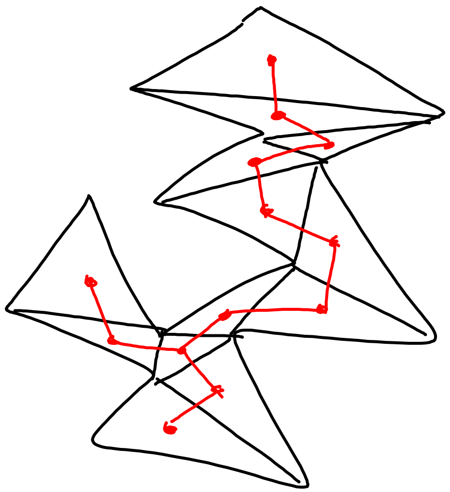
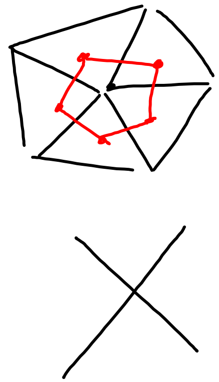
1

# Triangulace jednoduchých mnohoúhelníků

Triangulace má vrcholy ve vrcholech mnohoúhelníka a duální graf k ní je vždy strom





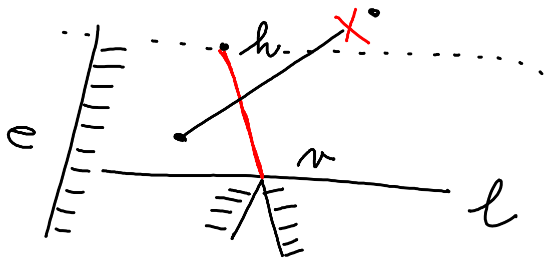
2

## Rozdělení mnohoúhelníka na monotónní části

Rozdělení na monotónní části zabere čas $O(n \log n)$,
kde $n$ je počet vrcholů

Algoritmus je korektní .... ~~A~~ k důkazu algoritmu pro různé typy vrcholů musíme ukázat, že přidávané diagonály se navzájem neprotínají

Typ split vrcholu



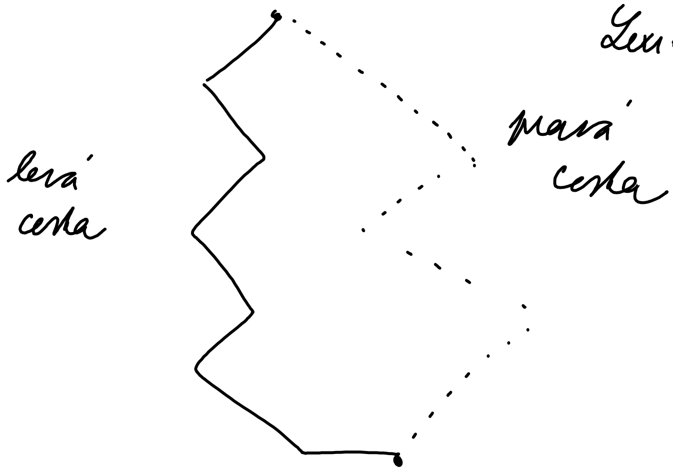
(3)

# Triangulace monotónních mnohoúhelníků

- opět metodou zametací přímky
- časová náročnost je $O(n)$, kde $n$ je počet vrcholů

