

4.3.10. Plane Sweep Alg. zum Mischen von zwei Konturen A und B.

→ 4.3.10.1. Problemstellung:

Eingabe: Zwei Konturen A, B in den Ebenen durch zwei Mengen von Liniensegmenten.

Ausgabe: Menge von Segmenten, die die Kanten von A ∪ B aufzweigen.

→ 4.3.10.2. Definition: Ein Segment heißt sichtbar, wenn es zur Ausgabe gehört, d.h. auf Rand von A ∪ B liegt.

→ 4.3.10.3. Idee:

Wir modifizieren den Plane Sweep Alg. zum Linienschritt.

Y-Struktur: Folge der von SL geschnittenen Segmenten (von unten nach oben sortiert). Wir speichern für jedes Segment in der Y-Struktur, ob es zur Zeit sichtbar oder unsichtbar ist.

Außerdem speichern wir für jedes Paar (s1, s2) von in Y benachbarten Segmenten, ob und von wem das Gebiet zwischen s1 und s2 zur Zeit überdeckt wird.

$$cover(s_1, s_2) = \begin{cases} \emptyset, & \text{nicht überdeckt} \\ \{A\}, & \text{von A} \\ \{B\}, & \text{von B} \\ \{A \cup B\}, & \text{von beiden.} \end{cases}$$

→ 4.3.10.4. Aktionen:

1. Linker Endpunkt p ∈ A (B analog) d.h. linke Ecke von zwei Segmenten. s1, s2 ∈ A.

s' = Y.succ(s1), s'' = Y.pred(s2)

Falls cover(s', s'') = ∅ oder cover(s', s'') = A dann

markiere s1, s2 als sichtbar

sonst unsichtbar

Falls A ∈ cover(s', s'') (d.h. bei p beginnt ein Loch) dann:

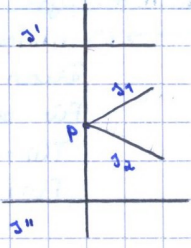
cover(s1, s2) ← cover(s', s'') \ {A}

sonst cover(s1, s2) ← cover(s', s'') ∪ {A}

cover(s', s1) ← cover(s', s'')

cover(s2, s'') ← cover(s', s'')

Y ← Y ∪ {s1, s2}



2. Rechter Endpunkt p ∈ A (B analog)

Falls s1 bzw. s2 sichtbar, dann Ausgabe der betreffenden Segmente.

Y ← Y \ {s1, s2}

cover(s', s'') ← cover(s', s1) (= cover(s2, s''))



3. Schnittpunkt p = s1 ∩ s2

oBd A s1 ∈ A, s2 ∈ B

Falls s1 (s2) sichtbar ⇒ Ausgabe bis p (spalten!)

Y ← Y \ {s1, s2}, Y ← Y ∪ {s1', s2'} in umgekehrter Reihenfolge mit negativer Sichtbarkeitsinformation

cover(s1', s2') ← cover(s1', s1)

cover(s1', s'') ← cover(s2, s'')

Falls A ∈ cover(s1, s2) dann

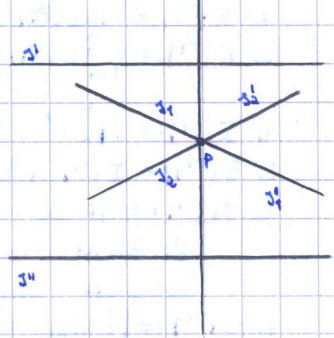
cover(s2', s1') ← cover(s1, s2) \ {A}

sonst cover(s2', s1') ← cover(s1, s2) ∪ {A}

Falls B ∈ cover(s1, s2) dann

cover(s2', s2') ← cover(s1, s2) \ {B}

sonst cover(s2', s2') ← cover(s1, s2) ∪ {B}



4. Durchgang p ∈ A (B analog)

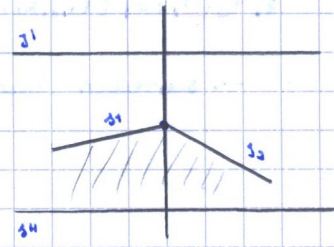
s1 endet und s2 beginnt

Y ← Y \ {s1}, Y ← Y ∪ {s2}

Falls s1 sichtbar ⇒ Ausgabe

sichtbarkeit(s2) ← sichtbarkeit(s1)

cover(s1', s2) ← cover(s1', s1); cover(s2, s'') ← cover(s1, s'')



→ 4.3.10.5 Bem:

- a). Es gibt nur Schnittpunkte zwischen Segmenten aus verschiedenen Kurven.
- b). Segmente werden an Schnittpunkten gespalten (→ entweder ganz sichtbar oder ganz unsichtbar.)
- c). Die Sichtbarkeitsinformation ändert sich nur bei Schnittpunkten (wechselt von sichtbar zu unsichtbar oder umgekehrt).

