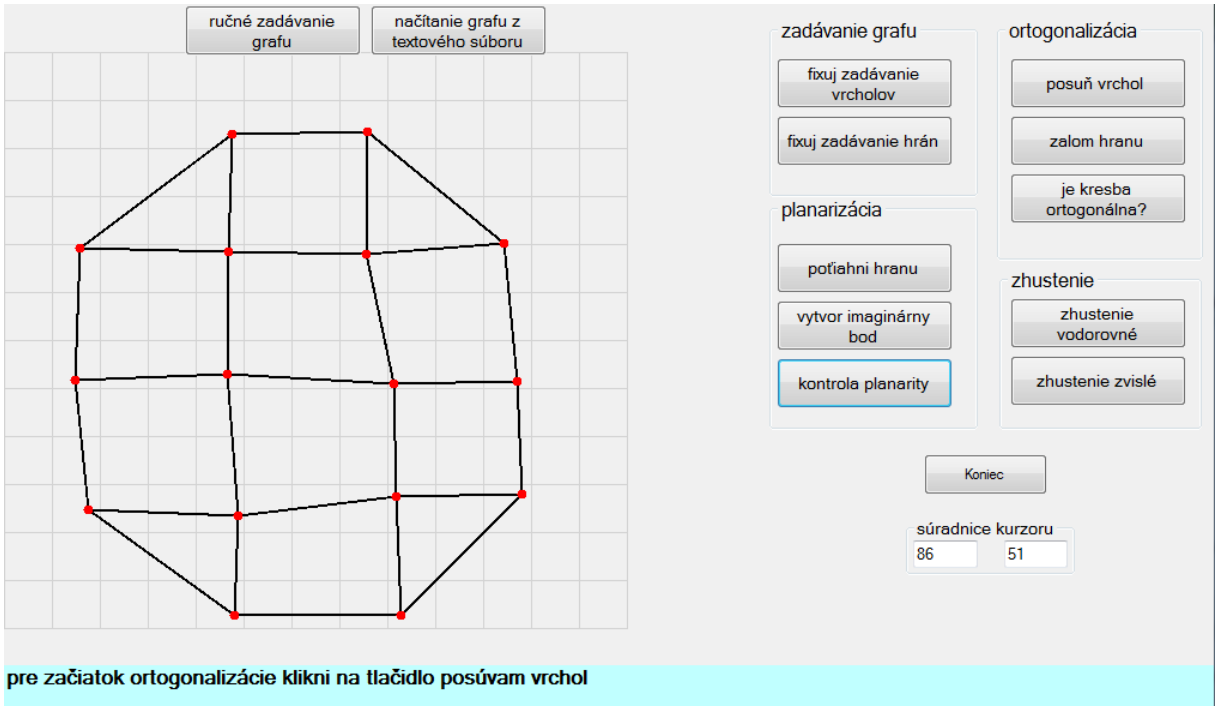
V triede *Graph* zisťujem, či nejaké dve hrany sa pretínajú s využitím funkcie *isIntersect*. Výpočet sa robí cez parametrické vyjadrenie priamok (obrázok 3.1). Túto funkciu používame pri zisťovaní, či graf je planárny alebo nie.

```
if (!this.Parallel(a,c))
{
    double t,s;
    double Ax = Convert.ToDouble(a.V1.GetPoint().X);
    double Ay = Convert.ToDouble(a.V1.GetPoint().Y);
    double Bx = Convert.ToDouble(a.V2.GetPoint().X);
    double By = Convert.ToDouble(a.V2.GetPoint().Y);
    double Cx = Convert.ToDouble(c.V1.GetPoint().X);
    double Cy = Convert.ToDouble(c.V1.GetPoint().Y);
    double Dx = Convert.ToDouble(c.V2.GetPoint().X);
    double Dy = Convert.ToDouble(c.V2.GetPoint().Y);

    s = (Ay +((Cx-Ax)*(By-Ay))/(Bx-Ax)-Cy)/((Dy-Cy)-((Dx-Cx)*(By-Ay))/(Bx-Ax));
    t = (Cx-Ax)/(Bx-Ax)+s*((Dx-Cx)/(Bx-Ax));

    if (t > 0.01 && t < 0.99 && s > 0.01 && s < 0.99)
    {
        return true;
    }
```

