Zusammenfassung Pledge Alg./Bug Alg

Elmar Langetepe
University of Bonn

Beweis Korrektheit!

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

- Falls der Agent keinen Ausweg finden kann, ist er eingeschlossen!
- Annahme: Roboter erreicht Rand nicht.
- **Lemma 7.3:** Pfad $\Pi_\circ$ stets aufs Neue
- **Lemma 7.4:** Hat keine Kreuzungen
- Zwei Orientierungen: 1) Im UZS 2) Gegen den UZS
- 2) stets $+2\pi$ pro Runde, wird positiv, Widerspruch **Lemma 7.2**
- Also 1) stets -2π pro Runde
- Irgendwann nur noch negativ, Hindernis wird nicht verlassen
- Orientierung: Im UZS $\Rightarrow$ Innenhof! Ausweglos!

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

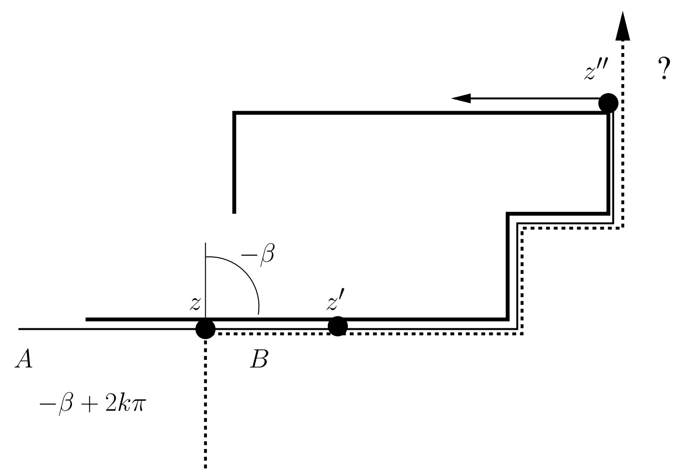
Lemma 7.3 Falls der Roboter nicht ins Freie gelangt, durchläuft er — bis auf ein endliches Anfangsstück — stets den gleichen Weg aufs Neue.

- Weg des Roboters ist polygonale 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 nicht mehr $\Rightarrow$ gleicher Weg immer wieder
- 2. Fall: Löst sich mehr als $|S|$ mal (unendlich oft)
- $\Rightarrow$ zweimal Zählerstand 0 an gleicher Ecke, Widerspruch zur Ann.!

Beweis Korrektheit: Endlosweg schnittfrei

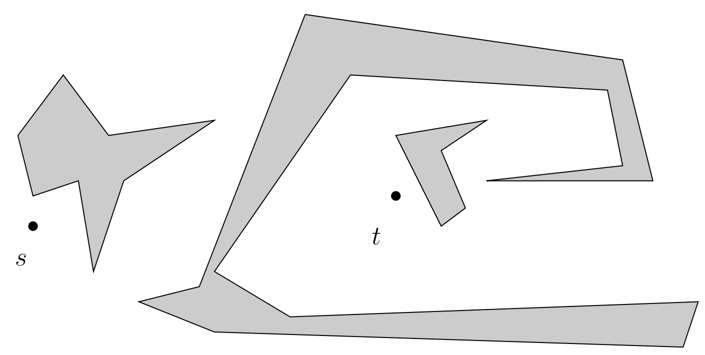
Lemma 7.4 Annahme Roboter kann nicht entkommen. $\Pi_\circ$ stets aufs Neue. $\Pi_\circ$ ist kreuzungsfrei, darf berühren!

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



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

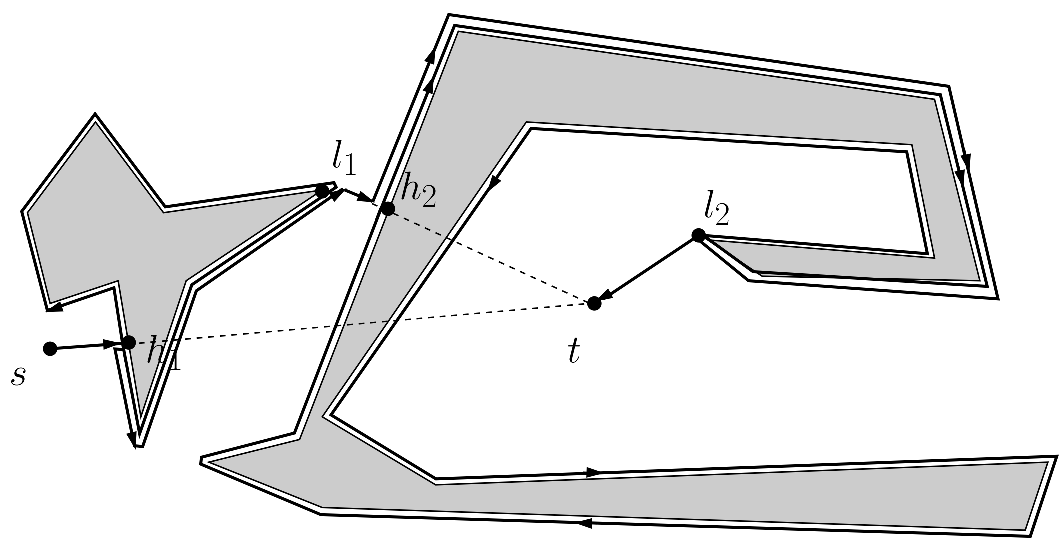


BUG Strategie: Lumelsky/Stepanov

- Aktionen:

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

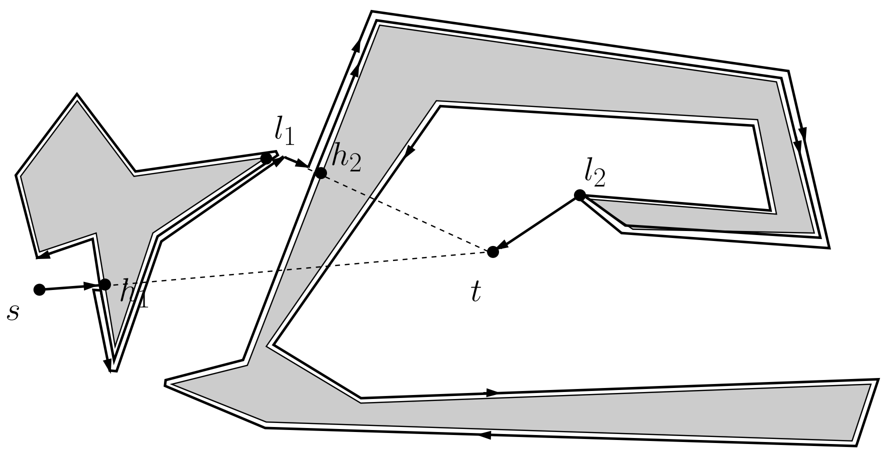
- Leave-Points l_i, Hit-Points h_i, Simpleste BUG Variante



Korrektheit BUG Strategie

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

- Folge von Hit- und Leave-Points h_i, l_i
- $|st| \geq |h_1 t| \geq |l_1 t| \ldots \geq |h_k t| \geq |l_k t|$
- $l_i \neq l_j$, neues Hindernis, sonst keine freie Bew.
- Endlich viele Polygone $\Rightarrow$ Korrektheit

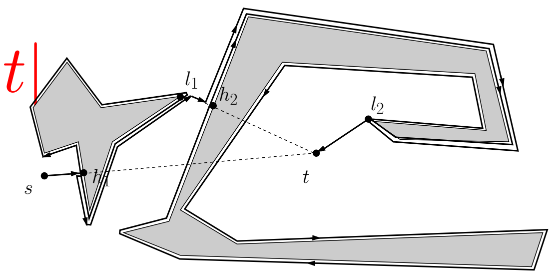


Laufzeit BUG1 Strategie

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

- Umrundungen $\frac{3}{2}\sum \mathtt{UP_i}$

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

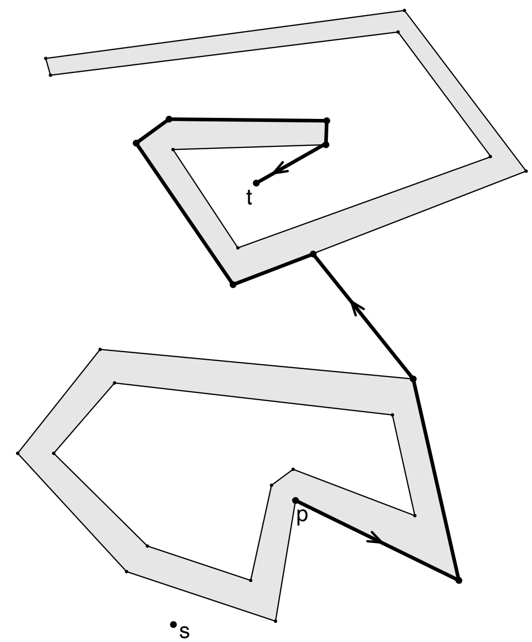


Einfache Bewegungen genügen stets!

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!

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.

Worte sukz. erweitern! $w_i = w(p_1, \ldots, p_i)$ angew. auf p_{i+1} landet bei q_{i+1}, verw. $w(q_{i+1})$: $w_{i+1} = w_i w(q_{i+1}) = w(p_1, \ldots, p_i, p_{i+1})$

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

Alg: Alle Wörter wachsender Länge sukzessive anwenden, irgendw. ist w_m am Zug!

Kapitel Buch

Kapitel 7 Seite 321 oben – S. 332 mitte