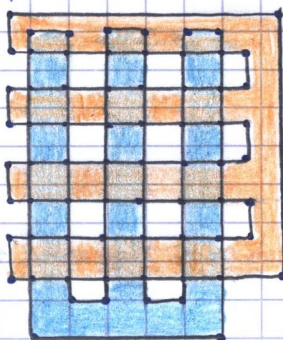
→ 4.3.10.6 Lemma:

Zwei Konturen A und B mit insgesamt n Ecken können in Zeit $O((n+s)\log n)$ berechnet werden, wobei $s = \#$ Schnittpunkte zw. Segmenten in A und B.

Beweis: siehe Segmentsticht.

→ 4.3.10.7 Bem: Im Allg. ist $s = O(n^2)$...

→ 4.3.10.8 Bsp:



z.z: $T(n) = O(n \log^2 n)$.

Indukt. Beh. gelte für $\frac{n}{2}$

Induktionsschritt: dann gilt sie auch für n !

$$\begin{aligned}
 T(n) &\leq c_1 n \log n + 2T\left(\frac{n}{2}\right) \leq c_1 n \log n + 2 \cdot c_2 \frac{n}{2} \cdot \log^2\left(\frac{n}{2}\right) \\
 &= c_1 n \log n + c_2 n (\log n - \log 2)^2 = c_1 n \log n + c_2 n (\log n - 1)^2 \\
 &= c_1 n \log n + c_2 n \log^2 n - 2c_2 n \log n + c_2 n \\
 &= n \log^2 n \left[c_1 \cdot \frac{1}{\log n} + c_2 - \frac{2c_2}{\log n} + \frac{c_2}{\log^2 n} \right] = \\
 &= n \log^2 n \cdot \left[\underbrace{c_1 - 2c_2}_{\xrightarrow{n \rightarrow \infty} 0} + c_2 + \underbrace{\frac{c_2}{\log^2 n}}_{\xrightarrow{n \rightarrow \infty} 0} \right] \leq \\
 &\leq M \cdot n \log^2 n \quad \forall n \geq n_0
 \end{aligned}$$

4.3.11. Analyse der Laufzeit:

→ 4.3.11.1. Idee: Wir zeigen nun, dass bei unserer Anwendung $s = O(n)$

Dann kostet der Merschritt $O(n \log n)$

Für die Gesamtlaufzeit $T(n)$:

$T(1) = 1$

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

Übung 8.

Um die $O(n)$ Schranke für s zu zeigen, verwenden wir die Tatsache, dass sich die Ränder von jeweils zwei aufgetragenen Hindernissen δ_i und δ_j in höchstens zwei Punkten schneiden.

Bew. Übung.

Wir betrachten ein etwas allgemeineres Problem:

Sei $\Gamma = \delta_1 \dots \delta_m$ eine Menge einfacher geschlossener Kurven (= Jordankurven) mit $|\delta_i \cap \delta_j| \leq 2 \quad \forall i \neq j$. d.h. zwei Kurven schneiden sich höchstens zweimal.

→ 4.3.11.2 Definitionen:

- 1) $I(\delta_i) :=$ Innere Regionen von δ_i
- 2) $K(\delta_i) := I(\delta_i) \cup \delta_i$ (also Abschluss)
- 3) $K(\Gamma) := \bigcup_{i=1}^m K(\delta_i)$
- 4) $I(\Gamma) := \bigcup_{i=1}^m (I(\delta_i) \cap I(\delta_j))$ (alle Schnittpunkte)
- 5) $E(\Gamma) := I(\Gamma) \cap \text{Rand}(K(\Gamma))$
- 6) $r_1(\Gamma) := \#$ der redundanten Kurven in Γ , wobei:
 δ_i heißt redundant, falls $\delta_i \subset \bigcup_{j \neq i} K(\delta_j)$, d.h. δ_i wird komplett von anderen Kurvenflächen überdeckt. also $\Rightarrow E(\Gamma \setminus \{\delta_i\}) = E(\Gamma)$
- 7) $r_2(\Gamma) := |\{ (i,j) : i \neq j \wedge \delta_i \cap \delta_j \neq \emptyset \}|$
- 8) $r_3(\Gamma) := |\{ (i,j,k) : i,j,k \text{ paarw. versch.}, K(\delta_i) \cap K(\delta_j) \cap K(\delta_k) \neq \emptyset \}|$

→ schreiben nun einfacher r_i statt $r_i(\Gamma)$ für $i=1,2,3$.