Monotónní mnohoúhelník

-

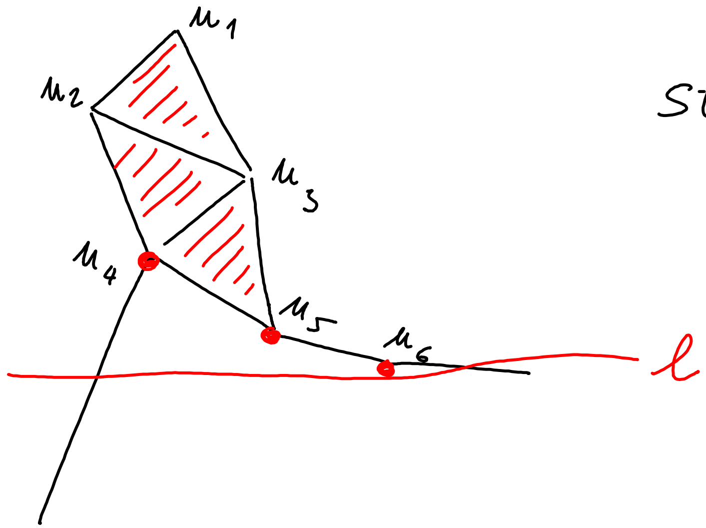



Lexikograf. uspořádání všech vrcholů je dáno pořadím vrcholů v levé a pravé cestě (k tomu stačí čas $O(n)$ !)

4

Procházíme vrcholy shora dolů a děláme △ směrem nahoru, kdykoliv je to možné.

Zásobník (stack) ... pro vrcholy na zametací přímkou té části mnohoúhelníka, která již/ještě nebyla triangulována




stack je $\begin{pmatrix} u_6 \\ u_5 \\ u_4 \end{pmatrix}$

Na začátku bude zásobníkem dvojice vrcholů $\begin{pmatrix} u_2 \\ u_1 \end{pmatrix}$

Všechny vrcholy stacku leží buď v levé nebo v pravé cestě zbylého mnohoúhelníka

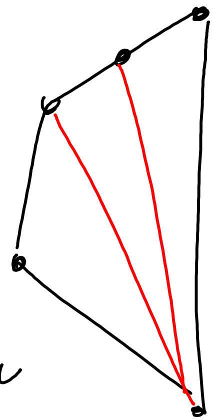
Demonstrace algoritmu na obr. 20 a 21.

5

Konec algoritmu

[illegible] poslední vrchol ve stacku

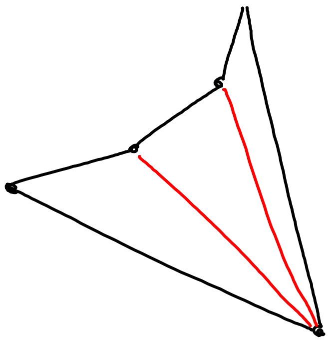
$\varphi$



1. vrchol ve stacku

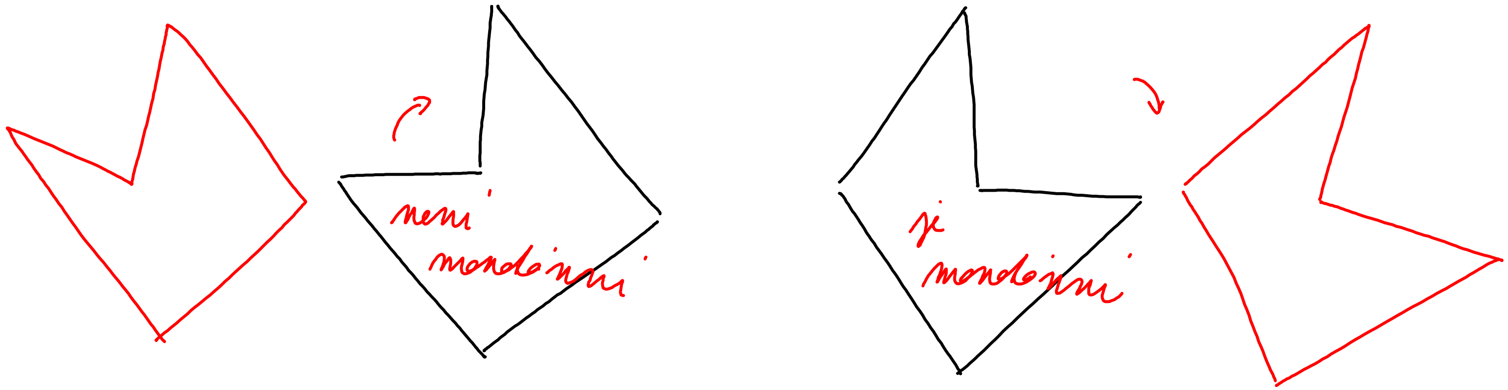
$u_n$ poslední vrchol mnohoúhelníka

Správně to má vypadat takto:



6

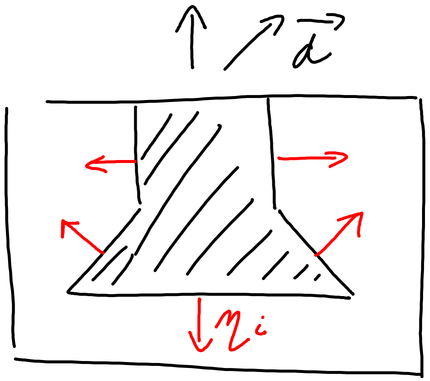
Který z těchto 2 mnohoúhelníků je monotónní?

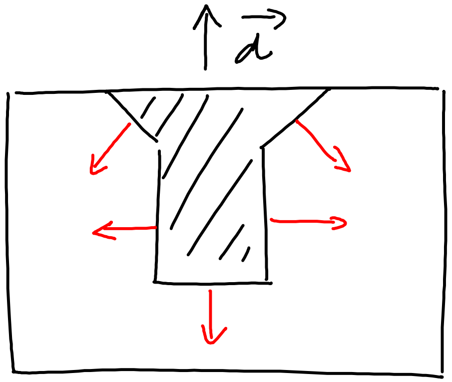


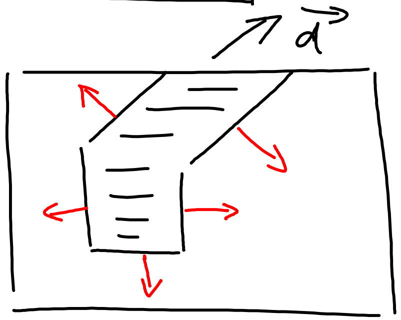

Podle geometrické definice oba, ale uvažme si lexikografické uspořádání

7

# Průnik polorovin a úloha lineárního programování v rovině




$\vec d$ je směr vytahování

$\vec n_i$ je normála k i-té straně

Kritérium pro vytažení je, že

$$\measuredangle(\vec d, \vec n_i) \geq 90^\circ$$

pro všechna i

8

$\vec{n}_i = (a_i, b_i, c_i)$ je dán pevně hranou plochy

$\vec{d} = (x, y, 1)$ hledáme $x, y$ tak, aby

$\sphericalangle(\vec{d}, \vec{n}_i) \geq 90$ pro všechna $i$

Skalární součin. $\langle \vec{d}, \vec{n}_i \rangle \leq 0$

$a_i x + b_i y + c_i 1 \leq 0$

$a_i x + b_i y \leq -c_i$

Dostáváme soustavu lineárních nerovnic v neznámých $x$ a $y$.
V dané úloze nám stačí jediné řešení.

Množina $\{(x,y) \in \mathbb{R}^2;\ a_i x + b_i y \leq -c_i\}$ je polorovina $h_i$.

<u>Geometricky:</u> Hledáme nějaký bo[d] z průniku polorovin.

9

Průnik n polorovin je konvexní množina

Možný algoritmus:

- najdeme $C_1$ a $C_2$ dva konvexní mnohoúhelníky, kde $C_1$ je průnik prvních $\frac{n}{2}$ polorovin a $C_2$ je průnik zbývajících $\frac{n}{2}$ polorovin
- pak najdeme průnik $C_1 \cap C_2$

Průnik 2 obecně nekonvexních mnohoúhelníků umíme (překryv map). Trvá obecně čas $O(n \log n)$

