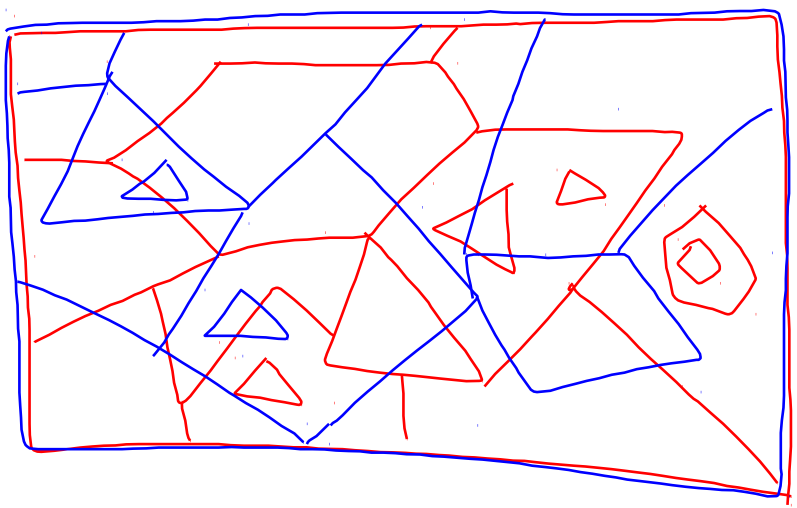
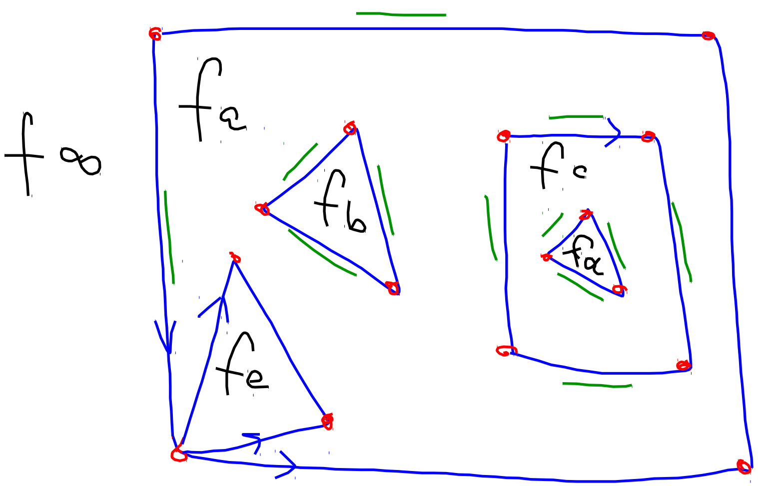
Překryvy map



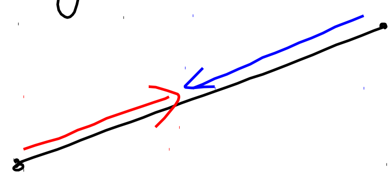


Rovinné podrozdělení
(bude pro nás mapou)
skládá se

- z vrcholů
- z hran, které spojují dvojice vrcholů a navzájem se neprotínají
- z oblastí (ty jsou ohraničeny hranami

Popis uložíme pomocí dvojitě souvislého seznamu
(doubly connected edge list)