Obrázok 3.1 funkcia *isIntersect*

Na zistenie, či sú dve hrany rovnobežné, máme funkciu *Parallel*. Ďalej *Graph* obsahuje metódy na vykreslenie vrcholov *DrawVertices*, na vykreslenie hrán *DrawEdge*. Metóda DrawVertices vykresľuje vrcholy, pričom regulárne vrcholy sú červené, imaginárne vrcholy sú čierne a vrcholy vzniknuté zalomením hrán sú modré. V metóde *DrawEdge* vieme nastaviť farbu, ktorou odlišujeme planárne a neplanárne hrany.

3.2 Pracovná plocha

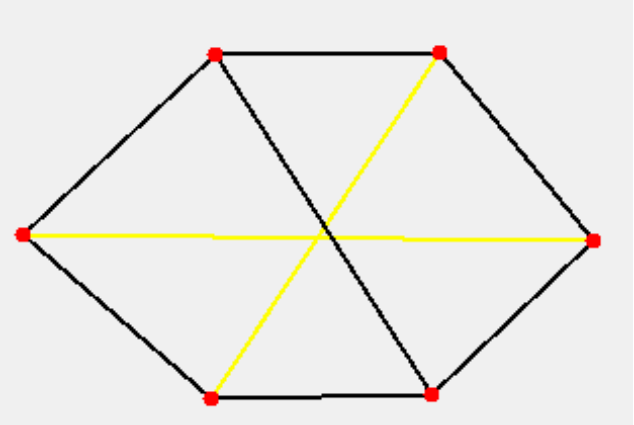


Obrázok 3.2 Vzhľad plochy aplikácie

Pracovná plocha je rozdelená na viacero sekcii (obrázok 3.2). Hore vľavo si používateľ na začiatku vyberie, či chce graf kresliť ručne alebo ho načíta z textového súboru. V pravej časti pracovnej plochy sú jednotlivé tlačidlá rozdelené do sekcii, podľa toho v akej fáze úpravy grafu sme. Dole vpravo sa zobrazujú súradnice kurzoru. V dolnej časti pracovnej plochy sa vypisuje nápoveda k jednotlivým krokom úpravy grafu. Graf sa dá kresliť a upravovať iba vo vymedzenej časti pracovnej plochy.

3.3 Vstup

Na vstupe načítame niekoľko grafov z textových súborov, ktoré sú spravené tak, aby sa dali na nich dobre porozumieť jednotlivé kroky algoritmov. Alebo má používateľ možnosť vytvorenia vlastného grafu, a to tak, že si nakliká body do vymedzenej plochy. Ak už používateľ nechce zadávať žiadne iné vrcholy, tak stlačí tlačidlo *fixuj pridávanie vrcholov*. Následne ich pospája hranami, a ak nechce pridávať ďalšie hrany, stlačí tlačidlo *fixuj zadávanie hrán*. Pri zadávaní hrán sa pomocou funkcie *isIntersect* testuje, či daná hrana pretína už existujúce planárne hrany grafu alebo nie. Podľa výsledku sa pridá do planárnych alebo neplanárnych hrán. Planárne hrany sú vykreslené čiernou farbou, neplanárne žltou farbou, ako je znázornené na obrázku 3.3 .



