Übung 3

Steffen Planthaber

DFKI - Labor Bremen & Universität Bremen

Forschungsgruppe Robotik

Director: Prof. Dr. Frank Kirchner

www.dfki.de/robotics

robotics@dfki.de

A*

- A* ist ein heuristischer Suchalgorithmus
- Knoten, die wahrscheinlich schneller zum Ziel führen, werden zuerst untersucht (expandiert)
  - Abschätzung durch die Heuristik
  - $f(x) = g(x) + h(x)$

Genutzt werden dabei:

- Bekannte Knoten
- Expandierte knoten

A*

Expandieren heißt, alle möglichen Folgeknoten in die Liste der bekannten Knoten aufzunehmen (openList)

- Bekannte Knoten
  - Knoten die noch nicht untersucht wurden
- Expandierte knoten
  - Knoten die untersucht wurden

A*

- Nach dem Expandieren wird der Knoten in die Liste der expandierten (untersuchten) Knoten aufgenommen (closedList)
- Alle Knoten speichern ihren Vorgänger.
- Wenn beim expandieren ein Knoten bereits in der closedList ist, wird nicht erneut expandiert, aber ggf. und der Vorgänger aktualisiert, wenn g(x) kleiner ist

A*

Anforderungen an die Heuristik

- Unterschätzend!
- Monoton (für die Übung, Ausnahmen möglich)

Planen

Arten von Planung

1. Klassisches Planen in festen Umgebungen
2. Planen in veränderlichen bzw. nicht überschaubaren Umgebungen

Planen

Voraussetzungen für klassisches Planen

Die Umgebung muss folgende Voraussetzungen erfüllen:

- Überschaubar
- Deterministisch
- Endlich
- Statisch (Nur der Agent bewirkt Veränderungen)
- Diskret (Zeit,Aktionen,Objekte und Effekte)

Unterschieden werden dabei:

- Zustände (Gegebenheiten)
- Ziele
- Aktionen

STRIPS - Zustände und Ziele

STRIPS (Stanford Research Institute Problem Solver) ist eine Sparache um Probleme zu definieren.

Zustände werden mittels positiver Literale beschrieben, es sind weder Variablen noch Funktionen erlaubt:
$Home \wedge Happy$ oder $At(Plane_1, Melbourne) \wedge At(Plane_2, Sydney)$

Ziele sind besondere Zustände, die erfüllt sind, wenn alle Literale des Zieles im aktuellem Zustand enthalten sind.
$Reich \wedge$ Berühmt $\wedge$ Unglücklich erfüllt das Ziel $Reich \wedge$ Berühmt

STRIPS - Aktionen

$Action(Fly(p, from, to),$
$PRECOND : At(p, from) \wedge Plane(p) \wedge Airport(from) \wedge Airport(to)$
$EFFECT : \neg At(p, from) \wedge At(p, to))$
“p”,“from” und “to” sind hierbei Variablen

Um eine Aktion auszuführen müssen alle Literale der PRECOND(ition) erfüllt sein, nach der Ausführung trifft der Effekt ein, wobei $\neg$ ein Löschen aus dem Zustand bedeutet. Positive Literale werden hinzugefügt. Literale im Zustand können nicht doppelt vorkommen.

Planen

$$
\begin{aligned}
&Init(At(C_1, SFO) \land At(C_2, JFK) \land At(P_1, SFO) \land At(P_2, JFK) \\
&\quad \land Cargo(C_1) \land Cargo(C_2) \land Plane(P_1) \land Plane(P_2) \\
&\quad \land Airport(JFK) \land Airport(SFO)) \\
&Goal(At(C_1, JFK) \land At(C_2, SFO)) \\
&Action(Load(c, p, a), \\
&\quad \text{PRECOND: } At(c, a) \land At(p, a) \land Cargo(c) \land Plane(p) \land Airport(a) \\
&\quad \text{EFFECT: } \neg At(c, a) \land In(c, p)) \\
&Action(Unload(c, p, a), \\
&\quad \text{PRECOND: } In(c, p) \land At(p, a) \land Cargo(c) \land Plane(p) \land Airport(a) \\
&\quad \text{EFFECT: } At(c, a) \land \neg In(c, p)) \\
&Action(Fly(p, from, to), \\
&\quad \text{PRECOND: } At(p, from) \land Plane(p) \land Airport(from) \land Airport(to) \\
&\quad \text{EFFECT: } \neg At(p, from) \land At(p, to))
\end{aligned}
$$

Wie sieht die Lösung aus?

Beispiel aus “Artificial Intelligence A Modern Approach”

STRIPS - Beispiel

$Load(C_1, P_1, SFO), Fly(P_1, SFO, JFK),$
$Load(C_2, P_2, JFK), Fly(P_2, JFK, SFO)$

STRIPS vs. ADL (Action Description Language)

STRIPS **Language**	ADL **Language**
Only positive literals in states: $Poor \wedge Unknown$	Positive and negative literals in states: $\neg Rich \wedge \neg Famous$
Closed World Assumption: Unmentioned literals are false.	Open World Assumption: Unmentioned literals are unknown.
Effect $P \wedge \neg Q$ means add P and delete Q.	Effect $P \wedge \neg Q$ means add P and $\neg Q$ and delete $\neg P$ and Q.
Only ground literals in goals: $Rich \wedge Famous$	Quantified variables in goals: $\exists x At(P_1, x) \wedge At(P_2, x)$ is the goal of having P_1 and P_2 in the same place.
Goals are conjunctions: $Rich \wedge Famous$	Goals allow conjunction and disjunction: $\neg Poor \wedge (Famous \vee Smart)$
Effects are conjunctions.	Conditional effects allowed: **when** P: E means E is an effect only if P is satisfied.
No support for equality.	Equality predicate $(x = y)$ is built in.
No support for types.	Variables can have types, as in $(p : Plane)$.