3 tabulky
- pro vrcholy
- pro hrany (hrana = orientovaná úsečka



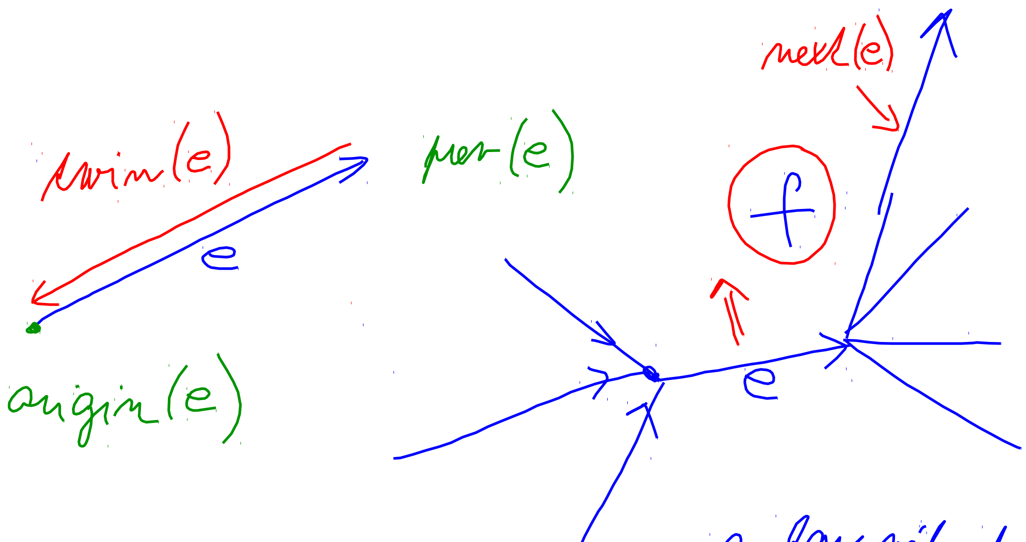
- pro oblasti

TABULKA PRO VRCHOLY

jméno vrcholu, souřadnice vrcholu, jedna hrana, která z vrcholu vychází
ukazatel (pointer) na tuto hranu

TABULKA PRO HRANY

jméno hrany, ukazatel na počáteční uzel (origin), ukazatel na dvojče (twin), ukazatel na přilehlou oblast, ukazatel na následníka, ukazatel na předchůdce



přilehlá oblast hrany e je oblast sousední zleva od e

následník hrany e

next(e) = hrana vycházející z koncového bodu hrany e se stejnou přilehlou oblastí

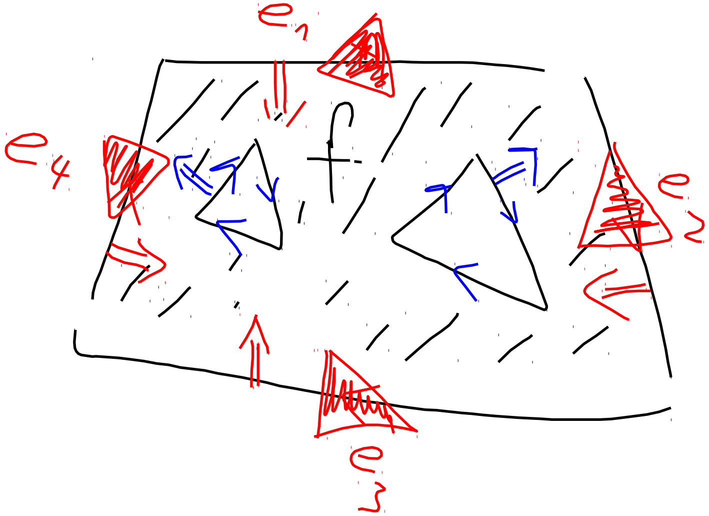
Předchůdce hrany e je hrana přicházející do poč. bodu hrany e, která má stejnou přilehlou oblast.

TABULKA PRO OBLASTI (faces)

jméno oblasti, ukazatel na jednu hranu z vnějšího cyklu

ukazatel na jednu hranu z každého vnitřního cyklu

Každá omezená oblast má vnější hranici



$e_1\, e_2\, e_3\, e_4\, e_5$ je cyklus, jestliže

e_{i+1} je následník e_i

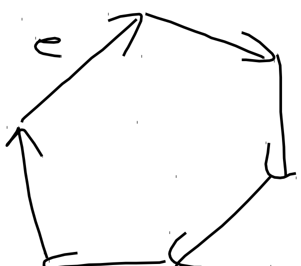
a $e_5 = e_1$.

Vnější cyklus

Vnitřní cyklus

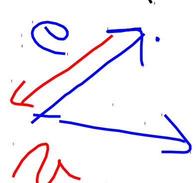
Jde o minimální popis, z kterého dostaneme všechny potřebné údaje.

– k dané oblasti f najít všechny hrany mezi jejího cyklu
máme e, najdem next(e), e = next(e), a děláme
tak dlouho až se dostaneme do původního e.



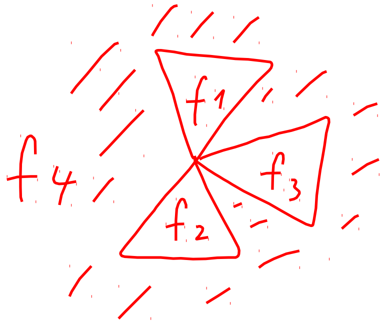
Časová náročnost této operace je
$O(\text{počet hran cyklu})$

– najít všechny hrany vycházející z vrcholu v a vypsat je ve směru hod. ručiček



e —> twin(e) => následník(twin(e)) atd až se dostaneme do e.

