Grundlagen der Künstlichen Intelligenz

19. Klassische Suche: A*: Optimalität, Teil II

Malte Helmert

Universität Basel

30. März 2015

Klassische Suche: Überblick

Kapitelüberblick klassische Suche:

- 5.–7. Grundlagen
- 8.–12. Basisalgorithmen
- 13.–20. heuristische Algorithmen
  - 13. Heuristiken
  - 14. Analyse von Heuristiken
  - 15. Bestensuche als Graphensuche
  - 16. Gierige Bestensuche, A*, Weighted A*
  - 17. IDA*
  - 18. A*: Optimalität, Teil I
  - 19. A*: Optimalität, Teil II
  - 20. A*: Vollständigkeit und Komplexität

Optimalität von A* ohne Reopening

Optimalität von A* ohne Reopening (1)

Satz (Optimalität von A* ohne Reopening)

A ohne Reopening ist optimal, wenn die verwendete Heuristik konsistent und zielerkennend ist.*

Beweis.

Wir verändern (als Gedankenexperiment) A* so,
dass es die Suche fortsetzt, wenn ein Zielzustand expandiert wird.

$\rightsquigarrow$ expandiert weiter, bis *open* leer ist

$\rightsquigarrow$ alle erreichbaren Zustände werden erreicht
(ausser solchen "hinter" Zuständen mit $h(s) = \infty$,
und hinter diesen können keine Lösungen liegen) ...

Optimalität von A* ohne Reopening (2)

Beweis (Fortsetzung).

Sei n der erste expandierte Zielknoten, s sein (Ziel-) Zustand.

- Optimale-Pfade-Lemma: $f(n) = f_h^*(s) = g^*(s) + h(s)$
- h ist zielerkennend: $h(s) = 0$
- zusammen: $f(n) = g^*(s)$

Optimalität von A* ohne Reopening (2)

Beweis (Fortsetzung).

Sei n der erste expandierte Zielknoten, s sein (Ziel-) Zustand.

- Optimale-Pfade-Lemma: $f(n) = f_h^*(s) = g^*(s) + h(s)$
- h ist zielerkennend: $h(s) = 0$
- zusammen: $f(n) = g^*(s)$

Sei n' ein später expandierter Zielknoten mit Zielzustand s'.

- analog: $f(n') = g^*(s')$
- Monotonielemma (Teil 3): $f(n) \leq f(n')$
- zusammen: $g^*(s) \leq g^*(s')$

Optimalität von A* ohne Reopening (2)

Beweis (Fortsetzung).

Sei n der erste expandierte Zielknoten, s sein (Ziel-) Zustand.

- Optimale-Pfade-Lemma: $f(n) = f_h^*(s) = g^*(s) + h(s)$
- h ist zielerkennend: $h(s) = 0$
- zusammen: $f(n) = g^*(s)$

Sei n' ein später expandierter Zielknoten mit Zielzustand s'.

- analog: $f(n') = g^*(s')$
- Monotonielemma (Teil 3): $f(n) \leq f(n')$
- zusammen: $g^*(s) \leq g^*(s')$

Wir sehen, dass optimale Pfadkosten für später erreichte Zielzustände mindestens so hoch wie für s sind.

$\rightsquigarrow$ erste gefundene Lösung ist optimal

$\rightsquigarrow$ Fortsetzung der Suche an dieser Stelle ist unnötig □

Diskussion: warum Konsistenz?

- Wir haben Konsistenz und Zielerkennung gefordert.
- Wir wissen: aus diesen Eigenschaften folgt Zulässigkeit.
- Reicht Zulässigkeit allein aus, damit A* optimal ist?
- Im allgemeinen ohne Reopening nein:
  zulässige, aber inkonsistente Heuristiken können
  das Optimale-Pfade-Lemma verletzen,
  und das kann zu suboptimalen Lösungen führen.
- Abhilfe: Reopening

Optimalität von A* mit Reopening

Vorbemerkungen

- Wir beweisen nun die Optimalität von A* mit Reopening bei Verwendung von zulässigen Heuristiken.
- Dazu benötigen wir wieder einige Definitionen und zwei Lemmas.

Lösbare Zustände

Definition (lösbar)

Ein Zustand s eines Zustandsraums heisst lösbar, wenn $h^*(s) < \infty$ gilt.

Erledigte Zustände in A* mit Reopening

Definition (erledigt)

Ein Zustand s heisst zu einem gegebenen Zeitpunkt während der Ausführung von A* mit Reopening erledigt, wenn s in *distances* liegt und $\mathit{distances}[s] = g^*(s)$ gilt.

