Pledge and Bug

Elmar Langetepe
University of Bonn

Beweis Korrektheit: Winkelzähler nicht positiv

Beweis Korrektheit: Winkelzähler nicht positiv

Lemma 7.2 Der Winkelzähler im Pledge-Algorithmus nimmt niemals einen positiven Wert an.

Beweis Korrektheit: Winkelzähler nicht positiv

Lemma 7.2 Der Winkelzähler im Pledge-Algorithmus nimmt niemals einen positiven Wert an.

Beweis:

Beweis Korrektheit: Winkelzähler nicht positiv

Lemma 7.2 Der Winkelzähler im Pledge-Algorithmus nimmt niemals einen positiven Wert an.

Beweis:

- Zu Beginn Null

Beweis Korrektheit: Winkelzähler nicht positiv

Lemma 7.2 Der Winkelzähler im Pledge-Algorithmus nimmt niemals einen positiven Wert an.

Beweis:

- Zu Beginn Null
- Null bei Verlassen des Hindernisses

Beweis Korrektheit: Winkelzähler nicht positiv

Lemma 7.2 Der Winkelzähler im Pledge-Algorithmus nimmt niemals einen positiven Wert an.

Beweis:

- Zu Beginn Null
- Null bei Verlassen des Hindernisses
- Beim Auftreffen Rechtsdrehung $\Rightarrow$ negativ

Beweis Korrektheit: Winkelzähler nicht positiv

Lemma 7.2 Der Winkelzähler im Pledge-Algorithmus nimmt niemals einen positiven Wert an.

Beweis:

- Zu Beginn Null
- Null bei Verlassen des Hindernisses
- Beim Auftreffen Rechtsdrehung $\Rightarrow$ negativ
- Stetige Änderung: Null $\Rightarrow$ Weg frei

Beweis Korrektheit: Endlosstück

Beweis Korrektheit: Endlosstück

Lemma 7.3 Falls der Roboter nicht ins Freie gelangt, durchläuft er — bis auf ein endliches Anfangsstück — stets den gleichen Weg aufs Neue.

Beweis Korrektheit: Endlosstück

Lemma 7.3 Falls der Roboter nicht ins Freie gelangt, durchläuft er — bis auf ein endliches Anfangsstück — stets den gleichen Weg aufs Neue.

Beweis:

Beweis Korrektheit: Endlosstück

Lemma 7.3 Falls der Roboter nicht ins Freie gelangt, durchläuft er — bis auf ein endliches Anfangsstück — stets den gleichen Weg aufs Neue.

Beweis:

- Weg des Roboters ist polygonale Kette

Beweis Korrektheit: Endlosstück

Lemma 7.3 Falls der Roboter nicht ins Freie gelangt, durchläuft er — bis auf ein endliches Anfangsstück — stets den gleichen Weg aufs Neue.

Beweis:

- Weg des Roboters ist polygonale Kette
- Eckpunkte der Szene sind Knoten der Kette

Beweis Korrektheit: Endlosstück

Lemma 7.3 Falls der Roboter nicht ins Freie gelangt, durchläuft er — bis auf ein endliches Anfangsstück — stets den gleichen Weg aufs Neue.

Beweis:

- Weg des Roboters ist polygonale Kette
- Eckpunkte der Szene sind Knoten der Kette
- Auftreffpunkte auf Kanten sind Knoten der Kette

Beweis Korrektheit: Endlosstück

Lemma 7.3 Falls der Roboter nicht ins Freie gelangt, durchläuft er — bis auf ein endliches Anfangsstück — stets den gleichen Weg aufs Neue.

Beweis:

- Weg des Roboters ist polygonale Kette
- Eckpunkte der Szene sind Knoten der Kette
- Auftreffpunkte auf Kanten sind Knoten der Kette
- Zu jeder Ecke existiert genau ein Auftreffpunkt auf Kante

Beweis Korrektheit: Endlosstück

Lemma 7.3 Falls der Roboter nicht ins Freie gelangt, durchläuft er — bis auf ein endliches Anfangsstück — stets den gleichen Weg aufs Neue.

Beweis:

- Weg des Roboters ist polygonale Kette
- Eckpunkte der Szene sind Knoten der Kette
- Auftreffpunkte auf Kanten sind Knoten der Kette
- Zu jeder Ecke existiert genau ein Auftreffpunkt auf Kante
- Endliche Menge S von möglichen Ecken für Weg

Beweis Korrektheit: Endlosstück

Lemma 7.3 Falls der Roboter nicht ins Freie gelangt, durchläuft er — bis auf ein endliches Anfangsstück — stets den gleichen Weg aufs Neue.

Beweis:

- Weg des Roboters ist polygonale Kette
- Eckpunkte der Szene sind Knoten der Kette
- Auftreffpunkte auf Kanten sind Knoten der Kette
- Zu jeder Ecke existiert genau ein Auftreffpunkt auf Kante
- Endliche Menge S von möglichen Ecken für Weg
- Gleicher Zählerstand an Ecke $\Rightarrow$ gleicher Weg immer wieder

Beweis Korrektheit: Endlosstück

Lemma 7.3 Falls der Roboter nicht ins Freie gelangt, durchläuft er — bis auf ein endliches Anfangsstück — stets den gleichen Weg aufs Neue.

Beweis:

- Weg des Roboters ist polygonale Kette
- Eckpunkte der Szene sind Knoten der Kette
- Auftreffpunkte auf Kanten sind Knoten der Kette
- Zu jeder Ecke existiert genau ein Auftreffpunkt auf Kante
- Endliche Menge S von möglichen Ecken für Weg
- Gleicher Zählerstand an Ecke $\Rightarrow$ gleicher Weg immer wieder
- Annahme: Nie gleicher Zählerstand

- 1. Fall: Löst sich irgendwann nicht mehr $\Rightarrow$ gleicher Weg immer wieder

- 1. Fall: Löst sich irgendwann nicht mehr $\Rightarrow$ gleicher Weg immer wieder
- 2. Fall: Löst sich mehr als $|S|$ mal (unendlich oft)

- 1. Fall: Löst sich irgendwann nicht mehr $\Rightarrow$ gleicher Weg immer wieder
- 2. Fall: Löst sich mehr als $|S|$ mal (unendlich oft)
- $\Rightarrow$ zweimal mit gleichem Zählerstand 0 an gleicher Ecke, Widerspruch!

Beweis Korrektheit: Endlosweg schnittfrei

Beweis Korrektheit: Endlosweg schnittfrei

Lemma 7.4 Angenommen, der Roboter kann dem Labyrinth nicht entkommen. Sei $\Pi_\circ$ der Teil des Weges, den der Roboter immer wieder durchläuft. $\Pi_\circ$ ist kreuzungsfrei.

Beweis Korrektheit: Endlosweg schnittfrei

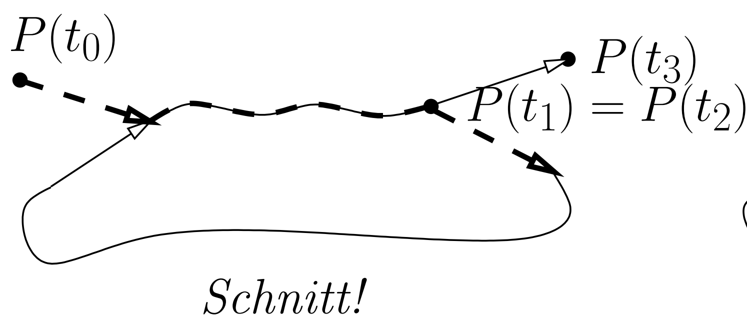
Lemma 7.4 Angenommen, der Roboter kann dem Labyrinth nicht entkommen. Sei $\Pi_\circ$ der Teil des Weges, den der Roboter immer wieder durchläuft. $\Pi_\circ$ ist kreuzungsfrei.

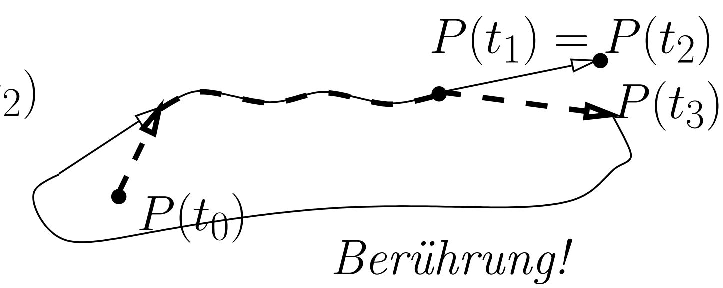
Unterschied: Kreuzen/Berühren

Beweis Korrektheit: Endlosweg schnittfrei

Lemma 7.4 Angenommen, der Roboter kann dem Labyrinth nicht entkommen. Sei $\Pi_\circ$ der Teil des Weges, den der Roboter immer wieder durchläuft. $\Pi_\circ$ ist kreuzungsfrei.

Unterschied: Kreuzen/Berühren

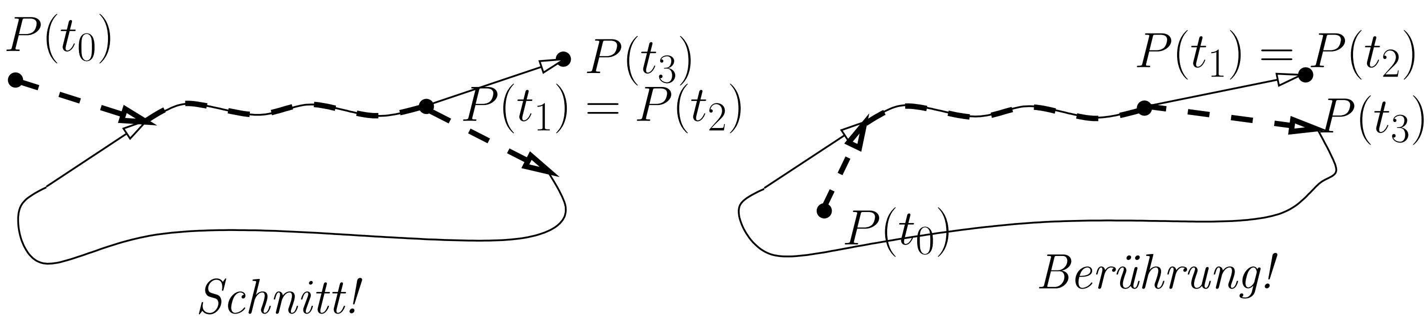




Beweis Korrektheit: Endlosweg schnittfrei

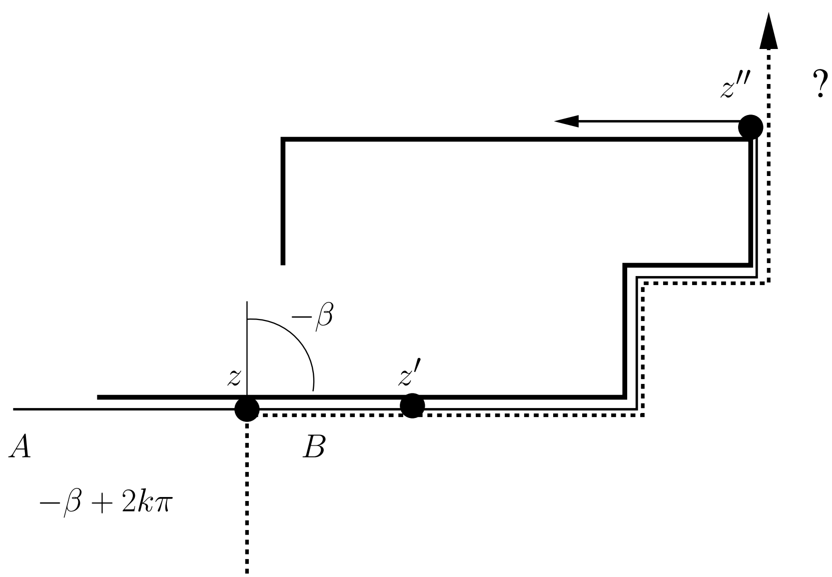
Lemma 7.4 Angenommen, der Roboter kann dem Labyrinth nicht entkommen. Sei $\Pi_\circ$ der Teil des Weges, den der Roboter immer wieder durchläuft. $\Pi_\circ$ ist kreuzungsfrei.