- najít všechny oblasti, které mají na hranici vrchol v a vypsat je ve směru hod. ručiček



f1 f4 f3 f4 f2 f4

Doubly connected edge list

Tabulky

vrcholy $\rightleftarrows$ hrany $\rightleftarrows$ oblasti

Algoritmus pro překryv map

Rovinné podrozdělení $\mathcal{S}_1$ .... červená mapa DCEL $\mathcal{D}(\mathcal{S}_1)$

Rovinné podrozdělení $\mathcal{S}_2$ .... modrá mapa $\mathcal{D}(\mathcal{S}_2)$

Překryv (overlay) $\mathcal{O}(\mathcal{S}_1, \mathcal{S}_2)$ ..... černá mapa $\mathcal{D}$

ALGORITMUS

- vstup $\mathcal{D}(\mathcal{S}_1)$, $\mathcal{D}(\mathcal{S}_2)$
- výstup $\mathcal{D}$ + ke každé nové oblasti f z $\mathcal{D}$ dáme její původní oblasti z červené i modré mapy.

ALGORITMUS

začne tím, že do $\mathcal{D}$ zahrneme $\mathcal{D}(\varphi_1)$ a $\mathcal{D}(\varphi_2)$.
Tento samozřejmě špatný popis pro přehyb budeme postupně upravovat. Úpravy probíhají ve dvou krocích:

(1) Opravíme a doplníme značky týkající se uzlů a hran. To provedeme metodou zametací přímky, kde při zpracování událostí provedeme více úkonů než v jedn. algoritmu pro průsečíky úseček.

(2) Stanovíme oblasti a jejich převod v černé a modré mapě. K tomu už zametací přímku nepotřebujeme.

1. část algoritmu

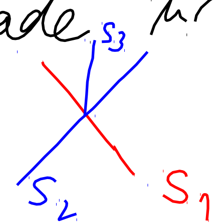
- z hran modrých i červených vytvoříme úsečky.
- uděláme frontu událostí a vyvážený binární strom jako v algoritmu pro průsečíky úseček.
- část algoritmu nazvaná Handle event point teď bude obsahovat více

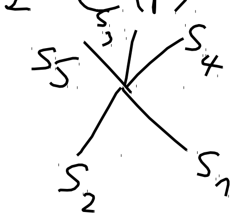
Mějme událost p

$L(p), U(p), C(p)$.

Jestliže všechny úsečky z $L(p) \cup U(p) \cup C(p)$ mají stejnou barvu, neděláme nic.

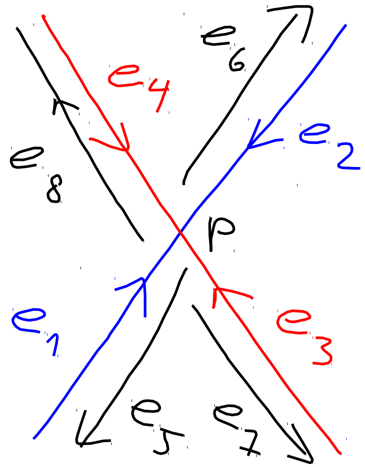
V opačném případě úsečky z $C(p)$ rozdělíme na 2.





a zařadíme do $L(p)$ a $U(p)$

Teď k vrcholům přiřadíme hrany (některé staré, některé nové)
a uděláme změny v tabulce pro vrcholy a pro hrany.



řádek pro p

p, souřadnice, e_5

změny v řádcích pro e_1, e_2, e_3, e_4

e_1, počátek stejný, dojde e_5, následník je e_8
předchůdce stejný, s oblastmi nic neděláme

U(p)

před p $\{e_4, e_8\} < \{e_2, e_6\}$

pod p $\{e_1, e_5\} < \{e_3, e_7\}$

(11)

Nový řádek pro e_5

$$\boxed{\begin{array}{l} e_5,\ \text{počátek } p,\ \text{dvojče } e_1,\ \text{následník} = \text{následník } e_2, \\ \text{předchůdce} = e_3,\ \text{oblast ponecháme} \end{array}}$$

Tímto postupem dostaneme v D vždy informace o vlastních vrcholů a hran pro průhyb.