Optimale-Fortsetzungs-Lemma

Wir zeigen nun das erste wichtige Ergebnis für A* mit Reopening:

Lemma (Optimale-Fortsetzungs-Lemma)

Betrachte A mit Reopening und einer sicheren Heuristik zu Beginn einer beliebigen Iteration der* **while***-Schleife.*

Wenn

- *Zustand s erledigt ist,*
- *Zustand s' ein lösbarer Nachfolger von s ist und*
- *ein optimaler Pfad von s_0 zu s' der Form $\langle s_0, \ldots, s, s' \rangle$ existiert,*

dann

- *ist s' erledigt oder*
- *open enthält einen Knoten n' mit $n'.\text{state} = s'$ und $g(n') = g^*(s')$.*

Optimale-Fortsetzungs-Lemma: Intuition

(Beweis folgt auf den nächsten Folien.)

Intuitiv sagt das Lemma so etwas aus wie:

> *Wenn noch kein optimaler Pfad*
> *zu einem Zustand gefunden wurde,*
> *dann enthält open einen „guten“ Knoten,*
> *der zum Finden eines optimalen Pfads*
> *zu diesem Zustand beiträgt.*

(Dafür muss man das Lemma gegebenenfalls mehrmals entlang eines Pfades anwenden.)

Optimale-Fortsetzungs-Lemma: Beweis (1)

Beweis.

Da s erledigt ist, wurde in einer früheren Iteration („Iteration A“) $\mathit{distances}[s] := g^*(s)$ gesetzt.

Optimale-Fortsetzungs-Lemma: Beweis (1)

Beweis.

Da s erledigt ist, wurde in einer früheren Iteration („Iteration A“) $\mathit{distances}[s] := g^*(s)$ gesetzt.

Dann muss in Iteration A ein Knoten n mit $n.\text{state} = s$ und $g(n) = g^*(s)$ aus *open* entnommen worden sein.

Optimale-Fortsetzungs-Lemma: Beweis (1)

Beweis.

Da s erledigt ist, wurde in einer früheren Iteration („Iteration A“) $\mathit{distances}[s] := g^*(s)$ gesetzt.

Dann muss in Iteration A ein Knoten n mit $n.\text{state} = s$ und $g(n) = g^*(s)$ aus *open* entnommen worden sein.

Da Iteration A eine frühere Iteration als die betrachtete ist, hat A* nicht in Iteration A terminiert.
Folglich wurde n in Iteration A expandiert.

Optimale-Fortsetzungs-Lemma: Beweis (1)

Beweis.

Da s erledigt ist, wurde in einer früheren Iteration („Iteration A“) $\mathit{distances}[s] := g^*(s)$ gesetzt.

Dann muss in Iteration A ein Knoten n mit $n.\text{state} = s$ und $g(n) = g^*(s)$ aus *open* entnommen worden sein.

Da Iteration A eine frühere Iteration als die betrachtete ist, hat A* nicht in Iteration A terminiert.
Folglich wurde n in Iteration A expandiert.

Bei dieser Expansion wurde der Nachfolger s' von s betrachtet.
Da s' lösbar ist, gilt $h^*(s') < \infty$. Da h sicher ist, folgt $h(s') < \infty$.
Es wurde also ein Nachfolgerknoten n' für s' erzeugt.

Optimale-Fortsetzungs-Lemma: Beweis (1)

Beweis.

Da s erledigt ist, wurde in einer früheren Iteration („Iteration A“) $\mathit{distances}[s] := g^*(s)$ gesetzt.

Dann muss in Iteration A ein Knoten n mit $n.\text{state} = s$ und $g(n) = g^*(s)$ aus *open* entnommen worden sein.

Da Iteration A eine frühere Iteration als die betrachtete ist, hat A* nicht in Iteration A terminiert.
Folglich wurde n in Iteration A expandiert.

Bei dieser Expansion wurde der Nachfolger s' von s betrachtet.
Da s' lösbar ist, gilt $h^*(s') < \infty$. Da h sicher ist, folgt $h(s') < \infty$.
Es wurde also ein Nachfolgerknoten n' für s' erzeugt.

Der Knoten n' genügt den Kriterien des Lemmas.
Die Kriterien des Lemmas waren also nach Iteration A erfüllt.

Optimale-Fortsetzungs-Lemma: Beweis (2)

Beweis (Fortsetzung).

Um den Beweis abzuschliessen zeigen wir: wenn die Konsequenz des Lemmas zu Beginn einer Iteration erfüllt ist,
ist sie auch zu Beginn der folgenden Iteration erfüllt.

Optimale-Fortsetzungs-Lemma: Beweis (2)

Beweis (Fortsetzung).

Um den Beweis abzuschliessen zeigen wir: wenn die Konsequenz des Lemmas zu Beginn einer Iteration erfüllt ist,
ist sie auch zu Beginn der folgenden Iteration erfüllt.

- Wenn s' zu Beginn einer Iteration erledigt ist,
  bleibt s' bis zur Terminierung erledigt.

