Offline Bewegungsplanung: Objekte bewegen

Elmar Langetepe
University of Bonn

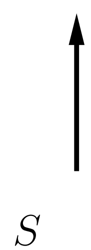
Liniensegmente statt Punkte

Liniensegmente statt Punkte

- Liniensegment in der Ebene

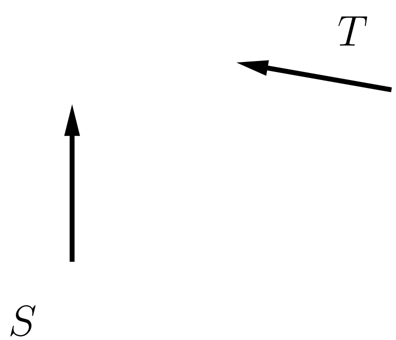
Liniensegmente statt Punkte

- Liniensegment in der Ebene
- Startlage S



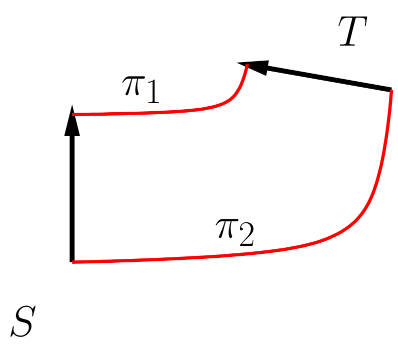
Liniensegmente statt Punkte

- Liniensegment in der Ebene
- Startlage S und Ziellage T



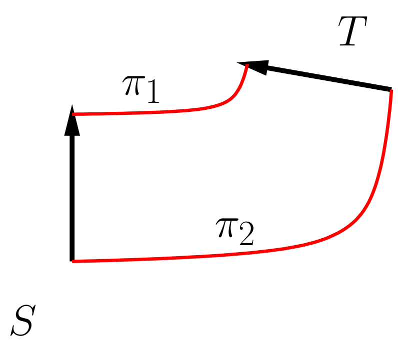
Liniensegmente statt Punkte

- Liniensegment in der Ebene
- Startlage S und Ziellage T
- Stetige Bewegung von S nach T



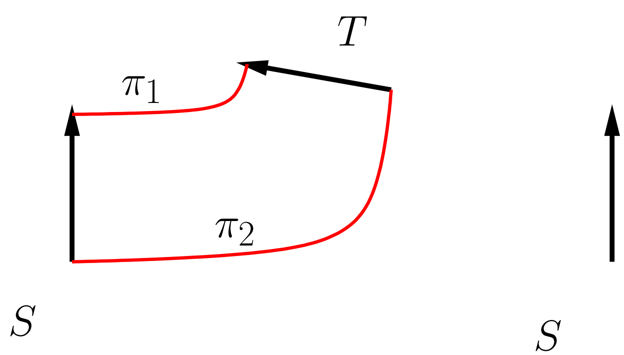
Liniensegmente statt Punkte

- Liniensegment in der Ebene
- Startlage S und Ziellage T
- Stetige Bewegung von S nach T
- Möglichst kurz!



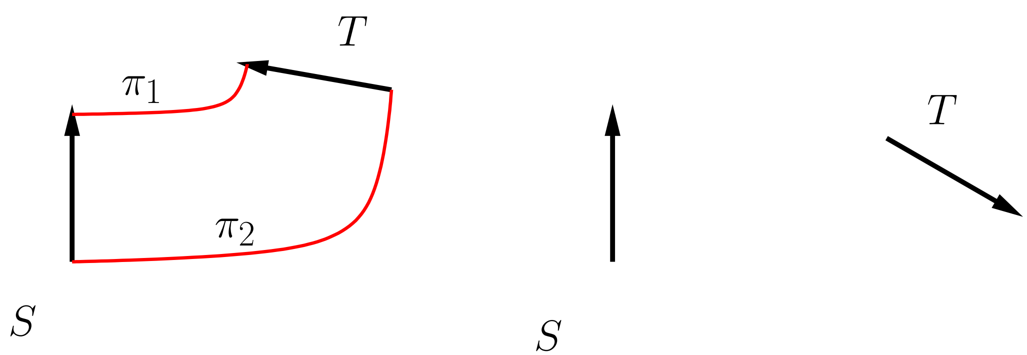
Liniensegmente statt Punkte

- Liniensegment in der Ebene
- Startlage S und Ziellage T
- Stetige Bewegung von S nach T
- Möglichst kurz!



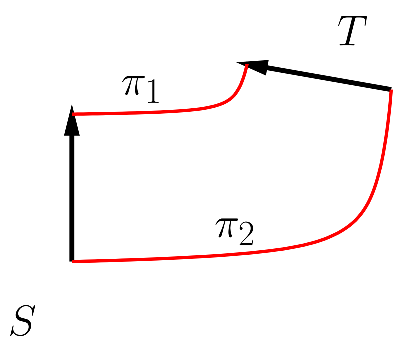
Liniensegmente statt Punkte

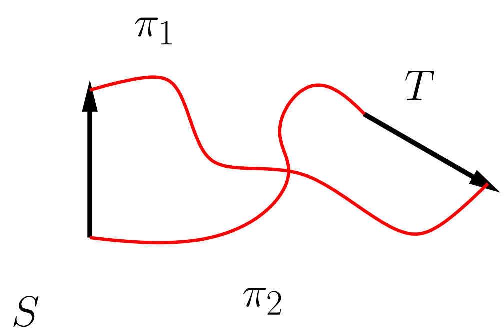
- Liniensegment in der Ebene
- Startlage S und Ziellage T
- Stetige Bewegung von S nach T
- Möglichst kurz!



Liniensegmente statt Punkte

- Liniensegment in der Ebene
- Startlage S und Ziellage T
- Stetige Bewegung von S nach T
- Möglichst kurz!





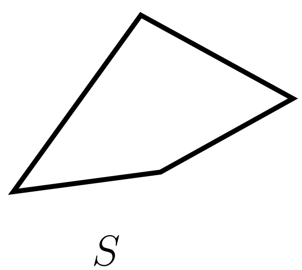
Was ist eine kurze Bewegung?

Was ist eine kurze Bewegung?

- Bei ausgedehnten Objekten schwierig

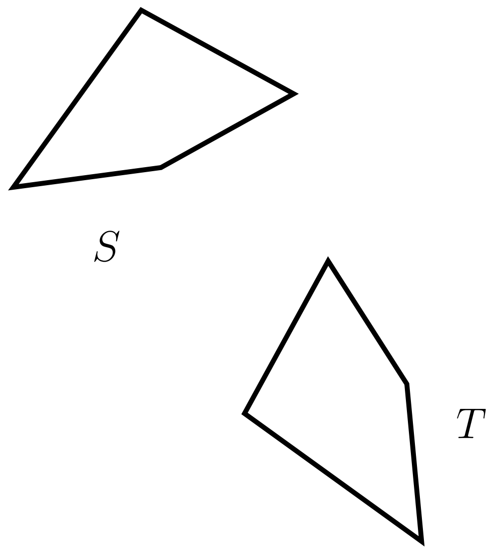
Was ist eine kurze Bewegung?

- Bei ausgedehnten Objekten schwierig



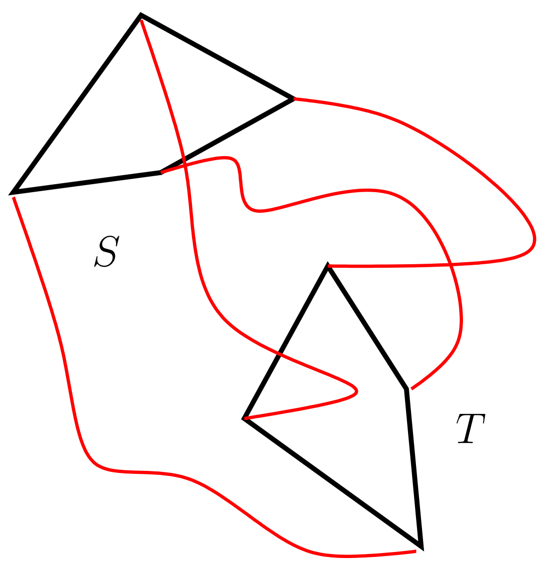
Was ist eine kurze Bewegung?

- Bei ausgedehnten Objekten schwierig



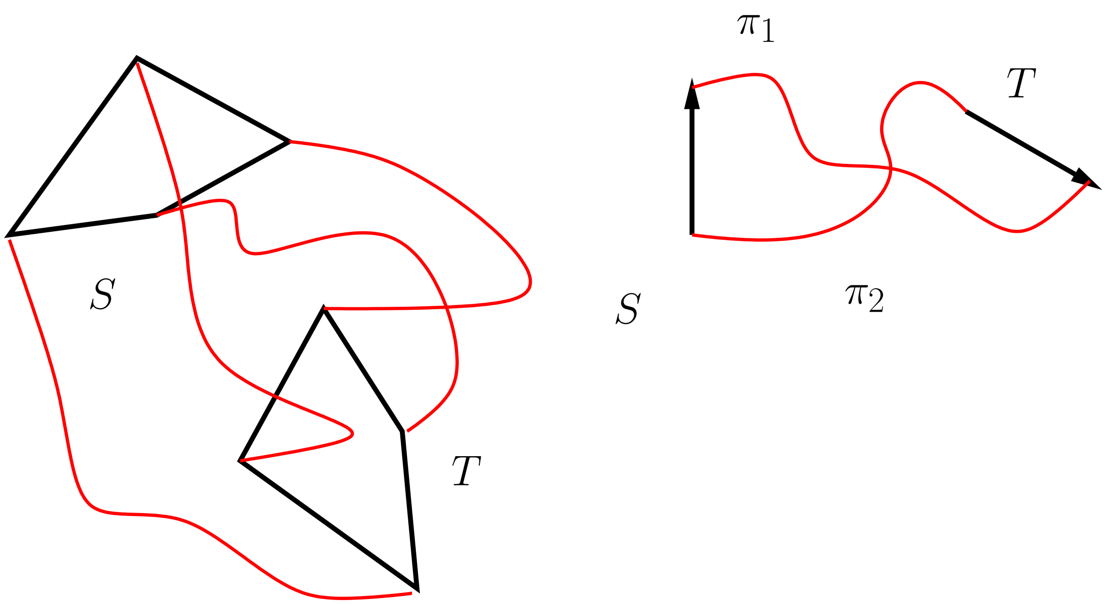
Was ist eine kurze Bewegung?

- Bei ausgedehnten Objekten schwierig



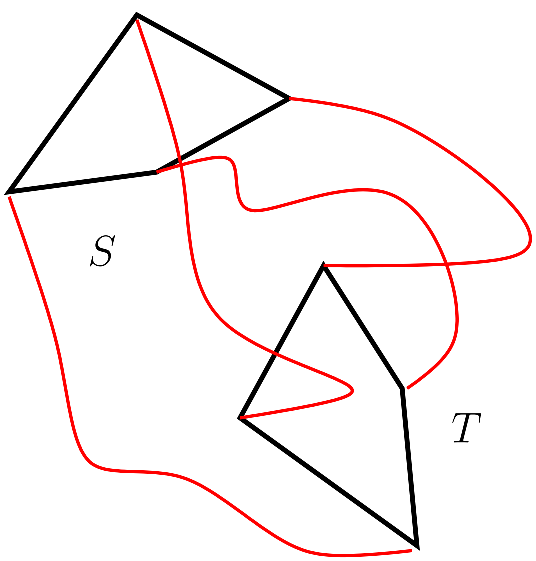
Was ist eine kurze Bewegung?

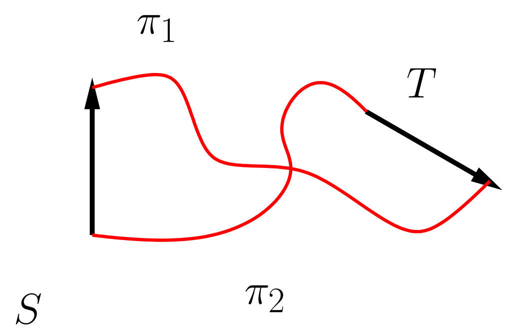
- Bei ausgedehnten Objekten schwierig
- Hier: Weglänge zweier(!) Träger: Summe



Was ist eine kurze Bewegung?

- Bei ausgedehnten Objekten schwierig
- Hier: Weglänge zweier(!) Träger: Summe
- Andere Maße: Mittlerer Weg, Min/Max eines Trägers





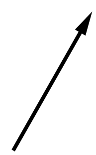
Formale Beschreibung!

Formale Beschreibung!

- Lage des Segmentes durch Tripel (x, y, α)

Formale Beschreibung!

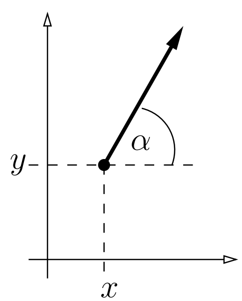
- Lage des Segmentes durch Tripel (x, y, α)



(i)

Formale Beschreibung!

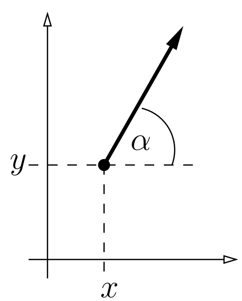
- Lage des Segmentes durch Tripel (x, y, α)
- Normiert!



(i)

Formale Beschreibung!

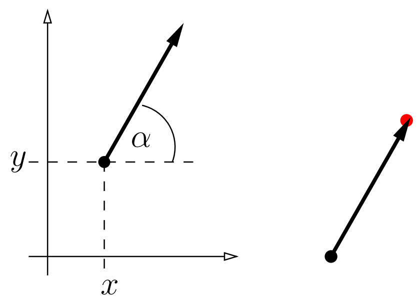
- Lage des Segmentes durch Tripel (x, y, α)
- Normiert!
- Optimale Bewegungen: Rotation/Translation(Klar!!)



(i)

Formale Beschreibung!

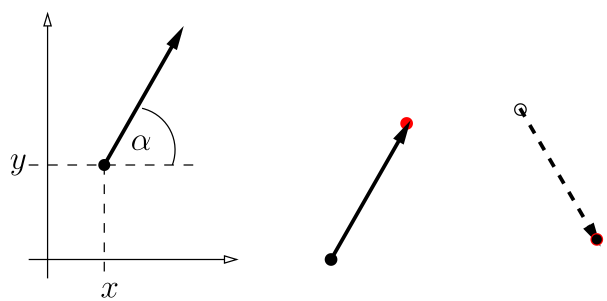
- Lage des Segmentes durch Tripel (x, y, α)
- Normiert!
- Optimale Bewegungen: Rotation/Translation(Klar!!)



(i)

Formale Beschreibung!

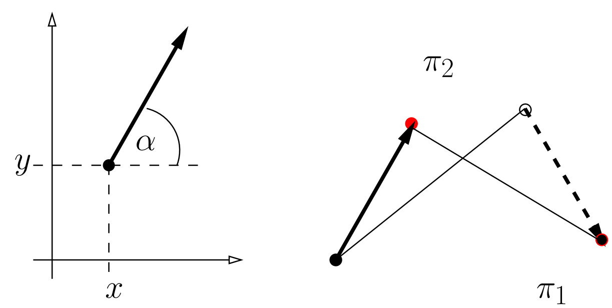
- Lage des Segmentes durch Tripel (x, y, α)
- Normiert!
- Optimale Bewegungen: Rotation/Translation(Klar!!)



(i)

Formale Beschreibung!

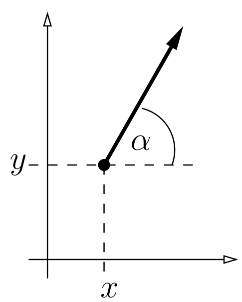
- Lage des Segmentes durch Tripel (x, y, α)
- Normiert!
- Optimale Bewegungen: Rotation/Translation(Klar!!)



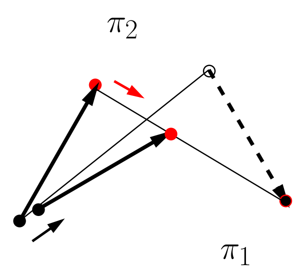
(i)

Formale Beschreibung!

- Lage des Segmentes durch Tripel (x, y, α)
- Normiert!
- Optimale Bewegungen: Rotation/Translation(Klar!!)

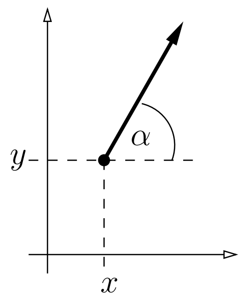


(i)

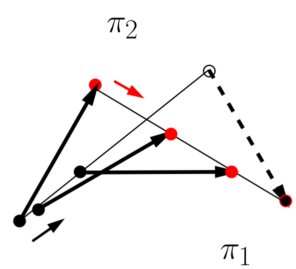


Formale Beschreibung!

- Lage des Segmentes durch Tripel (x, y, α)
- Normiert!
- Optimale Bewegungen: Rotation/Translation(Klar!!)

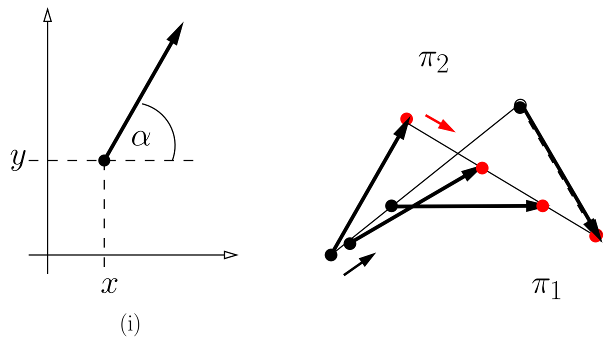


(i)



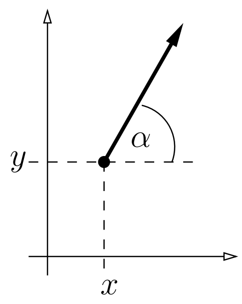
Formale Beschreibung!

- Lage des Segmentes durch Tripel (x, y, α)
- Normiert!
- Optimale Bewegungen: Rotation/Translation(Klar!!)

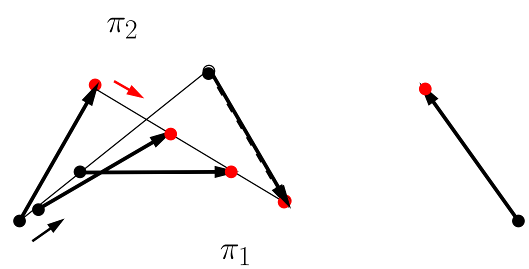


Formale Beschreibung!

- Lage des Segmentes durch Tripel (x, y, α)
- Normiert!
- Optimale Bewegungen: Rotation/Translation(Klar!!)

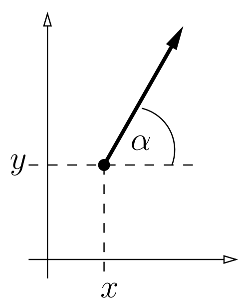


(i)

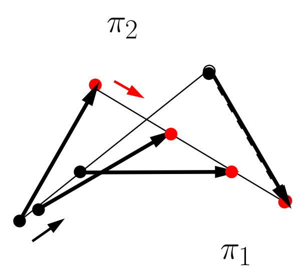


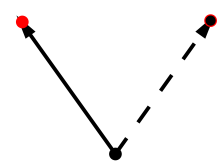
Formale Beschreibung!

- Lage des Segmentes durch Tripel (x, y, α)
- Normiert!
- Optimale Bewegungen: Rotation/Translation(Klar!!)



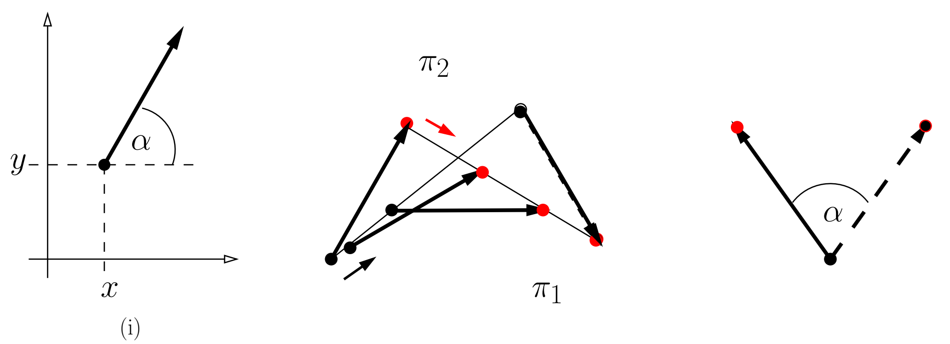
(i)





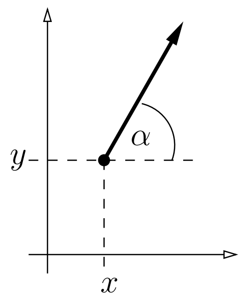
Formale Beschreibung!

- Lage des Segmentes durch Tripel (x, y, α)
- Normiert!
- Optimale Bewegungen: Rotation/Translation(Klar!!)

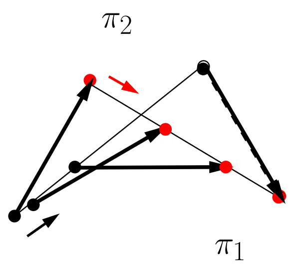


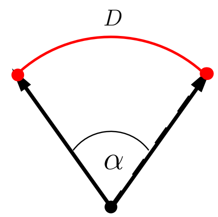
Formale Beschreibung!

