3. Vorlesung am 22.04.05

Wiederholung

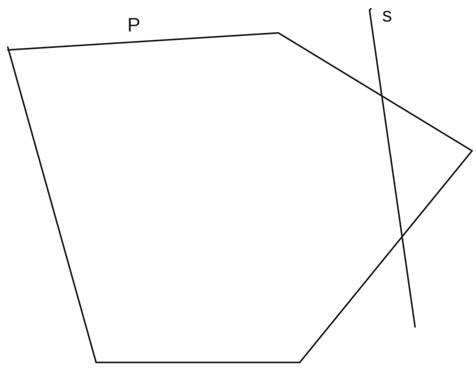
Die Betrachtung von konvexen Polygonen bringt uns zu zwei Fragestellungen:

- Wie schneidet sich eine Gerade g mit dem Rand eines konvexen Polygons P?
- Liegt in Punkt p in dem konvexen Polygon P?

Ist man in der Situation, dass man für ein bestimmtes Polgon mehrerer solcher Fragestellungen beantworten will, so ist eine Vorverabeitung des Polygons hilfreich. Denn dann kann man diese in linearer Zeit lösen. Die geignete Datenstrucktuktur hierfür ist die BHD aus der 2. Vorlesung.

Korollar

Gegeben seien Polygon P in BHD und eine Strecke s. Dann kann $P \cap s$ in $O(\log n)$ Zeit berechent werden. Dies tut man, indem man s zu einer Gerade fortsetzt und den deren Schnittpunkt mir dem Polygon sucht.



Beispiel für Schnitt von s mit P

Bezeichnung

Meistens meint man mit "konvexem Polygon" die Punkte auf den Kanten des Polygons vereingit mit der Menge der Punkte im Inneren. (Also die konvexe Hülle des Polygons)

Schnitt konvexer Polygone

Seien P und Q konvexe Polygone, mit n- bzw. m-Ecken, dann ist $P \cap Q$ auch ein konvexes Polygon mit maximal $n + m$ Ecken. (Es können sogar 0 Ecken seien, dies ist ein konvexes 0-Eck)

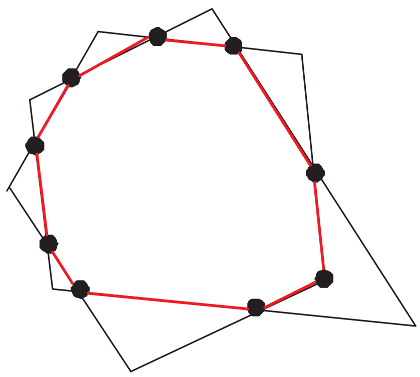
Die folgenden Fragen sind von Interesse bei dem Arbeiten mit konvexen Polygonen.

Entscheidungsproblem

Ist $P \cap Q = \emptyset$.
Dies ist in $O(\log n + m)$ Zeit lösbar, wenn P und Q in BHD gegeben sind.

Berechnungsproblem

Berechne $P \cap Q$.
Dies ist nicht in besserer Zeit als $\Omega(n + m)$ lösbar, da die Ausgabe alleine schon diese Größe haben kann.



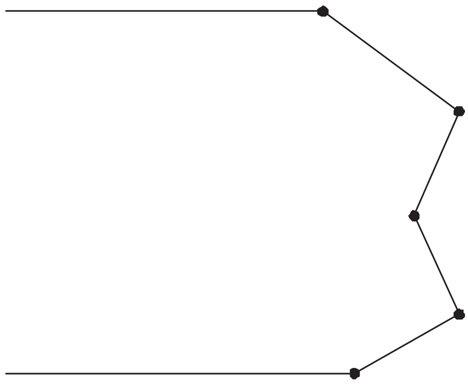
Beispielsweise für Schnitt von Q (4-Eck) mit P (5-Eck)

Satz 2

Seien mit P und Q konvexe Polygone in BHD mit n bzw. m Ecken gegeben. Dann läßt sich in $O(\log(n+m))$ Zeit entscheiden, ob $P \cap Q = \emptyset$ ist.