Unterschied: Kreuzen/Berühren



Kreuzung nur am Hindernis, freie Wege parallel!

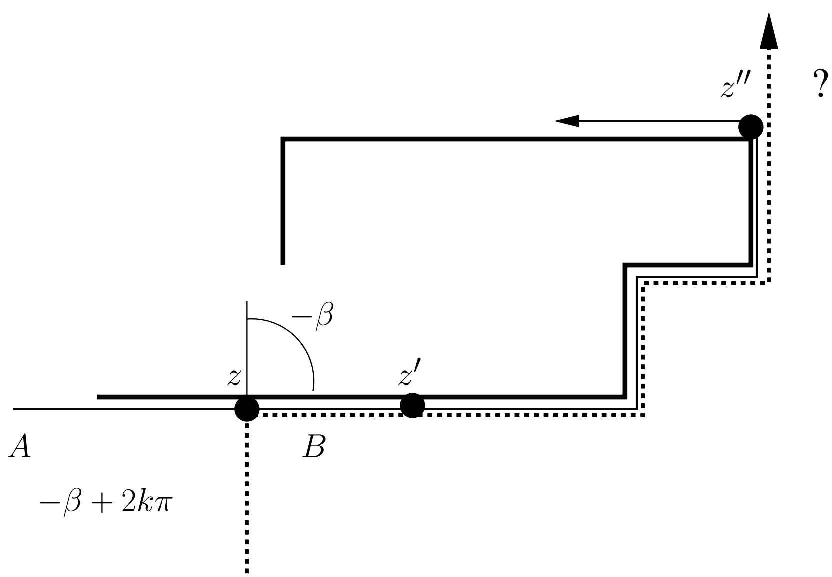
Beweis Korrektheit: Endlosweg schnittfrei



Beweis Korrektheit: Endlosweg schnittfrei

Lemma 7.4 Annahme Roboter kann nicht entkommen. $\Pi_\circ$ stets aufs Neue. $\Pi_\circ$ ist kreuzungsfrei.

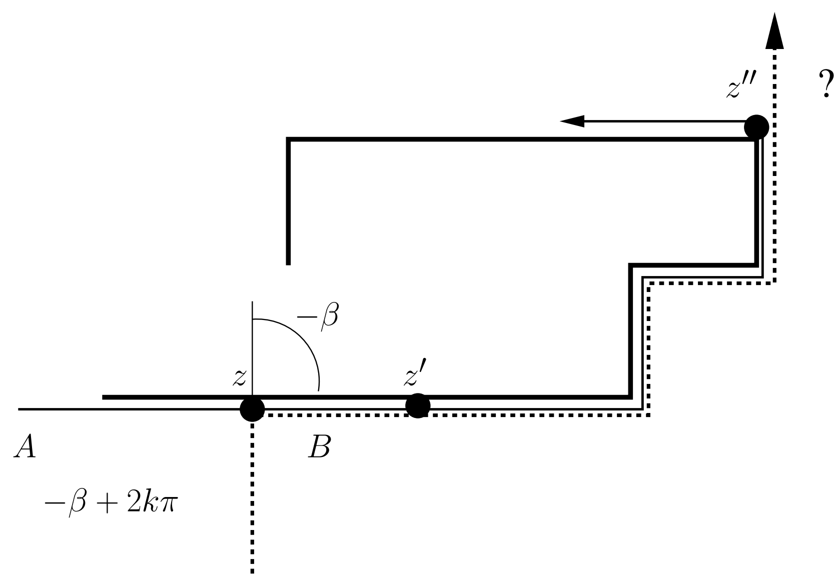
- Beweis:



Beweis Korrektheit: Endlosweg schnittfrei

Lemma 7.4 Annahme Roboter kann nicht entkommen. $\Pi_\circ$ stets aufs Neue. $\Pi_\circ$ ist kreuzungsfrei.

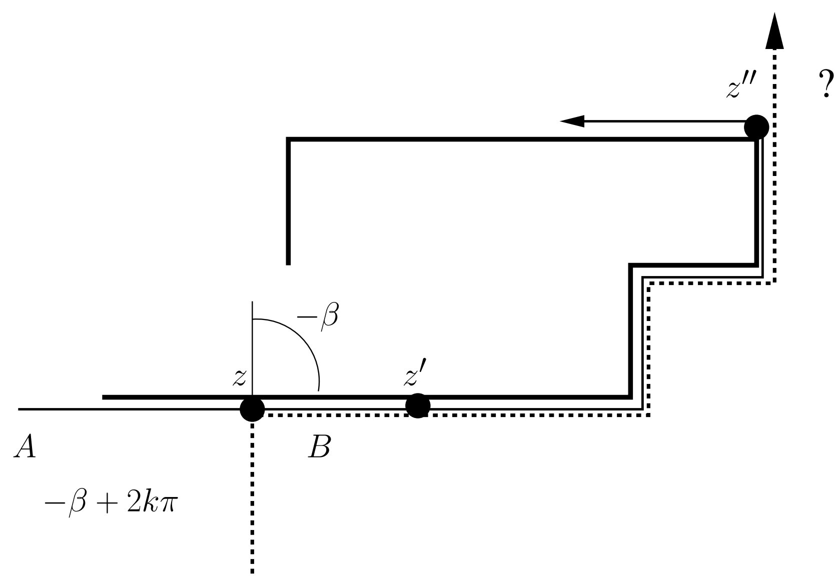
- Beweis: Einer der beiden Segmente z.B. B ist frei



Beweis Korrektheit: Endlosweg schnittfrei

Lemma 7.4 Annahme Roboter kann nicht entkommen. $\Pi_\circ$ stets aufs Neue. $\Pi_\circ$ ist kreuzungsfrei.

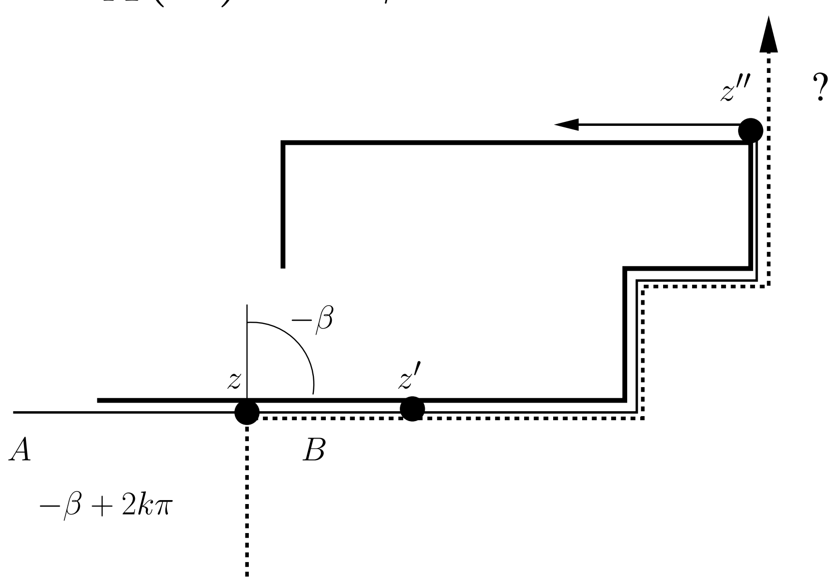
- Beweis: Einer der beiden Segmente z.B. B ist frei
- Kurz hinter z Winkelzähler $C_A(z')$ und $C_B(z')$



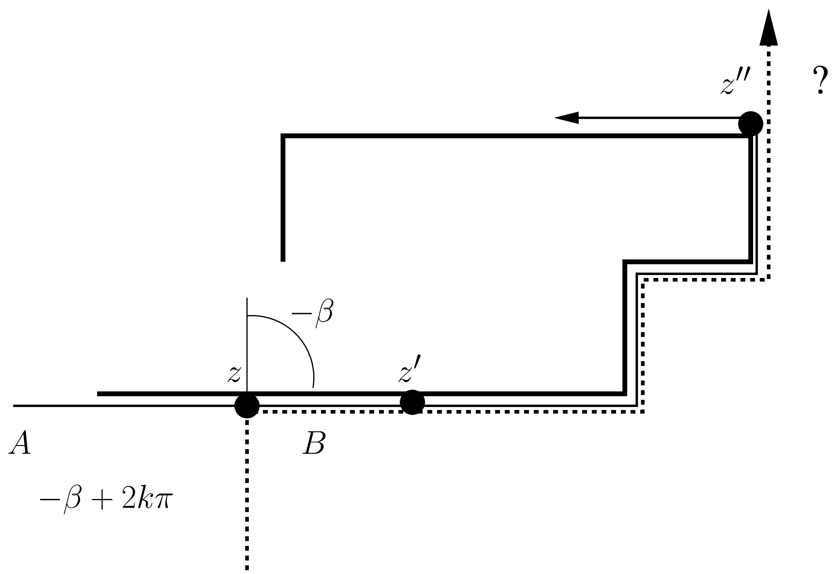
Beweis Korrektheit: Endlosweg schnittfrei

Lemma 7.4 Annahme Roboter kann nicht entkommen. $\Pi_\circ$ stets aufs Neue. $\Pi_\circ$ ist kreuzungsfrei.

- Beweis: Einer der beiden Segmente z.B. B ist frei
- Kurz hinter z Winkelzähler $C_A(z')$ und $C_B(z')$
- $C_B(z') = -\beta$ und $C_A(z') = -\beta + 2k\pi$ mit $k \in Z$

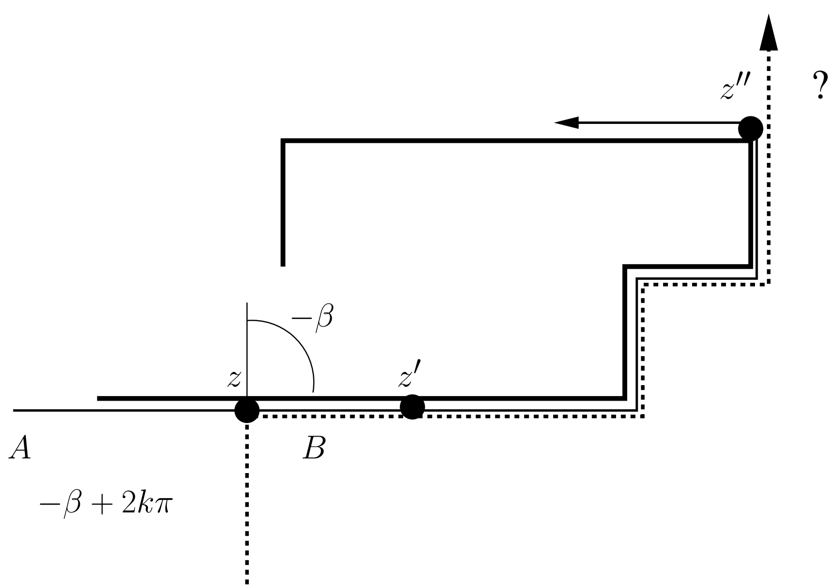


Beweis Lemma 7.4 Endlosweg schnittfrei



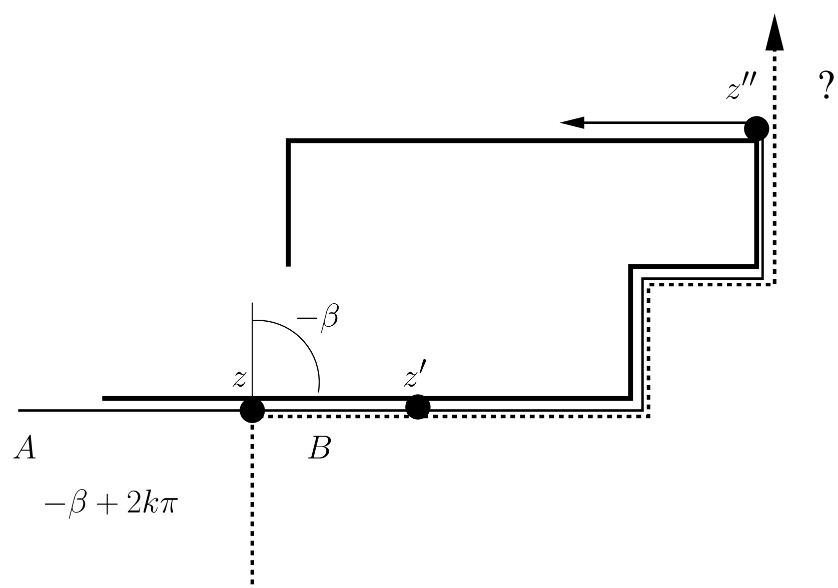
Beweis Lemma 7.4 Endlosweg schnittfrei

- $C_B(z') = -\beta$ und $C_A(z') = -\beta + 2k\pi$ mit $k \in Z$
- $k = 0$? Geht nicht wg. determ. Widerspruch!