- Lage des Segmentes durch Tripel (x, y, α)
- Normiert!
- Optimale Bewegungen: Rotation/Translation(Klar!!)
- Translation geht nicht immer



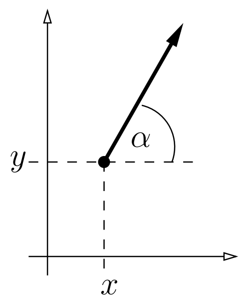
(i)



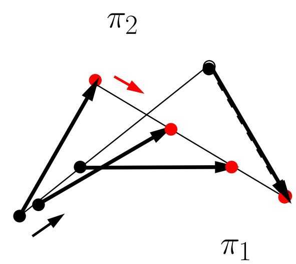


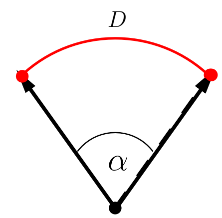
Formale Beschreibung!

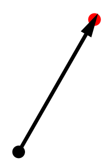
- Lage des Segmentes durch Tripel (x, y, α)
- Normiert!
- Optimale Bewegungen: Rotation/Translation(Klar!!)
- Translation geht nicht immer



(i)

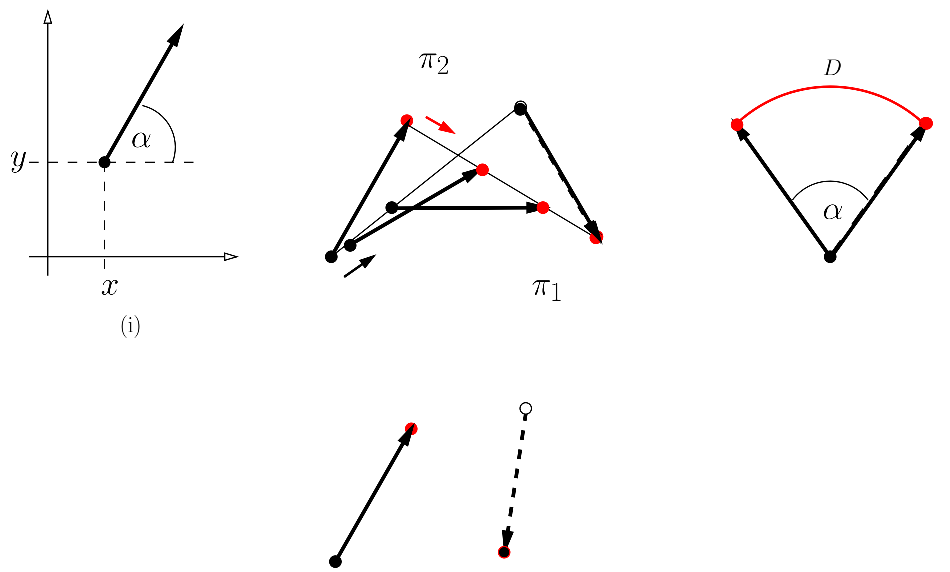






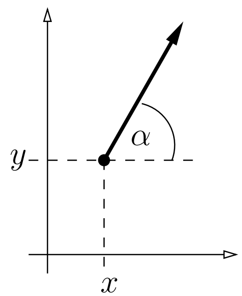
Formale Beschreibung!

- Lage des Segmentes durch Tripel (x, y, α)
- Normiert!
- Optimale Bewegungen: Rotation/Translation(Klar!!)
- Translation geht nicht immer

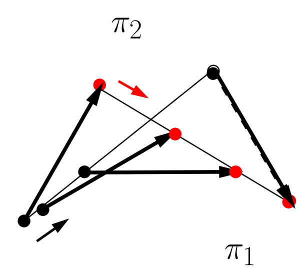


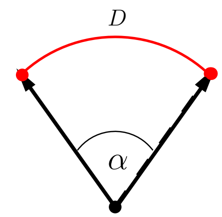
Formale Beschreibung!

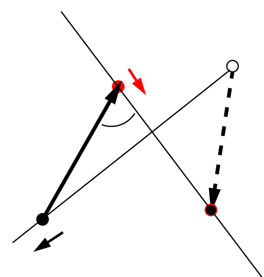
- Lage des Segmentes durch Tripel (x, y, α)
- Normiert!
- Optimale Bewegungen: Rotation/Translation(Klar!!)
- Translation geht nicht immer



(i)







Kombinationen Th. 1.47

Zwischen je zwei Positionen von Liniensegmenten gibt es eine optimale Bewegung von einem der folgenden Typen:

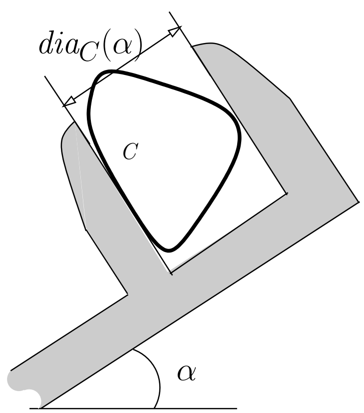
1. maximal drei Rotationen,
2. maximal zwei Rotationen und eine geradlinige Bewegung,
3. eine Rotation zwischen zwei geradlinigen Bewegungen.

Die Bewegungen lassen sich effizient berechnen.

Rotation? Hilfe: Surface-Area Th. 1.46!

Rotation? Hilfe: Surface-Area Th. 1.46!

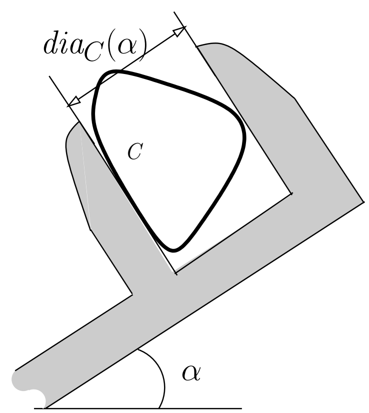
- Geschlossene konvexe Kurve C: Durchm. Funktion: $\mathtt{dia}_\mathtt{C} : [0, 2\pi) \rightarrow \mathtt{R}$



(i)

Rotation? Hilfe: Surface-Area Th. 1.46!

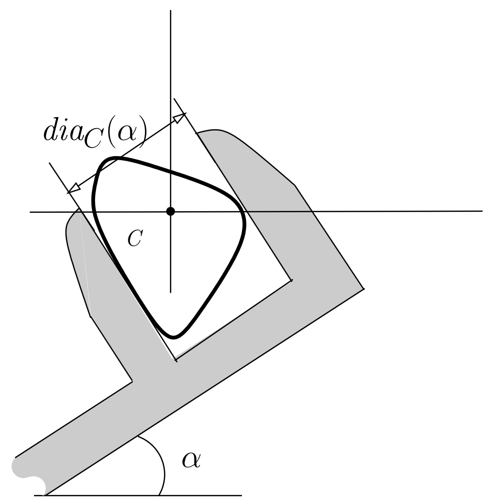
- Geschlossene konvexe Kurve C: Durchm. Funktion: $\mathtt{dia_C} : [0, 2\pi) \to \mathtt{R}$
- Support Funktion: $\mathtt{h_C}(\alpha) = \sup\{\mathtt{x}\cos\alpha + \mathtt{y}\sin\alpha | (\mathtt{x}, \mathtt{y}) \in \mathtt{C}\}$
- X-Koordinate nach Rotation mit α



(i)

Rotation? Hilfe: Surface-Area Th. 1.46!

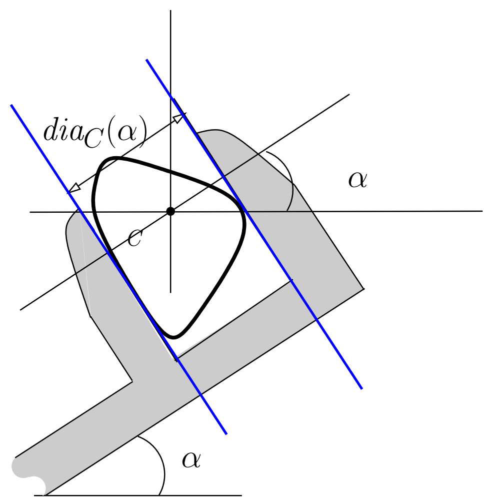
- Geschlossene konvexe Kurve C: Durchm. Funktion: $\mathtt{dia_C} : [0, 2\pi) \rightarrow \mathtt{R}$
- Support Funktion: $\mathtt{h_C}(\alpha) = \sup\{\mathtt{x}\cos\alpha + \mathtt{y}\sin\alpha | (\mathtt{x}, \mathtt{y}) \in \mathtt{C}\}$
- X-Koordinate nach Rotation mit α



(i)

Rotation? Hilfe: Surface-Area Th. 1.46!

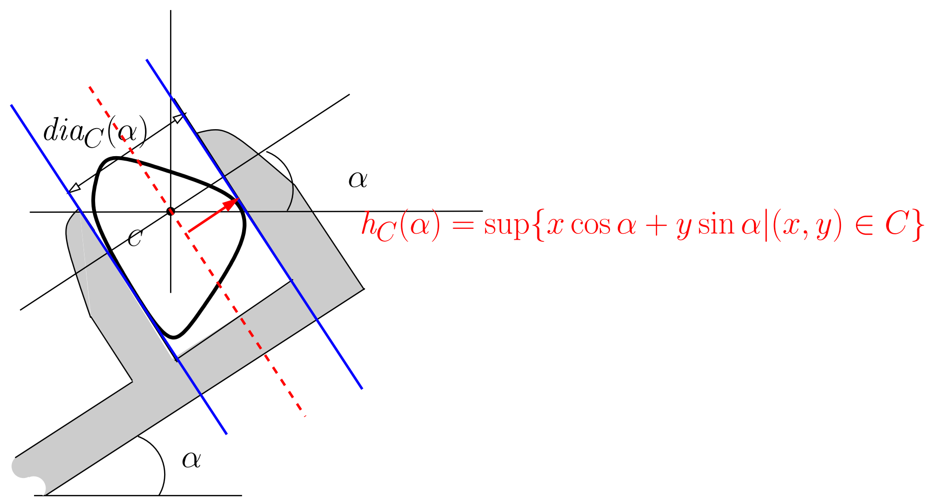
- Geschlossene konvexe Kurve C: Durchm. Funktion: $\mathtt{dia_C} : [0, 2\pi) \rightarrow \mathtt{R}$
- Support Funktion: $\mathtt{h_C}(\alpha) = \sup\{\mathtt{x}\cos\alpha + \mathtt{y}\sin\alpha | (\mathtt{x}, \mathtt{y}) \in \mathtt{C}\}$
- X-Koordinate nach Rotation mit α



(i)

Rotation? Hilfe: Surface-Area **Th. 1.46!**

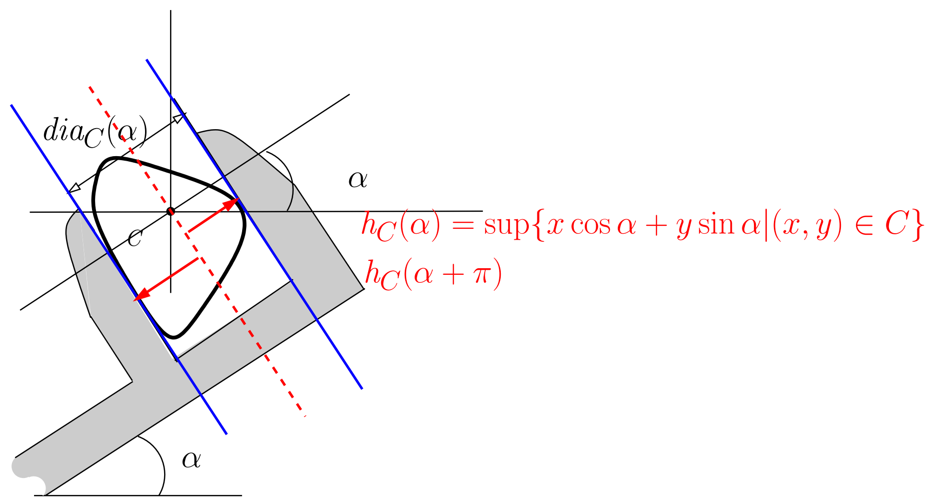
- Geschlossene konvexe Kurve C: Durchm. Funktion: $\mathtt{dia_C} : [0, 2\pi) \to \mathtt{R}$
- Support Funktion: $\mathtt{h_C}(\alpha) = \sup\{\mathtt{x}\cos\alpha + \mathtt{y}\sin\alpha | (\mathtt{x}, \mathtt{y}) \in \mathtt{C}\}$
- X-Koordinate nach Rotation mit α



(i)

Rotation? Hilfe: Surface-Area Th. 1.46!

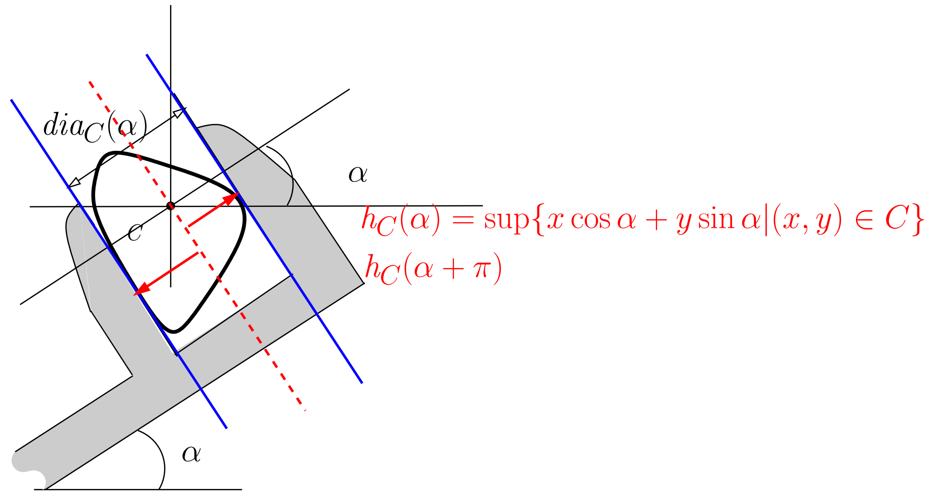
- Geschlossene konvexe Kurve C: Durchm. Funktion: $\mathtt{dia_C} : [0, 2\pi) \to \mathtt{R}$
- Support Funktion: $\mathtt{h_C}(\alpha) = \sup\{\mathtt{x}\cos\alpha + \mathtt{y}\sin\alpha | (\mathtt{x}, \mathtt{y}) \in \mathtt{C}\}$
- X-Koordinate nach Rotation mit α
- Offensichtlich: $\mathtt{dia_C}(\alpha) = \mathtt{h_C}(\alpha) + \mathtt{h_C}(\alpha + \pi)$



(i)

Rotation? Hilfe: Surface-Area Th. 1.46!

- Geschlossene konvexe Kurve C: Durchm. Funktion: $\mathtt{dia_C} : [0, 2\pi) \to \mathtt{R}$
- Support Funktion: $\mathtt{h_C}(\alpha) = \sup\{\mathtt{x}\cos\alpha + \mathtt{y}\sin\alpha | (\mathtt{x}, \mathtt{y}) \in \mathtt{C}\}$
- X-Koordinate nach Rotation mit α
- Offensichtlich: $\mathtt{dia_C}(\alpha) = \mathtt{h_C}(\alpha) + \mathtt{h_C}(\alpha + \pi)$
- Es gilt: $\mathtt{Length(C)} = \int_0^\pi \mathtt{dia_C}(\alpha)\,\mathtt{d}\alpha$



(i)

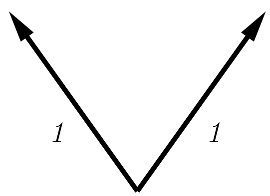
Beweis: Rotation ist optimal!

Beweis: Rotation ist optimal!

- Zwei Kurven C,

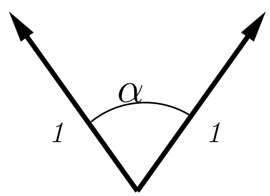
Beweis: Rotation ist optimal!

- Zwei Kurven C,



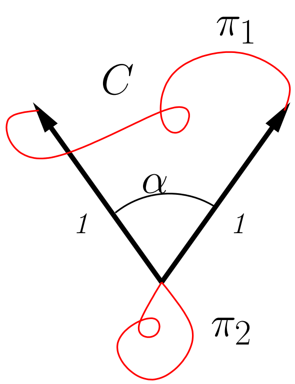
Beweis: Rotation ist optimal!

- Zwei Kurven C,



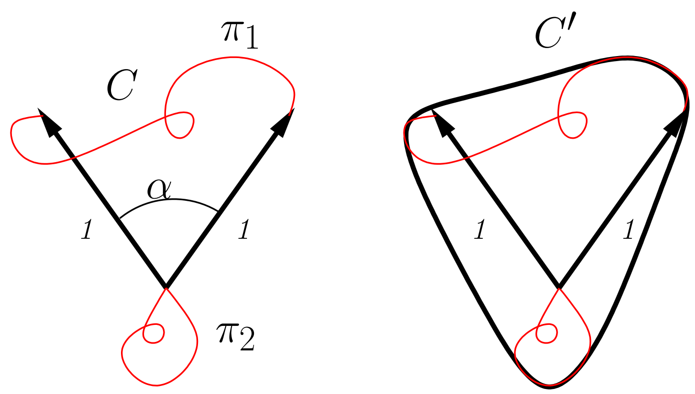
Beweis: Rotation ist optimal!

- Zwei Kurven C, konvexe Hülle C',



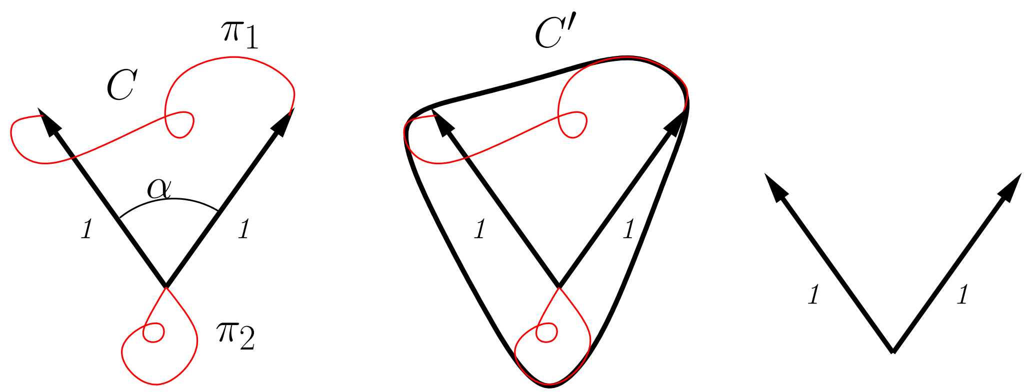
Beweis: Rotation ist optimal!

- Zwei Kurven C, konvexe Hülle C', D einfache Rotation



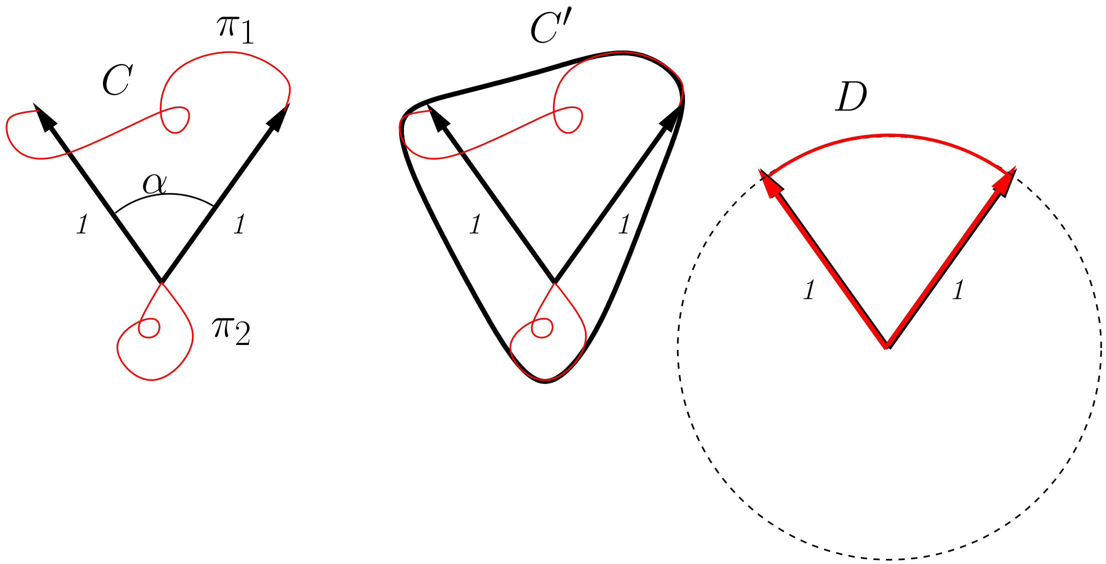
Beweis: Rotation ist optimal!