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n \log n)$$

tato rekurentní formule vede k času $T(n) = O(n \log^2 n)$

10

Budeme-li mít algoritmus na případy $C_1$ a $C_2$ časové náročnosti $O(n)$,

pak celková časová náročnost algoritmu se bude řídit rekurentní formulí

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

a to vede k

$$T(n) = O(n \log n) \qquad n = 2^k \qquad n \log_2 n = 2^k \cdot k$$

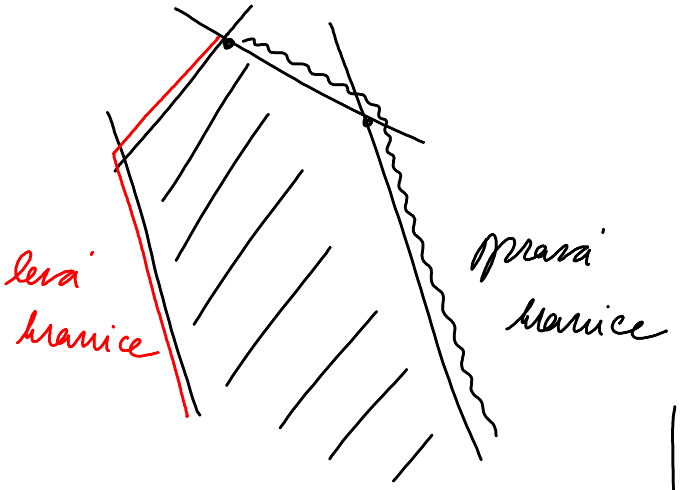
11

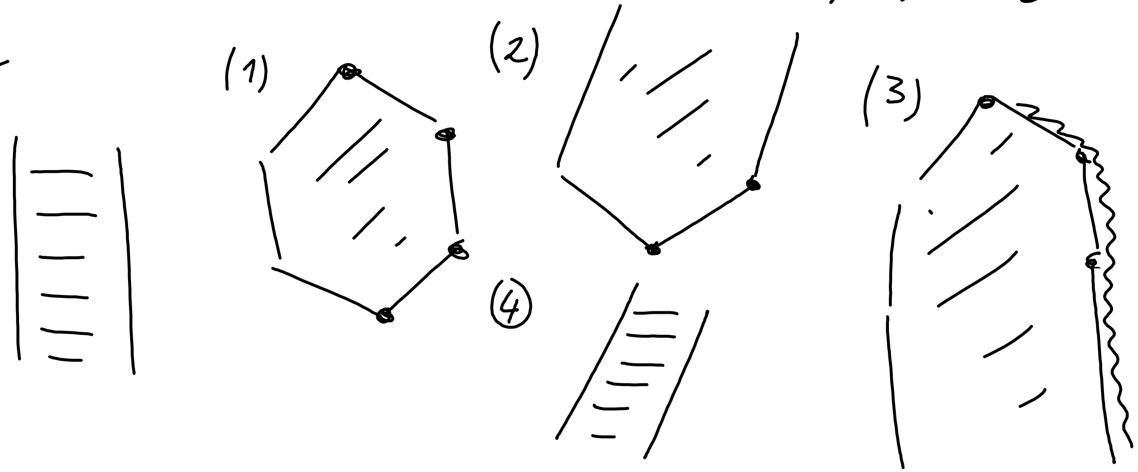
V našem případě jsou $C_1$ a $C_2$ průniky polorovin, tj. konvexní množiny, které nemusí být omezené a tedy nejsou obecně mnohoúhelníky

12

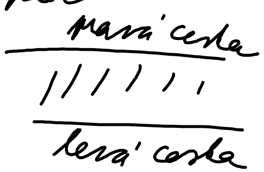
Průniky polorovin $C_1$ a $C_2$ mají levou a pravou hranici

Každá z nich je určena posloupností vrcholů nebo polopřímkou a vrcholy nebo vrcholy a polopřímkou nebo přímkou




Spec. případ



(13)

Algoritmus na průnik $C_1 \cap C_2$ je metodou zametací přímky. V ní nepotřebujeme strom ani frontu. Pořadí vrcholů neboli hraničních úseček, podepírek, průniků je dáno popisem levých a pravých hranic $C_1$ a $C_2$. Chceme stejným způsobem popsat hranici $C_1 \cap C_2$