4.3.11.3 Satz: Für $n \geq 3$ gilt: $E(\Gamma) \leq 6n - 12$

Beweis:

1. Vorbemerkungen:

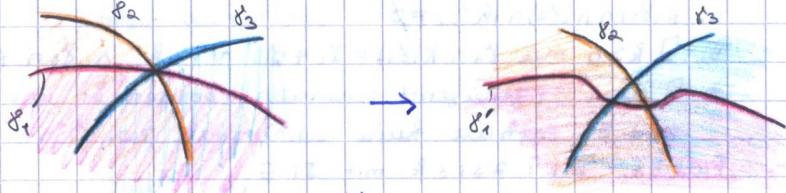
Wir nehmen an, dass sich die Kurven in allgemeiner Lage befinden:

- a) Keine drei Kurven schneiden sich in einem Pkt.
- b) Zwei Kurven schneiden sich entweder in zwei Pkten oder gar nicht. (d.h. keine Berührungspkte).

Dies sind keine echten Einschränkungen, da sich jede Menge Γ durch minimale lokale Verfeinerungen in allgemeine Lage versetzen lässt ohne die Zahl der äußeren Ecken $E(\Gamma)$ zu ändern.

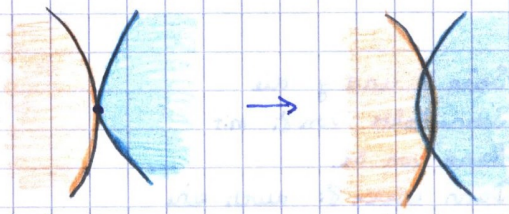
Beispiel Kurve:

Zu a): Seien $\delta_1, \delta_2, \delta_3 \in \Gamma$ mit $\delta_1 \cap \delta_2 \cap \delta_3 = \{p\}$



ersetze δ_i durch δ_i'
 vollkommen verdeckt.

Zu b):



Berührungspkt p Zwei Schnittpkte.

2. Wir zeigen nun den Satz durch Induktion über die Tripel (r_1, r_2, r_3) in lexikographischer Ordnung. Die Idee für den Induktionsschritt besteht darin, dass eine der Zahlen r_1, r_2, r_3 durch Streichen oder Verfeinern einer Kurve δ_i vermindert wird, so dass $E(\Gamma)$ höchstens wächst. (sich nicht verkleinert also).

Induktion: Wir zeigen die Beh für $r_1 = r_2 = 0, r_3 \geq 0$.

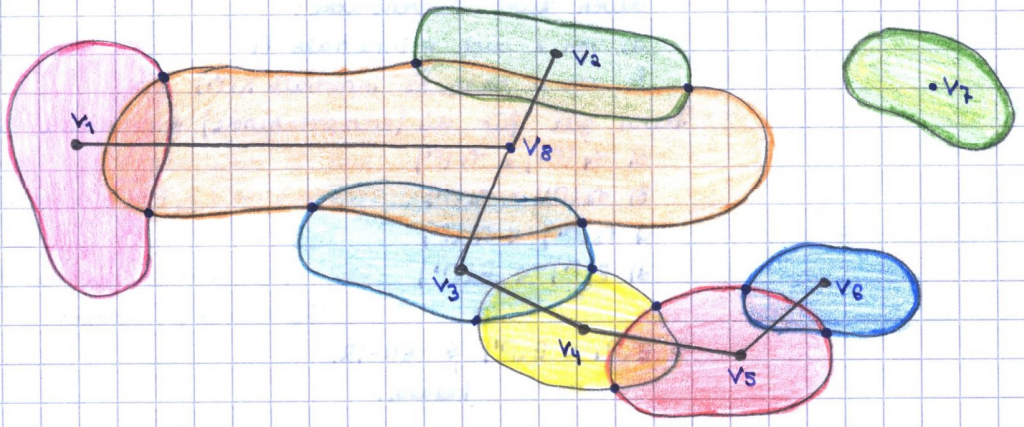
- Es gilt: a) keine redundanten Kurven
- b) $K(\delta_i) \cap K(\delta_j) \cap K(\delta_k) = \emptyset \quad \forall 1 \leq i < j < k \leq n$.
- c) $E(\Gamma) = I(\Gamma)$ (folgt aus b)).