- Zwei Kurven C, konvexe Hülle C', D einfache Rotation

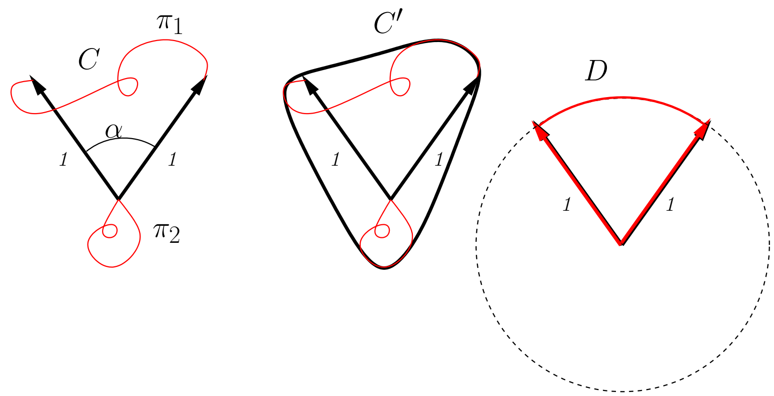


Beweis: Rotation ist optimal!

- Zwei Kurven C, konvexe Hülle C', D einfache Rotation
- Rotation optimal?

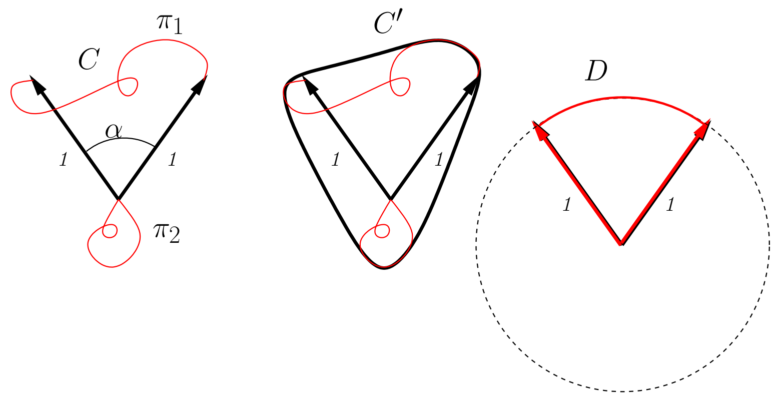


Beweis: Rotation ist optimal!



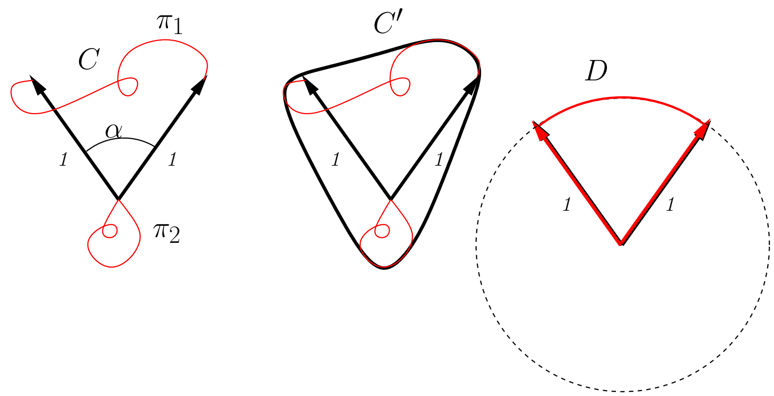
Beweis: Rotation ist optimal!

$2 + \text{Kosten}(C)$



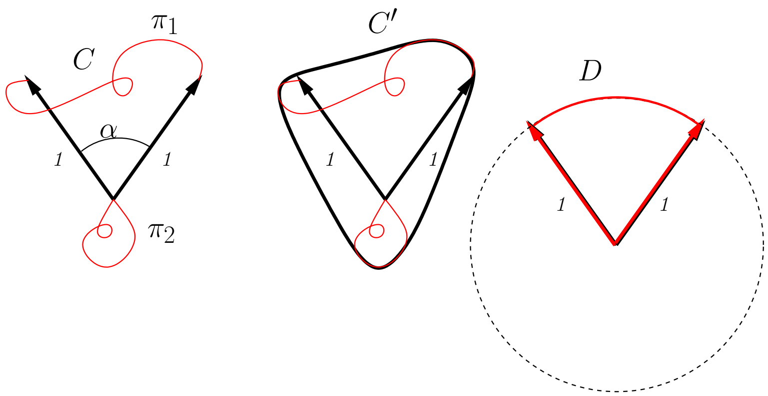
Beweis: Rotation ist optimal!

$2 + \text{Kosten}(C) \;\; \geq \;\; \text{Länge}(C')$



Beweis: Rotation ist optimal!

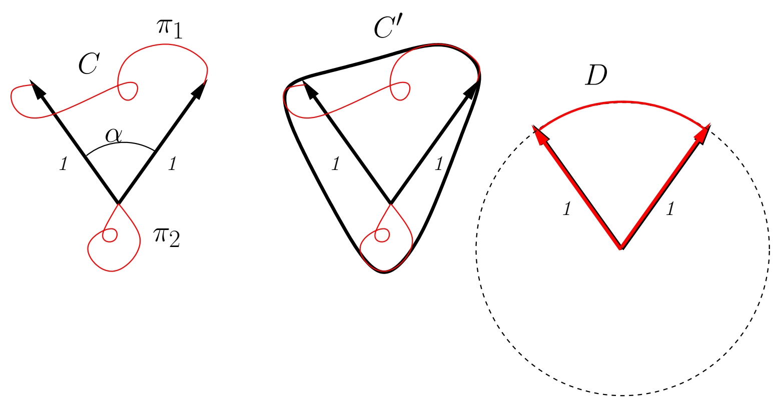
$$2 + \text{Kosten}(C) \;\geq\; \text{Länge}(C') = \int_0^\pi \mathtt{dia}_{C'}(\alpha)\,\mathtt{d}\alpha \qquad (\text{Th. } 1.46)$$



Beweis: Rotation ist optimal!

$$2 + \text{Kosten}(C) \quad \geq \quad \text{Länge}(C') = \int_0^\pi \mathtt{dia}_{\mathtt{C'}}(\alpha)\, \mathtt{d}\alpha \qquad (\text{Th. 1.46})$$

$$\geq \quad \int_0^\pi \mathtt{dia}_{\mathtt{D}}(\alpha)\, \mathtt{d}\alpha \qquad (\text{wegen } \mathtt{dia}_{\mathtt{C'}}(\alpha) \geq \mathtt{dia}_{\mathtt{D}}(\alpha))$$

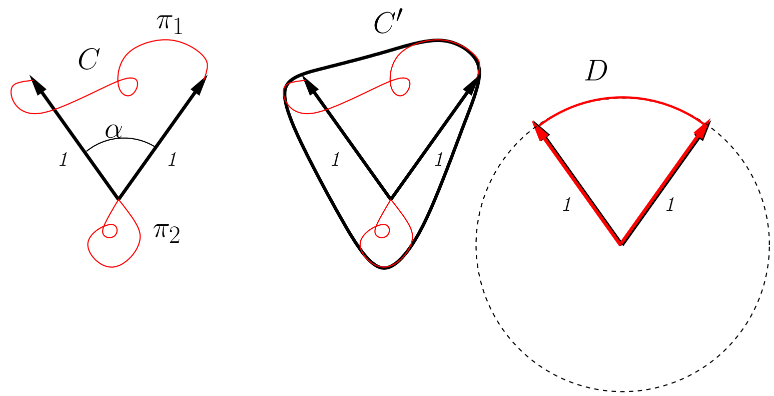


Beweis: Rotation ist optimal!

$$2 + \text{Kosten}(C) \geq \text{Länge}(C') = \int_0^{\pi} \mathtt{dia}_{\mathtt{C'}}(\alpha)\, \mathtt{d}\alpha \quad (\text{Th. } 1.46)$$

$$\geq \int_0^{\pi} \mathtt{dia}_{\mathtt{D}}(\alpha)\, \mathtt{d}\alpha \quad (\text{wegen } \mathtt{dia}_{\mathtt{C'}}(\alpha) \geq \mathtt{dia}_{\mathtt{D}}(\alpha))$$

$$= \text{Länge}(D) \quad (\text{Th. } 1.46)$$

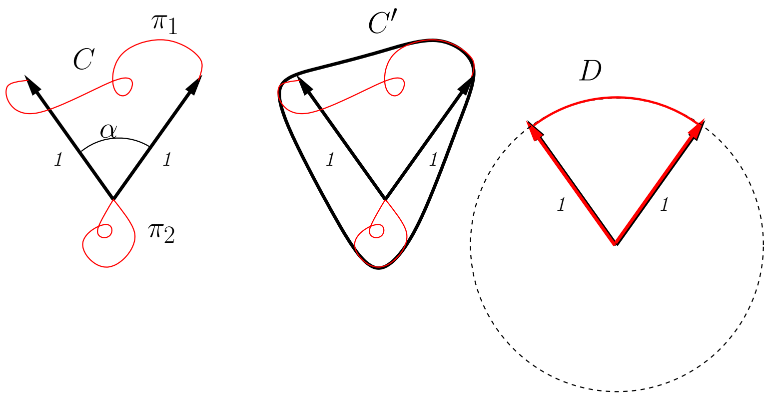


Beweis: Rotation ist optimal!

$$2 + \text{Kosten}(C) \geq \text{Länge}(C') = \int_0^\pi \mathtt{dia}_{C'}(\alpha)\,\mathtt{d}\alpha \quad (\text{Th. } 1.46)$$

$$\geq \int_0^\pi \mathtt{dia}_{D}(\alpha)\,\mathtt{d}\alpha \quad (\text{wegen } \mathtt{dia}_{C'}(\alpha) \geq \mathtt{dia}_{D}(\alpha))$$

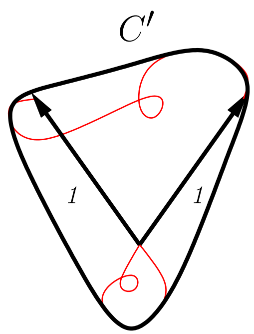
$$= \text{Länge}(D) \quad (\text{Th. } 1.46) = 2 + \text{Kosten}(\text{Rotation})$$

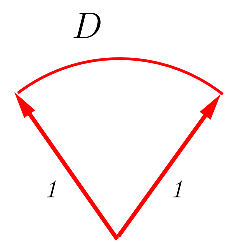


Beweis: $\mathtt{dia}_{\mathtt{C'}}(\alpha) \geq \mathtt{dia}_{\mathtt{D}}(\alpha)$, $\alpha \in [0, \pi]$

Beweis: $\mathtt{dia}_{\mathtt{C}'}(\alpha) \geq \mathtt{dia}_{\mathtt{D}}(\alpha)$, $\alpha \in [0, \pi]$

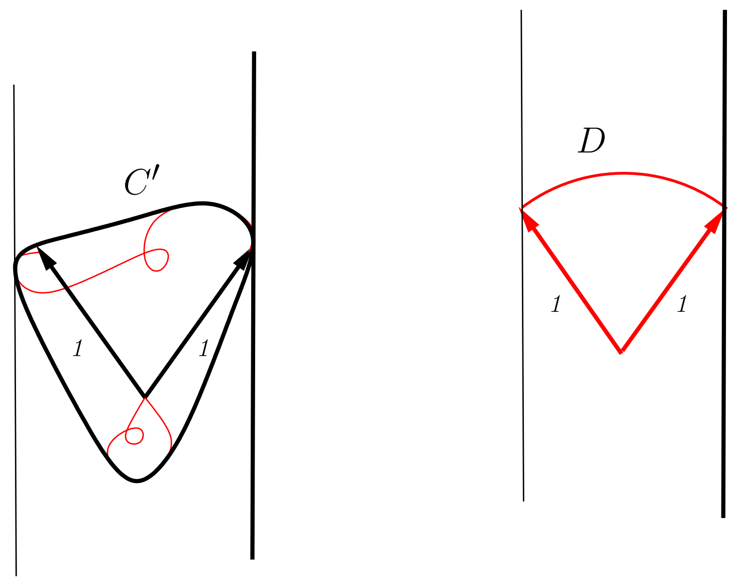
- Drei Winkelbereiche!!





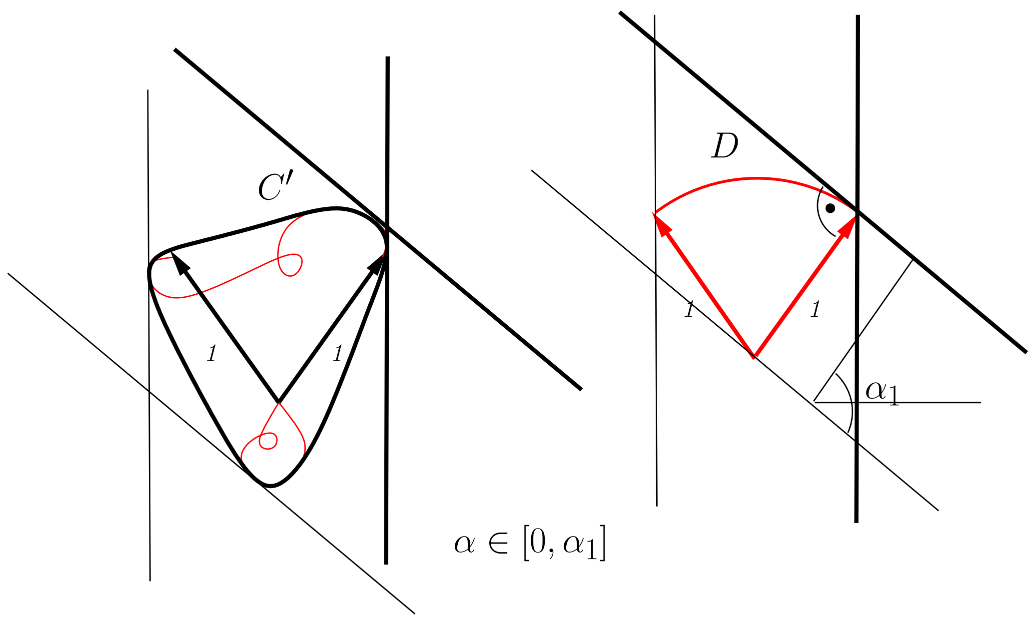
Beweis: $\mathtt{dia}_{\mathtt{C}'}(\alpha) \geq \mathtt{dia}_{\mathtt{D}}(\alpha)$, $\alpha \in [0, \pi]$

- Drei Winkelbereiche!!



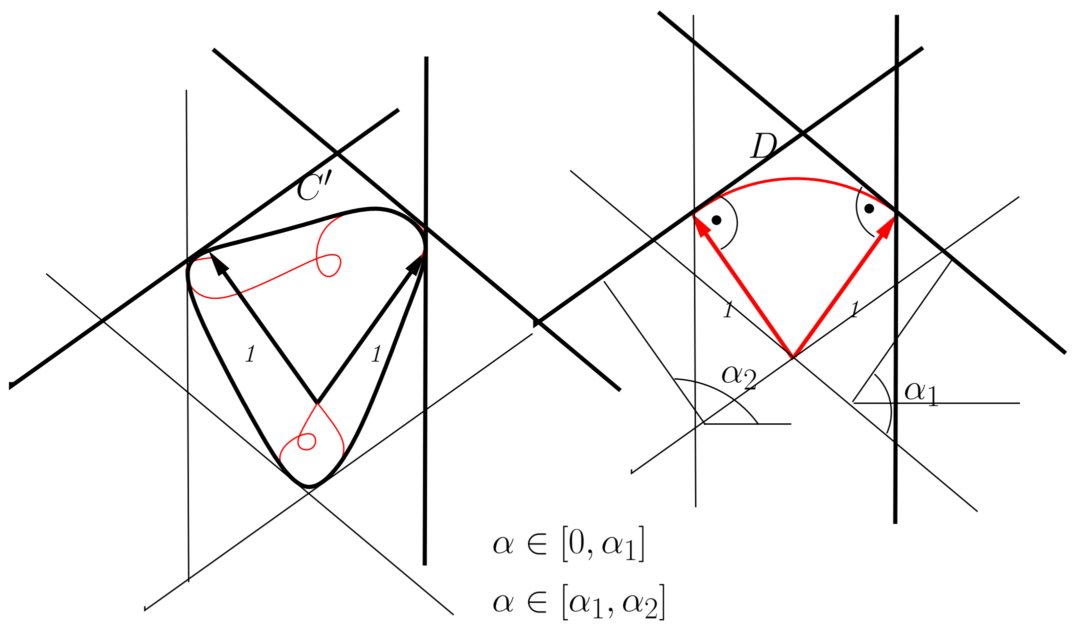
Beweis: $\mathtt{dia}_{\mathtt{C'}}(\alpha) \geq \mathtt{dia}_{\mathtt{D}}(\alpha)$, $\alpha \in [0, \pi]$

- Drei Winkelbereiche!!



Beweis: $\mathtt{dia}_{\mathtt{C}'}(\alpha) \geq \mathtt{dia}_{\mathtt{D}}(\alpha)$, $\alpha \in [0, \pi]$

- Drei Winkelbereiche!!

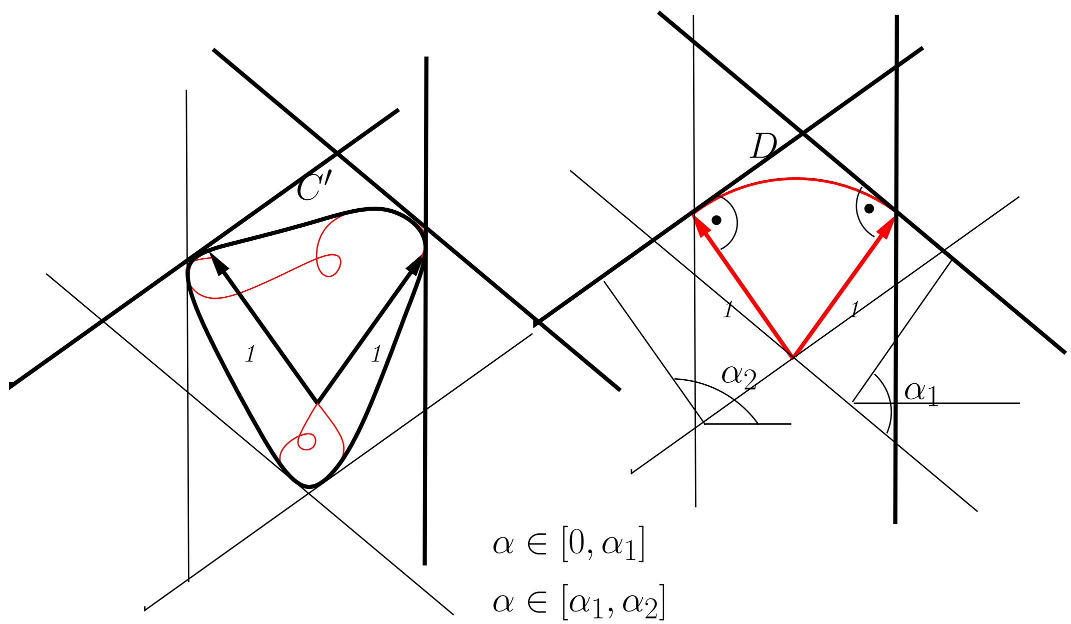


$\alpha \in [0, \alpha_1]$

$\alpha \in [\alpha_1, \alpha_2]$

Beweis: $\mathtt{dia}_{\mathtt{C'}}(\alpha) \geq \mathtt{dia}_{\mathtt{D}}(\alpha)$, $\alpha \in [0, \pi]$

- Drei Winkelbereiche!!