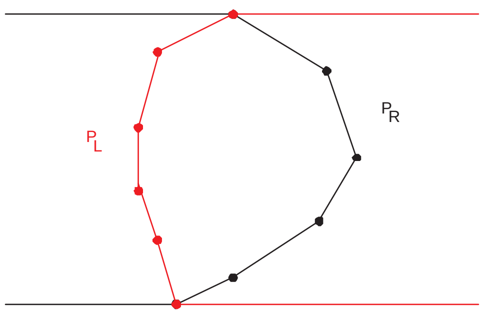
Definition

Ein Polygonzug $p_0, \ldots, p_k$ heißt genau dann *y-Monoton*, wenn $y(p_i) \leq y(p_{i+1}) \quad i = 0, \ldots k-1$, wobei $y(p_i)$ die Y-Koordinatenfunktion ist.



Beispiel für einen y-monotonen Polygonzug mit offenem Rand

Der Polygonzug mit den angestzten Strahlen umschließt ein unbeschränktes Gebiet. Jedes konvexes Polygon läßt sich als Schnitt zweier solcher Gebiete darstellen. Wir bezeichnen diese mit P_r und P_l, da P_r nach rechts und P_l nach links beschränkt ist.



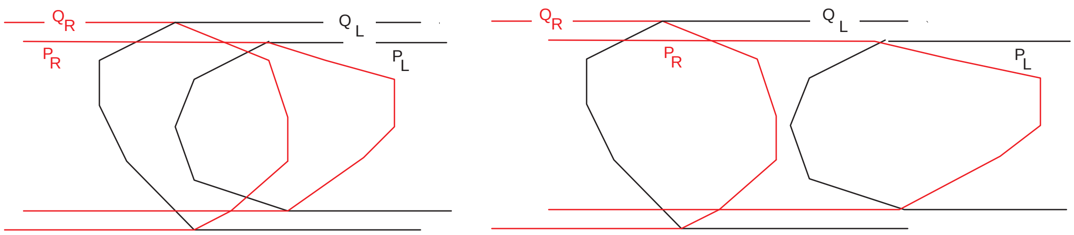
Aufteilung eines Polygones in halboffene Polygone

Lemma 1

Seien P,Q, P_l, Q_l, P_r und Q_r analog definiert. Dann ist:
$(P \cap Q \neq \emptyset) \Longleftrightarrow (P_l \cap Q_l \neq \emptyset$ und $P_l \cap Q_l \neq \emptyset)$.

Beweis von Lemma 1

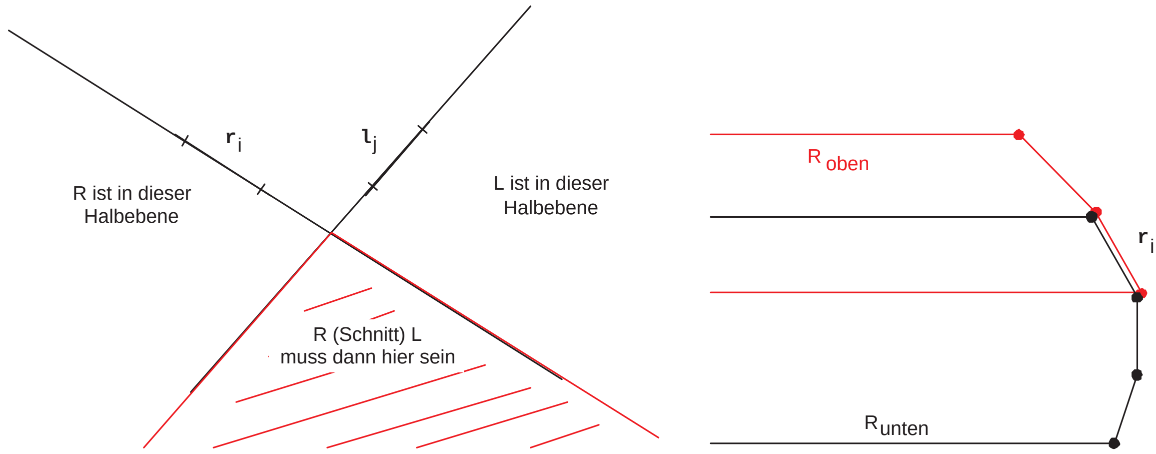
Übungsaufgabe.