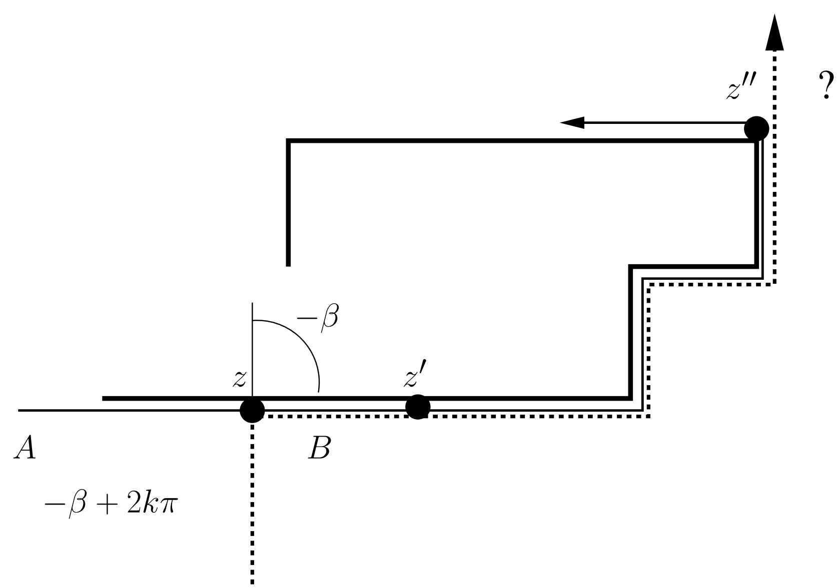
Beweis Lemma 7.4 Endlosweg schnittfrei

- $C_B(z') = -\beta$ und $C_A(z') = -\beta + 2k\pi$ mit $k \in Z$
- $k = 0$? Geht nicht wg. determ. Widerspruch!
- $k > 0$? Geht nicht wg. Lemma 7.2, $C_A(z')$ negativ



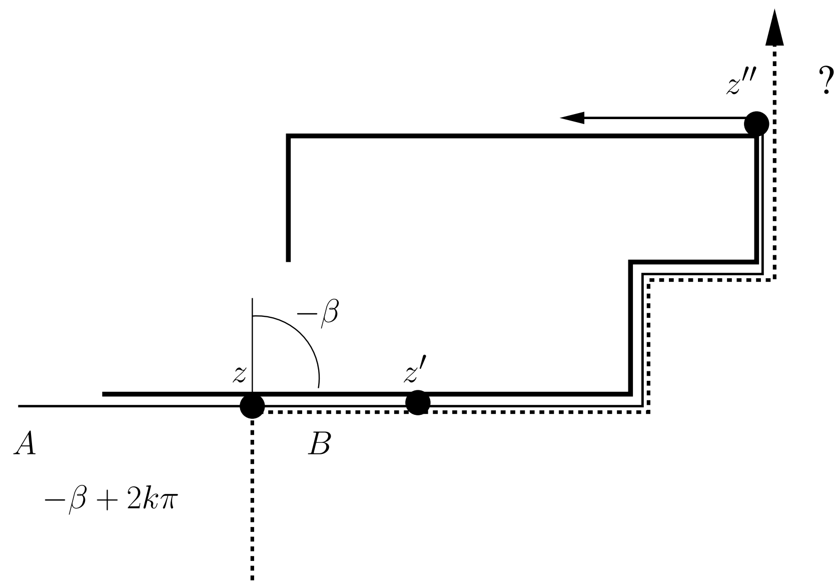
Beweis Lemma 7.4 Endlosweg schnittfrei

- $C_B(z') = -\beta$ und $C_A(z') = -\beta + 2k\pi$ mit $k \in Z$
- $k = 0$? Geht nicht wg. determ. Widerspruch!
- $k > 0$? Geht nicht wg. Lemma 7.2, $C_A(z')$ negativ
- Also $k < 0$ dann $C_A(p) < C_B(p)$ für alle p zwischen z' und z''



Beweis Lemma 7.4 Endlosweg schnittfrei

- $C_B(z') = -\beta$ und $C_A(z') = -\beta + 2k\pi$ mit $k \in Z$
- $k = 0$? Geht nicht wg. determ. Widerspruch!
- $k > 0$? Geht nicht wg. Lemma 7.2, $C_A(z')$ negativ
- Also $k < 0$ dann $C_A(p) < C_B(p)$ für alle p zwischen z' und z''
- Weg B trennt sich zuerst, kein Schnitt!!!



Beweis Korrektheit!

Beweis Korrektheit!

Theorem 7.1 Der Pledge-Algorithmus findet in jedem Labyrinth und von jeder Startposition einen Ausweg, falls überhaupt einer existiert.

Beweis Korrektheit!

Theorem 7.1 Der Pledge-Algorithmus findet in jedem Labyrinth und von jeder Startposition einen Ausweg, falls überhaupt einer existiert.

Beweis:

Beweis Korrektheit!

Theorem 7.1 Der Pledge-Algorithmus findet in jedem Labyrinth und von jeder Startposition einen Ausweg, falls überhaupt einer existiert.

Beweis:

- Annahme: Roboter erreicht Rand nicht.

Beweis Korrektheit!

Theorem 7.1 Der Pledge-Algorithmus findet in jedem Labyrinth und von jeder Startposition einen Ausweg, falls überhaupt einer existiert.

Beweis:

- Annahme: Roboter erreicht Rand nicht.
- **Lemma 7.3:** Pfad $\Pi_\circ$ stets aufs Neue

Beweis Korrektheit!

Theorem 7.1 Der Pledge-Algorithmus findet in jedem Labyrinth und von jeder Startposition einen Ausweg, falls überhaupt einer existiert.

Beweis:

- Annahme: Roboter erreicht Rand nicht.
- **Lemma 7.3:** Pfad $\Pi_\circ$ stets aufs Neue
- **Lemma 7.4:** Ohne echte Kreuzungen

Beweis Korrektheit!

Theorem 7.1 Der Pledge-Algorithmus findet in jedem Labyrinth und von jeder Startposition einen Ausweg, falls überhaupt einer existiert.

Beweis:

- Annahme: Roboter erreicht Rand nicht.
- **Lemma 7.3:** Pfad $\Pi_\circ$ stets aufs Neue
- **Lemma 7.4:** Ohne echte Kreuzungen
- Zwei Orientierungen: 1) Im UZS 2) Gegen den UZS

Beweis Korrektheit!

Theorem 7.1 Der Pledge-Algorithmus findet in jedem Labyrinth und von jeder Startposition einen Ausweg, falls überhaupt einer existiert.

Beweis:

- Annahme: Roboter erreicht Rand nicht.
- **Lemma 7.3:** Pfad $\Pi_\circ$ stets aufs Neue
- **Lemma 7.4:** Ohne echte Kreuzungen
- Zwei Orientierungen: 1) Im UZS 2) Gegen den UZS
- 2) stets $+2\pi$ pro Runde, irgendwann positiv, Widerspruch

Beweis Korrektheit!

Theorem 7.1 Der Pledge-Algorithmus findet in jedem Labyrinth und von jeder Startposition einen Ausweg, falls überhaupt einer existiert.

Beweis:

- Annahme: Roboter erreicht Rand nicht.
- **Lemma 7.3:** Pfad $\Pi_\circ$ stets aufs Neue
- **Lemma 7.4:** Ohne echte Kreuzungen
- Zwei Orientierungen: 1) Im UZS 2) Gegen den UZS
- 2) stets $+2\pi$ pro Runde, irgendwann positiv, Widerspruch
- Also 1) stets -2π pro Runde

Beweis Korrektheit!

Theorem 7.1 Der Pledge-Algorithmus findet in jedem Labyrinth und von jeder Startposition einen Ausweg, falls überhaupt einer existiert.

Beweis:

- Annahme: Roboter erreicht Rand nicht.
- **Lemma 7.3:** Pfad $\Pi_\circ$ stets aufs Neue
- **Lemma 7.4:** Ohne echte Kreuzungen
- Zwei Orientierungen: 1) Im UZS 2) Gegen den UZS
- 2) stets $+2\pi$ pro Runde, irgendwann positiv, Widerspruch
- Also 1) stets -2π pro Runde
- Irgendwann nur noch negativ, Hindernis wird nicht verlassen

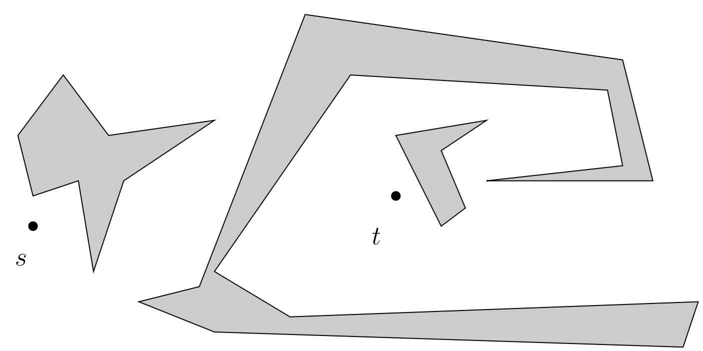
Beweis Korrektheit!

Theorem 7.1 Der Pledge-Algorithmus findet in jedem Labyrinth und von jeder Startposition einen Ausweg, falls überhaupt einer existiert.

Beweis:

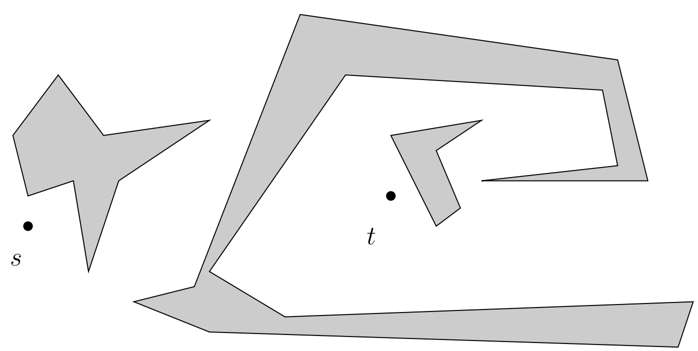
- Annahme: Roboter erreicht Rand nicht.
- **Lemma 7.3:** Pfad $\Pi_\circ$ stets aufs Neue
- **Lemma 7.4:** Ohne echte Kreuzungen
- Zwei Orientierungen: 1) Im UZS 2) Gegen den UZS
- 2) stets $+2\pi$ pro Runde, irgendwann positiv, Widerspruch
- Also 1) stets -2π pro Runde
- Irgendwann nur noch negativ, Hindernis wird nicht verlassen
- Orientierung: Im UZS $\Rightarrow$ Innenhof!