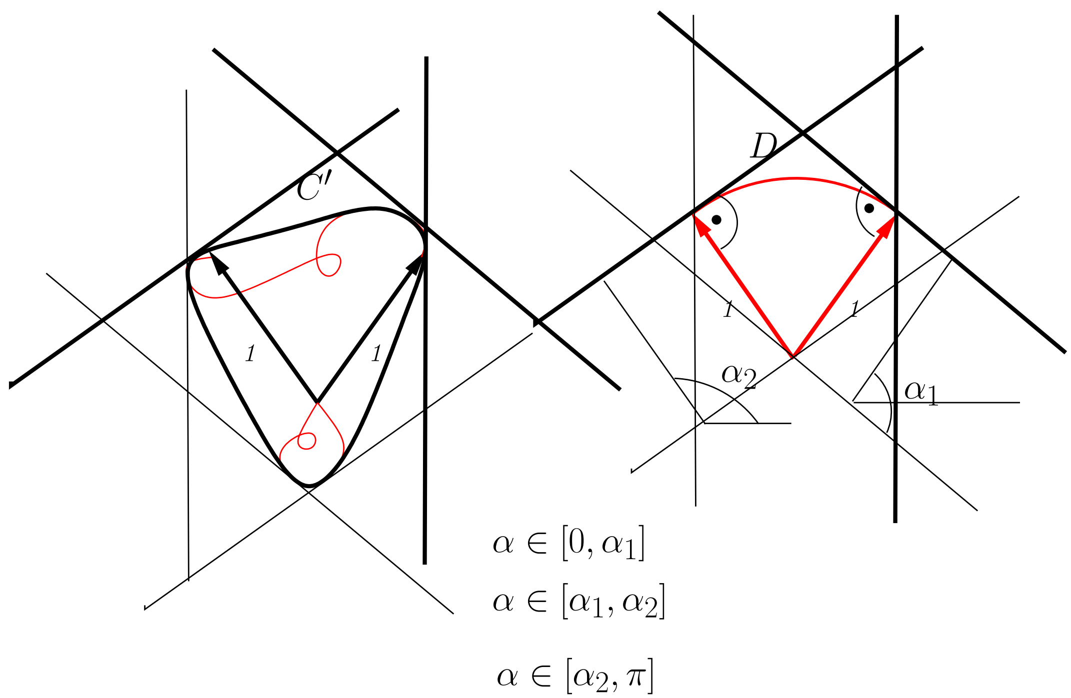
$\alpha \in [0, \alpha_1]$

$\alpha \in [\alpha_1, \alpha_2]$

$\alpha \in [\alpha_2, \pi]$

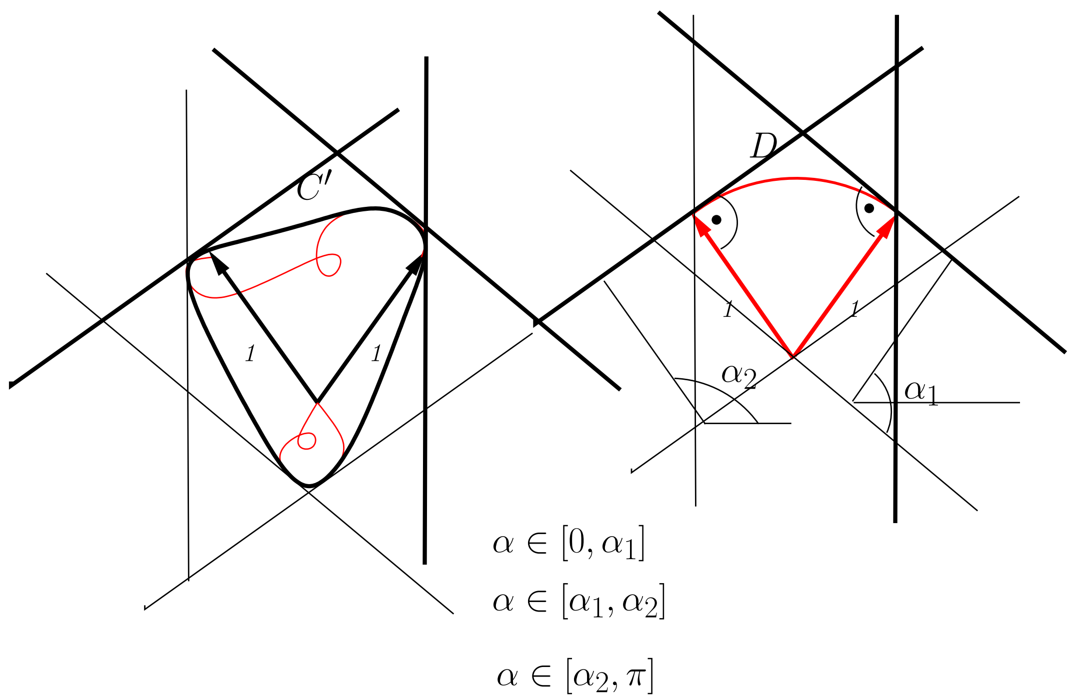
Beweis: $\mathtt{dia}_{C'}(\alpha) \geq \mathtt{dia}_{D}(\alpha)$, $\alpha \in [0, \pi]$

- Drei Winkelbereiche!!
- $\alpha \in [0, \alpha_1] \cup [\alpha_2, \pi] \Rightarrow \mathtt{dia}_{C'}(\alpha) \geq \mathtt{dia}_{D}(\alpha)$



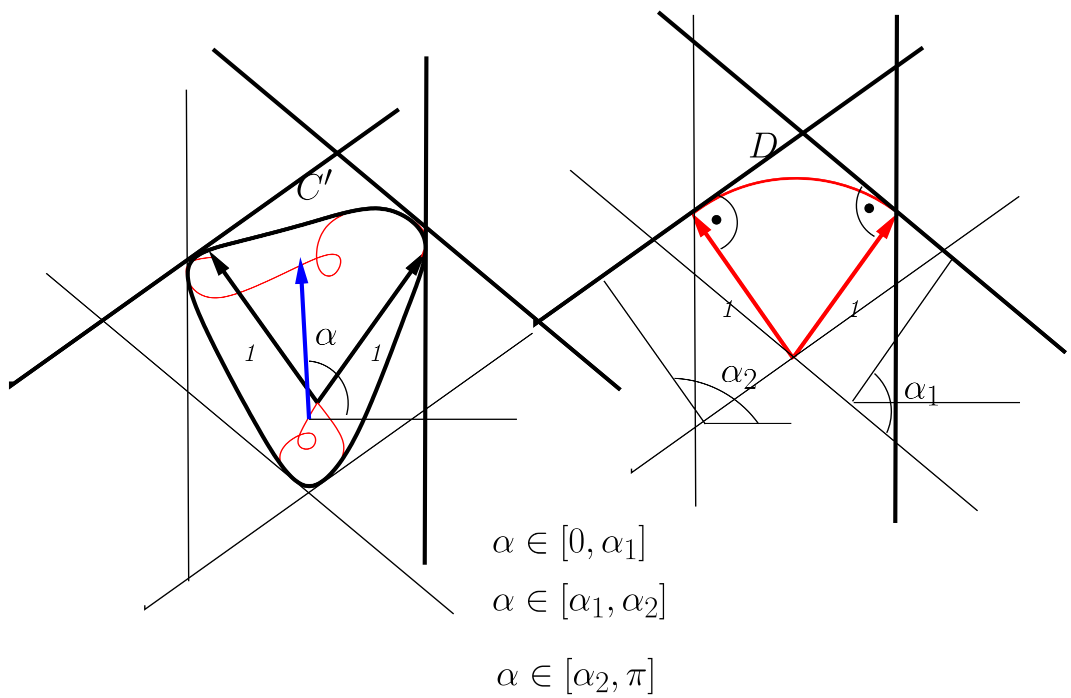
Beweis: $\mathtt{dia}_{C'}(\alpha) \geq \mathtt{dia}_{D}(\alpha)$, $\alpha \in [0,\pi]$

- Drei Winkelbereiche!!
- $\alpha \in [0,\alpha_1] \cup [\alpha_2,\pi] \Rightarrow \mathtt{dia}_{C'}(\alpha) \geq \mathtt{dia}_{D}(\alpha)$
- Für $\alpha \in [\alpha_1,\alpha_2]$ ex. stets Platzierung (x,y,α) in C'



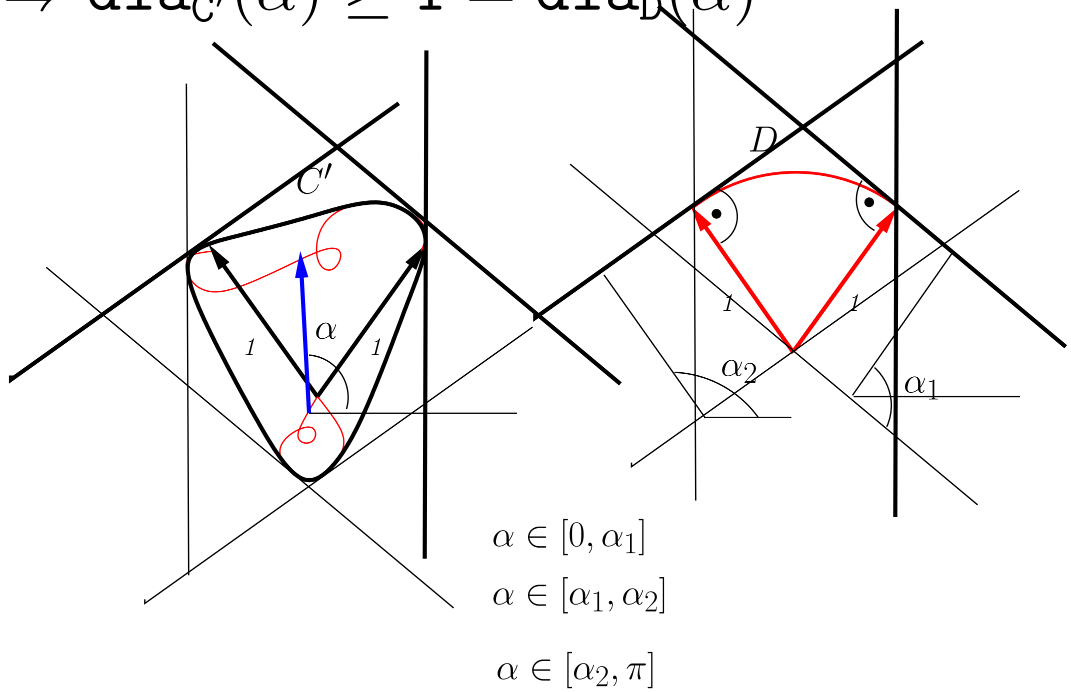
Beweis: $\mathtt{dia}_{\mathtt{C}'}(\alpha) \geq \mathtt{dia}_{\mathtt{D}}(\alpha)$, $\alpha \in [0, \pi]$

- Drei Winkelbereiche!!
- $\alpha \in [0, \alpha_1] \cup [\alpha_2, \pi] \Rightarrow \mathtt{dia}_{\mathtt{C}'}(\alpha) \geq \mathtt{dia}_{\mathtt{D}}(\alpha)$
- Für $\alpha \in [\alpha_1, \alpha_2]$ ex. stets Platzierung (x, y, α) in C'



Beweis: $\mathtt{dia}_{C'}(\alpha) \geq \mathtt{dia}_{D}(\alpha)$, $\alpha \in [0, \pi]$

- Drei Winkelbereiche!!
- $\alpha \in [0, \alpha_1] \cup [\alpha_2, \pi] \Rightarrow \mathtt{dia}_{C'}(\alpha) \geq \mathtt{dia}_{D}(\alpha)$
- Für $\alpha \in [\alpha_1, \alpha_2]$ ex. stets Platzierung (x, y, α) in C'
- $\alpha \in [\alpha_1, \alpha_2] \Rightarrow \mathtt{dia}_{C'}(\alpha) \geq 1 = \mathtt{dia}_{D}(\alpha)$



Kombinationen Th. 1.47

Zwischen je zwei Positionen von Liniensegmenten gibt es eine optimale Bewegung von einem der folgenden Typen:

1. maximal drei Rotationen,
2. maximal zwei Rotationen und eine geradlinige Bewegung,
3. eine Rotation zwischen zwei geradlinigen Bewegungen.

Die Bewegungen lassen sich effizient berechnen.

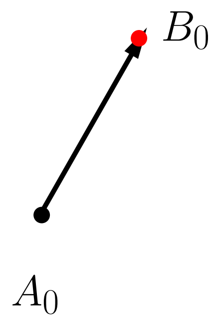
Beispiel: Nahe Placements Th. 1.47

Beispiel: Nahe Placements Th. 1.47

- Abstände $|A_0 - B_1|$ und $|A_1 - B_0|$ kleiner gleich 1

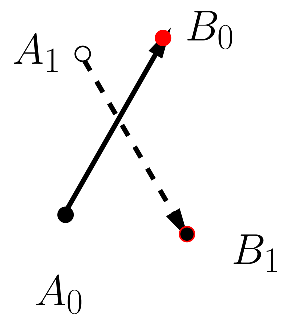
Beispiel: Nahe Placements Th. 1.47

- Abstände $|A_0 - B_1|$ und $|A_1 - B_0|$ kleiner gleich 1
- Drei Rotationen!!



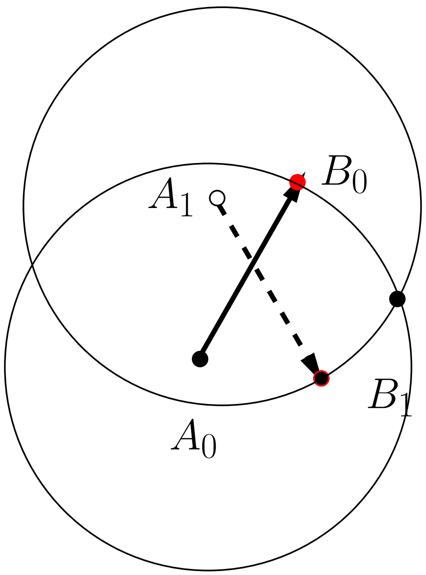
Beispiel: Nahe Placements Th. 1.47

- Abstände $|A_0 - B_1|$ und $|A_1 - B_0|$ kleiner gleich 1
- Drei Rotationen!!



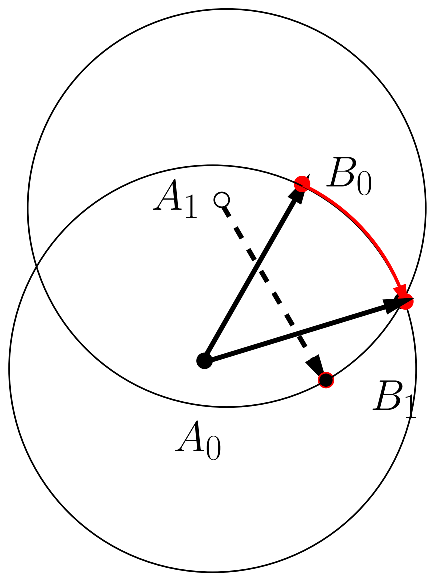
Beispiel: Nahe Placements Th. 1.47

- Abstände $|A_0 - B_1|$ und $|A_1 - B_0|$ kleiner gleich 1
- Drei Rotationen!!



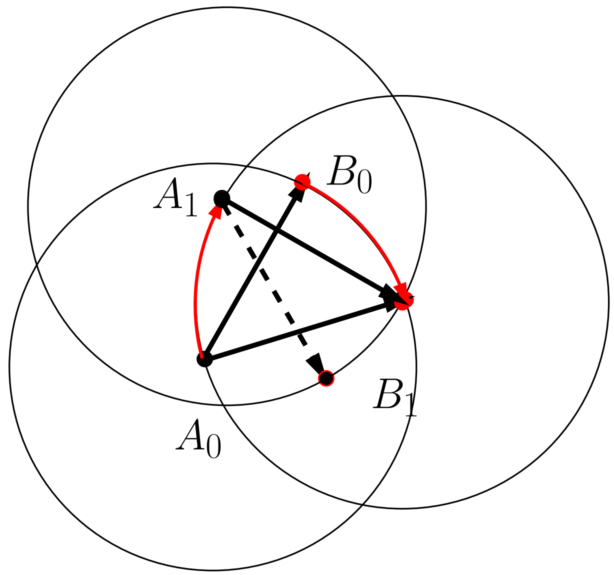
Beispiel: Nahe Placements Th. 1.47

- Abstände $|A_0 - B_1|$ und $|A_1 - B_0|$ kleiner gleich 1
- Drei Rotationen!!



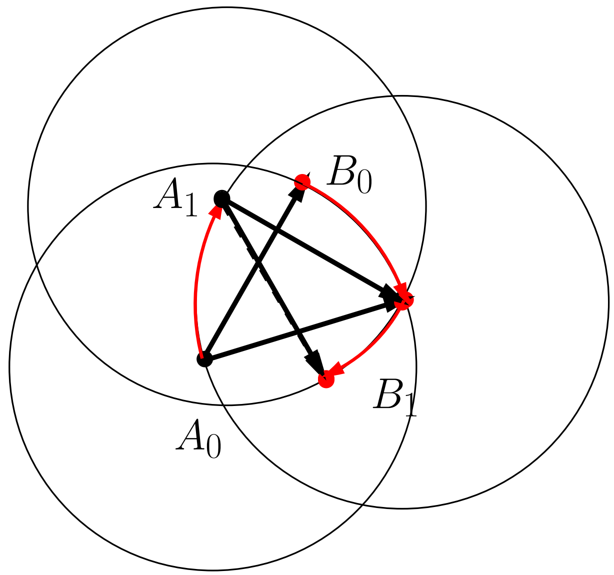
Beispiel: Nahe Placements Th. 1.47

- Abstände $|A_0 - B_1|$ und $|A_1 - B_0|$ kleiner gleich 1
- Drei Rotationen!!



Beispiel: Nahe Placements Th. 1.47

- Abstände $|A_0 - B_1|$ und $|A_1 - B_0|$ kleiner gleich 1
- Drei Rotationen!!