Obrázok 3.3 Zadaný graf na vstupe

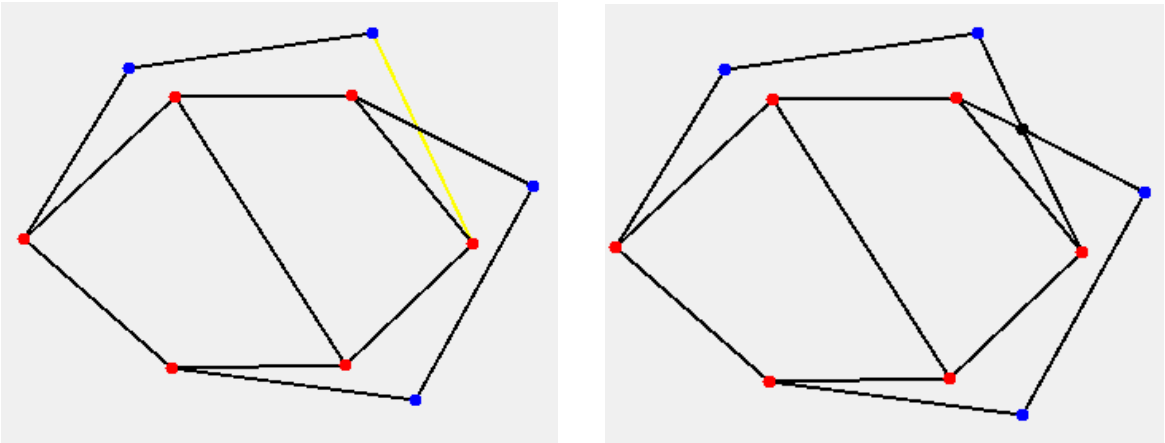
3.4 Ako funguje program

V hlavnom programe sme vytvorili globálnu premennú *status*, ktorá má pre každú časť programu inú hodnotu. Pomocou tejto premennej zabezpečujeme, aby používateľ nemohol prejsť od jedného kroku k akémukoľvek inému kroku. Napríklad, ak používateľ klikne na tlačidlo *načítanie z textového súboru*, nedá sa už vykonať ručné zadávanie grafu. Keď je používateľ vo fáze ortogonalizácie grafu, nemôže sa vrátiť k planarizácii.

Pri každom kroku je v spodnej časti pracovnej plochy vypísaná nápoveda, ktorá používateľa informuje o tom, čo by mal v danom kroku urobiť.

3.4.1 Planarizácia

Ak už je graf vytvorený, začne sa proces planarizácie. Po stlačení tlačidla *potiahni hranu* môže používateľ uchopiť iba neplanárnu hranu stlačením myši a posunúť ju tak, aby sa minimalizoval počet priesečníkov (obrázok 3.4 (a)). Ak sa už posúvaním hrán nedá zbaviť žiadnych ďalších priesečníkov, treba pridať imaginárny bod stlačením tlačidla *vytvor imaginárny bod* a kliknutím na priesečník planárnej a neplanárnej hrany (obrázok 3.4 (b)). Imaginárny bod je pridaný do grafu s vlastnosťou *dummy* nastavenou na true. Pôvodné hrany sú rozdelené a otestované, či sú planárne alebo neplanárne.



Obrázok 3.4 (a) neplanárny graf po potiahnutí neplanárnych hrán, (b) splanarizovaný graf pridaním imaginárneho vrcholu

Po stlačení tlačidla *kontrola planarity* sa testuje, či existujú ešte nejaké neplanárne hrany. Ak už žiadne neexistujú, vykreslí sa mriežka. Inak treba neplanárne hrany splanarizovať posúvaním hrán a pridávaním imaginárnych bodov.