Finden eines Zielpunktes



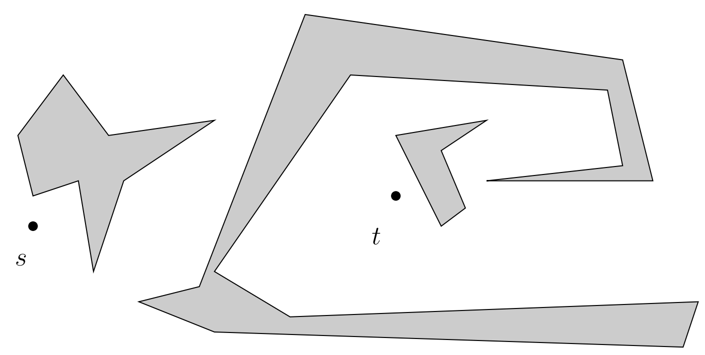
Finden eines Zielpunktes

- Andere Aufgabenstellung: Suchen eines vorgegebenen Ziels



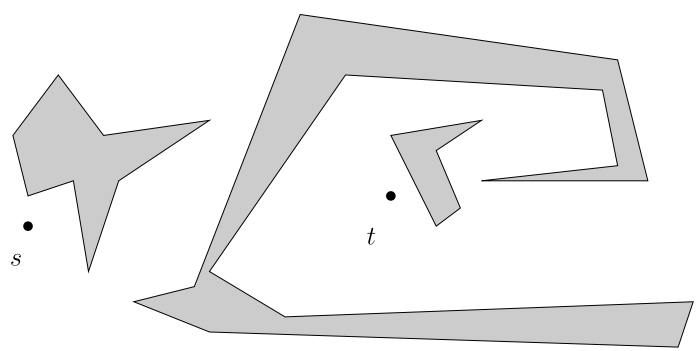
Finden eines Zielpunktes

- Andere Aufgabenstellung: Suchen eines vorgegebenen Ziels
- Polygonale Umgebung, endlich viele disjunkte Polygone



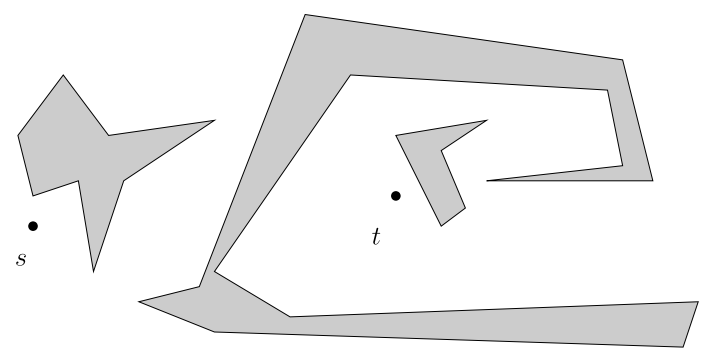
Finden eines Zielpunktes

- Andere Aufgabenstellung: Suchen eines vorgegebenen Ziels
- Polygonale Umgebung, endlich viele disjunkte Polygone
- Tastsensor, entlang der Wand



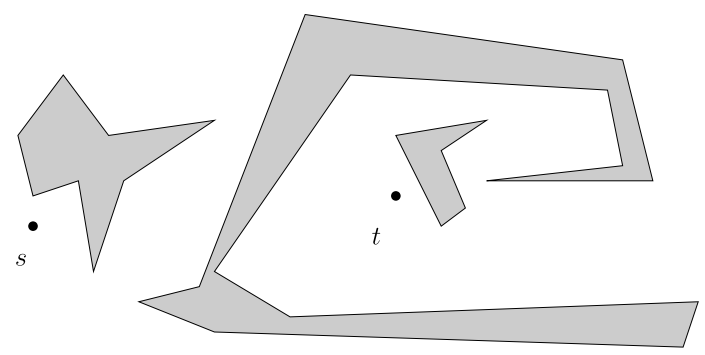
Finden eines Zielpunktes

- Andere Aufgabenstellung: Suchen eines vorgegebenen Ziels
- Polygonale Umgebung, endlich viele disjunkte Polygone
- Tastsensor, entlang der Wand
- Startpunkt s, Zielpunkt t, Koordinaten des Zielpunktes



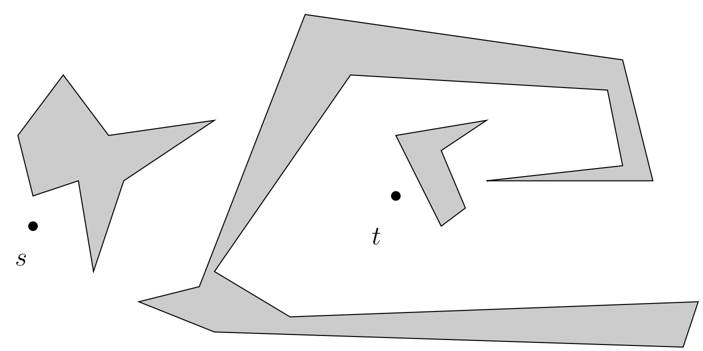
Finden eines Zielpunktes

- Andere Aufgabenstellung: Suchen eines vorgegebenen Ziels
- Polygonale Umgebung, endlich viele disjunkte Polygone
- Tastsensor, entlang der Wand
- Startpunkt s, Zielpunkt t, Koordinaten des Zielpunktes
- Endlicher Speicher: z.B. Eigene Koordinaten



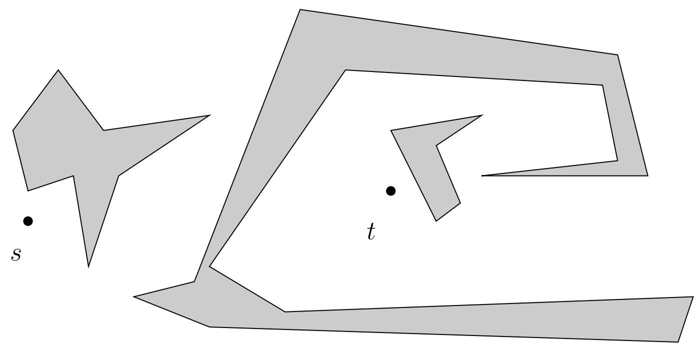
Finden eines Zielpunktes

- Andere Aufgabenstellung: Suchen eines vorgegebenen Ziels
- Polygonale Umgebung, endlich viele disjunkte Polygone
- Tastsensor, entlang der Wand
- Startpunkt s, Zielpunkt t, Koordinaten des Zielpunktes
- Endlicher Speicher: z.B. Eigene Koordinaten
- BUG Algorithmen:



Finden eines Zielpunktes

- Andere Aufgabenstellung: Suchen eines vorgegebenen Ziels
- Polygonale Umgebung, endlich viele disjunkte Polygone
- Tastsensor, entlang der Wand
- Startpunkt s, Zielpunkt t, Koordinaten des Zielpunktes
- Endlicher Speicher: z.B. Eigene Koordinaten
- BUG Algorithmen: Sojourner



Bezeichnungen

Bezeichnungen

- $|pq|$ den Abstand zwischen zwei Punkten p und q

Bezeichnungen

- $|pq|$ den Abstand zwischen zwei Punkten p und q
- $D := |st|$ die Distanz vom Start zum Zielpunkt

Bezeichnungen

- $|pq|$ den Abstand zwischen zwei Punkten p und q
- $D := |st|$ die Distanz vom Start zum Zielpunkt
- Π_S Weg der Strategie S vom Start zum Ziel

Bezeichnungen

- $|pq|$ den Abstand zwischen zwei Punkten p und q
- $D := |st|$ die Distanz vom Start zum Zielpunkt
- Π_S Weg der Strategie S vom Start zum Ziel
- $|\Pi_S|$ Länge des Weges Π_S

Bezeichnungen

- $|pq|$ den Abstand zwischen zwei Punkten p und q
- $D := |st|$ die Distanz vom Start zum Zielpunkt
- Π_S Weg der Strategie S vom Start zum Ziel
- $|\Pi_S|$ Länge des Weges Π_S
- $\mathtt{UP_i}$ Umfang des Hindernisses P_i.

Bezeichnungen

- $|pq|$ den Abstand zwischen zwei Punkten p und q
- $D := |st|$ die Distanz vom Start zum Zielpunkt
- Π_S Weg der Strategie S vom Start zum Ziel
- $|\Pi_S|$ Länge des Weges Π_S
- $\mathtt{UP_i}$ Umfang des Hindernisses P_i.
- Aktionen:

Bezeichnungen

- $|pq|$ den Abstand zwischen zwei Punkten p und q
- $D := |st|$ die Distanz vom Start zum Zielpunkt
- Π_S Weg der Strategie S vom Start zum Ziel
- $|\Pi_S|$ Länge des Weges Π_S
- $\mathtt{UP_i}$ Umfang des Hindernisses P_i.
- Aktionen:

1. Bewegung in Richtung zum Ziel

Bezeichnungen

- $|pq|$ den Abstand zwischen zwei Punkten p und q
- $D := |st|$ die Distanz vom Start zum Zielpunkt
- Π_S Weg der Strategie S vom Start zum Ziel
- $|\Pi_S|$ Länge des Weges Π_S
- $\mathtt{UP_i}$ Umfang des Hindernisses P_i.
- Aktionen:

1. Bewegung in Richtung zum Ziel
2. Bewegung entlang des Randes

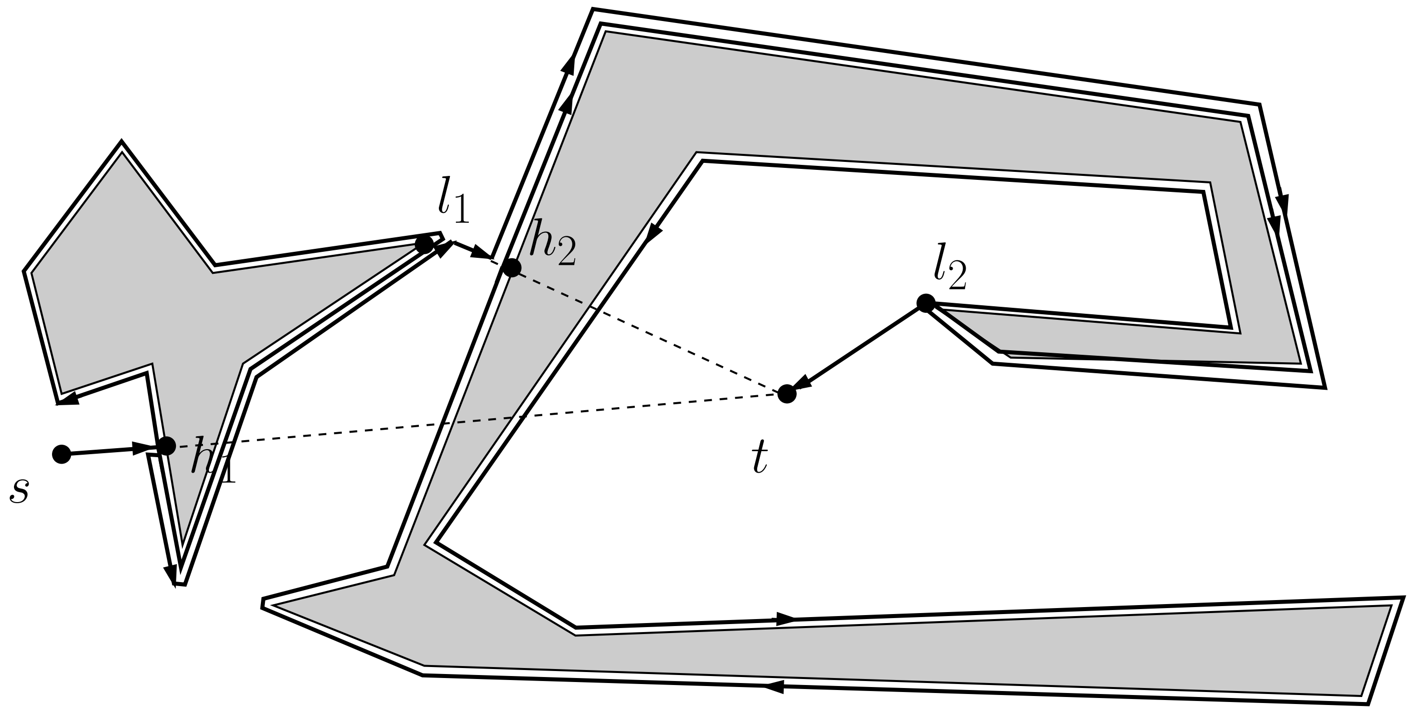
Bezeichnungen

- $|pq|$ den Abstand zwischen zwei Punkten p und q
- $D := |st|$ die Distanz vom Start zum Zielpunkt
- Π_S Weg der Strategie S vom Start zum Ziel
- $|\Pi_S|$ Länge des Weges Π_S
- $\mathtt{UP_i}$ Umfang des Hindernisses P_i.
- Aktionen:

  1. Bewegung in Richtung zum Ziel
  2. Bewegung entlang des Randes

- Leave-Points l_i, Hit-Points h_i

BUG Strategie: Lumelsky/Stepanov

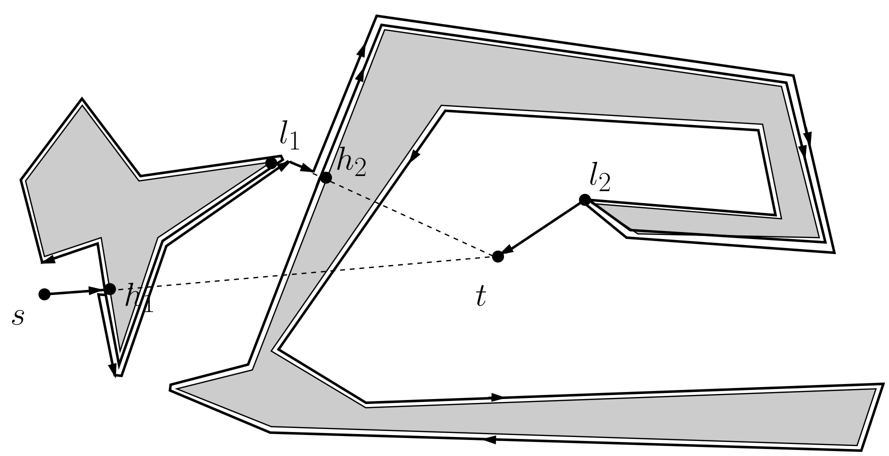


BUG Strategie: Lumelsky/Stepanov

BUG Strategie: Lumelsky/Stepanov

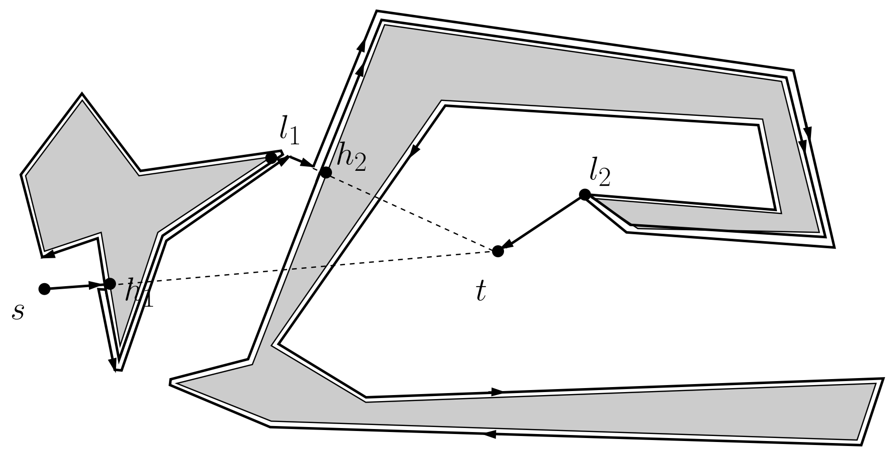
0. $l_0 := s$, $i := 1$
1. Von l_{i-1} gehe in Richtung Ziel, bis
   (a) Das Ziel wird erreicht: halte an.
   (b) Ein Hindernis wird getroffen. h_i sei der Punkt, an dem der Roboter auf das Hindernis trifft.
2. Umrunde das Hindernis im Uhrzeigersinn — beim Umrunden berechne den Punkt l_i auf dem Rand des Hindernisses mit kleinstem Abstand zu t —, bis
   (a) Das Ziel wird erreicht: halte an.
   (b) h_i wird erreicht.
3. Gehe auf kürzestem Weg am Hindernis entlang zu l_i.
4. Inkrementiere i, GOTO 1.

Korrektheit BUG Strategie



Korrektheit BUG Strategie

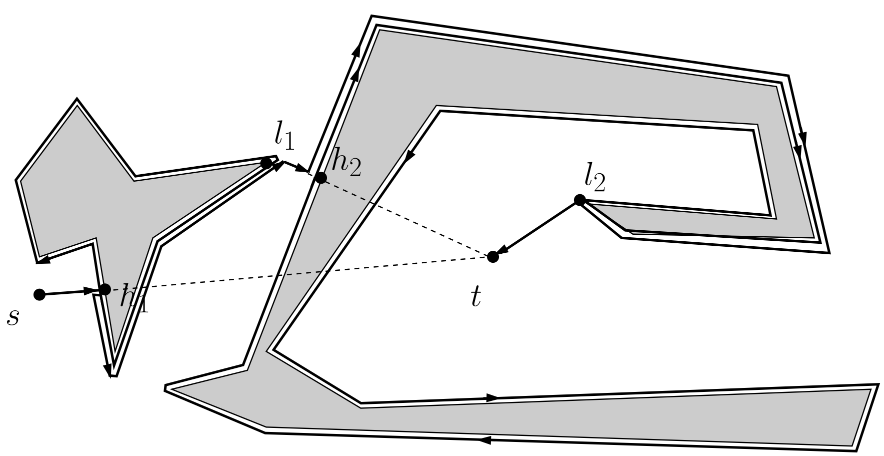
Theorem 7.5 Die Strategie BUG findet einen Weg vom Startpunkt s zum Zielpunkt t, falls ein solcher Weg existiert.



Korrektheit BUG Strategie

Theorem 7.5 Die Strategie BUG findet einen Weg vom Startpunkt s zum Zielpunkt t, falls ein solcher Weg existiert.

Beweis:

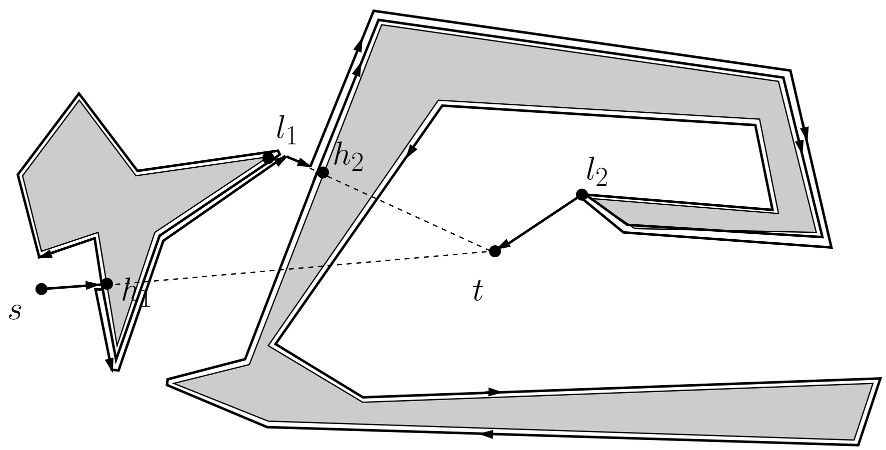


Korrektheit BUG Strategie

Theorem 7.5 Die Strategie BUG findet einen Weg vom Startpunkt s zum Zielpunkt t, falls ein solcher Weg existiert.

Beweis:

- Folge von Hit- und Leave-Points h_i, l_i

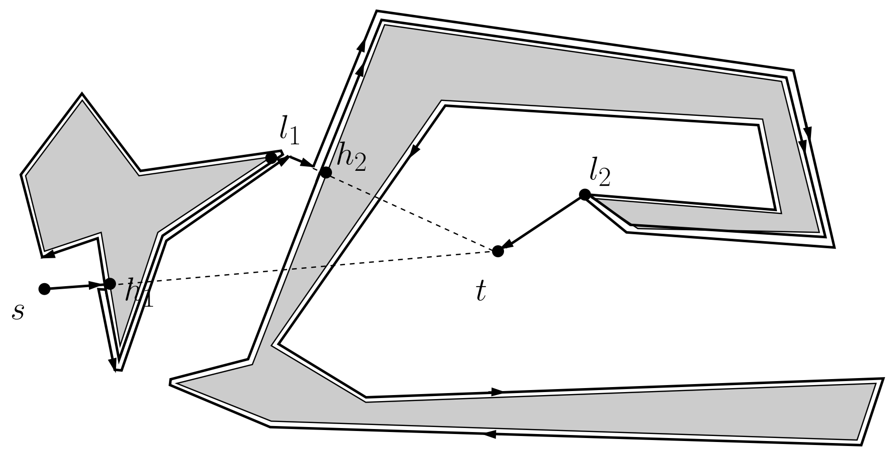


Korrektheit BUG Strategie

Theorem 7.5 Die Strategie BUG findet einen Weg vom Startpunkt s zum Zielpunkt t, falls ein solcher Weg existiert.

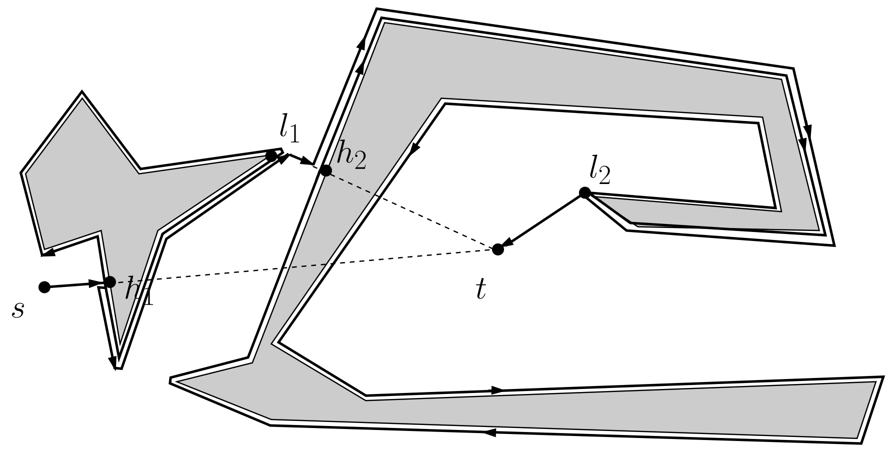
Beweis:

- Folge von Hit- und Leave-Points h_i, l_i
- $|st| \geq |h_1 t| \geq |l_1 t| \ldots \geq |h_k t| \geq |l_k t|$



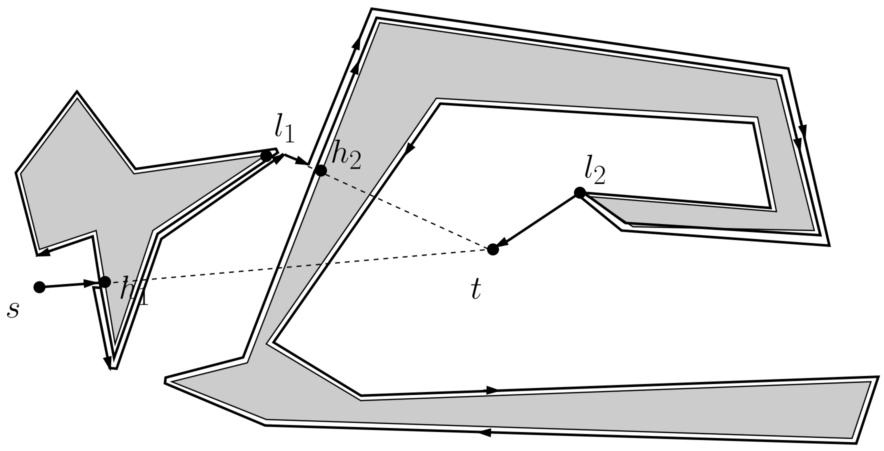
Theorem 7.5 Korrektheit BUG Strategie

Theorem 7.5 Korrektheit BUG Strategie



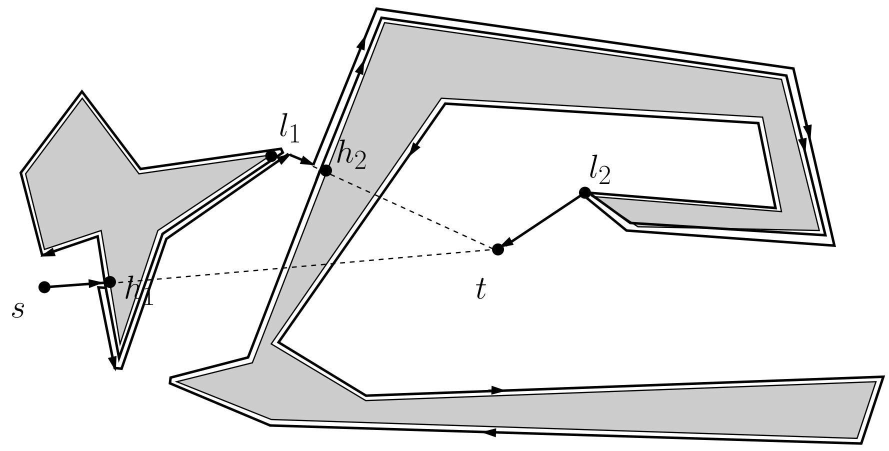
- Punkt mit kleinster Distanz als Leave-Point