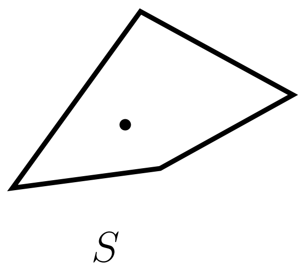
Bewegungen mit ausgedehnten Objekten

Bewegungen mit ausgedehnten Objekten

- Schwierig zu lösen

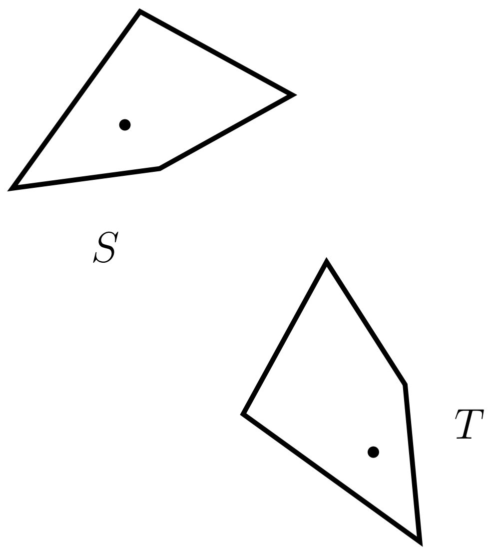
Bewegungen mit ausgedehnten Objekten

- Schwierig zu lösen



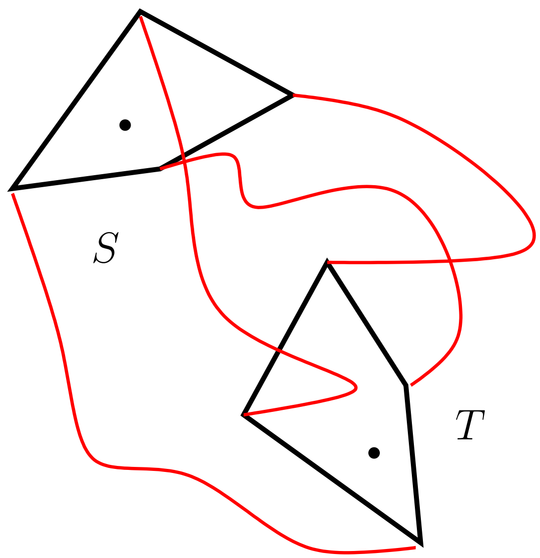
Bewegungen mit ausgedehnten Objekten

- Schwierig zu lösen



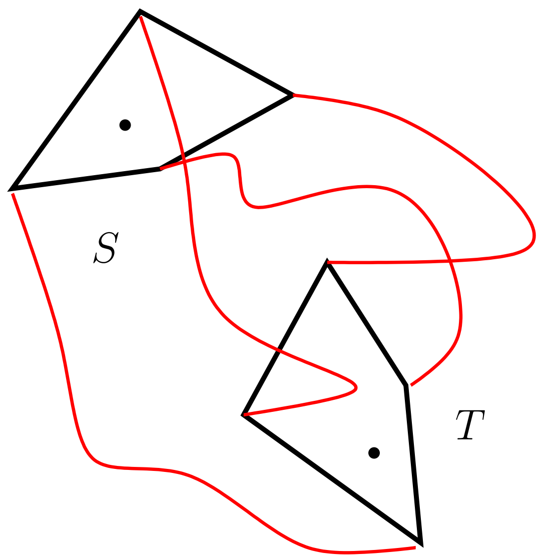
Bewegungen mit ausgedehnten Objekten

- Schwierig zu lösen
- Bewegungen über Referenzpunkt



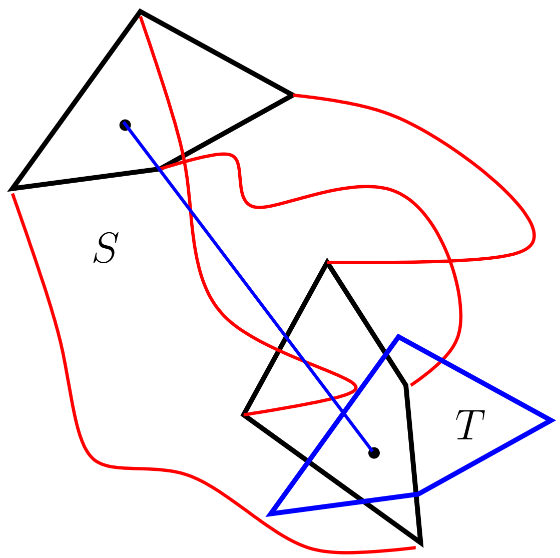
Bewegungen mit ausgedehnten Objekten

- Schwierig zu lösen
- Bewegungen über Referenzpunkt



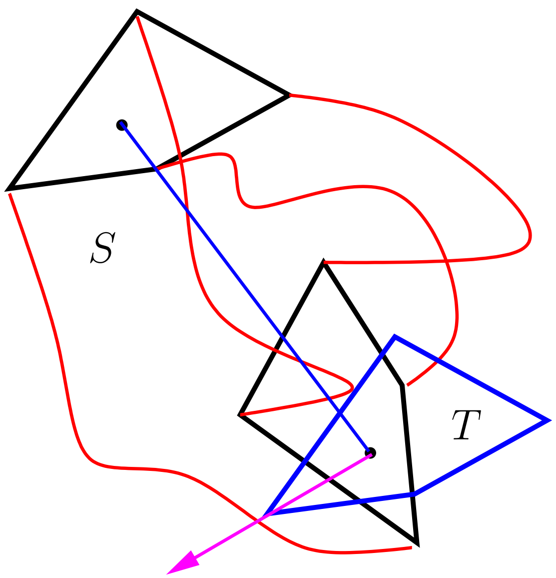
Bewegungen mit ausgedehnten Objekten

- Schwierig zu lösen
- Bewegungen über Referenzpunkt



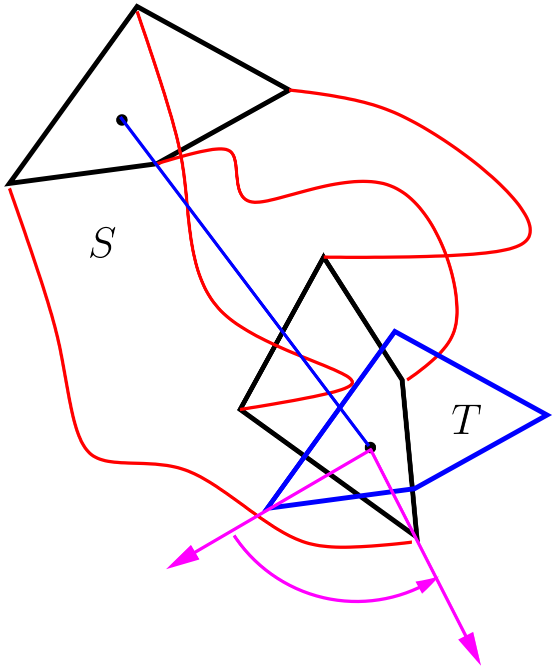
Bewegungen mit ausgedehnten Objekten

- Schwierig zu lösen
- Bewegungen über Referenzpunkt



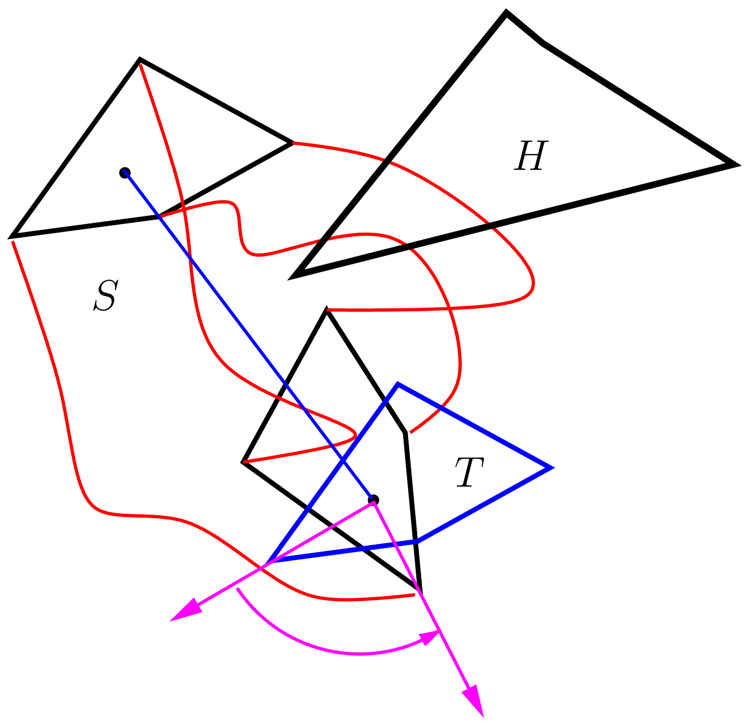
Bewegungen mit ausgedehnten Objekten

- Schwierig zu lösen
- Bewegungen über Referenzpunkt



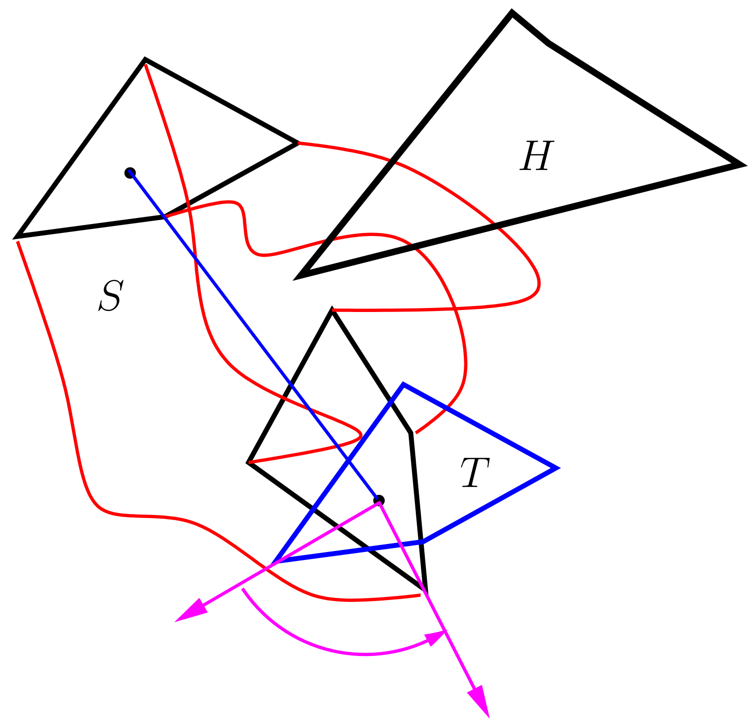
Bewegungen mit ausgedehnten Objekten

- Schwierig zu lösen
- Bewegungen über Referenzpunkt



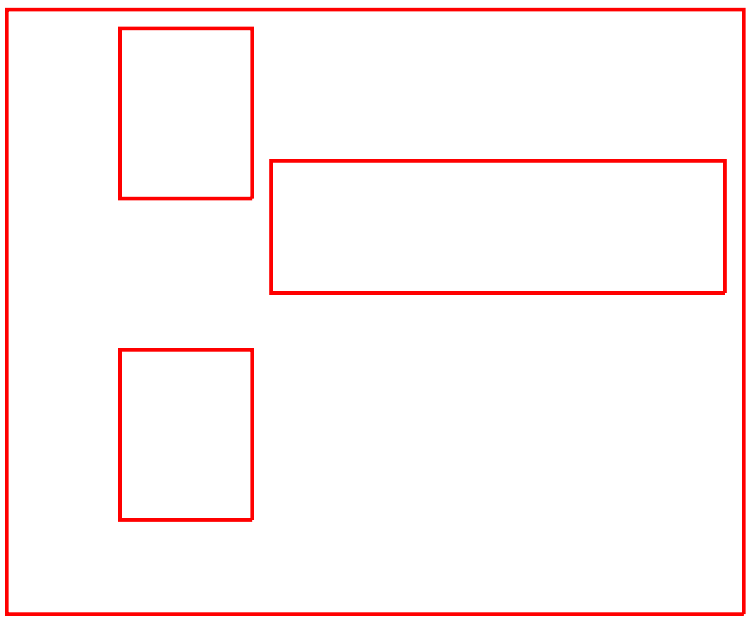
Bewegungen mit ausgedehnten Objekten

- Schwierig zu lösen
- Bewegungen über Referenzpunkt
- Kollisionsfrei??

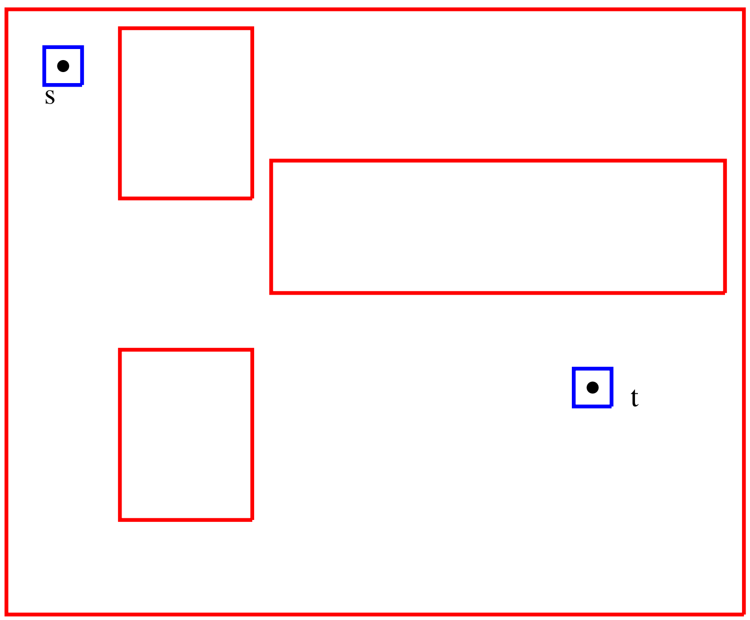


Kollisionsfreie Bahnen: Startbeispiel!!

Kollisionsfreie Bahnen: Startbeispiel!!



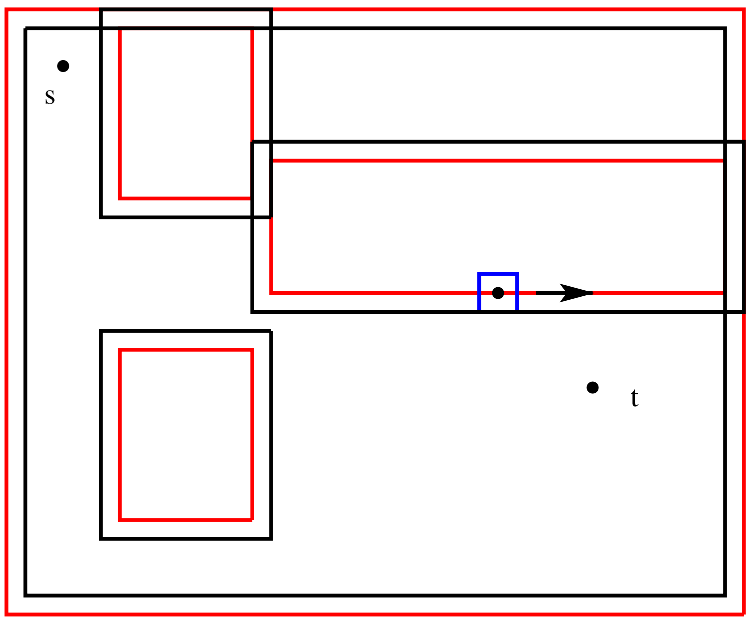
Kollisionsfreie Bahnen: Startbeispiel!!



Translation: Verschieben Referenzpunkt

Aufblasen der Hindernisse

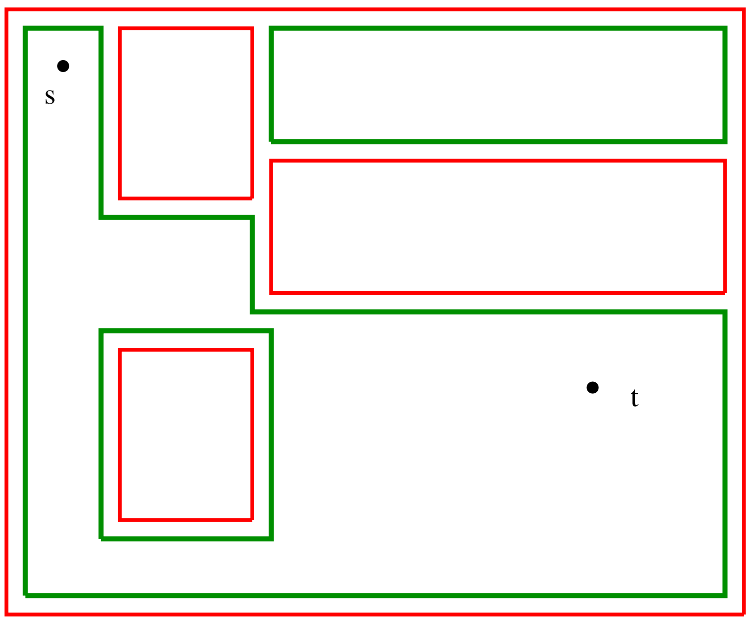
Kollisionsfreie Bahnen: Startbeispiel!!



Translation: Verschieben Referenzpunkt

Aufblasen der Hindernisse

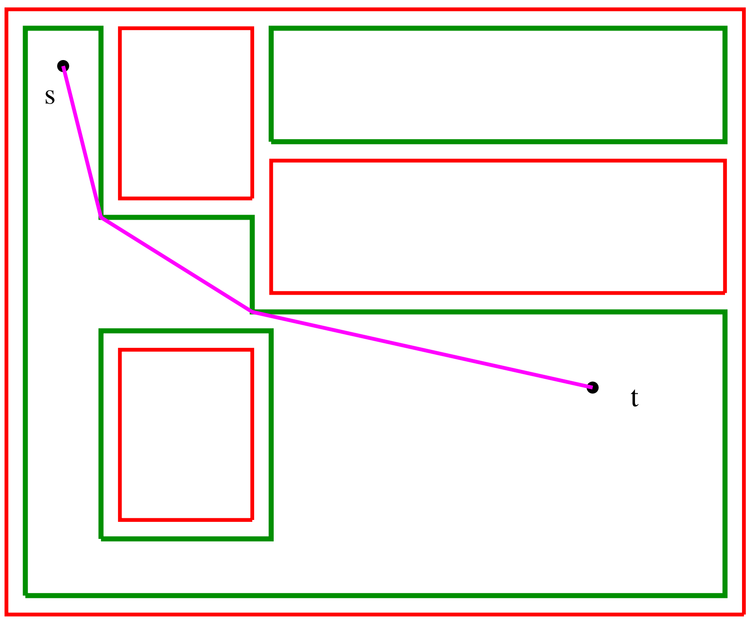
Kollisionsfreie Bahnen: Startbeispiel!!



Translation: Verschieben Referenzpunkt

Aufblasen der Hindernisse
Vereinigung Sweep

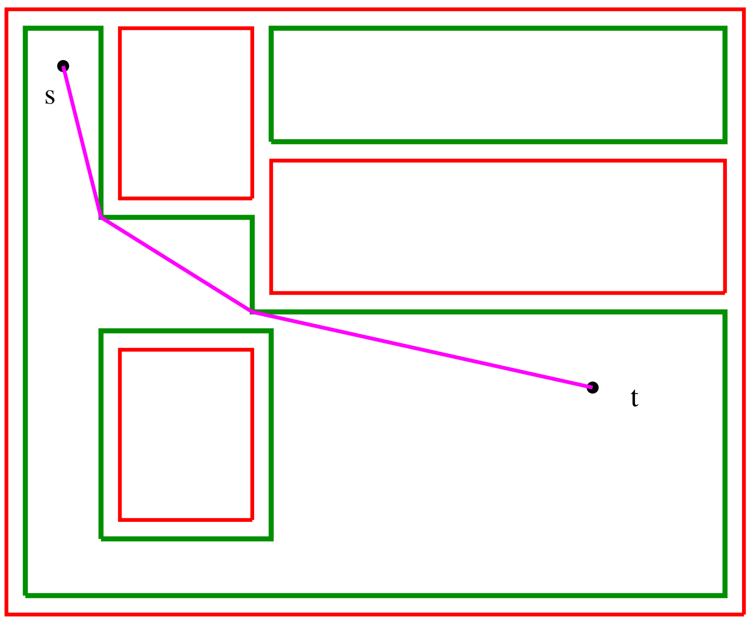
Kollisionsfreie Bahnen: Startbeispiel!!



Translation: Verschieben Referenzpunkt

Aufblasen der Hindernisse
Vereinigung Sweep
Kürzester Weg????

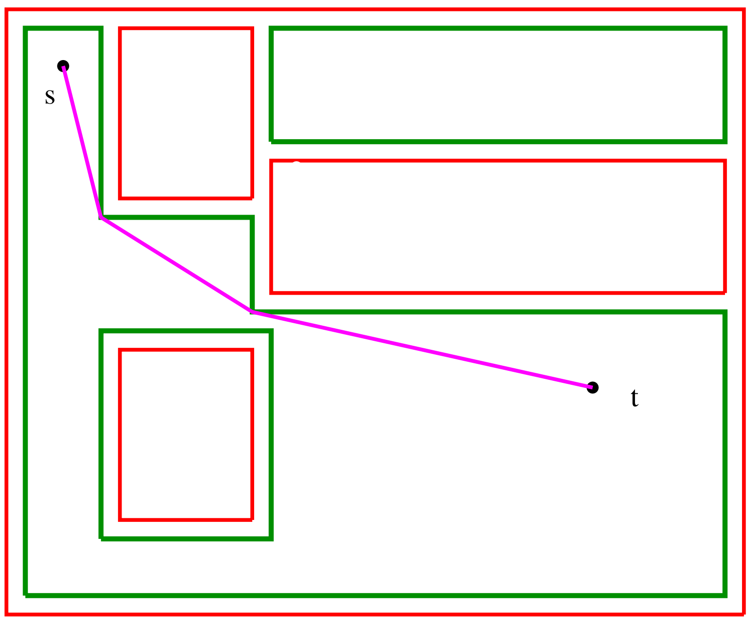
Kollisionsfreie Bahnen: Startbeispiel!!



Translation: Verschieben Referenzpunkt

Aufblasen der Hindernisse
Vereinigung Sweep
Kürzester Weg????

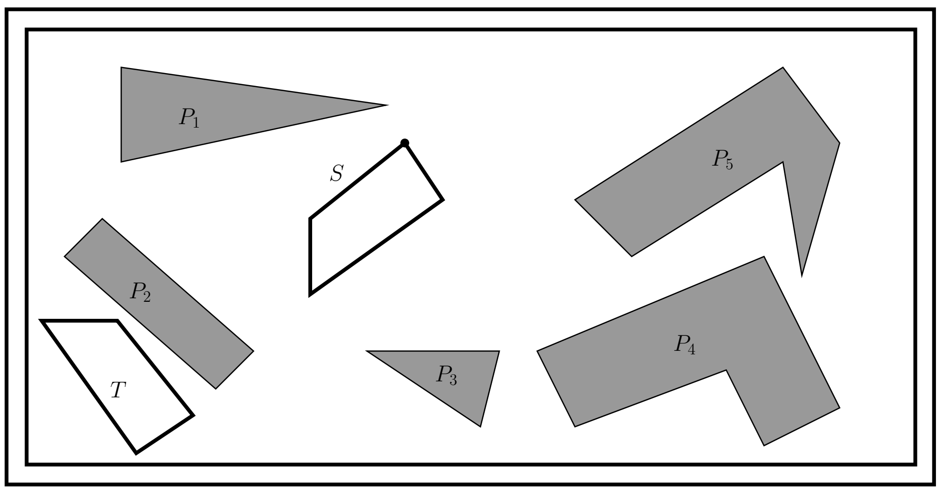
Kollisionsfreie Bahnen: Startbeispiel!!



Translation: Verschieben Referenzpunkt

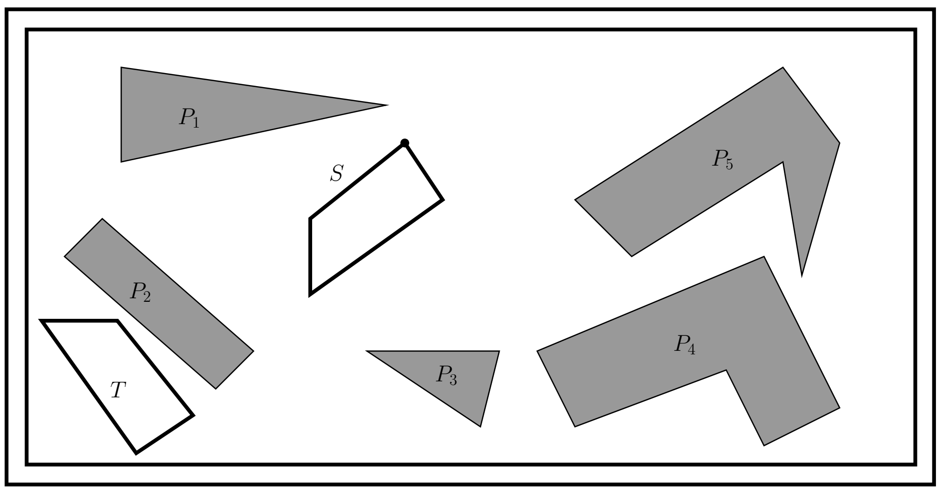
Aufblasen der Hindernisse
Vereinigung Sweep
Kürzester Weg????

Kollisionsfreie Bahnen: Allgemeiner!!



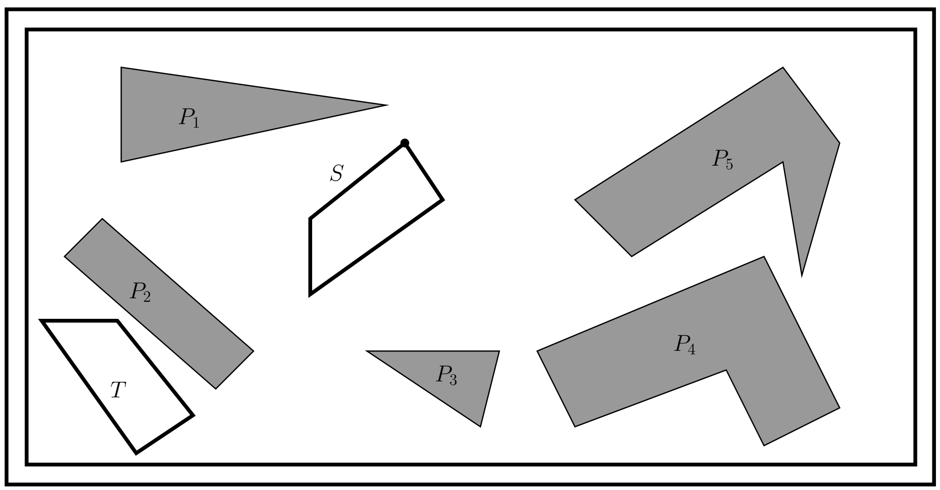
Kollisionsfreie Bahnen: Allgemeiner!!