Optimale-Fortsetzungs-Lemma: Beweis (2)

Beweis (Fortsetzung).

Um den Beweis abzuschliessen zeigen wir: wenn die Konsequenz des Lemmas zu Beginn einer Iteration erfüllt ist,
ist sie auch zu Beginn der folgenden Iteration erfüllt.

- Wenn s' zu Beginn einer Iteration erledigt ist,
  bleibt s' bis zur Terminierung erledigt.
- Wenn *open* zu Beginn einer Iteration einen Knoten n'
  mit $n'.\text{state} = s'$ und $g(n') = g^*(s')$ enthält,
  dann verbleibt dieser in dieser Iteration entweder in *open*,
  oder er wird entnommen und am Ende der Iteration
  ist s' erledigt. (Dies schliesst den Fall ein,
  dass s' schon zu Beginn der Iteration erledigt war.)

f-Schranken-Lemma

Wir benötigen noch ein weiteres Lemma:

Lemma (f-Schranken-Lemma)

Betrachte A mit Reopening und einer zulässigen Heuristik, angewandt auf einen lösbaren Zustandsraum mit optimalen Lösungskosten c^*.*

Dann enthält open zu Beginn jeder Iteration der **while***-Schleife einen Knoten n mit $f(n) \leq c^*$.*

f-Schranken-Lemma: Beweis (1)

Beweis.

Betrachte die Situation zu Beginn einer beliebigen Iteration der **while**-Schleife.

Sei $\langle s_0, \ldots, s_n \rangle$ eine optimale Lösung.

f-Schranken-Lemma: Beweis (1)

Beweis.

Betrachte die Situation zu Beginn einer beliebigen Iteration der **while**-Schleife.

Sei $\langle s_0, \ldots, s_n \rangle$ eine optimale Lösung.

Sei s_i der erste Zustand in dieser Folge, der nicht erledigt ist.

(Nicht alle Zustände in der Folge können erledigt sein:
s_n ist ein Zielzustand, und wenn ein Zielzustand
in *distances* eingetragen wird, terminiert A*.)

f-Schranken-Lemma: Beweis (2)

Beweis (Fortsetzung).

Fall 1: $i = 0$

Da s_0 nicht erledigt ist, befinden wir uns vor der ersten Iteration der **while**-Schleife.

f-Schranken-Lemma: Beweis (2)

Beweis (Fortsetzung).

Fall 1: $i = 0$

Da s_0 nicht erledigt ist, befinden wir uns
vor der ersten Iteration der **while**-Schleife.

Da der Zustandsraum lösbar und h zulässig ist, gilt $h(s_0) < \infty$.

f-Schranken-Lemma: Beweis (2)

Beweis (Fortsetzung).

Fall 1: $i = 0$

Da s_0 nicht erledigt ist, befinden wir uns vor der ersten Iteration der **while**-Schleife.

Da der Zustandsraum lösbar und h zulässig ist, gilt $h(s_0) < \infty$.

Daher enthält *open* den Wurzelknoten n_0.

f-Schranken-Lemma: Beweis (2)

Beweis (Fortsetzung).

Fall 1: $i = 0$

Da s_0 nicht erledigt ist, befinden wir uns
vor der ersten Iteration der **while**-Schleife.

Da der Zustandsraum lösbar und h zulässig ist, gilt $h(s_0) < \infty$.

Daher enthält *open* den Wurzelknoten n_0.

Wir erhalten: $f(n_0) = g(n_0) + h(s_0) = 0 + h(s_0) \leq h^*(s_0) = c^*$,
wobei „$\leq$“ die Zulässigkeit von h verwendet.

Damit ist der Beweis für diesen Fall erbracht.

f-Schranken-Lemma: Beweis (3)

Beweis (Fortsetzung).

Fall 2: $i > 0$

Dann ist s_{i-1} erledigt und s_i nicht erledigt.
Ferner ist s_i ein lösbarer Nachfolger von s_{i-1}
und $\langle s_0, \ldots, s_{i-1}, s_i \rangle$ ein optimaler Pfad von s_0 zu s_i.

f-Schranken-Lemma: Beweis (3)

Beweis (Fortsetzung).

Fall 2: $i > 0$

Dann ist s_{i-1} erledigt und s_i nicht erledigt.
Ferner ist s_i ein lösbarer Nachfolger von s_{i-1}
und $\langle s_0, \ldots, s_{i-1}, s_i \rangle$ ein optimaler Pfad von s_0 zu s_i.

Wir können somit das Optimale-Fortsetzungs-Lemma anwenden (mit $s = s_{i-1}$ und $s' = s_i$). Es folgt:

(A) s_i ist erledigt, oder

(B) *open* enthält n' mit $n'.\text{state} = s_i$ und $g(n') = g^*(s_i)$.

f-Schranken-Lemma: Beweis (3)

Beweis (Fortsetzung).