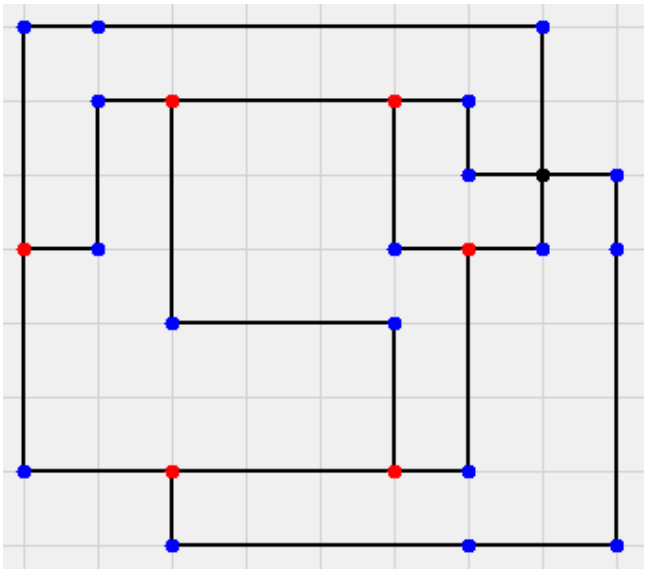
3.4.2 Ortogonalizácia

Po vykreslení mriežky začína proces ortogonalizácie. Ak má nejaký vrchol menej ako dvoch alebo viacej ako piatich susedov (počet prvkov v zozname Neighbours daného vrcholu), vypíše sa oznam, že kresba nie je ortogonálna a program je ukončený. Ak kresba môže byť ortogonálna, treba vrcholy posunúť do priesečníkov mriežky tak, aby všetky uhly v grafe boli násobkami 90 stupňov. Tento proces sa začne kliknutím na tlačidlo *posuň vrchol*. Používateľ by mal uchopiť myšou vrchol a posunúť ho na potrebné miesto. Vrcholy je možné prichytiť iba na priesečníky mriežky, ktorej odstupy sú násobkami čísla

40. Ak používateľ nepresunie vrchol na správnu pozíciu, vrchol je vrátený na pôvodné miesto. S vrcholom sú presúvané aj všetky hrany, ktoré sú s ním incidentné. (Na zistenie incidentných hrán sa používa zoznam *Neighbours*.)

Ak posúvaním vrcholov nie je možné dosiahnuť uhly s veľkosťou 90 stupňov a násobkov 90, treba zalomiť hranu. Po stlačení tlačidla *zalom hranu,* je možné rozdeliť hranu na dve časti a prichytiť ju do priesečníku čiar mriežky.(obrázok 3.5)

Stlačením tlačidla *je kresba ortogonálna?*, môže používateľ skontrolovať, či kresba je alebo nie je ortogonálna. Ak kresba ortogonálna nie je, je možné pokračovať v procese ortogonalizácie.



Obrázok 3.5 Zortogonalizovaná kresba 3.4(b)