- Existiert Weg von S nach T



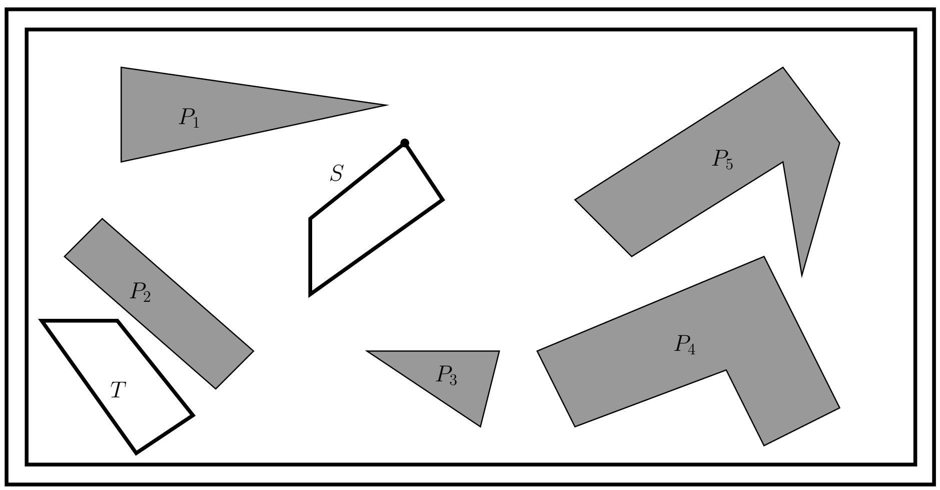
Kollisionsfreie Bahnen: Allgemeiner!!

- Existiert Weg von S nach T
- Berechne Weg von S nach T



Kollisionsfreie Bahnen: Allgemeiner!!

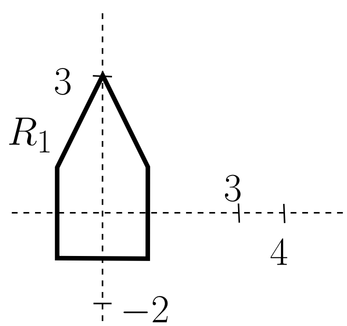
- Existiert Weg von S nach T
- Berechne Weg von S nach T
- Kürzester Weg



Formal: Rotation und Translation

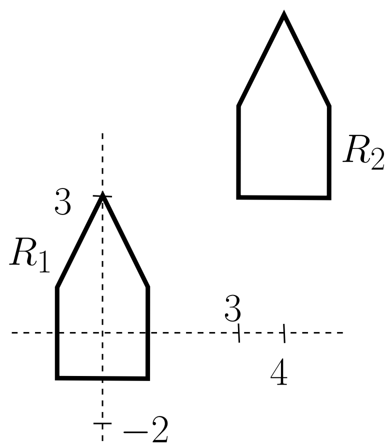
Formal: Rotation und Translation

- Platzierung Endpunkte
  $R_1 = \{(1,-1),(1,1),(0,3),(-1,1),(-1,-1)\}$



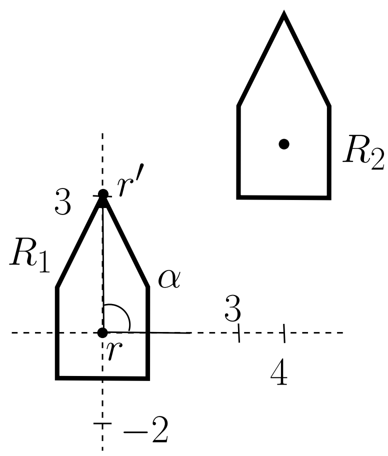
Formal: Rotation und Translation

- Platzierung Endpunkte
  $R_1 = \{(1,-1),(1,1),(0,3),(-1,1),(-1,-1)\}$
- Verschiebung $R_2 = \{(5,3),(5,5),(4,7),(3,5),(3,3)\}$



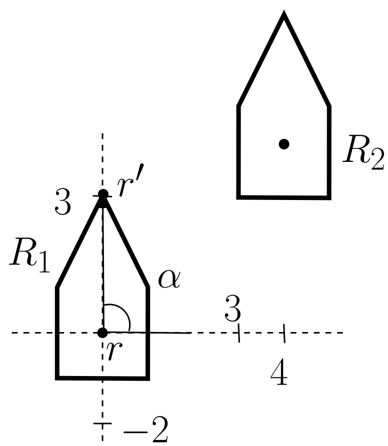
Formal: Rotation und Translation

- Platzierung Endpunkte
  $R_1 = \{(1,-1),(1,1),(0,3),(-1,1),(-1,-1)\}$
- Verschiebung $R_2 = \{(5,3),(5,5),(4,7),(3,5),(3,3)\}$
- Referenzpunkt r



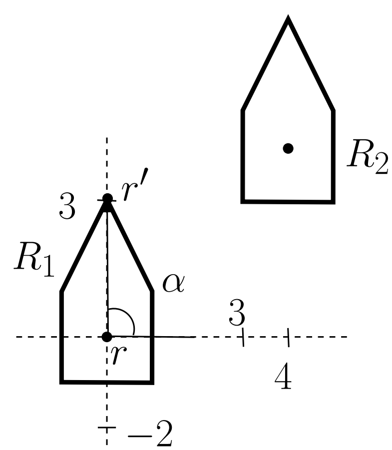
Formal: Rotation und Translation

- Platzierung Endpunkte
  $R_1 = \{(1,-1),(1,1),(0,3),(-1,1),(-1,-1)\}$
- Verschiebung $R_2 = \{(5,3),(5,5),(4,7),(3,5),(3,3)\}$
- Referenzpunkt r Winkelreferenzpunkt r'



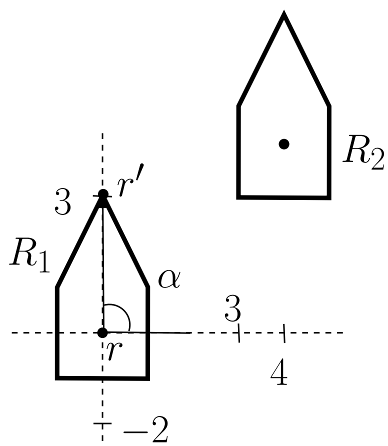
Formal: Rotation und Translation

- Platzierung Endpunkte
  $R_1 = \{(1,-1),(1,1),(0,3),(-1,1),(-1,-1)\}$
- Verschiebung $R_2 = \{(5,3),(5,5),(4,7),(3,5),(3,3)\}$
- Referenzpunkt r Winkelreferenzpunkt r'
- $R_1 = R(0,0,\pi/2) = \{(1,-1),(1,1),(0,3),(-1,1),(-1,-1)\}$



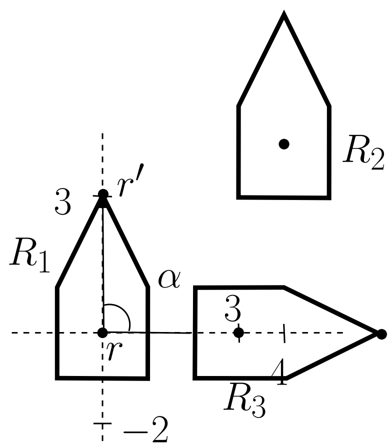
Formal: Rotation und Translation

- Platzierung Endpunkte
  $R_1 = \{(1,-1),(1,1),(0,3),(-1,1),(-1,-1)\}$
- Verschiebung $R_2 = \{(5,3),(5,5),(4,7),(3,5),(3,3)\}$
- Referenzpunkt r Winkelreferenzpunkt r'
- $R_1 = R(0,0,\pi/2) = \{(1,-1),(1,1),(0,3),(-1,1),(-1,-1)\}$
- $R(x,y,\alpha)$,



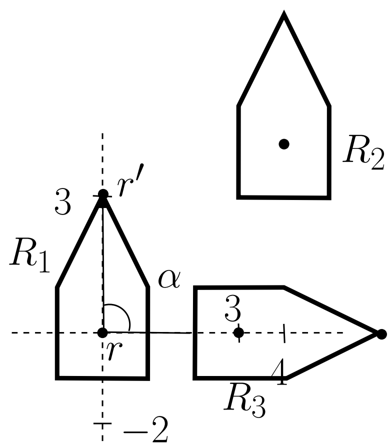
Formal: Rotation und Translation

- Platzierung Endpunkte
  $R_1 = \{(1,-1),(1,1),(0,3),(-1,1),(-1,-1)\}$
- Verschiebung $R_2 = \{(5,3),(5,5),(4,7),(3,5),(3,3)\}$
- Referenzpunkt r Winkelreferenzpunkt r'
- $R_1 = R(0,0,\pi/2) = \{(1,-1),(1,1),(0,3),(-1,1),(-1,-1)\}$
- $R(x,y,\alpha)$, Translation/Rotation:



Formal: Rotation und Translation

- Platzierung Endpunkte
  $R_1 = \{(1,-1),(1,1),(0,3),(-1,1),(-1,-1)\}$
- Verschiebung $R_2 = \{(5,3),(5,5),(4,7),(3,5),(3,3)\}$
- Referenzpunkt r Winkelreferenzpunkt r'
- $R_1 = R(0,0,\pi/2) = \{(1,-1),(1,1),(0,3),(-1,1),(-1,-1)\}$
- $R(x,y,\alpha)$, Translation/Rotation: $R_3 = R(3,0,0)$



Formale Definitionen: Def. 2.1

Formale Definitionen: Def. 2.1

- Tripel $(x, y, \alpha) \in \mathbb{R} \times \mathbb{R} \times [0, 2\pi)$ Platzierung, 3 DOF (Freiheitsgrade) Referenzpunkt+Winkel!!!

Formale Definitionen: Def. 2.1

- Tripel $(x, y, \alpha) \in \mathbb{R} \times \mathbb{R} \times [0, 2\pi)$ Platzierung, 3 DOF (Freiheitsgrade) Referenzpunkt+Winkel!!!
- Konfigurationsraum $C = \mathbb{R} \times \mathbb{R} \times [0, 2\pi)$

Formale Definitionen: Def. 2.1

- Tripel $(x, y, \alpha) \in \mathbb{R} \times \mathbb{R} \times [0, 2\pi)$ Platzierung, 3 DOF (Freiheitsgrade) Referenzpunkt+Winkel!!!
- Konfigurationsraum $C = \mathbb{R} \times \mathbb{R} \times [0, 2\pi)$
- Arbeitsraum: $R(x, y, \alpha)$, $P_1, P_2, \ldots, P_k$

Formale Definitionen: Def. 2.1

- Tripel $(x, y, \alpha) \in \mathbb{R} \times \mathbb{R} \times [0, 2\pi)$ Platzierung, 3 DOF (Freiheitsgrade) Referenzpunkt+Winkel!!!
- Konfigurationsraum $C = \mathbb{R} \times \mathbb{R} \times [0, 2\pi)$
- Arbeitsraum: $R(x, y, \alpha)$, $P_1, P_2, \ldots, P_k$
- Verbotene Platzierungen $C_{\text{verb}} := \left\{ c \in C | R(c) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$

Formale Definitionen: Def. 2.1

- Tripel $(x, y, \alpha) \in \mathbb{R} \times \mathbb{R} \times [0, 2\pi)$ Platzierung, 3 DOF (Freiheitsgrade) Referenzpunkt+Winkel!!!
- Konfigurationsraum $C = \mathbb{R} \times \mathbb{R} \times [0, 2\pi)$
- Arbeitsraum: $R(x, y, \alpha)$, $P_1, P_2, \ldots, P_k$
- Verbotene Platzierungen $C_{\text{verb}} := \left\{ c \in C | R(c) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Freie Platzierungen $C_{\text{frei}} := C \setminus C_{\text{verb}}$

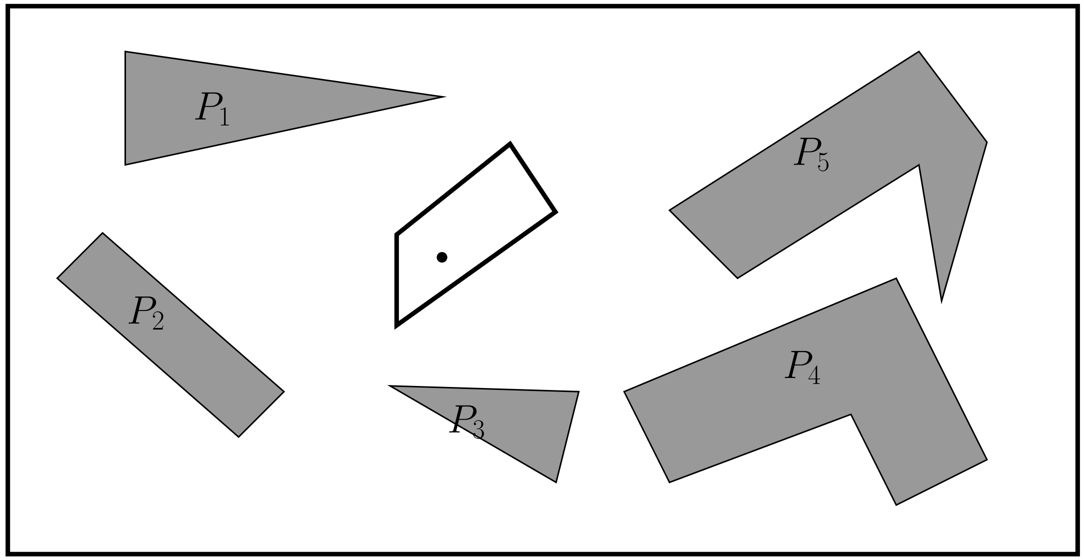
Formale Definitionen: Def. 2.1

- Tripel $(x, y, \alpha) \in \mathbb{R} \times \mathbb{R} \times [0, 2\pi)$ Platzierung, 3 DOF (Freiheitsgrade) Referenzpunkt+Winkel!!!
- Konfigurationsraum $C = \mathbb{R} \times \mathbb{R} \times [0, 2\pi)$
- Arbeitsraum: $R(x, y, \alpha)$, $P_1, P_2, \ldots, P_k$
- Verbotene Platzierungen $C_{\text{verb}} := \left\{ c \in C | R(c) \cap \bigcup \overset{\circ}{P}_i \neq \emptyset \right\}$
- Freie Platzierungen $C_{\text{frei}} := C \setminus C_{\text{verb}}$
- Halbfreie Platzierungen

Beispiel!!

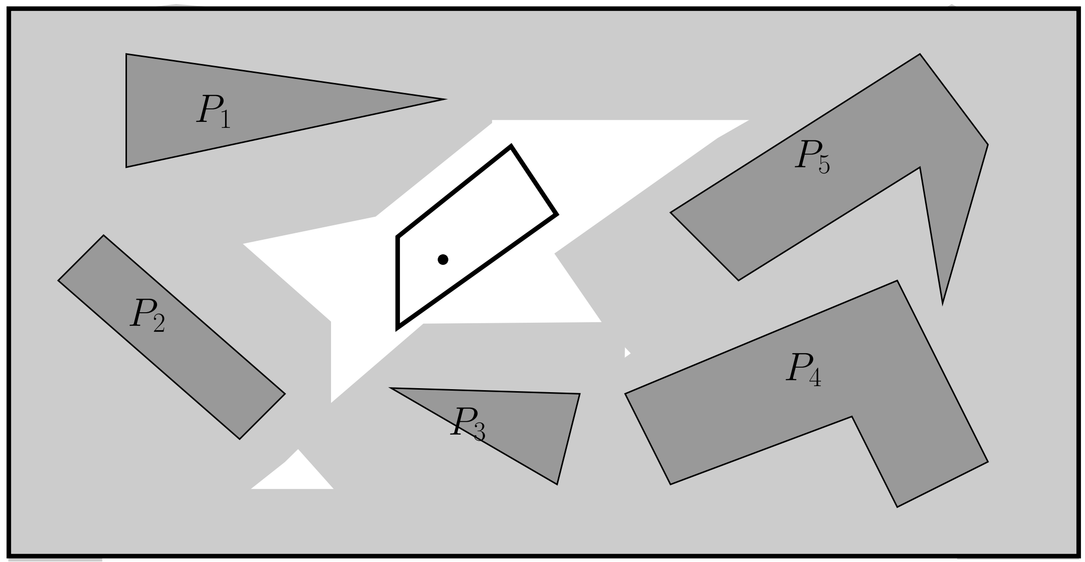
- Reine Translation
- Bereich der gültigen Platzierungen

Beispiel!!



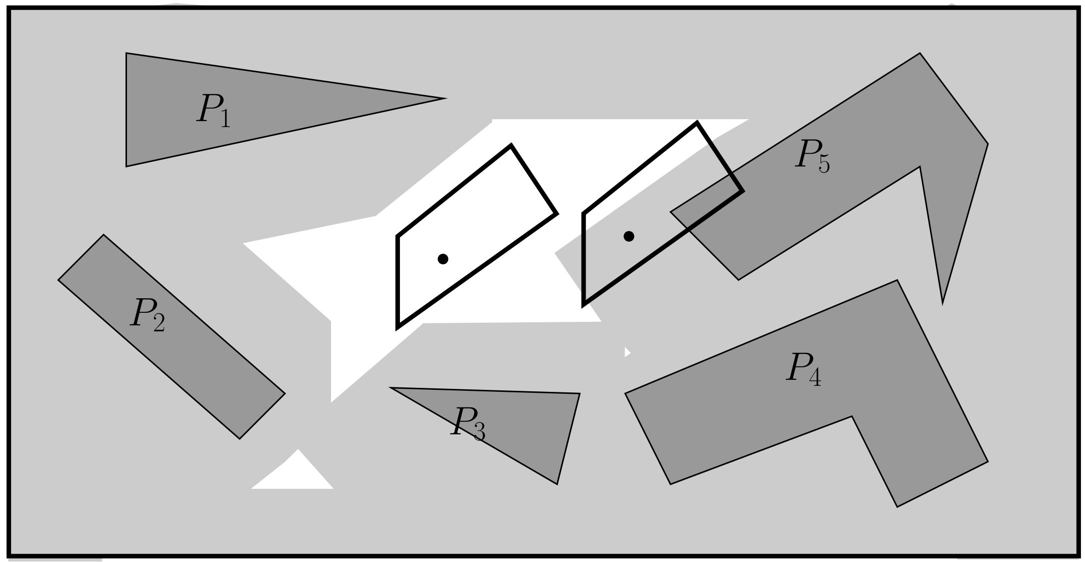
- Reine Translation
- Bereich der gültigen Platzierungen

Beispiel!!



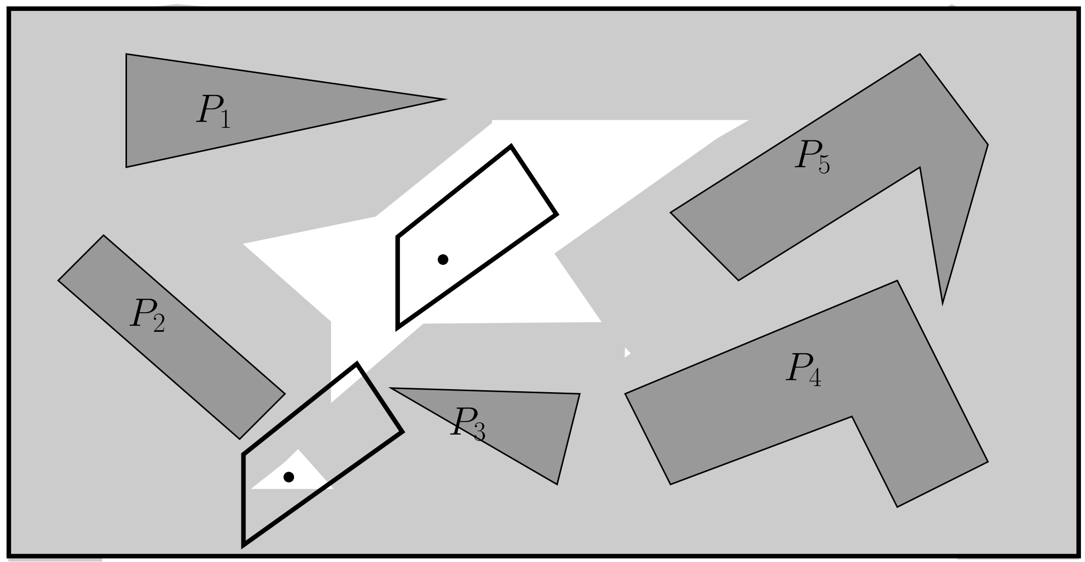
- Reine Translation
- Bereich der gültigen Platzierungen

Beispiel!!