Fall 2: $i > 0$

Dann ist s_{i-1} erledigt und s_i nicht erledigt.
Ferner ist s_i ein lösbarer Nachfolger von s_{i-1}
und $\langle s_0, \ldots, s_{i-1}, s_i \rangle$ ein optimaler Pfad von s_0 zu s_i.

Wir können somit das Optimale-Fortsetzungs-Lemma anwenden (mit $s = s_{i-1}$ und $s' = s_i$). Es folgt:

(A) s_i ist erledigt, oder

(B) *open* enthält n' mit $n'.\text{state} = s_i$ und $g(n') = g^*(s_i)$.

Da (A) falsch ist, muss (B) gelten.

f-Schranken-Lemma: Beweis (3)

Beweis (Fortsetzung).

Fall 2: $i > 0$

Dann ist s_{i-1} erledigt und s_i nicht erledigt.
Ferner ist s_i ein lösbarer Nachfolger von s_{i-1}
und $\langle s_0, \ldots, s_{i-1}, s_i \rangle$ ein optimaler Pfad von s_0 zu s_i.

Wir können somit das Optimale-Fortsetzungs-Lemma anwenden (mit $s = s_{i-1}$ und $s' = s_i$). Es folgt:

(A) s_i ist erledigt, oder

(B) *open* enthält n' mit $n'.\text{state} = s_i$ und $g(n') = g^*(s_i)$.

Da (A) falsch ist, muss (B) gelten.

Wir erhalten:
$f(n') = g(n') + h(s_i) = g^*(s_i) + h(s_i) \leq g^*(s_i) + h^*(s_i) = c^*$,
wobei „$\leq$“ die Zulässigkeit von h verwendet. □

Optimalität von A* mit Reopening

Wir können nun das Hauptergebnis zeigen:

Satz (Optimalität von A* mit Reopening)

A mit Reopening ist optimal,*
wenn die verwendete Heuristik zulässig ist.

Optimalität von A* mit Reopening: Beweis

Beweis.

Angenommen, A* wäre nicht optimal.

Es gibt also einen Zustandsraum mit optimalen Lösungskosten c^*, für den A* eine Lösung mit Kosten $c > c^*$ zurückliefert.

Optimalität von A* mit Reopening: Beweis

Beweis.

Angenommen, A* wäre nicht optimal.

Es gibt also einen Zustandsraum mit optimalen Lösungskosten c^*, für den A* eine Lösung mit Kosten $c > c^*$ zurückliefert.

Dann entnimmt A* in der letzten Iteration einen Zielknoten n mit $g(n) = c > c^*$ aus *open*.

Optimalität von A* mit Reopening: Beweis

Beweis.

Angenommen, A* wäre nicht optimal.

Es gibt also einen Zustandsraum mit optimalen Lösungskosten c^*, für den A* eine Lösung mit Kosten $c > c^*$ zurückliefert.

Dann entnimmt A* in der letzten Iteration einen Zielknoten n mit $g(n) = c > c^*$ aus *open*.

Mit $h(n.\text{state}) = 0$ (da h zulässig und damit zielerkennend) folgt:

Optimalität von A* mit Reopening: Beweis

Beweis.

Angenommen, A* wäre nicht optimal.

Es gibt also einen Zustandsraum mit optimalen Lösungskosten c^*, für den A* eine Lösung mit Kosten $c > c^*$ zurückliefert.

Dann entnimmt A* in der letzten Iteration einen Zielknoten n mit $g(n) = c > c^*$ aus *open*.

Mit $h(n.\text{state}) = 0$ (da h zulässig und damit zielerkennend) folgt:

$f(n) = g(n) + h(n.\text{state}) = g(n) + 0 = g(n) = c > c^*$.

Optimalität von A* mit Reopening: Beweis

Beweis.

Angenommen, A* wäre nicht optimal.

Es gibt also einen Zustandsraum mit optimalen Lösungskosten c^*, für den A* eine Lösung mit Kosten $c > c^*$ zurückliefert.

Dann entnimmt A* in der letzten Iteration einen Zielknoten n mit $g(n) = c > c^*$ aus *open*.

Mit $h(n.\text{state}) = 0$ (da h zulässig und damit zielerkennend) folgt:

$$f(n) = g(n) + h(n.\text{state}) = g(n) + 0 = g(n) = c > c^*.$$

A* entnimmt immer einen Knoten mit minimalem f-Wert aus *open*. Da dieser f-Wert grösser als c^* ist, ergibt sich ein Widerspruch zum f-Schranken-Lemma. □

Zusammenfassung

Zusammenfassung

- A* ohne Reopening ist optimal, wenn die verwendete Heuristik konsistent und zielerkennend ist.
- A* mit Reopening ist optimal,
  wenn die verwendete Heuristik zulässig ist.