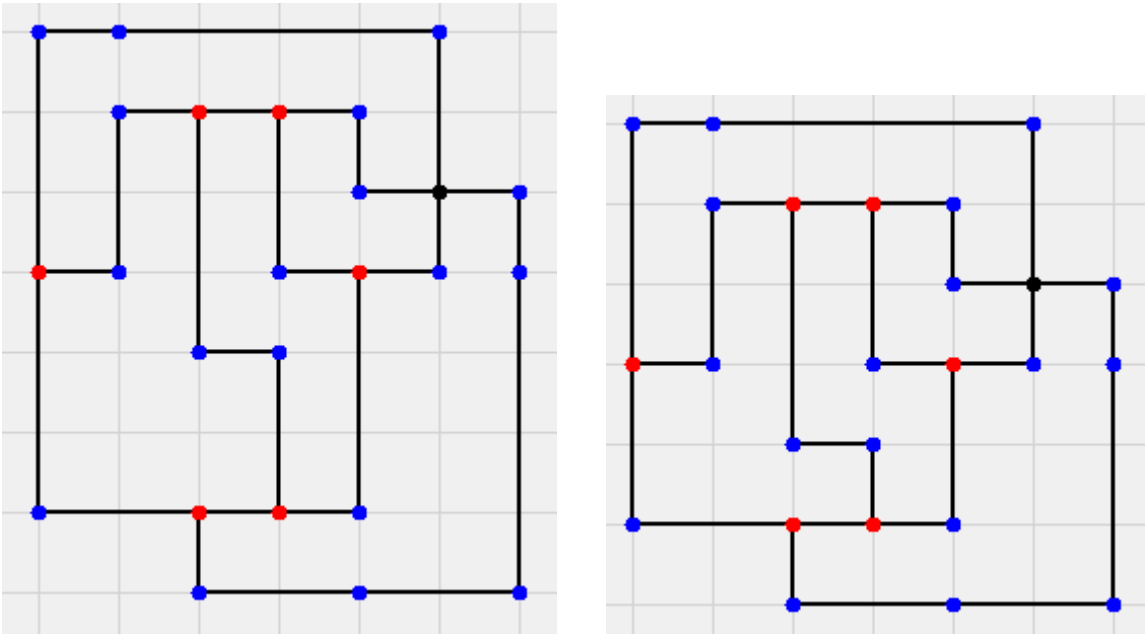
3.4.3 Zhustenie

Ak je kresba ortogonálna, pokračuje sa procesom zhustenia. Zhustenie najskôr prebieha vo vodorovnom smere, a potom v zvislom smere.

Používateľ klikne na tlačidlo *zhustenie vodorovné*. V tomto kroku sa postupuje zľava doprava. Vrcholy, ktoré majú rovnakú súradnicu x a zvislé hrany s tou istou súradnicou x nazývame *úroveň*. Program automaticky nájde úroveň, ktorá je najviac vľavo. Následne ju zafixuje. Používateľ by mal kliknúť na ľubovoľný vrchol alebo hranu v kresbe, ktorú chce posúvať doľava. Ak tento bod leží na najľavejšej úrovni, ktorá nie je zafixovaná, kontroluje sa, či jej vzdialenosť od poslednej zafixovanej úrovne je väčšia ako *diferenceX* (čo je najmenšia vzdialenosť dvoch susedných vrcholov zo všetkých dvojíc v grafe). Ak je väčšia ako *diferenceX*, program zmení súradnicu x celej úrovne nasledovným spôsobom: k poslednej zafixovanej úrovni sa pripočíta diferenceX a na túto

súradnicu sa presunie vybraná úroveň. Následne je úroveň zafixovaná. Ak vzdialenosť vybranej úrovne nie je väčšia ako *diferenceX*, úroveň je zafixovaná. Tento postup sa opakuje, kým nie sú zafixované všetky úrovne (obrázok 3.6 (a)).

Po ukončení vodorovného zhustenia treba kliknúť na tlačidlo *zhustenie zvislé* a podobne sa postupuje pri zhustení v zvislom smere. Postupuje sa v smere zdola nahor a fixuje sa úroveň s najväčšou y-ovou súradnicou. Ďalší postup je podobný ako pri vodorovnom zhustení (obrázok 3.6 (b)).



Obrázok 3.6 (a) vodorovné zhustenie obrázku 3.5, (b) zvislé zhustenie obrázku 3.6(a)

Po tom, ako je kresba zhustená, treba odstrániť imaginárne vrcholy, ktoré boli pridané pri procese planarizácie.

3.5 Výstup

Kresba podľa prístupu topológia – tvar – metrika.

Program spolu s textovými súbormi budeme postupne aktualizovať na webe http://flurry.dg.fmph.uniba.sk/web_studenti/kobliska/index.html. Stránka sme vytvorili v HTML 5 a štýl stránky sme programovali v CSS.

Záver

Výsledkom tejto práce je aplikácia, na ktorej si používateľ vie manuálne vyskúšať jednotlivé fázy prístupu topológia – tvar – metrika.

Motiváciou tejto práce bolo podporiť výučbu kreslenia grafov v predmetoch, ako je Webovská grafika alebo iné predmety, kde sa vyučuje táto problematika.