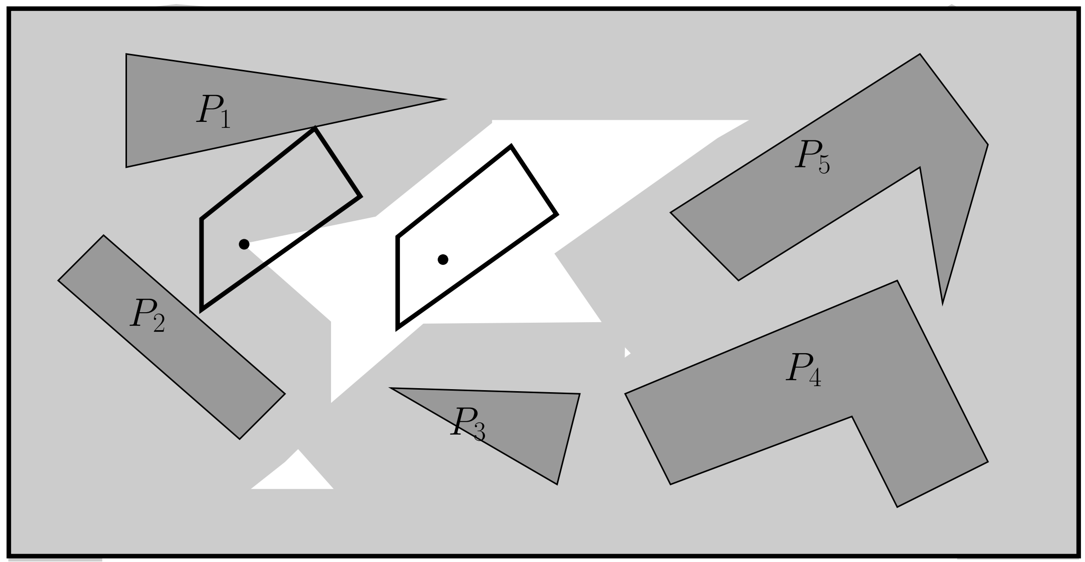
- Reine Translation
- Bereich der gültigen Platzierungen

Beispiel!!



- Reine Translation
- Bereich der gültigen Platzierungen

Beispiel!!



- Reine Translation
- Bereich der gültigen Platzierungen

Weg-zusammenhängend Def. 2.2

Weg-zusammenhängend Def. 2.2

- Menge Z ($Z \subseteq \mathbb{R}^d$)

Weg-zusammenhängend Def. 2.2

- Menge Z ($Z \subseteq \mathbb{R}^d$)
- Pfad π von a nach b:

Weg-zusammenhängend Def. 2.2

- Menge Z ($Z \subseteq \mathbb{R}^d$)
- Pfad π von a nach b:
  - Stetige Abbildung $\pi : [0,1] \to Z$

Weg-zusammenhängend Def. 2.2

- Menge Z ($Z \subseteq \mathbb{R}^d$)
- Pfad π von a nach b:
  - Stetige Abbildung $\pi : [0,1] \to Z$
  - $\pi(0) = a$, $\pi(1) = b$

Weg-zusammenhängend Def. 2.2

- Menge Z ($Z \subseteq \mathbb{R}^d$)
- Pfad π von a nach b:
  - Stetige Abbildung $\pi : [0,1] \to Z$
  - $\pi(0) = a$, $\pi(1) = b$
- weg-zusammenhängend:

Weg-zusammenhängend Def. 2.2

- Menge Z ($Z \subseteq \mathbb{R}^d$)
- Pfad π von a nach b:
  - Stetige Abbildung $\pi : [0,1] \to Z$
  - $\pi(0) = a$, $\pi(1) = b$
- weg-zusammenhängend:
  - $\forall a, b \in Z : \exists$ Pfad π von a nach b in Z.

Zusammenhangskomponenten Lemma 2.3

Zusammenhangskomponenten Lemma 2.3

- Relation auf $A \times A$ ($A \subseteq \mathbb{R}^d$):

Zusammenhangskomponenten Lemma 2.3

- Relation auf $A \times A$ ($A \subseteq \mathbb{R}^d$): $a \sim b :\Leftrightarrow \exists$ Pfad $\pi \subset A$ von a nach b in A mit $\pi(0) = a$ und $\pi(1) = b$

Zusammenhangskomponenten Lemma 2.3

- Relation auf $A \times A$ ($A \subseteq \mathbb{R}^d$): $a \sim b :\Leftrightarrow \exists$ Pfad $\pi \subset A$ von a nach b in A mit $\pi(0) = a$ und $\pi(1) = b$
- Äquivalenzrelation:

Zusammenhangskomponenten Lemma 2.3

- Relation auf $A \times A$ ($A \subseteq \mathbb{R}^d$): $a \sim b :\Leftrightarrow \exists$ Pfad $\pi \subset A$ von a nach b in A mit $\pi(0) = a$ und $\pi(1) = b$
- Äquivalenzrelation:
  - Reflexiv: $a \sim a$

Zusammenhangskomponenten Lemma 2.3

- Relation auf $A \times A$ ($A \subseteq \mathbb{R}^d$): $a \sim b :\Leftrightarrow \exists$ Pfad $\pi \subset A$ von a nach b in A mit $\pi(0) = a$ und $\pi(1) = b$
- Äquivalenzrelation:
  - Reflexiv: $a \sim a$
  - Symmetrisch: $a \sim b \Rightarrow b \sim a$

Zusammenhangskomponenten Lemma 2.3

- Relation auf $A \times A$ ($A \subseteq \mathbb{R}^d$): $a \sim b :\Leftrightarrow \exists$ Pfad $\pi \subset A$ von a nach b in A mit $\pi(0) = a$ und $\pi(1) = b$
- Äquivalenzrelation:
  - Reflexiv: $a \sim a$
  - Symmetrisch: $a \sim b \Rightarrow b \sim a$
  - Transitiv: $a \sim b,\ b \sim c \Rightarrow a \sim c$

Zusammenhangskomponenten Lemma 2.3

- Relation auf $A \times A$ ($A \subseteq \mathbb{R}^d$): $a \sim b :\Leftrightarrow \exists$ Pfad $\pi \subset A$ von a nach b in A mit $\pi(0) = a$ und $\pi(1) = b$
- Äquivalenzrelation:
  - Reflexiv: $a \sim a$
  - Symmetrisch: $a \sim b \Rightarrow b \sim a$
  - Transitiv: $a \sim b,\ b \sim c \Rightarrow a \sim c$
- Äquivalenzklassen (Zusammenhangskomponenten)
  - größte weg-zusammenhängende Teilmengen

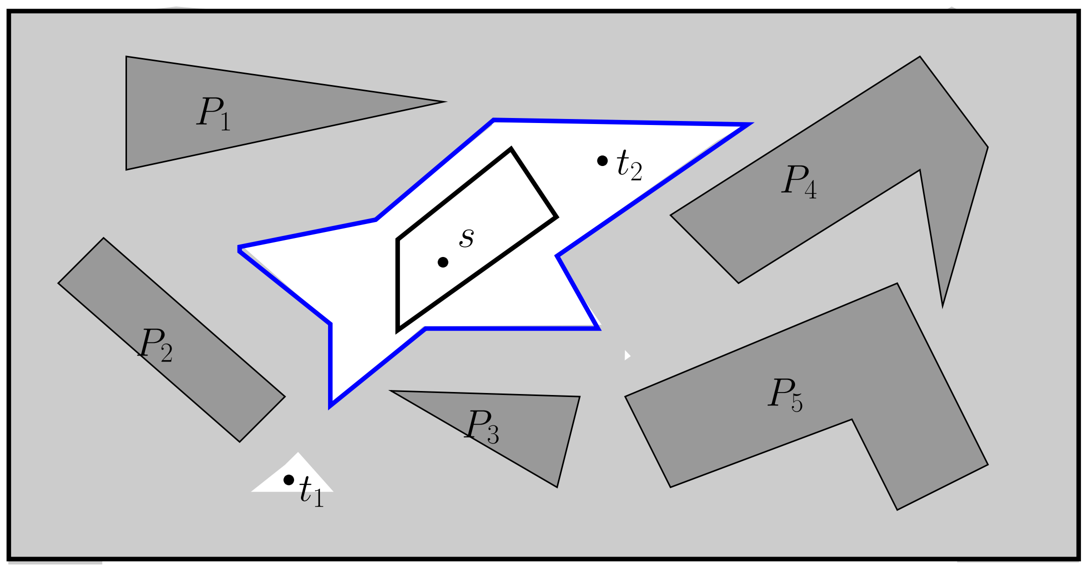
Zusammenhangskomponenten Lemma 2.3

- Relation auf $A \times A$ ($A \subseteq \mathbb{R}^d$): $a \sim b :\Leftrightarrow \exists$ Pfad $\pi \subset A$ von a nach b in A mit $\pi(0) = a$ und $\pi(1) = b$
- Äquivalenzrelation:
  - Reflexiv: $a \sim a$
  - Symmetrisch: $a \sim b \Rightarrow b \sim a$
  - Transitiv: $a \sim b,\ b \sim c \Rightarrow a \sim c$
- Äquivalenzklassen (Zusammenhangskomponenten)
  - größte weg-zusammenhängende Teilmengen
  - überdecken A disjunkt:

Zusammenhangskomponenten Lemma 2.3

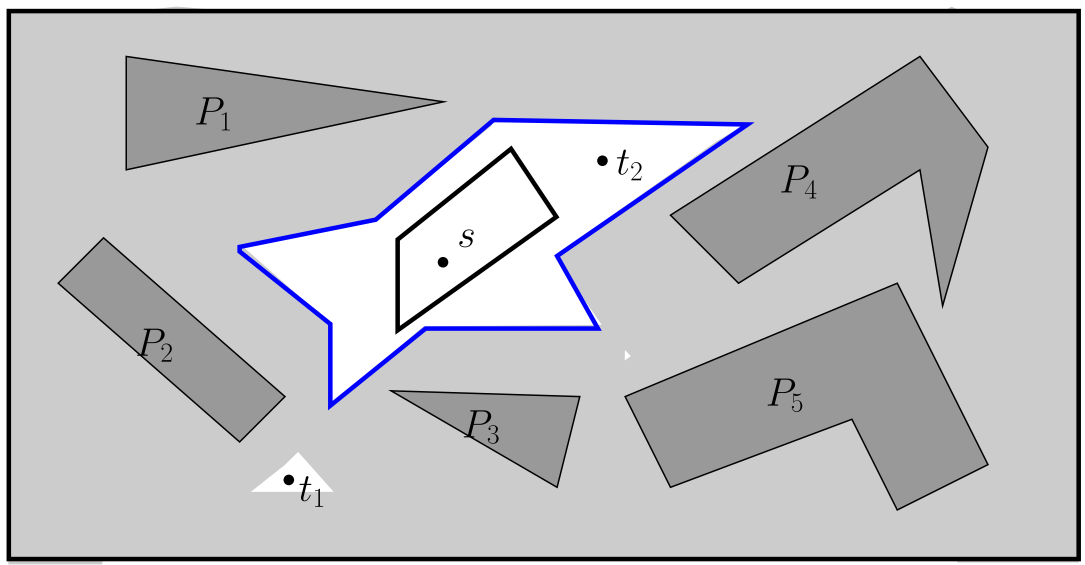
- Relation auf $A \times A$ ($A \subseteq \mathbb{R}^d$): $a \sim b :\Leftrightarrow \exists$ Pfad $\pi \subset A$ von a nach b in A mit $\pi(0) = a$ und $\pi(1) = b$
- Äquivalenzrelation:
  - Reflexiv: $a \sim a$
  - Symmetrisch: $a \sim b \Rightarrow b \sim a$
  - Transitiv: $a \sim b,\ b \sim c \Rightarrow a \sim c$
- Äquivalenzklassen (Zusammenhangskomponenten)
  - größte weg-zusammenhängende Teilmengen
  - überdecken A disjunkt: $A = \dot{\bigcup} Z_i$

Aufgabe: Beispiel Translation!



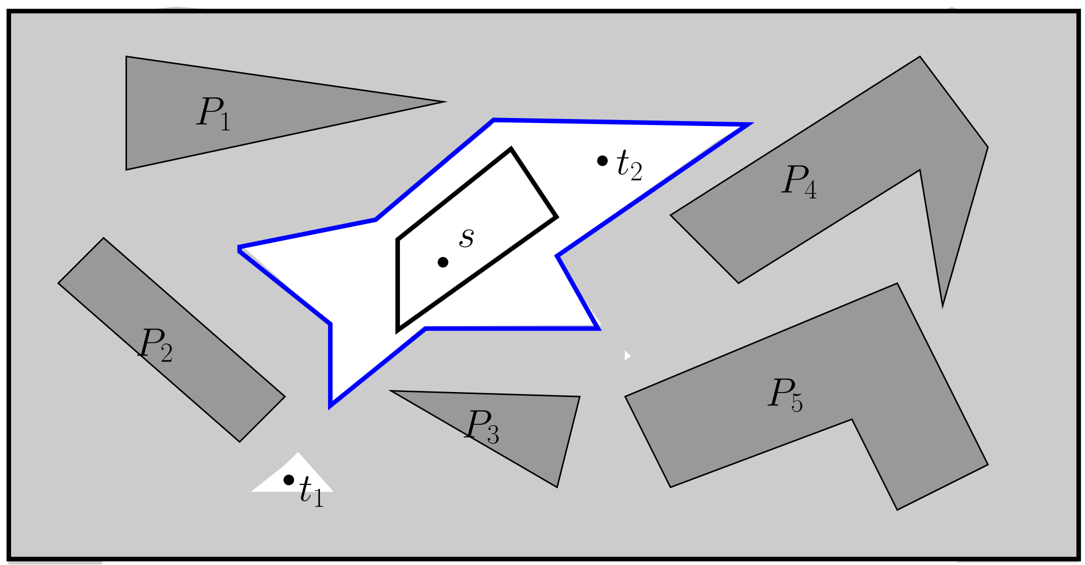
Aufgabe: Beispiel Translation!

- Berechne Zusammenhangskomponente von Z_s, s Startkoordinaten



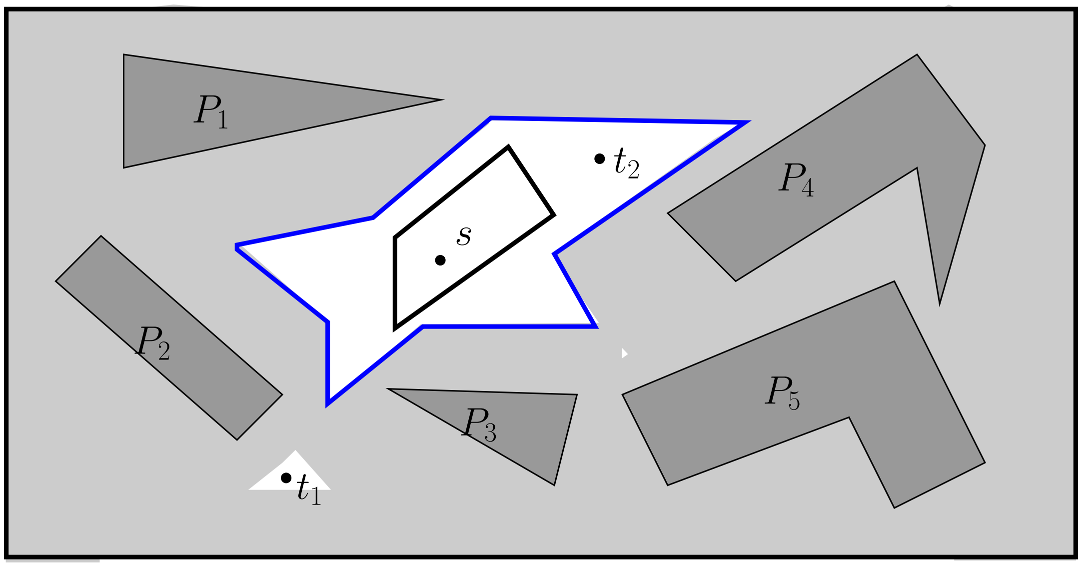
Aufgabe: Beispiel Translation!

- Berechne Zusammenhangskomponente von Z_s, s Startkoordinaten
- Liegt t auch in Z_s?



Aufgabe: Beispiel Translation!

- Berechne Zusammenhangskomponente von Z_s, s Startkoordinaten
- Liegt t auch in Z_s?
- Berechne ggf. Weg s nach t in Z_s



Reine Translationsbewegung, konvexe Roboter 2.1.2

Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$

Reine Translationsbewegung, konvexe Roboter 2.1.2

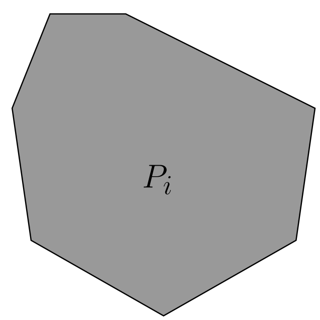
- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$

Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i

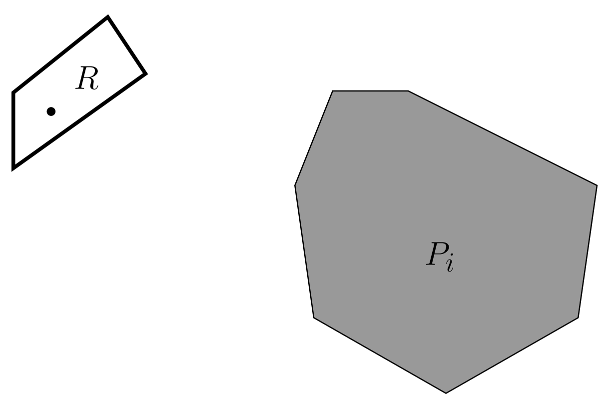
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i



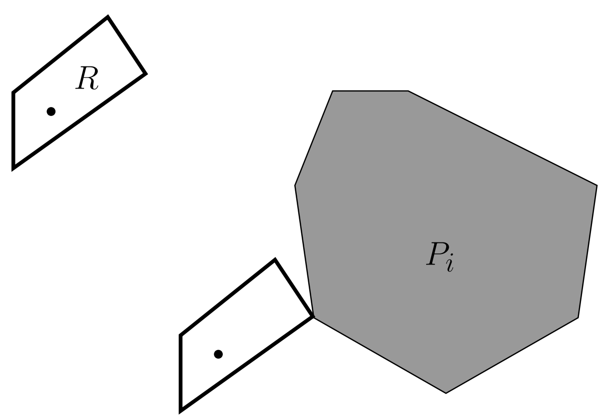
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i



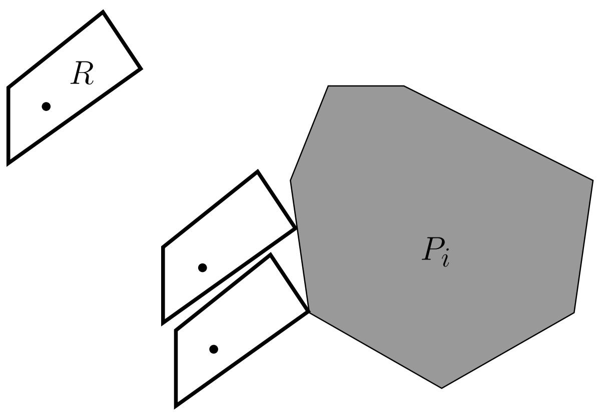
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i



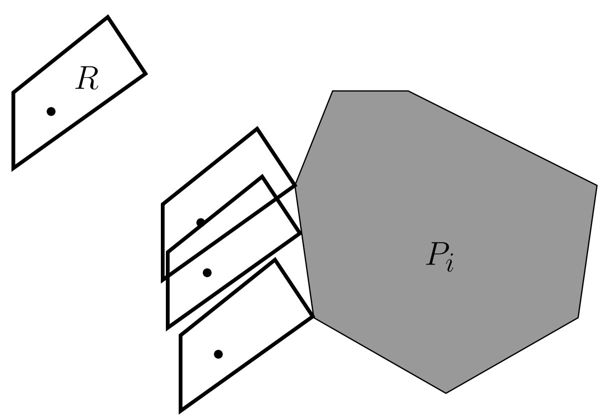
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i



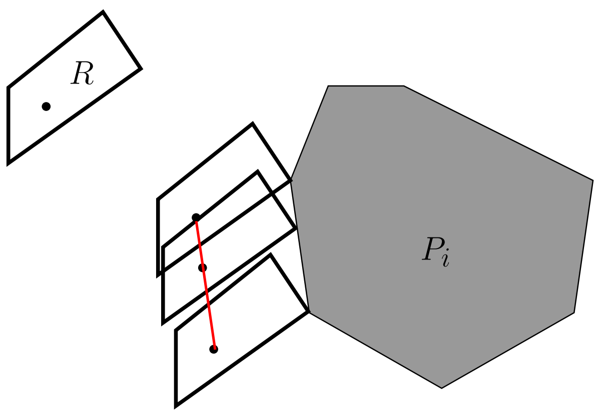
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i



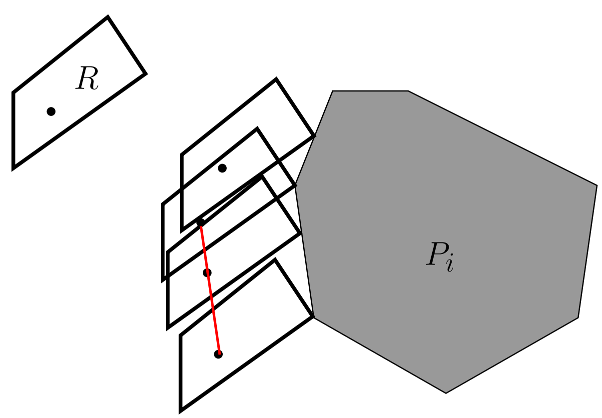
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i