$\forall \delta_i$: sieht die Situation wie folgt aus:

Wir konstruieren einen planaren Graphen $G = (V, E)$ mit $V = \{v_1, \dots, v_n\}$, wobei v_i beliebiger Pkt in $K(\delta_i) \setminus \bigcup_{i \neq j} K(\delta_j)$ und $E = \{(v_i, v_j) : \delta_i \cap \delta_j \neq \emptyset\}$

Planarität: Zeichne Kanten als Kurven durch Schnittflächen $K(\delta_i) \cap K(\delta_j)$.

- \Rightarrow a) G ist planar
- b) jede Kante werden genau zwei Schnittpkte aus $E(\Gamma)$ zugeordnet.
- $\Rightarrow |E(\Gamma)| = 2 \cdot |E| \leq 2 \cdot (3n - 6) = 6n - 12$.



Induktion: Sei Γ eine beliebige Menge von n Jordankurven, die sich paarw. nicht oder zweimal schneiden.

Sei $R(\Gamma) := (r_1, r_2, r_3)$.

Induktion: Beh. gilt $\forall \Gamma'$ mit $R(\Gamma') \leq_L R(\Gamma)$

Fall 1: $r_1 > 0$

$\Rightarrow \exists \delta_i \in \Gamma$ mit δ_i redundant

hier nur r_1 verdeckert $\rightarrow <_L$

Sei $\Gamma' := \Gamma \setminus \{\delta_i\} \Rightarrow E(\Gamma') = E(\Gamma) \wedge R(\Gamma') <_L R(\Gamma), |\Gamma'| = n-1$

$\Rightarrow E(\Gamma) = E(\Gamma') \leq 6(n-1) - 12 = 6n - 6 - 12 \leq 6n - 12$.

Induktion

Fall 2: $r_1 = 0 \wedge r_3 > 0$ (Beh: Fall $r_1 = r_3 = 0$ was Ind.anf.)

Wir können oBdA annehmen (durch geeignete Nummerierung der δ_i):

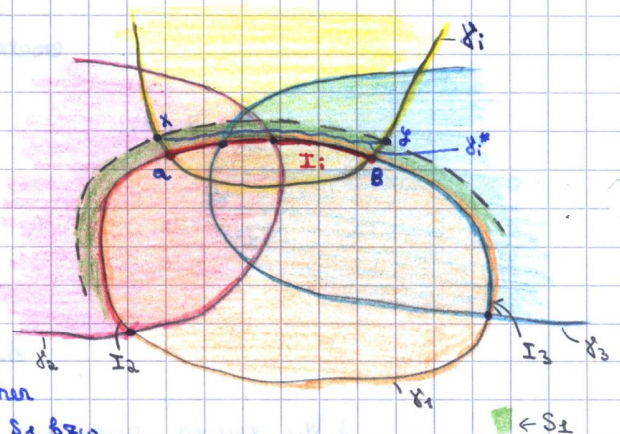
a) $K(\delta_1) \cap K(\delta_2) \cap K(\delta_3) \neq \emptyset$ weil $r_3 > 0 \Rightarrow$ möglich

b) $\exists k \geq 0$ mit $\delta_1 \cap K(\delta_i) = I_i \neq \emptyset \forall i \leq k \wedge \delta_1 \cap K(\delta_j) = \emptyset \forall j > k$.

$\Rightarrow \exists$ einen (genügend schmalen) Streifen S_1 um die Kurve δ_1 mit $S_1 \cap \delta_j = \emptyset \forall j > k$ und S_1 enthält keine Schnittpkte. zw. δ_i 's.

Fall 2.1: $\exists i: 2 \leq i \leq k$ mit $I_i \subseteq \bigcup_{\substack{j=2 \\ j \neq i}}^k I_j$

Bsp:



Seien x und y die Schnittpkte von δ_i mit Rand von S_1 .