Erklärende Zeichnungen zur Beweisidee

Algorithmus

Ein Algorithmus zum Entscheidungsproblem für halboffen konvexe Polygone (eins nach links L und das andere nach rechts R geöffnet) reicht offensichtlich aus. Der Algorithmus benötigt die Kanten von L und R sortiert. Dabei sind $r_1, \ldots, r_m$ sind die Kanten von R und $l_1, \ldots, l_n$ die von L. Diese kann man in logarithmischer Zeit aus dem BHD-Baum erhalten.
Nun bestimmt man den Schnitt durch folgende Art der Binärsuche:
Man betrachte die mitteleren Kanten r_i und l_j mit $i = \lceil \frac{m}{2} \rceil$ und $j = \lceil \frac{n}{2} \rceil$:



1. Beobachtung, Erklärung der Aufteilung

Teile R und L auf in R_{oben}, R_{unten}, L_{oben} und L_{unten} wie in obiger Abbildung.

Es können nun folgende Fälle auftreten:

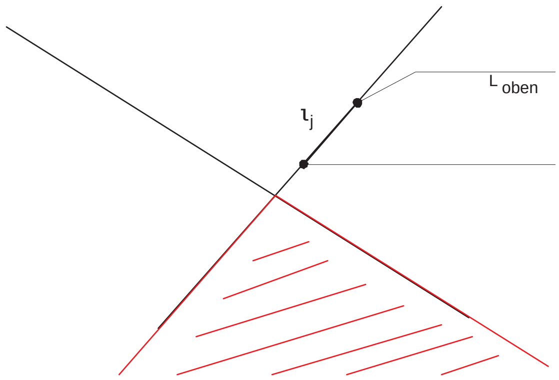
a) Dann ist $R \cap L = R \cap L_{\text{unten}}$

b) Dann ist $R \cap L = R_{\text{unten}} \cap L$

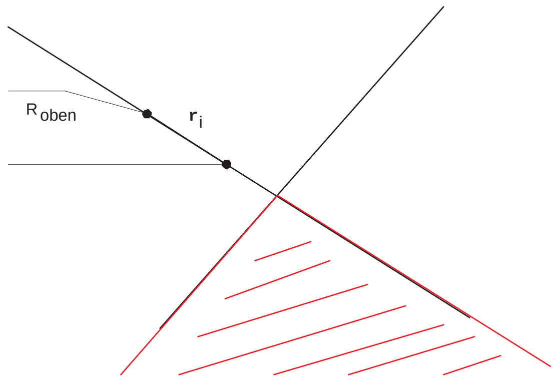
c) Beide r_i und r_j liegen unterhalb des Schnittpunktes. Wenn der obere Endpunkt von r_i höher als der von l_j liegt, dann gilt: $R \cap L \neq \emptyset \Longleftrightarrow R \cap L_{\text{oben}} \neq \emptyset$

d) $R \cap L \neq \emptyset \Longleftrightarrow R_{\text{oben}} \cap L \neq \emptyset$

a,b)

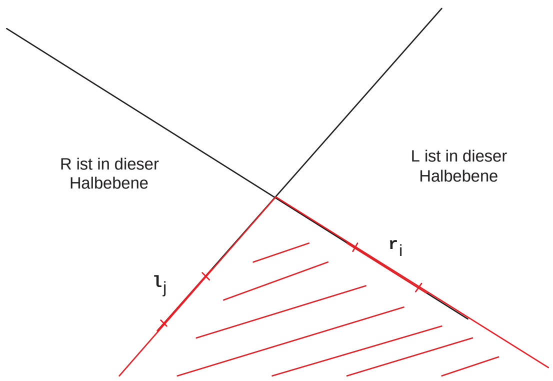


l_j ist über dem Schnittpunkt,

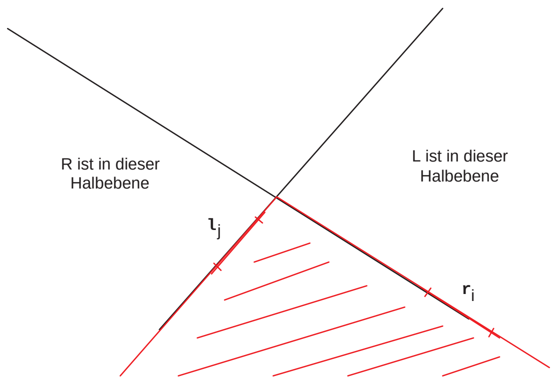


r_i ist über dem Schnittpunkt

c,d)