Theorem 7.5 Korrektheit BUG Strategie



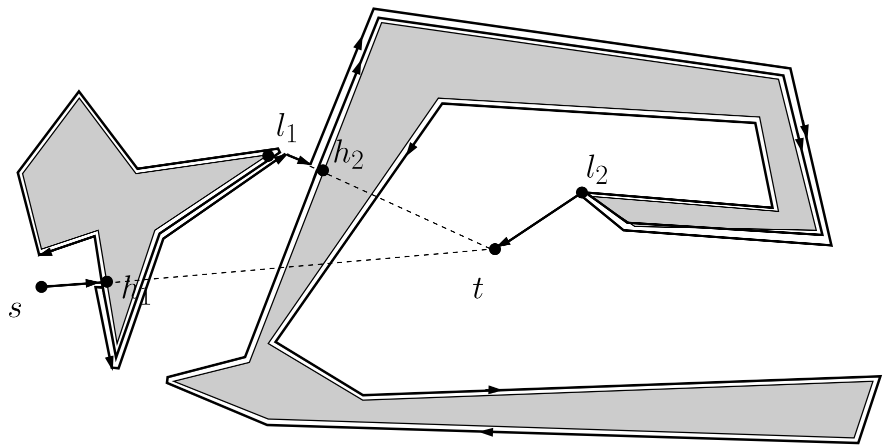
- Punkt mit kleinster Distanz als Leave-Point
- Keine freie Bewegung $\Rightarrow$ eingeschlossen

Theorem 7.5 Korrektheit BUG Strategie



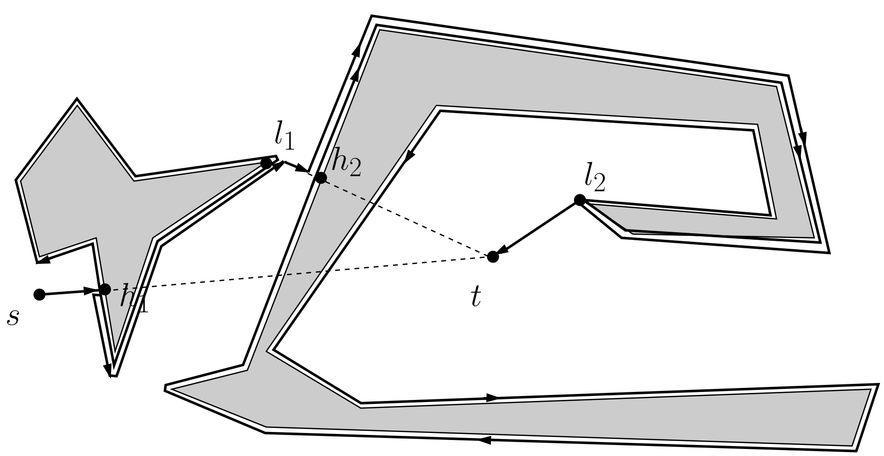
- Punkt mit kleinster Distanz als Leave-Point
- Keine freie Bewegung $\Rightarrow$ eingeschlossen
- $l_i \neq l_j$, neues Hindernis!

Theorem 7.5 Korrektheit BUG Strategie



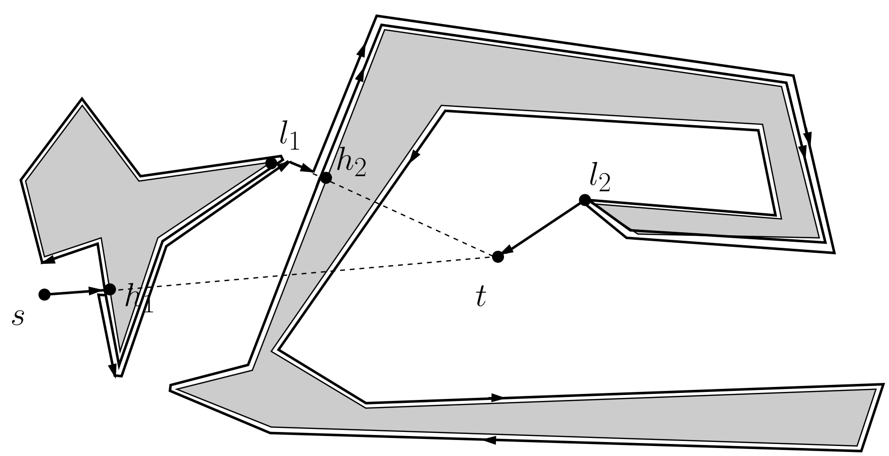
- Punkt mit kleinster Distanz als Leave-Point
- Keine freie Bewegung $\Rightarrow$ eingeschlossen
- $l_i \neq l_j$, neues Hindernis!
- Endlich viele Polygone $\Rightarrow$ Korrektheit

Laufzeit BUG1 Strategie



Laufzeit BUG1 Strategie

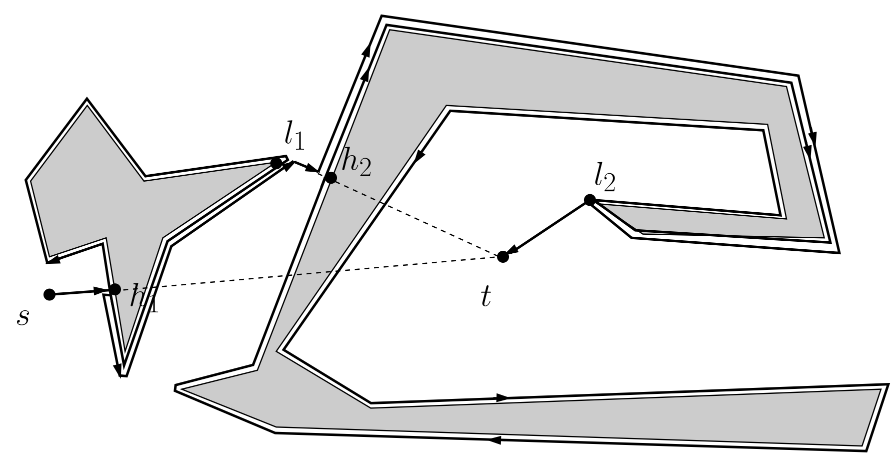
Theorem 7.6 Sei $\Pi_{\mathtt{Bug}}$ der Weg vom Start s zum Ziel t, den die Strategie Bug1 zurücklegt. Dann gilt $|\Pi_{\mathtt{Bug}}| \leq D + \frac{3}{2} \sum_i \mathtt{UP_i}$.



Laufzeit BUG1 Strategie

Theorem 7.6 Sei $\Pi_{\texttt{Bug}}$ der Weg vom Start s zum Ziel t, den die Strategie Bug1 zurücklegt. Dann gilt $|\Pi_{\texttt{Bug}}| \leq D + \frac{3}{2} \sum_i \texttt{UP}_\texttt{i}$.

Beweis:

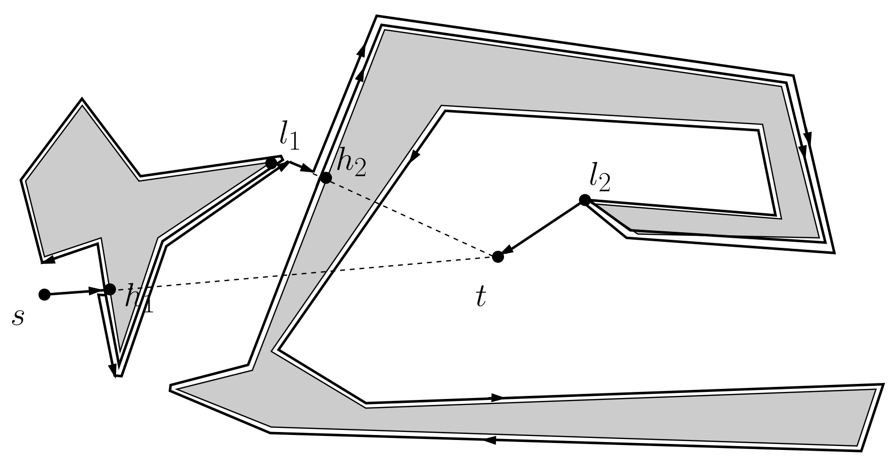


Laufzeit BUG1 Strategie

Theorem 7.6 Sei $\Pi_{\texttt{Bug}}$ der Weg vom Start s zum Ziel t, den die Strategie Bug1 zurücklegt. Dann gilt $|\Pi_{\texttt{Bug}}| \leq D + \frac{3}{2}\sum_i \texttt{UP}_\texttt{i}$.

Beweis:

- Einteilung: Direkter Weg, Umrundungen

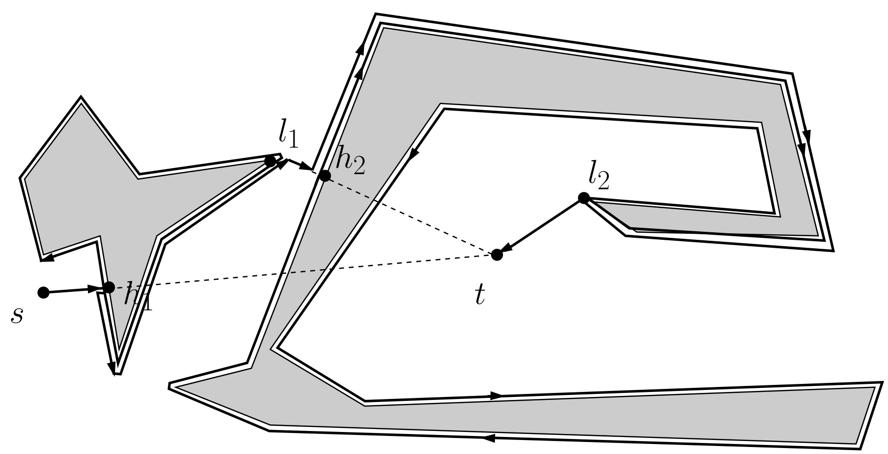


Laufzeit BUG1 Strategie

Theorem 7.6 Sei $\Pi_{\mathtt{Bug}}$ der Weg vom Start s zum Ziel t, den die Strategie Bug1 zurücklegt. Dann gilt $|\Pi_{\mathtt{Bug}}| \leq D + \frac{3}{2} \sum_i \mathtt{UP_i}$.

Beweis:

- Einteilung: Direkter Weg, Umrundungen
- Umrundung, danach kürzester Weg

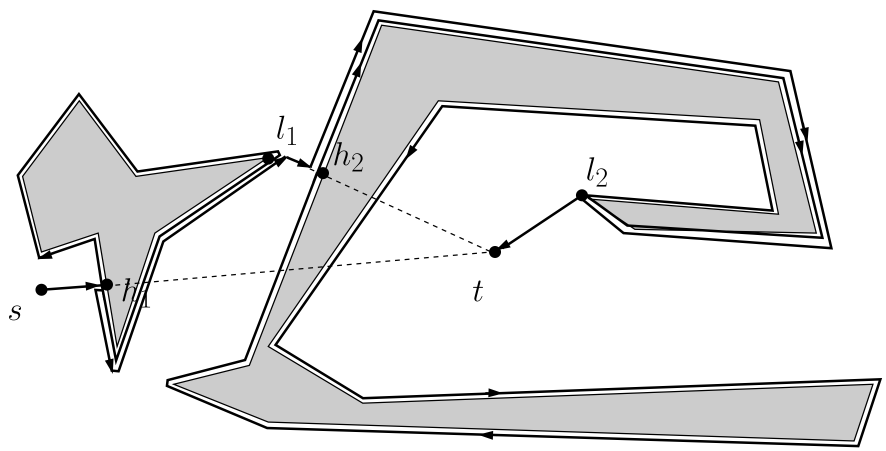


Laufzeit BUG1 Strategie

Theorem 7.6 Sei $\Pi_{\texttt{Bug}}$ der Weg vom Start s zum Ziel t, den die Strategie Bug1 zurücklegt. Dann gilt $|\Pi_{\texttt{Bug}}| \leq D + \frac{3}{2}\sum_i \texttt{UP}_\texttt{i}$.

