

# Grundlagen der Künstlichen Intelligenz

## 17. Klassische Suche: A\*: Vollständigkeit und Komplexität

Malte Helmert

Universität Basel

4. April 2014

# Klassische Suche: Überblick

## Kapitelüberblick klassische Suche:

- 3.–5. Einführung
- 6.–9. Basisalgorithmen
- 10.–17. heuristische Algorithmen
  - 10. Heuristiken
  - 11. Analyse von Heuristiken
  - 12. Bestensuche als Graphensuche
  - 13. Gierige Bestensuche,  $A^*$ , Weighted  $A^*$
  - 14. IDA $^*$
  - 15.  $A^*$ : Optimalität, Teil I
  - 16.  $A^*$ : Optimalität, Teil II
  - 17.  $A^*$ : Vollständigkeit und Komplexität

# Vollständigkeit

# Vollständigkeit von $A^*$

## Vollständigkeit von $A^*$ :

- $A^*$  ist vollständig für **sichere** Heuristiken  
(wie andere Bestensuchalgorithmen als Graphensuche)
- Anders als gierige Bestensuche ist  $A^*$  mit sicheren Heuristiken immer noch **semi-vollständig**, wenn **keine** Duplikateliminierung durchgeführt wird und alle Aktionskosten grösser als 0 sind.  
(Warum?)
- Das ist sogar für **unendliche Zustandsräume wahr**  
(in dieser Vorlesung nicht betrachtet),  
wenn jeder Zustand endlich viele Nachfolger hat  
und das Infimum der Aktionskosten grösser als 0 ist.

# Zeit- und Speicheraufwand von $A^*$

# Zeitaufwand von $A^*$ (1)

## Was ist der Zeitaufwand von $A^*$ ?

- hängt stark von der Qualität der Heuristik ab
- ein Extremfall:  $h = 0$  für alle Zustände
  - ↪  $A^*$  identisch mit uniformer Kostensuche
- ein anderer Extremfall:  $h = h^*$  und  $cost(a) > 0$  für alle Aktionen  $a$ 
  - ↪  $A^*$  expandiert nur Knoten entlang einer optimalen Lösung
  - ↪  $O(\ell^*)$  expandierte Knoten,  $O(\ell^* b)$  erzeugte Knoten, wobei
    - $\ell^*$ : Länge der gefundenen optimalen Lösung
    - $b$ : Verzweigungsgrad

# Zeitaufwand von $A^*$ (2)

## Genauere Analyse:

- Abhängigkeit der Laufzeit von  $A^*$  vom **Heuristik-Fehler**

## Beispiel:

- Einheitskosten-Probleme mit
- **konstantem Verzweigungsgrad** und
- **konstantem absoluten Fehler**:  $|h^*(s) - h(s)| \leq c$  für alle  $s \in S$

## Zeitaufwand:

- **Zustandsraum ist Baum**: Laufzeit von  $A^*$   
skaliert linear in Lösungslänge (Pohl 1969; Gaschnig 1977)
- **allgemeine Zustandsräume**: Laufzeit von  $A^*$   
skaliert exponentiell in Lösungslänge (Helmert & Röger 2008)

# Zeitaufwand durch Reopening

## Wie aufwändig ist Reopening?

- Für die meisten praktischen Probleme und inkonsistenten, aber zulässigen Heuristiken, ist die Zahl wiedereröffneter Knoten **vernachlässigbar**.
- **Ausnahmen** existieren:  
Martelli (1977) konstruiert Zustandsräume mit  $n$  Zuständen, bei denen **exponentiell** (in  $n$ ) viele Zustände von  $A^*$  wiedereröffnet werden  
( $\rightsquigarrow$  exponentiell schlechter als uniforme Kostensuche)



# Speicheraufwand von $A^*$

## Speicheraufwand von $A^*$ :

- alle erzeugten Knoten werden potenziell für immer gespeichert (Ausnahme: Duplikate)
- ↪ Speicheraufwand kann so gross wie Zeitaufwand sein (potenziell niedriger wegen Duplikaten und Reopening)

# Praktische Auswertung

# Praktische Auswertung von $A^*$ (1)

7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

Goal State

$h_1$ : Anzahl Kacheln in falscher Position (**misplaced tiles**)

$h_2$ : Summe der Distanzen der Kacheln zu deren Zielposition  
(**Manhattan-Distanz**)

## Praktische Auswertung von $A^*$ (2)

- jede Zeile mittelt über 100 Instanzen (Anfangszustände)
- **keine** Duplikateliminierung (Baumsuche)

	erzeugte Knoten		
$h^*$	iter. Tiefensuche	$A^*(h_1)$	$A^*(h_2)$
2	10	6	6
4	112	13	12
6	680	20	18
8	6384	39	25
10	47127	93	39
12	3644035	227	73
14	–	539	113
16	–	1301	211
18	–	3056	363
20	–	7276	676
22	–	18094	1219
24	–	39135	1641

# Zusammenfassung

# Zusammenfassung (1)

Weitere Eigenschaften von  $A^*$ :

- **Vollständigkeit:**
  - **vollständig** für sichere Heuristiken
  - **semi-vollständig**, wenn als Baumsuche statt Graphensuche und Aktionskosten positiv
  - **semi-vollständig** für unendliche Zustandsräume mit endlicher Verzweigung, wenn Infimum der Aktionskosten positiv

# Zusammenfassung (2)

Weitere Eigenschaften von  $A^*$ :

- **Zeitkomplexität:**
  - Null-Heuristik: wie uniforme Kostensuche
  - perfekte Heuristik, positive Aktionskosten:  
nur optimale Lösung wird expandiert
  - allgemein: abhängig von Qualität der Heuristik  
und Graphstruktur des Zustandsraums
  - Reopening normalerweise vernachlässigbar;  
in degenerierten Fällen extrem teuer
- **Speicherkomplexität:** ähnlich Zeitkomplexität