Dann ersetze δ_i durch eine Kurve δ_i^* .

Ersetze das Stück von δ_i zwischen x und y , das innerhalb von S_1 bzw. $K(\delta_i)$ verläuft, durch eine Kurve, die x und y im Inneren von S_1 verbindet.

Dann gilt für $\Gamma' := (\Gamma \setminus \{\delta_i\}) \cup \{\delta_i^*\}$:

a) $r_1(\Gamma') = r_1(\Gamma)$ (δ_i redundant $\Leftrightarrow \delta_i^*$ redundant).

b) $r_2(\Gamma') < r_2(\Gamma)$ da $\delta_i \cap \delta_1 \neq \emptyset$ aber $\delta_i^* \cap \delta_1 = \emptyset$ und alle anderen Schnittpkte bleiben erhalten.

$\Rightarrow E(\Gamma) = E(\Gamma')$ da die Schnittpkte a, b von $\delta_i \cap \delta_1$ verdeckt waren

$\Rightarrow E(\Gamma) = E(\Gamma') \leq 6n - 12$

Induktion.

Fall 2.2: $\forall 2 \leq i \leq k: I_i \not\subseteq \bigcup_{\substack{j=2 \\ j \neq i}}^k I_j$

bei entspr. Nummerierung

$\Rightarrow \exists p \in I_3$ mit $p \notin \Gamma_j$ für $j \neq 3$ wg

Sei $I_3 = (a_3, b_3)$ und x, y Schnittpkte von δ_3 mit Rand von S_1

Verfeinere δ_3 zu δ_3^* wie folgt:

ersetze das Stück zwischen x und b_3 durch zwei Kurvenstücke:

a) von x nach p innerhalb S_1

b) von p nach b_3 innerhalb $K(\delta_3)$

Dann gilt für die (entsprechende) erkrankene Menge Γ' :

a) $r_1(\Gamma') = r_1(\Gamma)$ weil δ_3 redundant $\Leftrightarrow \delta_3^*$ redundant.

b) $r_2(\Gamma') = r_2(\Gamma)$ ($(a_3, b_3) \rightsquigarrow (p, b_3)$, sonst nix wandert \Rightarrow # Schnittpkte Gleich.

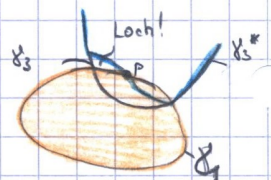
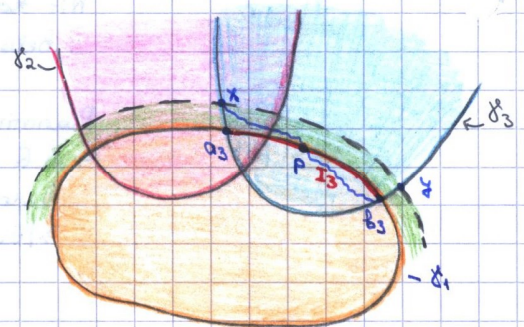
c) $r_3(\Gamma') < r_3(\Gamma)$

d) $E(\Gamma) < E(\Gamma')$ weil: p ist ein neuer externer Schnittpkt!

(wg Loch ist p Randpkt!)

$\Rightarrow E(\Gamma) < E(\Gamma') \leq 6n - 12$.

Induktion.



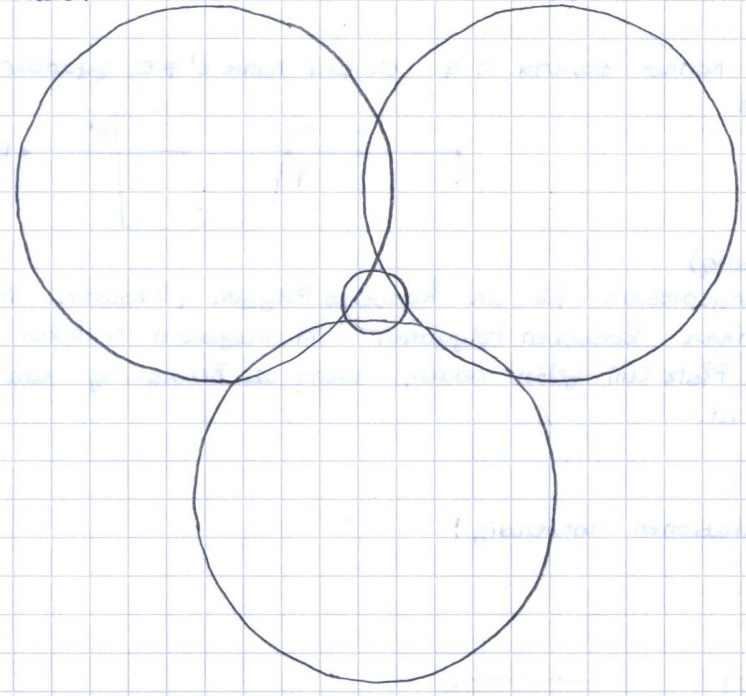
4.3.11.4. Bem:

$E(n) \leq 6n - 12$ ist eine obere Schranke für Komplexität des Konturs.

Es ex. Bsp, bei denen tatsächlich $6n - 12$ äußere Schnittpkte vorkommen.

4.3.11.5 Bsp:

n Kreise:



Die ersten drei Kreise haben 6 Schnittpkte. Jeder weitere Kreis erzeugt nochmals 6 Schnittpkte. $\rightarrow 6 + 6(n-3) = 6n - 12$

die ersten \exists noch $n-3$ Kreis, pro Kreis 6 Schnittpkte $\Rightarrow 6 \cdot (n-3)$

\Rightarrow Schnittpkte = $O(n)$!

4.3.11.6. Satz: Die Kontur von FP hat Größe $O(n)$ und kann in Zeit $O(n \log^2 n)$ berechnet werden.

4.3.12. Lösung des Bewegungsplanungsproblems:

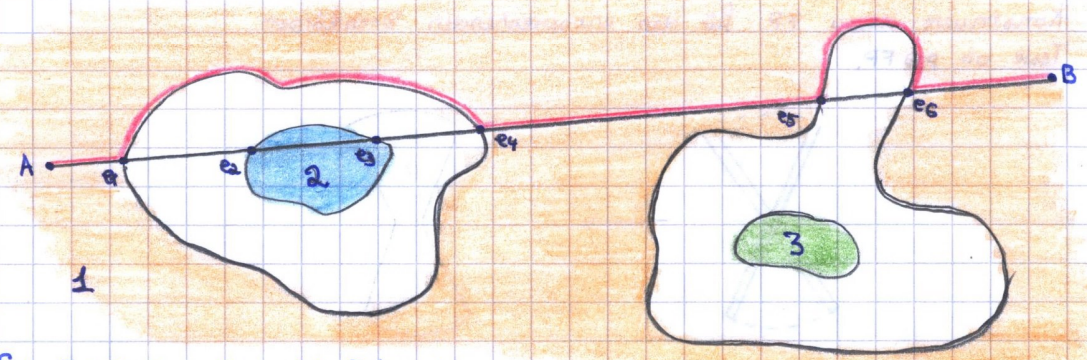
1. Berechne die Zusammenhangskomponenten des Konturs in Zeit $O(n)$.

Kontur: besteht aus geschlossenen Polygonzügen (evtl. ineinander verschachtelt), die sich nicht schneiden

\rightarrow planarer Graph.

2. Berechne die Schnittpkte der Strecke AB mit allen Konturkanten (nur echte Schnittpkte, keine Berührptkte).

$ZHK(e_1) = 1$	$ZHK(e_4) = 1$
$ZHK(e_2) = 2$	$ZHK(e_5) = 1$
$ZHK(e_3) = 2$	$ZHK(e_6) = 1$



1 und 2 kommen gerade oft vor!

Sei $F = e_1, e_2, \dots, e_k$ die Folge der geschnittenen Kanten entlang AB sortiert.

Zeit: $O(n \log n)$ $O(n)$ [für Finden der Kanten] + $O(P \log P)$ [für Sortieren der gefundenen Kanten] = $O(n \log n)$

3. Dann gilt: A und B liegen in derselben Komponente von FP

\Leftrightarrow In der Folge der ZHK's $ZHK(e_1) \dots ZHK(e_k)$ kommt jedes Element gerade oft vor.