Beide Abschnitte r_i, l_j liegen unterhalb



l_j liegt höher als r_i

Also durch Analyse der Lage von r_i und l_j läßt sich mindestens eines der beiden Polygone "halbieren"(bezüglich der Eckenzahl). Nun wird dies iteriert. Nach $O(\log n + \log m)$ Schritten hat mindestens eines die konstante Größe $c \leq 5$. Diese c Kanten testen wir auf Schnitt mit dem anderen Polygon in $O(\log m)$ bzw. $O(\log n)$. (Siehe auch Korollar zu Satz 1). Man erhält als Gesamtlaufzeit $O(\log n + \log m) = O(\log(n + m))$.

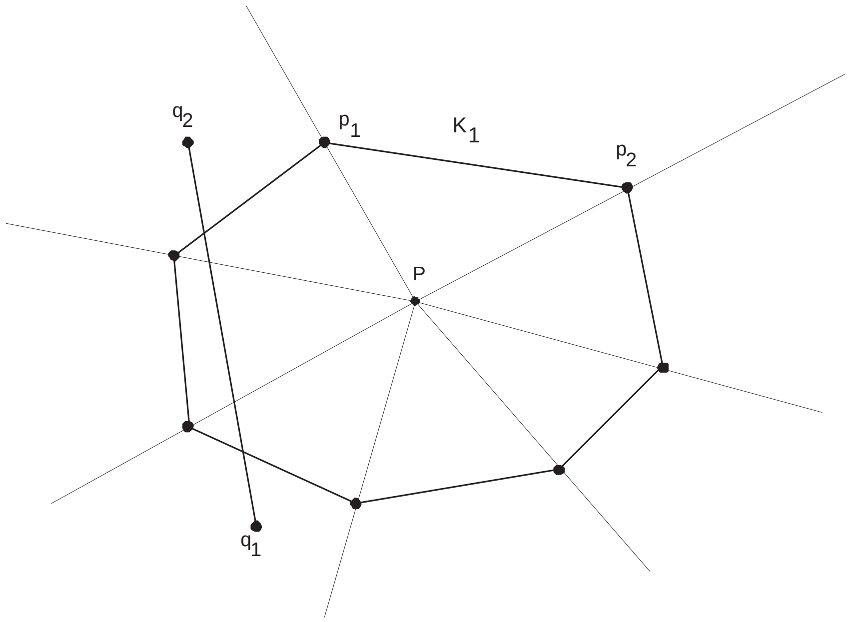
Satz 3

Seien P und Q konvexe Polygone mit m bzw. n Ecken, gegeben durch eine geordente Folge der Ecken (z.B. im mathematisch positiven Sinn)

Algorithmus

P habe die Ecken $p_1, \ldots, p_n$ und Q habe die Ecken $q_1, \ldots, q_n$.

(1) Bestimme einen Punkt im Inneren von P. (z.B.seien a, b und c drei bel. Ecken: deren Schwerpunkt $p := \frac{1}{3}(a + b + c)$) Wir ziehen Strecken von p durch alle Ecken p_i von P. Wir erhalten n Kegel $K_1, \ldots, K_n$. Für eine bel. Ecke q_j von Q sei $K(q_j)$ der Kegel, der q_j enthällt.



Polygon P mit konstruierten Kegeln

(2) Berechne $K(q_1)$ $(= K_r)$

(3) Berechne alle Schnitte von $\overline{q_1 q_2}$ mit P durch Durchlaufen von $K_r, K_{r+1}, \ldots, K_l = K(q_2)$ und mit den entsprechenden Kanten von P.

(4) von K_l ausgehend: analog mit $\overline{q_2 q_3}$ usw.

Laufzeit

(1) $O(n)$

(2) $O(n)$ (geht sogar in $O(\log n)$)

(3) $O(1 + k_1)$ wobei k_1 die Anzahl der geschnitten Strahlen

(4) $O(\Sigma_{i=1}^{m}(1 + k_i) = O(m)$ da jeder Strahl Q in höchstens 2 Punkten schneidet.