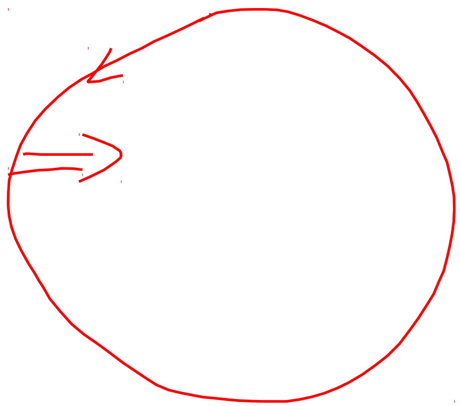
2. ČÁST O OBLASTECH

Každá oblast je určena vnějším cyklem a případně vnitřními cykly.
Cykly – vždy – umíme najít v naší tabulce D pomocí funkce následníka.

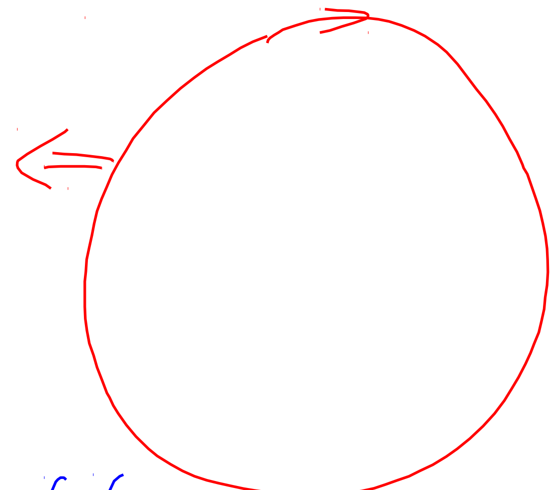
12

Najdeme tedy všechny cykly.

Vnější



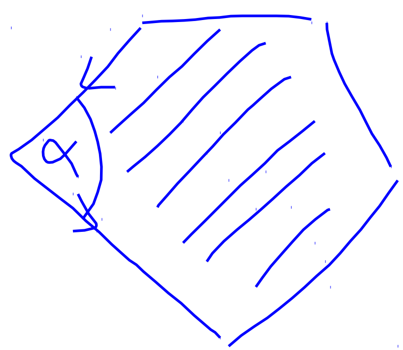
Vnitřní

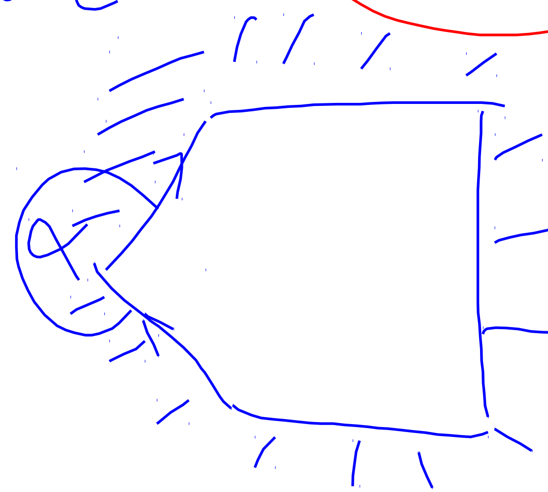


Vnitřní a vnější cesty se liší takto: nezměníme vchod v cyklu nejvíce vlevo:

vnější cyklus

$\alpha < 180^\circ$



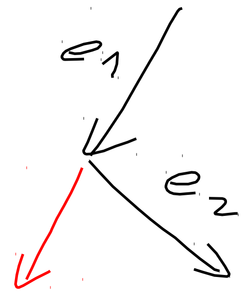


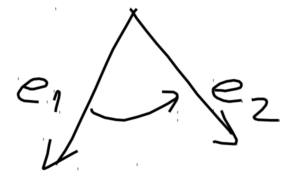
vnitřní cyklus

$\alpha > 180^\circ$

Vyprávěl

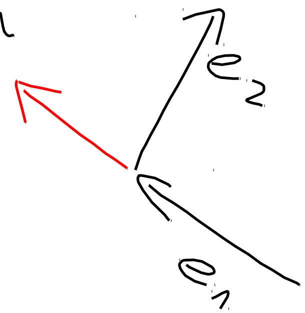
vnější

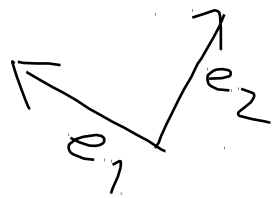




$$\det\begin{pmatrix} e_{1x} & e_{2x} \\ e_{1y} & e_{2y} \end{pmatrix} > 0$$

vnitřní



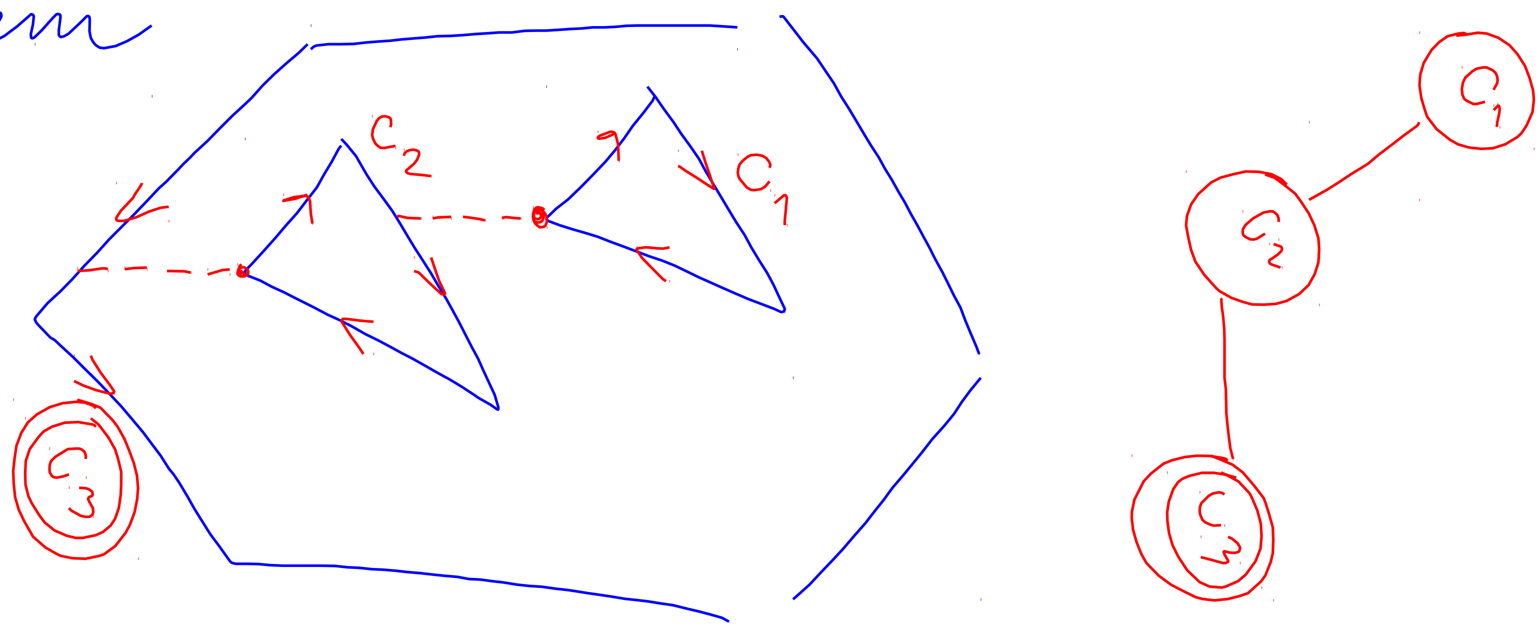


$$\det\begin{pmatrix} e_{1x} & e_{2x} \\ e_{1y} & e_{2y} \end{pmatrix} < 0$$

Známe vnější cykly a vnitřní cykly. Neomezená oblast nechť je omezena příslušným cyklem C_∞.

Každý vnější cyklus definuje právě jednu oblast.
Abychom k oblastem přiřadili i vnitřní cykly sestrojíme následující abstraktní graf
(ve smyslu teorie grafů)
Uzly v něm jsou všechny cykly.
Dva cykly jsou spojeny hranou, jestliže nastane tato situace.

Vezmeme minimální cyklus, v něm vrchol nejvíce vlevo a jdeme odtud doleva tak dlouho až narazíme na hranu ze stejné předchozí oblasti. Cyklus obsahující tuto hranu spojíme grafovou hranou s původním cyklem