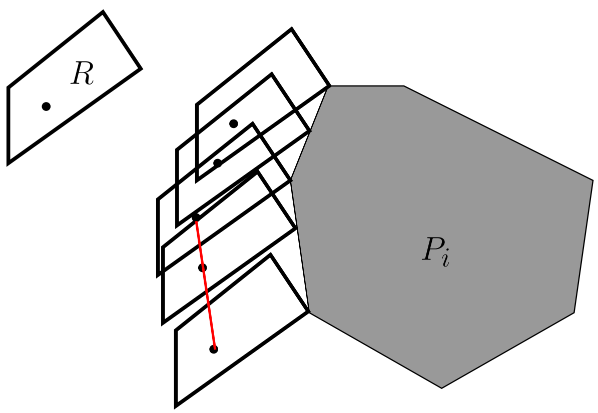
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i



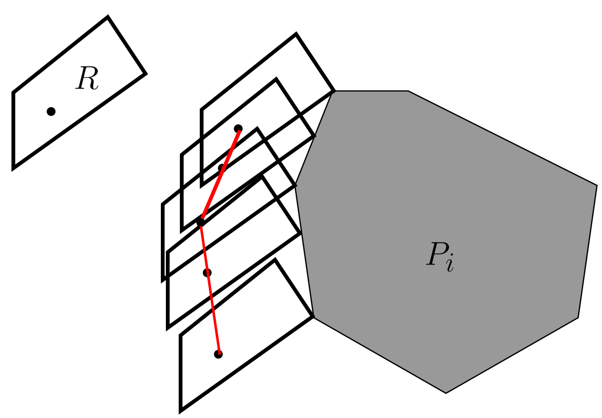
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x, y) \in \mathbb{R}^2 | R(x, y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i



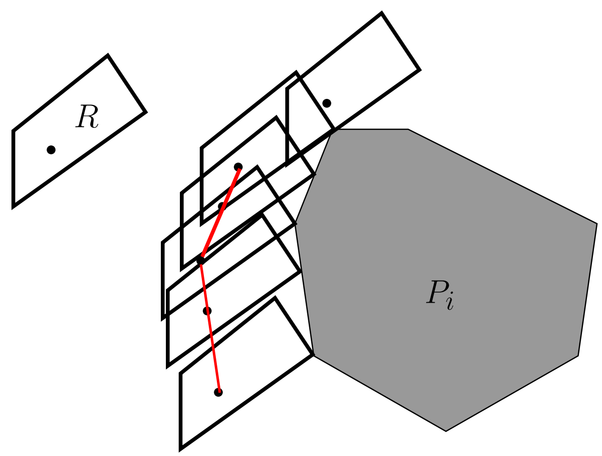
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \overset{\circ}{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i



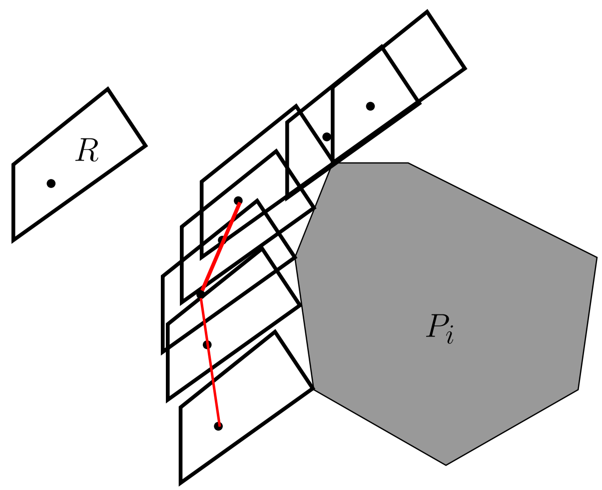
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i



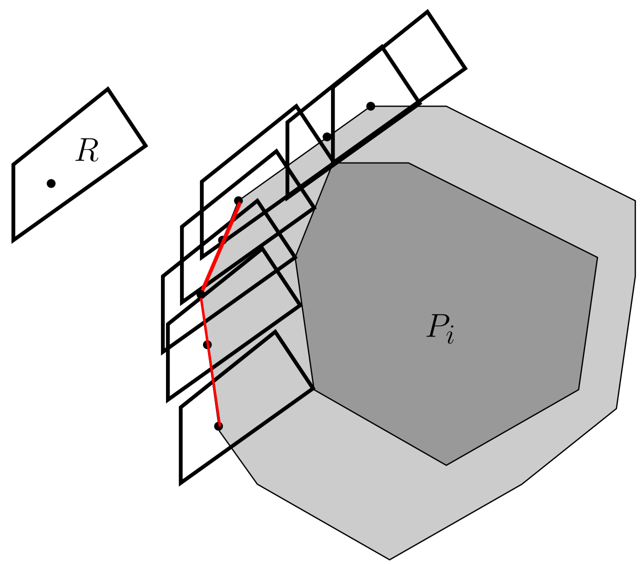
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i



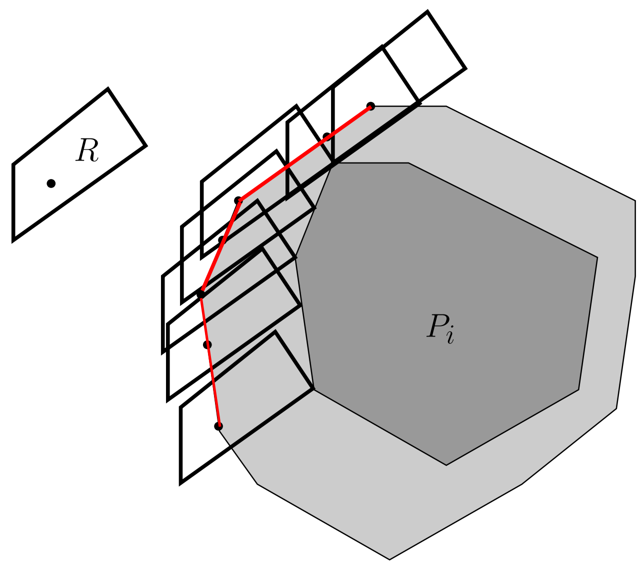
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i
- Konfigurationsraumhindernis
  $CP_i := \{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap P_i \neq \emptyset \}$



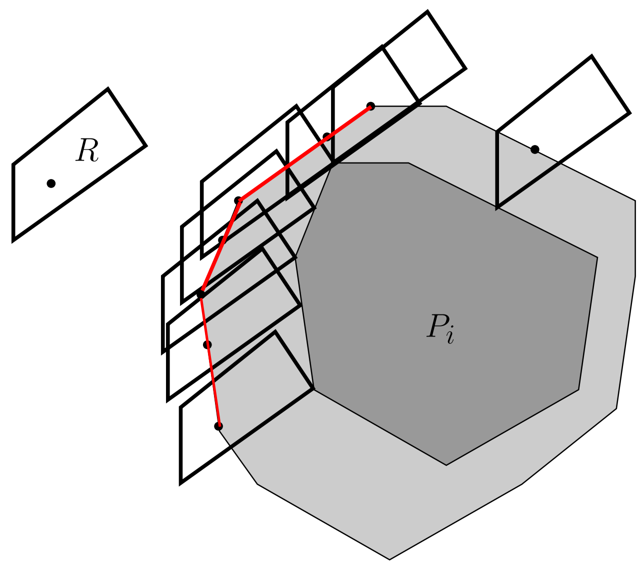
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i
- Konfigurationsraumhindernis
  $CP_i := \{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap P_i \neq \emptyset \}$



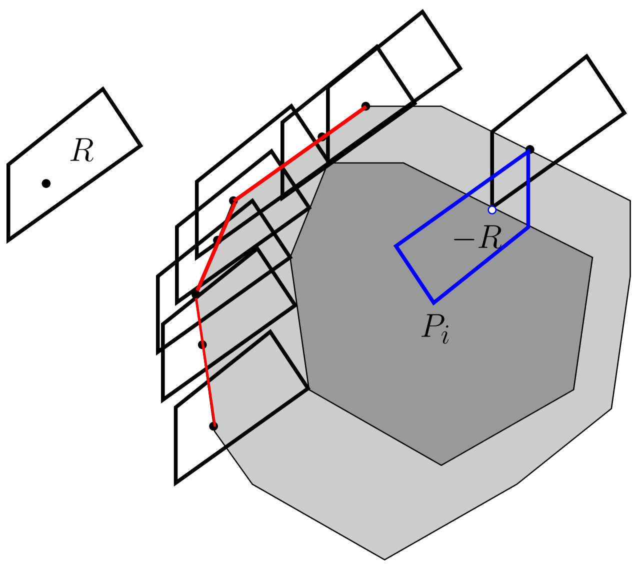
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x, y) \in \mathbb{R}^2 | R(x, y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i
- Konfigurationsraumhindernis
  $CP_i := \left\{ (x, y) \in \mathbb{R}^2 | R(x, y) \cap P_i \neq \emptyset \right\}$



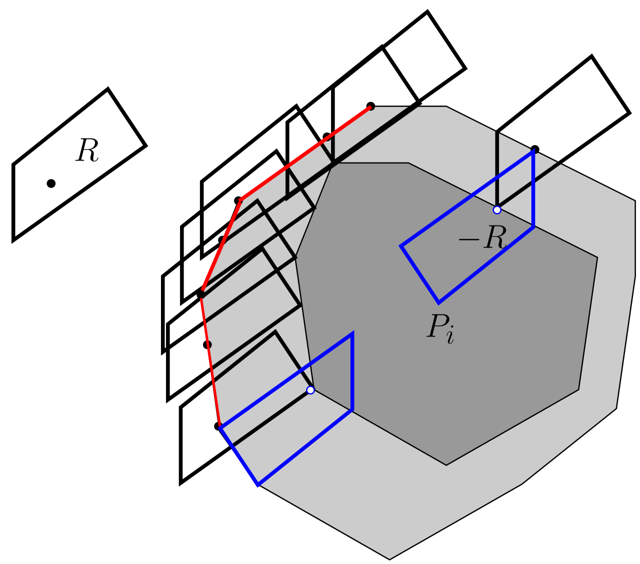
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x, y) \in \mathbb{R}^2 | R(x, y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i
- Konfigurationsraumhindernis
  $CP_i := \{ (x, y) \in \mathbb{R}^2 | R(x, y) \cap P_i \neq \emptyset \}$



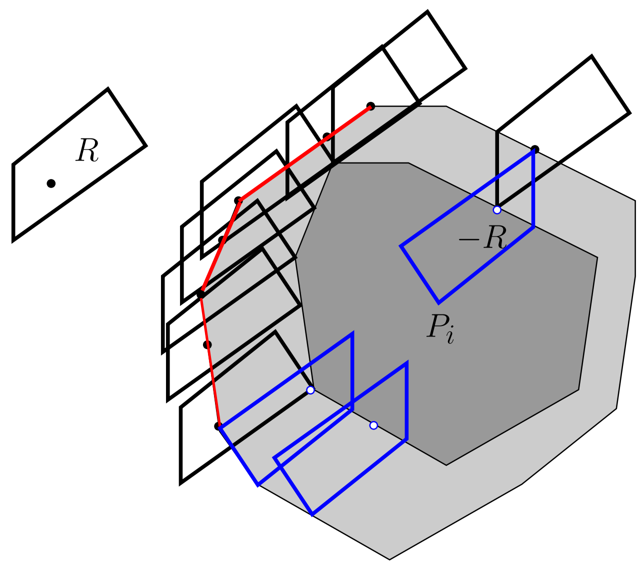
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\mathrm{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i
- Konfigurationsraumhindernis
  $CP_i := \{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap P_i \neq \emptyset \}$



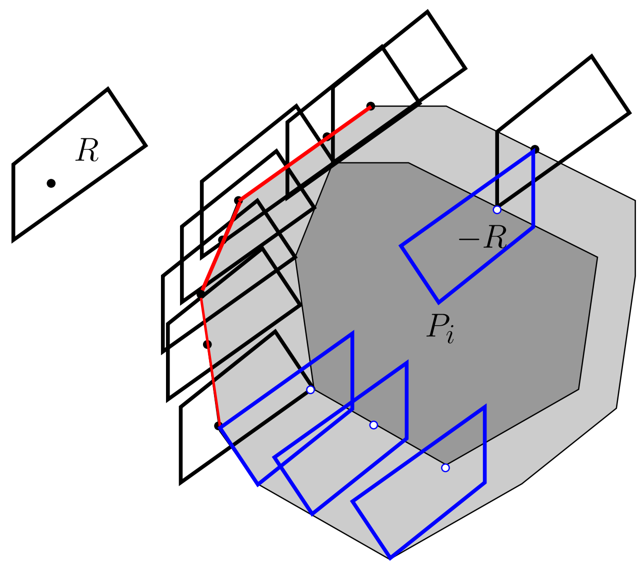
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i
- Konfigurationsraumhindernis
  $CP_i := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap P_i \neq \emptyset \right\}$



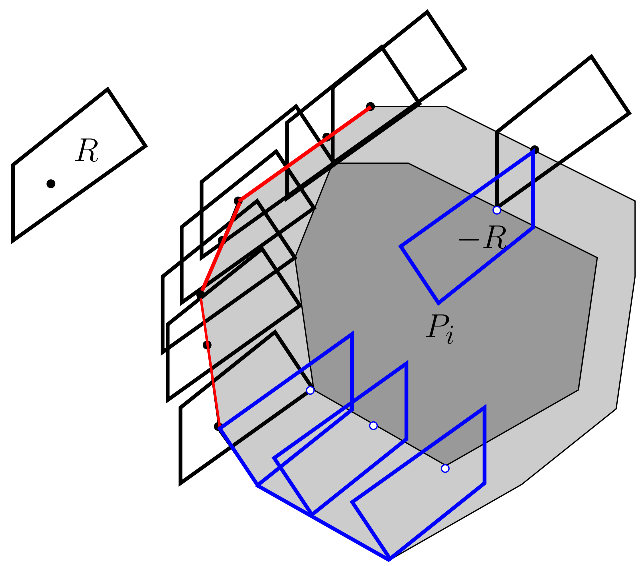
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i
- Konfigurationsraumhindernis
  $CP_i := \{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap P_i \neq \emptyset \}$



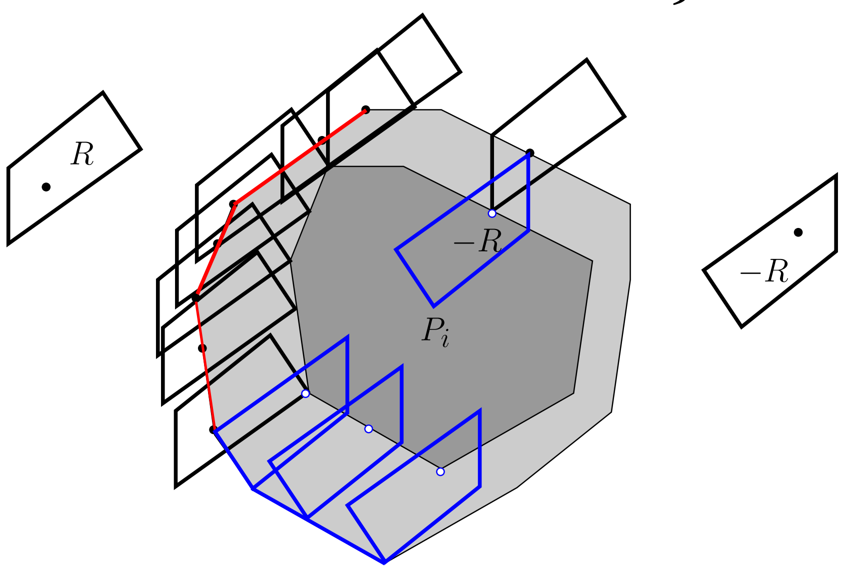
Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i
- Konfigurationsraumhindernis
  $CP_i := \{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap P_i \neq \emptyset \}$



Reine Translationsbewegung, konvexe Roboter 2.1.2

- Konvexer Roboter R, Hindernisse $P_1, P_2, \ldots, P_k$
- Berechne $C_{\text{verb}} := \left\{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap \bigcup \mathring{P}_i \neq \emptyset \right\}$
- Bsp: Für konvexes Polygon P_i
- Konfigurationsraumhindernis
  $CP_i := \{ (x,y) \in \mathbb{R}^2 | R(x,y) \cap P_i \neq \emptyset \}$



Minkowski-Summe Def. 2.4

Minkowski-Summe Def. 2.4

- $A, B \subset \mathbb{R}^2$: $A \oplus B := \{a + b | a \in A, b \in B\}$.

Minkowski-Summe Def. 2.4

- $A, B \subset \mathbb{R}^2$: $A \oplus B := \{a + b | a \in A, b \in B\}$.
- Bem. 2.4:

Minkowski-Summe Def. 2.4

- $A, B \subset \mathbb{R}^2$: $A \oplus B := \{a + b | a \in A, b \in B\}$.
- Bem. 2.4:

  i) kommutativ: $A \oplus B = B \oplus A$

Minkowski-Summe Def. 2.4

- $A, B \subset \mathbb{R}^2$: $A \oplus B := \{a + b | a \in A, b \in B\}$.
- Bem. 2.4:

  i) kommutativ: $A \oplus B = B \oplus A$

  ii) assoziativ: $A \oplus (B \oplus C) = (A \oplus B) \oplus C$

Minkowski-Summe Def. 2.4

- $A, B \subset \mathbb{R}^2$: $A \oplus B := \{a + b | a \in A, b \in B\}$.
- Bem. 2.4:

  i) kommutativ: $A \oplus B = B \oplus A$

  ii) assoziativ: $A \oplus (B \oplus C) = (A \oplus B) \oplus C$

  iii) distributiv bzgl. der Vereinigung:
  $A \oplus (B \cup C) = (A \oplus B) \cup (A \oplus C)$

Minkowski-Summe Def. 2.4

- $A, B \subset \mathbb{R}^2$: $A \oplus B := \{a + b | a \in A, b \in B\}$.
- Bem. 2.4:

  i) kommutativ: $A \oplus B = B \oplus A$

  ii) assoziativ: $A \oplus (B \oplus C) = (A \oplus B) \oplus C$

  iii) distributiv bzgl. der Vereinigung:
  $A \oplus (B \cup C) = (A \oplus B) \cup (A \oplus C)$

Anschaulich!!!

Minkowski-Summe Def. 2.4

- $A, B \subset \mathbb{R}^2$: $A \oplus B := \{a + b | a \in A, b \in B\}$.
- Bem. 2.4:

  i) kommutativ: $A \oplus B = B \oplus A$

  ii) assoziativ: $A \oplus (B \oplus C) = (A \oplus B) \oplus C$

  iii) distributiv bzgl. der Vereinigung:
  $A \oplus (B \cup C) = (A \oplus B) \cup (A \oplus C)$

Anschaulich!!! `MinkSum.html`

Lemma 2.6

Lemma 2.6

$$CP_i := \{\, (x,y) \in \mathbb{R}^2 | R(x,y) \cap P_i \; \neq \; \emptyset \,\}$$

Lemma 2.6

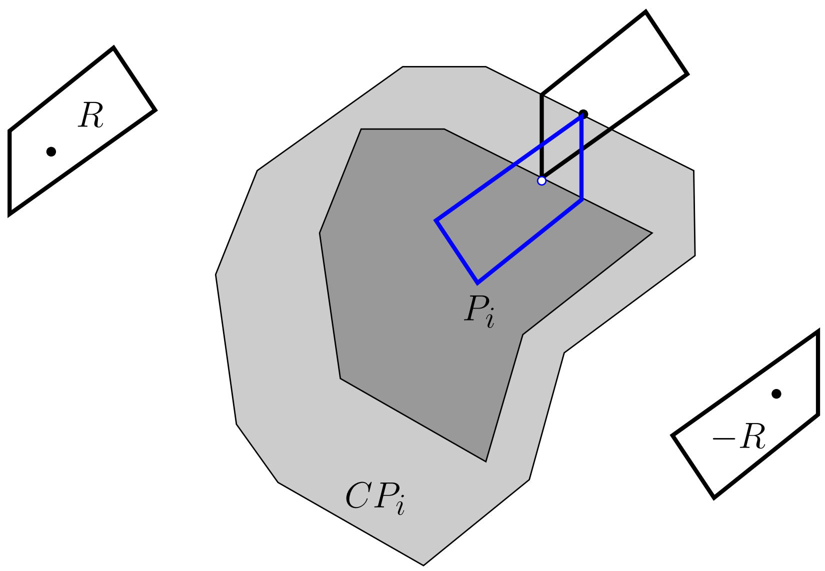
$$CP_i := \{\ (x,y) \in \mathbb{R}^2 | R(x,y) \cap P_i \ \neq\ \emptyset\ \}$$

$$CP_i = P_i \oplus (-R(0,0))$$

Lemma 2.6

$$CP_i := \{ \, (x,y) \in \mathbb{R}^2 | R(x,y) \cap P_i \neq \emptyset \, \}$$

$$CP_i = P_i \oplus (-R(0,0))$$



Beweis: Tafel!!!