(levá cesta = levá hranice)

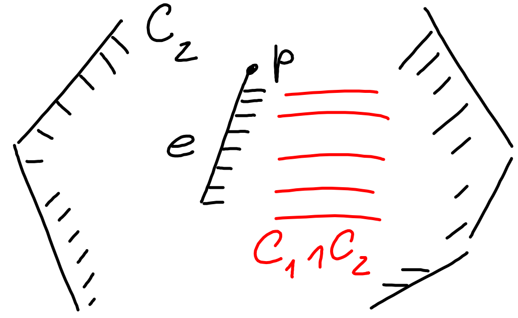
Nechť p je horní vrchol hrany (podepírky) e z levé cesty $C_1$.
Mohou nastat tyto možnosti

(1) e neprotíná hranici $C_2$
- p ∩ e leží uvnitř $C_2$
- p ∩ e leží vně $C_2$

(2) e protíná hranici $C_2$
- e protíná levou cestu
- e protíná pravou cestu

14

1a e leží mezi levou a pravou cestou $C_2$

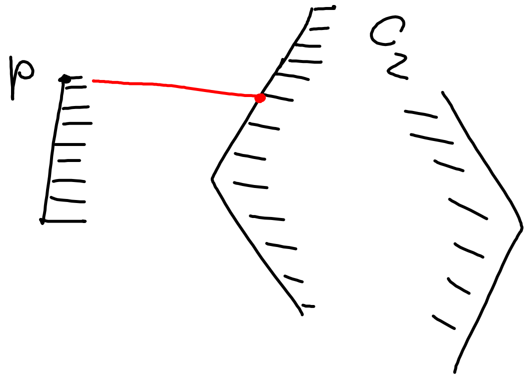



e přidáme do levé cesty

$C_1 \cap C_2$

Tuto cestu známe už shora do p

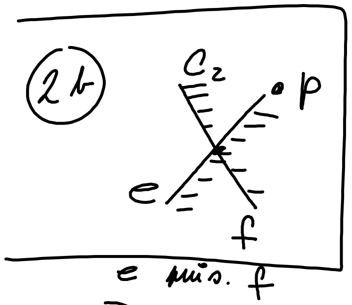
1b

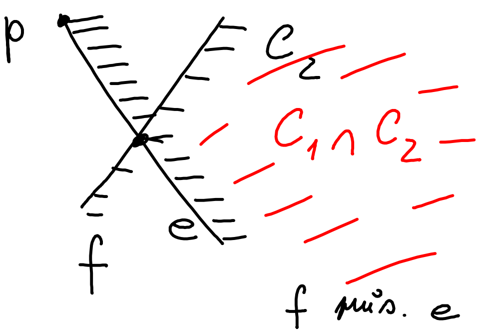



Nic se neděje

15

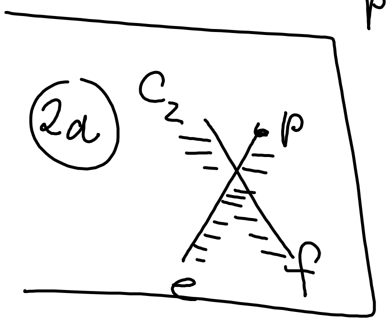
2a e protíná levou cestu $C_2$

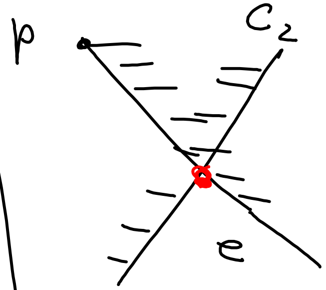



Za následující hranu f z levé cesty $C_1 \cap C_2$ (f z levé cesty $C_2$) přidáme hranu e (začínající v průsečíku $f \cap e$)

2c e protíná pravou cestu $C_2$




~~Obrazy~~ Do levé cesty $C_1 \cap C_2$ přidáme e a bod $e \cap f$ jako koncový vrchol. Do pravé cesty dáme f a $f \cap e$.