Beweis:

- Einteilung: Direkter Weg, Umrundungen
- Umrundung, danach kürzester Weg
- $\frac{3}{2}\sum \texttt{UP}_\texttt{i}$

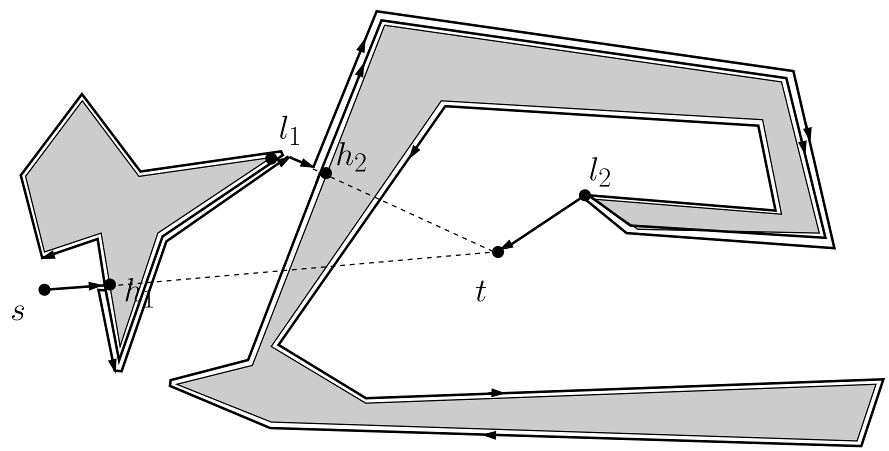


Laufzeit BUG1 Strategie

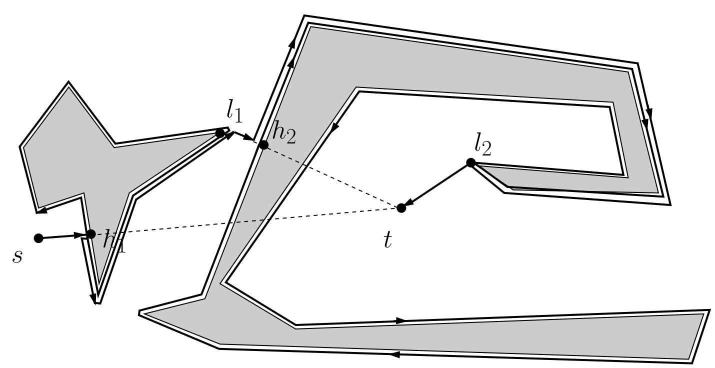
Theorem 7.6 Sei $\Pi_{\texttt{Bug}}$ der Weg vom Start s zum Ziel t, den die Strategie Bug1 zurücklegt. Dann gilt $|\Pi_{\texttt{Bug}}| \leq D + \frac{3}{2}\sum_i \texttt{UP}_\texttt{i}$.

Beweis:

- Einteilung: Direkter Weg, Umrundungen
- Umrundung, danach kürzester Weg
- $\frac{3}{2}\sum \texttt{UP}_\texttt{i}$
- Jetzt: Weg D' zwischen den Hindernissen

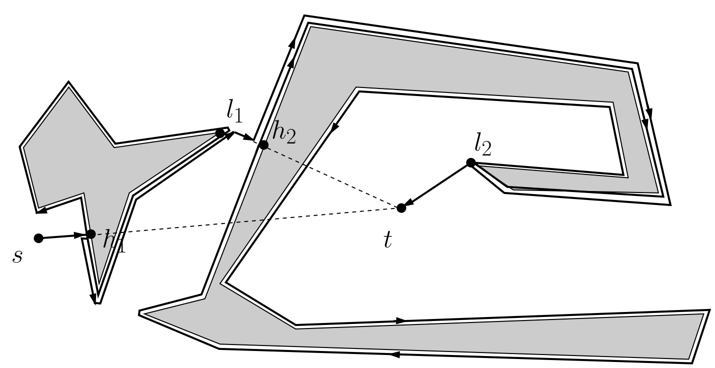


Theorem 7.6 $|\Pi_{\mathtt{Bug}}| \leq D + \frac{3}{2}\sum_i \mathtt{UP_i}$.



Theorem 7.6 $|\Pi_{\texttt{Bug}}| \leq D + \frac{3}{2}\sum_i \texttt{UP}_\texttt{i}$.

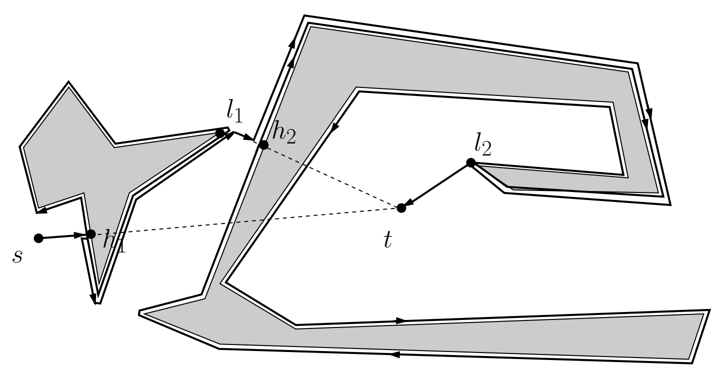
Beweis: D' zwischen Hindernissen



Theorem 7.6 $|\Pi_{\texttt{Bug}}| \leq D + \frac{3}{2}\sum_i \texttt{UP}_\texttt{i}$.

Beweis: D' zwischen Hindernissen

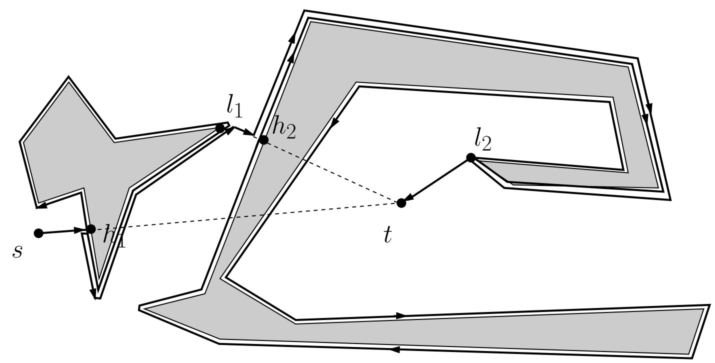
$$D' \;=\; |sh_1| + |\ell_1 h_2| + \ldots + |\ell_{k-1} h_k| + |\ell_k t|$$



Theorem 7.6 $|\Pi_{\texttt{Bug}}| \leq D + \frac{3}{2}\sum_i \texttt{UP}_\texttt{i}$.

Beweis: D' zwischen Hindernissen

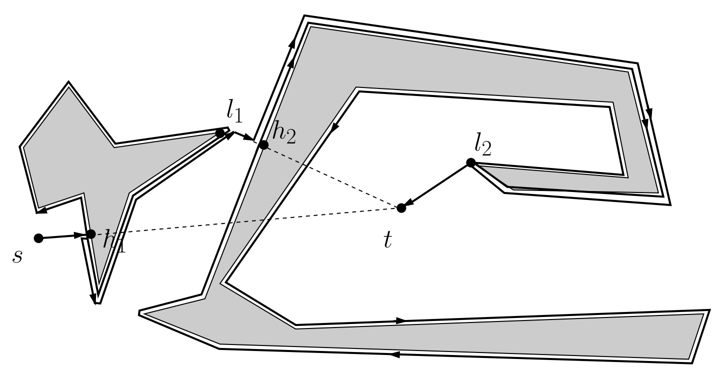
$$\begin{aligned} D' &= |sh_1| + |\ell_1 h_2| + \ldots + |\ell_{k-1} h_k| + |\ell_k t| \\ &\leq |sh_1| + |\ell_1 h_2| + \ldots + |\ell_{k-1} h_k| + |h_k t| \end{aligned}$$



Theorem 7.6 $|\Pi_{\texttt{Bug}}| \leq D + \frac{3}{2}\sum_i \texttt{UP}_\texttt{i}$.

Beweis: D' zwischen Hindernissen

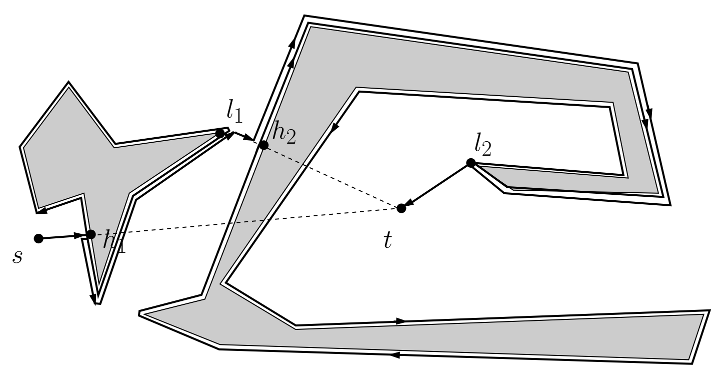
$$\begin{aligned} D' &= |sh_1| + |\ell_1 h_2| + \ldots + |\ell_{k-1} h_k| + |\ell_k t| \\ &\leq |sh_1| + |\ell_1 h_2| + \ldots + {\color{red}|\ell_{k-1} h_k| + |h_k t|} \\ &= |sh_1| + |\ell_1 h_2| + \ldots + {\color{red}|\ell_{k-1} t|} \end{aligned}$$



Theorem 7.6 $|\Pi_{\texttt{Bug}}| \leq D + \frac{3}{2}\sum_i \texttt{UP}_\texttt{i}$.

Beweis: D' zwischen Hindernissen

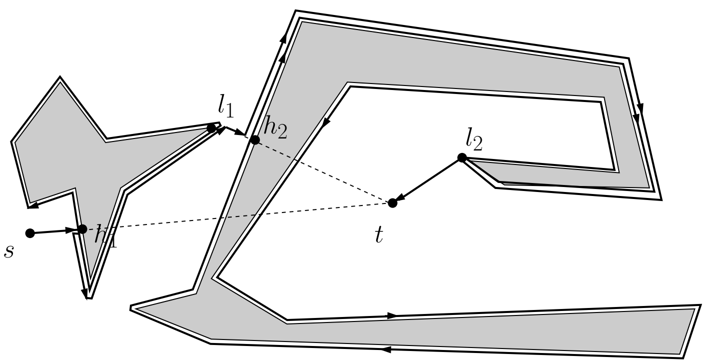
$$\begin{aligned} D' &= |sh_1| + |\ell_1 h_2| + \ldots + |\ell_{k-1} h_k| + |\ell_k t| \\ &\leq |sh_1| + |\ell_1 h_2| + \ldots + {\color{red}|\ell_{k-1} h_k| + |h_k t|} \\ &= |sh_1| + |\ell_1 h_2| + \ldots + {\color{red}|\ell_{k-1} t|} \\ &\ldots \\ &\leq |sh_1| + |\ell_1 t| \end{aligned}$$



Theorem 7.6 $|\Pi_{\texttt{Bug}}| \leq D + \frac{3}{2}\sum_i \texttt{UP}_\texttt{i}$.