Suche e_i in 1, 2, 3 \Rightarrow Zeit $O(n)$ \Rightarrow Zeit $\forall i: O(p \cdot n) \stackrel{p < n}{=} O(n)$

4.3.13 Grober Algorithmus zur Lösung des Bewegungsproblems:

Eine mögliche Bewegung des Roboters kann nun wie folgt konstruiert werden:

```

p ← A
while p ≠ B do
  sei e nächste Kante von p aus in der Folge F
  q ← e ∩ AB
  durchlaufe die Kontur absteigend in q bis eine Kante e' ≠ e gefunden wird
  mit: e' ∩ AB ≠ ∅
  p ← e' ∩ AB
  
```



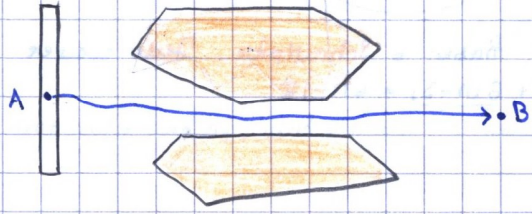
4.3.14 Satz: (Zusammenfassung)

Das Bewegungsplanungsproblem für ein konvexes Polygon (\cong Roboter) in einer Szene von paarweise disjunkten konvexen Polygonen mit insgesamt n Ecken kann in Zeit $O(n \log^2 n)$ und Platz $O(n)$ gelöst werden, wenn als Bewegung eine Folge von Translationen erlaubt ist.

Bew. s.o.

4.3.15 Bemerkung:

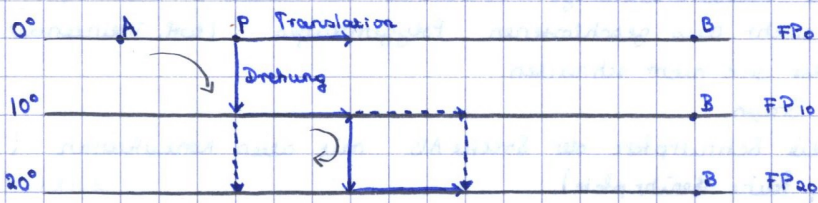
In der Praxis sind Rotationen notwendig!



Idee: Diskretisierung

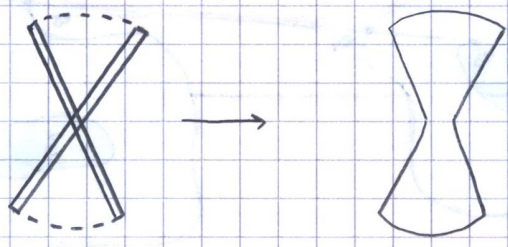
d.h. erlaube nur eine konstante Zahl von möglichen Winkeln (d.h. Orientierung des Roboters). z.B. $0^\circ, 10^\circ, 20^\circ, \dots, 360^\circ$

Nun konstruiere "Schichten" von FP-Konturen für jeden Winkel



Wichtig: Test, ob Drehung möglich ist!

Konstruiere dazu FP für den entsprechenden Drehkörper
 Teste ob $p \in FP$.



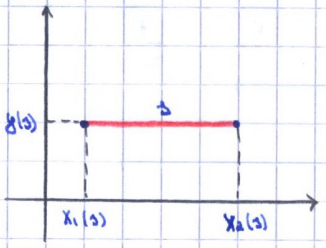
5.1. Segmentbaum

5.1.1 Definitionen + Bemerkungen

Sei $S = \{s_1, \dots, s_n\}$ Menge von n horizontalen Liniensegmenten in der Ebene.

Wir bezeichnen mit:

- $x_1(s_i)$: die x -Koord. des linken Endpunkts von s_i
- $x_2(s_i)$: die x -Koord. des rechten Endpunkts von s_i
- $y(s_i)$: die y -Koord. von s_i .



Wir nehmen zunächst an, dass alle x -Koord. ganze Zahlen aus $\{1..N\}$ sind und die y -Koord. beliebige reelle Zahlen sind. (→ sog. Halbdynamischer Fall).

Ein Segmentbaum T zur Speicherung von S ist ein blattorientierter, binärer Suchbaum für die x -Koord. $\{1..N\}$ der Höhe $\log N$.

Jeder Knoten erhält zusätzlich eine Liste $NL(v)$ von Segmenten nach y -Koord. sortiert, die Knotenliste von v .

Bea: Beim Segmentbaum müssen die Blätter nicht voll sein.

Segmente von S werden wie folgt abgespeichert:

Sei $s \in S$.