Cieľom práce bolo poskytnúť základný, ale relatívne komplexný úvod do vybraných algoritmov kreslenia planárnych grafov tak, aby boli zrozumiteľné pre študenta, či iného čitateľa, ktorého táto téma zaujíma. Preto sme v tejto práci opisovali viacero vzorov kreslenia grafov a taktiež typy a najdôležitejšie vlastnosti kresieb. Hlavným cieľom práce bolo popísanie prístupu topológia – tvar – metrika a taktiež jeho aktivizačná implementácia. V práci sa nám podarilo naplniť vytýčený cieľ.

Pri projekte sme použili 12 prameňov, 5 algoritmov. Vytvorili sme 3 textové súbory, ktoré slúžia ako vstupné údaje pre vytvorenie grafu, 1010 riadkov softvéru, 1 webovú stránku, 1 video, 28 obrázkov. Navrhli sme aktivizačnú verziu daného prístupu, 13 príkladov, ktoré na minimálnych dátach ilustrujú danú vlastnosť kresby (obrázky 2.1 – 2.8). Tieto obrázky boli vytvorené v programe Inkspace. Ďalej sme vytvorili 6 kresieb priamo v aplikácii, ktorú sme vytvorili, aby sme znázornili kroky prístupu topológia – tvar – metrika (obrázky 3.3 – 3.6).

Na túto prácu možno v budúcnosti nadviazať tak, že by sa zautomatizovali jednotlivé kroky prístupu topológia – tvar – metrika a uplatnili optimalizačné kroky, prípadne pridaním iných prístupov na kreslenie grafov a mobilnej verzie vzdelávacej hry.

Literatúra a internetové pramene

[CHi00] CHI, H. 2000. *A Taxonomy of Visualization Techniques using the Data State Reference Model.* [online] Dátum platnosti 23. 3. 2018.

[DiGi17] DI GIACOMO, E. et al. 2017. GRAPH DRAWING. Chapter 55 in [Toth17]. [online] https://www.csun.edu/~ctoth/Handbook/chap55.pdf. Dátum platnosti 15.3.2018.

[DETT99] Di Battista, G., Eades, P., Tamassia, R., Tollis, I. 1999 *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice Hall, Upper Saddle River, 1999.

[Fri00] FRISHMAN, Y. 2000. *Graph Drawing Algorithms in Information Visualization.* Haifa: Technion 2000. [online] http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2009/PHD/PHD-2009-02.pdf . Dátum platnosti 15. 3. 2018.

[Ger99] GERMANO, T. 1999. *Graph Drawing*
[online] http://davis.wpi.edu/~matt/courses/graphs Dátum platnosti 23. 3. 2018.

[Katr17] KATRENIAKOVA, J. 2017. *Vizualizácia a navigácia s použitím techník kreslenia grafov.* Prednáška z predmetu WEGA, 2017.
[online]http://www.sccg.sk/ferko/KreslenieGrafovDrKatreniakova.pdf. Dátum platnosti 15.3.2018.

[KvPo08] KVASNIČKA, V. - POSPÍCHAL, J. 2008. *Algebra a diskrétna matematika*. Kapitola 10. STU Press, Bratislava, 2008. [online] http://www2.fiit.stuba.sk/~kvasnicka/DiskretnaMatematika/Chapter_10/kapitola10.pdf. Dátum platnosti 15. 3. 2018.

[Nis04] NISHIZEKI, T. dal. 2004. *Planar Graph Drawing.* World Sientific, 2004.

[Tam13] TAMASSIA, R. *Ed.* 2013. *Handbook of Graph Drawing and Visualization.* CRC Press 2013. [online] https://cs.brown.edu/~rt/gdhandbook/. Dátum platnosti 15. 3. 2018.

[Toth17] TOTH, C. 2017. [online] *Handbook of Discrete and Computational Geometry.* Third Edition. https://www.csun.edu/~ctoth/Handbook/HDCG3.html. Datum platnosti 15.3.2018.