Aus "Artificial Intelligence A Modern Approach"

Spare Tire Problem

Man ist liegen geblieben, der Ersatzreifen ist im Auto, man kann das Auto aber auch über Nacht stehen lassen.

$Init(At(Flat, Axle) \wedge At(Spare, Trunk))$

$Goal(At(Spare, Axle))$

Welche Aktionen benötigt man?

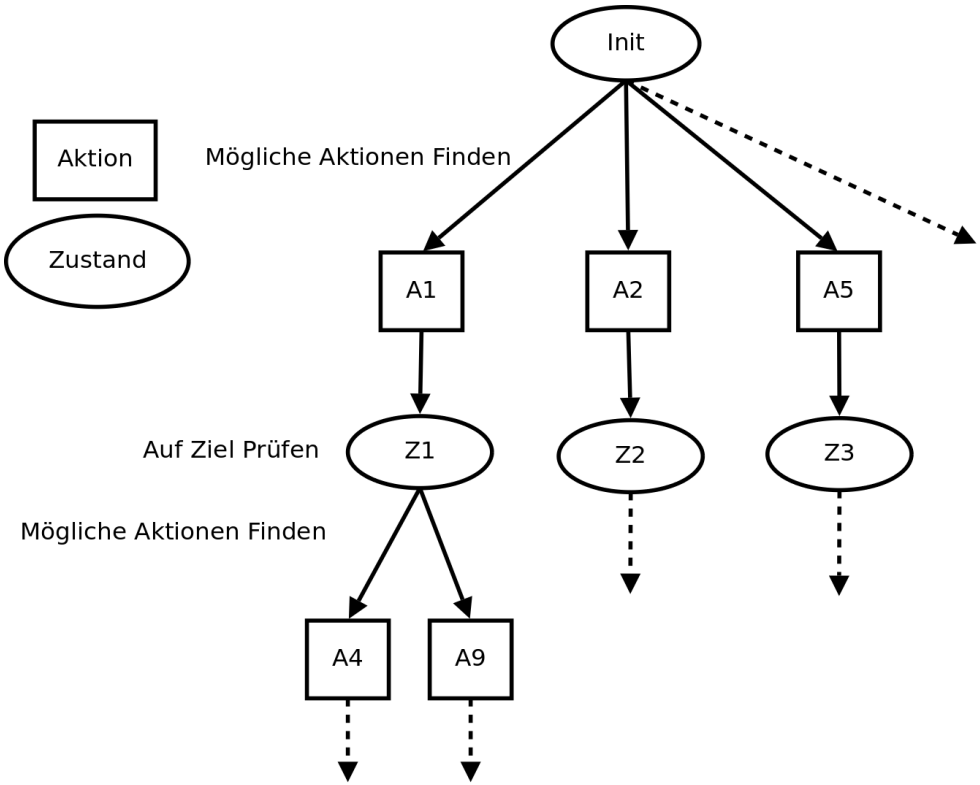
Spare Tire Problem

$$
\begin{aligned}
&Init(At(Flat, Axle) \ \wedge \ At(Spare, Trunk)) \\
&Goal(At(Spare, Axle)) \\
&Action(Remove(Spare, Trunk), \\
&\quad \text{PRECOND: } At(Spare, Trunk) \\
&\quad \text{EFFECT: } \neg\, At(Spare, Trunk) \ \wedge \ At(Spare, Ground)) \\
&Action(Remove(Flat, Axle), \\
&\quad \text{PRECOND: } At(Flat, Axle) \\
&\quad \text{EFFECT: } \neg\, At(Flat, Axle) \ \wedge \ At(Flat, Ground)) \\
&Action(PutOn(Spare, \ Axle), \\
&\quad \text{PRECOND: } At(Spare, Ground) \ \wedge \ \neg\, At(Flat, Axle) \\
&\quad \text{EFFECT: } \neg\, At(Spare, Ground) \ \wedge \ At(Spare, Axle)) \\
&Action(LeaveOvernight, \\
&\quad \text{PRECOND:} \\
&\quad \text{EFFECT: } \neg\, At(Spare, Ground) \ \wedge \ \neg\, At(Spare, Axle) \ \wedge \ \neg\, At(Spare, Trunk) \\
&\qquad\qquad \wedge \neg\, At(Flat, Ground) \ \wedge \ \neg\, At(Flat, Axle))
\end{aligned}
$$

Beispiel aus “Artificial Intelligence A Modern Approach”

Planen

Wie Lösung Suchen?



Planen

Quelle: Artificial Intelligence A Modern Approach
Stuart Russel und Peter Norvig
`http://aima.cs.berkeley.edu`

Planungskapitel ist frei zum Download
`http://aima.cs.berkeley.edu/newchap11.pdf`