$P_1(s)$ = Suchpfad nach $x_1(s)$ / Suchpfad nach $x_2(s)$

$P_2(s)$ = Suchpfad nach $x_2(s)$ / Suchpfad nach $x_1(s)$

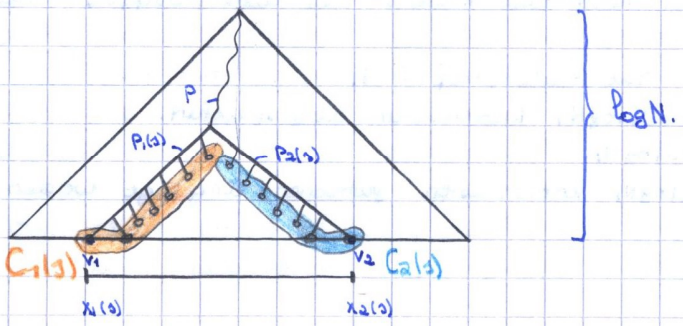
$C_1(s)$ = $\{v : v \text{ rechtes Kind eines Knotens von } P_1(s) \text{ und } v \notin P_1(s)\} \cup \{v_1\}$

$C_2(s)$ = $\{v : v \text{ linkes Kind eines Knotens von } P_2(s) \text{ und } v \notin P_2(s)\} \cup \{v_2\}$

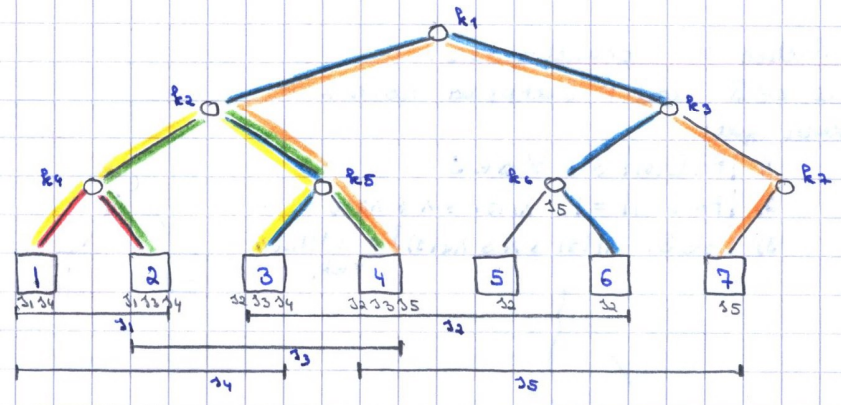
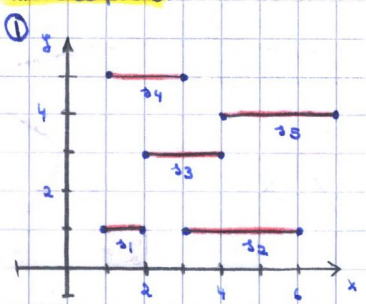
$C(s) = C_1(s) \cup C_2(s)$.

Dann wird $s_i, 1 \leq i \leq n$ in alle Knotenlisten $NL(v)$ mit $v \in C(s)$ abgespeichert.

Skizze:



5.1.2 Beispiele:



$P_1(s_1)$ =	$P_1(s_5)$ =	$C_1(s_1) = \{1\}$	$C_1(s_5) = \{4\}$	$NL(r_1) = \{s_1, s_4\}$	$NL(r_1, r_5) = \emptyset$
$P_2(s_1)$ =	$P_2(s_5)$ =	$C_2(s_1) = \{2\}$	$C_2(s_5) = \{r_6, 7\}$	$NL(2) = \{s_1, s_3, s_4\}$	$NL(r_6) = \{s_5\}$
$P_1(s_2)$ =		$C_1(s_2) = \{4, 3\}$		$NL(3) = \{s_2, s_3, s_4\}$	$NL(r_7) = \emptyset$
$P_2(s_2)$ =		$C_2(s_2) = \{5, 6\}$		$NL(4) = \{s_2, s_3, s_5\}$	
$P_1(s_3)$ =		$C_1(s_3) = \{2\}$		$NL(5) = \{s_2\}$	
$P_2(s_3)$ =		$C_2(s_3) = \{3, 4\}$		$NL(6) = \{s_2\}$	
$P_1(s_4)$ =		$C_1(s_4) = \{2, 1\}$		$NL(7) = \{s_5\}$	
$P_2(s_4)$ =		$C_2(s_4) = \{3\}$			