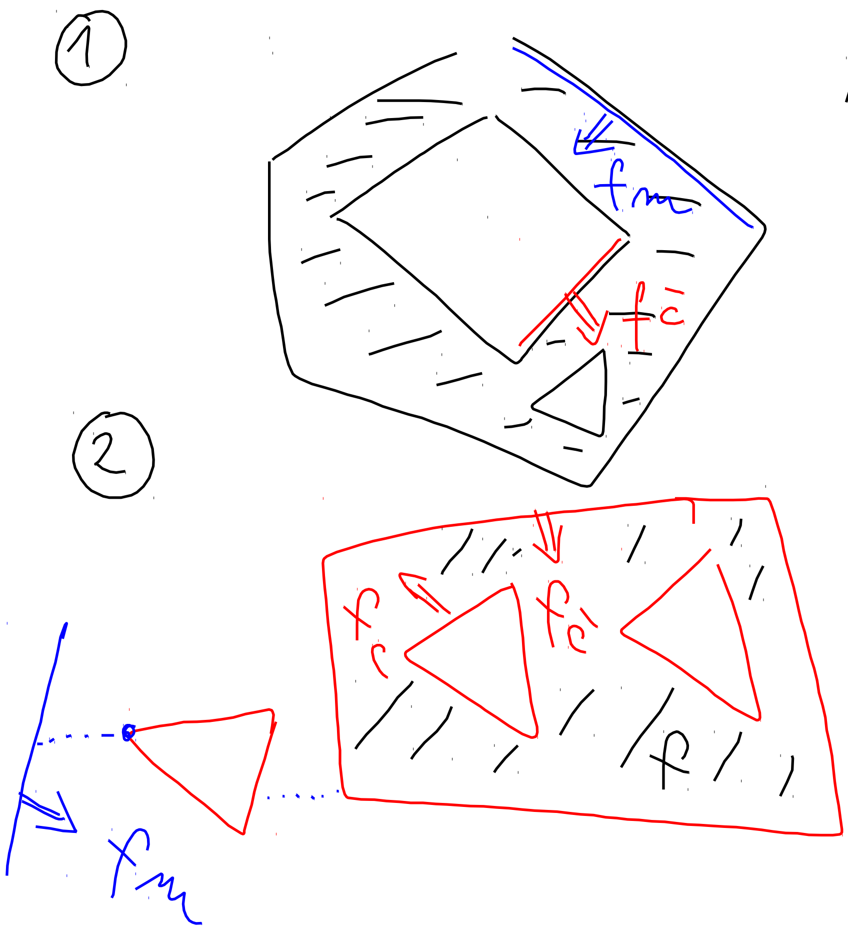


V každé komponentě souvislosti tohoto grafu leží právě jeden mezní cyklus. Ten odpovídá oblasti přehybu. Tedy mezi komponentami souvislosti grafu a oblastmi je vzájemně jednoznačná korespondence, která umožňuje definovat tabulky týkající se oblasti

f, mezní cyklus → vezmu jednu hranu
vnitřní cyklus → vezmu jednu hranu

e hrana → najdeme její cyklus → přilehlá oblast

Ke každé nové oblasti f překryvu chceme původní oblasti v červené a modré mapě.

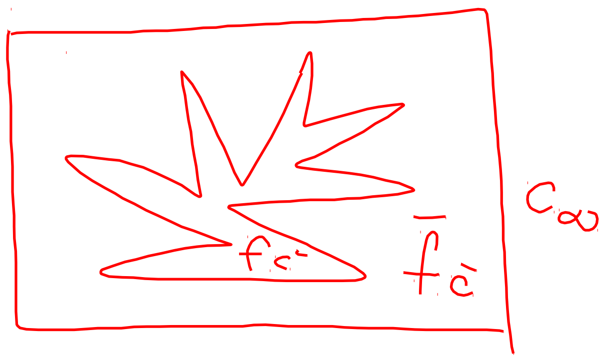


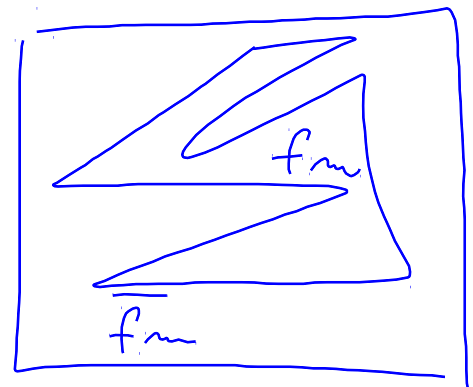
V hraničních cyklech jsou hrany obou barev – jejich přilehlé oblasti a původních map jsou ty správné, které hledáme.

$f \ldots f_č$ a f_m

Zvláštní případy

Průnik mnohoúhelníků





Jak najdeme $f_{\bar c} \cap f_m$. To jsou ty oblasti překryvu sítě dvou map, které leží v f_c i v f_m.

Obdobně sjednocení nebo rozdíl

Časová náročnost je $O((n+m+k)\log(n+m))$