Beweis: D' zwischen Hindernissen

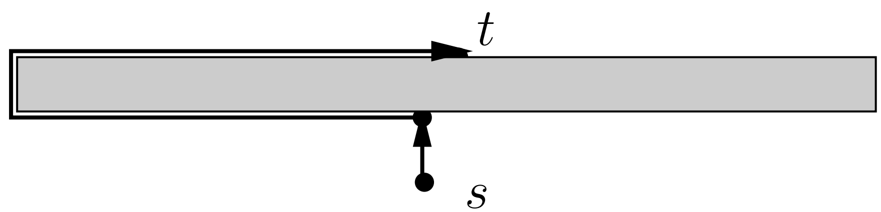
$$
\begin{aligned}
D' &= |sh_1| + |\ell_1 h_2| + \ldots + |\ell_{k-1} h_k| + |\ell_k t| \\
&\leq |sh_1| + |\ell_1 h_2| + \ldots + |\ell_{k-1} h_k| + |h_k t| \\
&= |sh_1| + |\ell_1 h_2| + \ldots + |\ell_{k-1} t| \\
&\ldots \\
&\leq |sh_1| + |\ell_1 t| \leq |sh_1| + |h_1 t| = |st| = D
\end{aligned}
$$



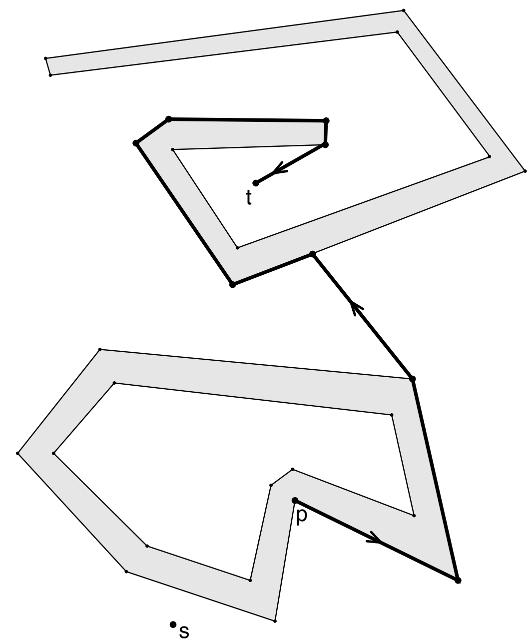
Theorem 7.6 $|\Pi_{\mathtt{Bug}}| \leq D + \frac{3}{2}\sum_i \mathtt{UP_i}$.

Theorem 7.6 $|\Pi_{\texttt{Bug}}| \leq D + \frac{3}{2}\sum_i \texttt{UP}_\texttt{i}$.

Beliebig langer Weg schon klar!

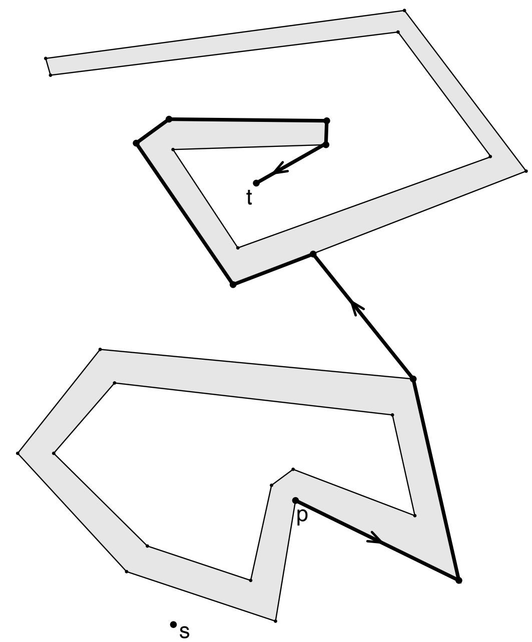


Einfache Bewegungen genügen stets!



Einfache Bewegungen genügen stets!

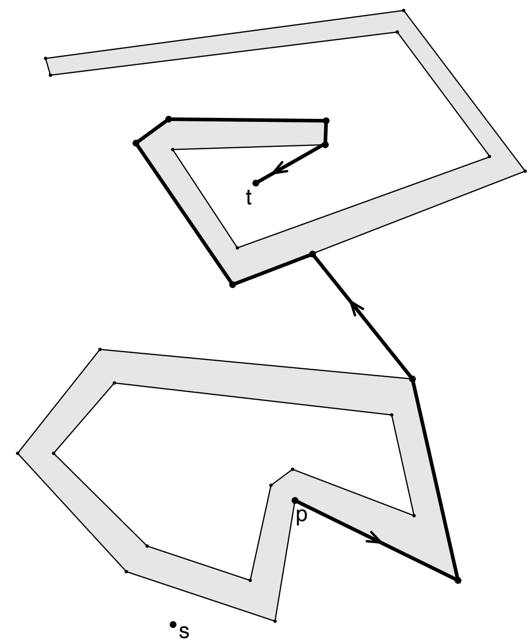
Zielkompass und Tastsensor



Einfache Bewegungen genügen stets!

Zielkompass und Tastsensor

Drei Steuerbefehle

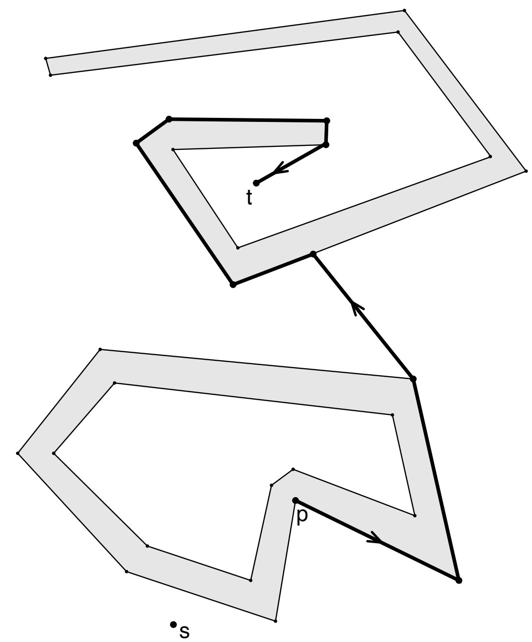


Einfache Bewegungen genügen stets!

Zielkompass und Tastsensor

Drei Steuerbefehle

1. T: Laufe in Richtung Ziel
   von Wandecke

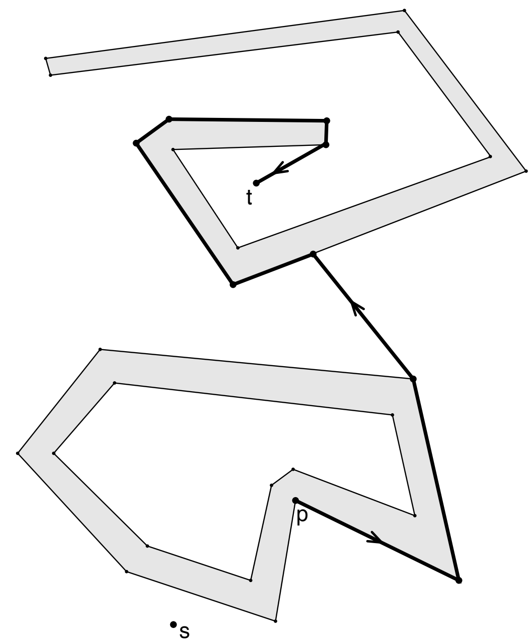


Einfache Bewegungen genügen stets!

Zielkompass und Tastsensor

Drei Steuerbefehle

1. T: Laufe in Richtung Ziel von Wandecke
2. L: Nächste Kante in gewohnte Richtung

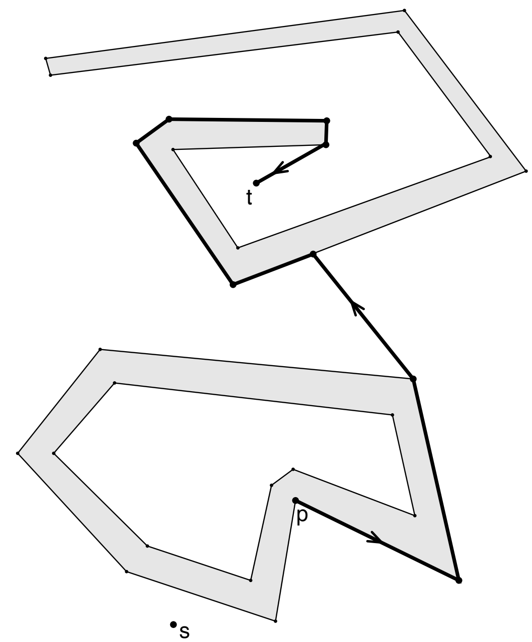


Einfache Bewegungen genügen stets!

Zielkompass und Tastsensor

Drei Steuerbefehle

1. T: Laufe in Richtung Ziel von Wandecke
2. L: Nächste Kante in gewohnte Richtung
3. R: Nächste Kante entg. gewohnte Richtung

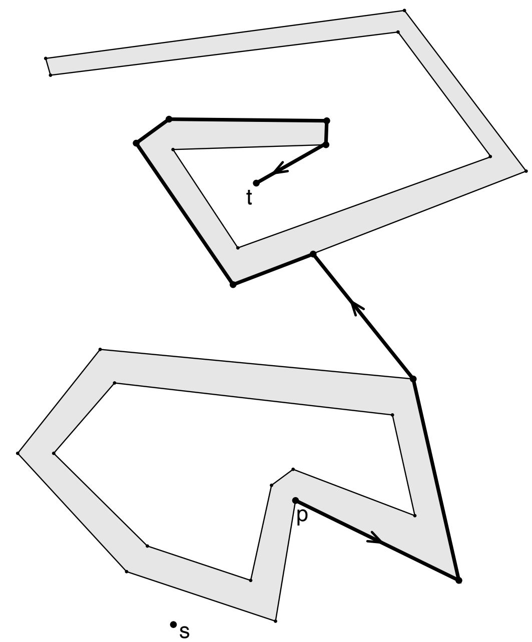


Einfache Bewegungen genügen stets!

Zielkompass und Tastsensor

Drei Steuerbefehle

1. T: Laufe in Richtung Ziel von Wandecke
2. L: Nächste Kante in gewohnte Richtung
3. R: Nächste Kante entg. gewohnte Richtung



Steuerwort-Beispiel:

$w(p) = L^2 T R^5 T$

Universelles Steuerwort!

Universelles Steuerwort!

Annahme: t gegeben, erreichbar, endliche Szene, endlich viele erreichbare Punkte $\{p_1, p_2, \ldots, p_m\}$

Universelles Steuerwort!

Annahme: t gegeben, erreichbar, endliche Szene, endlich viele erreichbare Punkte $\{p_1, p_2, \ldots, p_m\}$

Lemma 7.8 Es gibt ein universelles Steuerwort w über $\Sigma = \{T, L, R\}$, das den Agenten von jedem Startpunkt p_i zum Ziel t führt, falls es einen Weg zum Ziel t gibt.

Universelles Steuerwort!

Annahme: t gegeben, erreichbar, endliche Szene, endlich viele erreichbare Punkte $\{p_1, p_2, \ldots, p_m\}$

Lemma 7.8 Es gibt ein universelles Steuerwort w über $\Sigma = \{T, L, R\}$, das den Agenten von jedem Startpunkt p_i zum Ziel t führt, falls es einen Weg zum Ziel t gibt.

Beweis:

Universelles Steuerwort!

Annahme: t gegeben, erreichbar, endliche Szene, endlich viele erreichbare Punkte $\{p_1, p_2, \ldots, p_m\}$

Lemma 7.8 Es gibt ein universelles Steuerwort w über $\Sigma = \{T, L, R\}$, das den Agenten von jedem Startpunkt p_i zum Ziel t führt, falls es einen Weg zum Ziel t gibt.

Beweis: Worte sukzessive erweitern!

Universelles Steuerwort!

Annahme: t gegeben, erreichbar, endliche Szene, endlich viele erreichbare Punkte $\{p_1, p_2, \ldots, p_m\}$

Lemma 7.8 Es gibt ein universelles Steuerwort w über $\Sigma = \{T, L, R\}$, das den Agenten von jedem Startpunkt p_i zum Ziel t führt, falls es einen Weg zum Ziel t gibt.

Beweis: Worte sukzessive erweitern!

Theorem 7.7 Im Prinzip genügen Zielkompass und Tastsensor, um in unbekannter Umgebung einen Zielpunkt zu finden.