

# Künstliche Intelligenz

## Übungsblatt #4

### Planen

### Version 1.3

Prof. Dr. J. Fürnkranz, Dr. G. Grieser

## Aufgabe 4.1

Wir betrachten wieder eine Situation aus der Blocksworld:



beschrieben durch die Ausgangssituation:

```
on(a, table), on(b, table), on(c, a), clear(b), clear(c),  
handempty, block(a), block(b), block(c)
```

und die Zielbeschreibung

```
on(b, a), on(c, b)
```

Gegeben seien die folgenden Aktionen aus der Vorlesung:

action: <u>unstack(X,Y)</u>	action: <u>stack(X,Y)</u>
preconditions: handempty, block(X), block(Y), clear(X), on(X,Y)	preconditions: holding(X), block(X), block(Y), clear(Y)
add: holding(X) clear(Y)	add: handempty clear(X), on(X,Y)
delete: handempty, clear(X), on(X,Y)	delete: holding(X), clear(Y)

action: putdown(X)  
 preconditions: holding(X)  
 add: handempty,  
           clear(X),  
           on(X,table)  
 delete: holding(X)

action: pickup(X)  
 preconditions: handempty,  
           block(X),  
           clear(X),  
           on(X,table)  
 add: holding(X)  
 delete: handempty,  
           clear(X),  
           on(X,table)

Erzeugen sie einen Plan mittels *Partial Order Planning*. Wenn Sie diesen Plan ausführen wollen, welche Möglichkeiten der Abarbeitung gibt es?

**Lösungsvorschlag:**

siehe Folienpräsentation

## Aufgabe 4.2

Rechnen Sie das *Flat-Tire*-Beispiel aus der Vorlesung nach, allerdings nun für das Ziel  
 $at(spare, axle), at(Flat, ground)$ .

- a) Erzeugen Sie den Graphen Schritt für Schritt. Geben Sie die komplette Menge der *Mutual Exclusions* an und erklären Sie jeweils, warum dieser gegenseitige Ausschluß besteht.

**Lösungsvorschlag:**

Zur Erinnerung, wir haben folgende Aktionen:

	action: <u>remove(spare,trunk)</u> preconditions: at(spare,trunk) add: at(spare,ground) delete: at(spare,trunk)
action: <u>remove(Flat,axle)</u> preconditions: at(Flat,axle) add: at(Flat,ground) delete: at(Flat,axle)	
	action: <u>puton(spare,axle)</u> preconditions: at(spare,ground), ¬at(Flat,axle) add: at(spare,axle) delete: at(spare,ground)

action: leaveovernight  
 preconditions:  
 add:  
 delete:           at(spare,ground),  
                   at(spare,axle),  
                   at(spare,trunk),  
                   at(flat,ground),  
                   at(flat,axle)

Wir beginnen mit dem Level  $S_0$ .

Die Beschreibung des Anfangszustandes ist:

at(flat, axle), at(spare, trunk).

Aufgrund der Annahme der abgeschlossenen Welt impliziert dies, daß auch die folgenden logischen Fakten wahr sind:

$\neg$ at(spare, axle),  $\neg$ at(spare, ground),  $\neg$ at(flat, ground).

Diese 5 Fakten sind im Level 0 möglich. Als nächstes überlegen wir, welche Aktionen mit den möglichen Fakten möglich sind, nämlich: remove(flat, trunk), remove(flat, axle) und leaveovernight. Außerdem ist für jeden Fakt die Persistenzaktion möglich.

Nach der Ausführung der einzelnen Aktionen sind folgende Fakten möglich:

Vorbedingungen	Aktion	Nachbedingungen
at(spare, trunk)	1 remove(spare, trunk)	at(spare, ground), $\neg$ at(spare, trunk)
at(flat, axle)	2 remove(flat, axle)	at(flat, ground), $\neg$ at(flat, axle)
	3 leaveovernight	$\neg$ at(spare, ground), $\neg$ at(spare, axle), $\neg$ at(spare, trunk), $\neg$ at(flat, ground), $\neg$ at(flat, axle)
at(flat, axle)	4 nop <sub>1</sub>	at(flat, axle)
at(spare, trunk)	5 nop <sub>2</sub>	at(spare, trunk)
$\neg$ at(spare, axle)	6 nop <sub>3</sub>	$\neg$ at(spare, axle)
$\neg$ at(spare, ground)	7 nop <sub>4</sub>	$\neg$ at(spare, ground)
$\neg$ at(flat, ground)	8 nop <sub>5</sub>	$\neg$ at(flat, ground)

Diese Aktionen schließen sich nun wie folgt gegenseitig aus:

- remove(spare, trunk) und leaveovernight:
  - *Inconsistency* bzgl. at(spare, ground)
  - *Interference* bzgl. at(spare, trunk)
- remove(spare, trunk) und nop<sub>2</sub>:
  - *Inconsistency* bzgl. at(spare, trunk)
  - *Interference* bzgl. at(spare, trunk)
- remove(spare, trunk) und nop<sub>4</sub>:
  - *Inconsistency* bzgl. at(spare, ground)
  - *Interference* bzgl. at(spare, ground)
- remove(flat, axle) und leaveovernight:

- *Inconsistency* bzgl.  $at(flat, ground)$
- *Interference* bzgl.  $at(flat, axle)$
- $remove(flat, axle)$  und  $nop_1$ :
  - *Inconsistency* bzgl.  $at(flat, axle)$
  - *Interference* bzgl.  $at(flat, axle)$
- $remove(flat, axle)$  und  $nop_5$ :
  - *Inconsistency* bzgl.  $at(flat, ground)$
  - *Interference* bzgl.  $at(flat, ground)$
- $leaveovernight$  und  $nop_1$ :
  - *Inconsistency* bzgl.  $at(flat, axle)$
  - *Interference* bzgl.  $at(flat, axle)$
- $leaveovernight$  und  $nop_2$ :
  - *Inconsistency* bzgl.  $at(spare, trunk)$
  - *Interference* bzgl.  $at(spare, trunk)$

Aktionen	1	2	3	4	5	6	7
1	/	/	/	/	/	/	/
2		/	/	/	/	/	/
3	X	X	/	/	/	/	/
4		X	X	/	/	/	/
5	X		X		/	/	/
6						/	/
7	X						/
8		X					

Damit ergeben sich folgende *Mutual Exclusions*:

Die Menge aller Fakten, die sich durch die obigen Aktionen ergeben können, bilden das nächste Level  $S_1$ :

1.  $at(spare, trunk)$ , 2.  $\neg at(spare, trunk)$ ,
3.  $at(flat, axle)$ , 4.  $\neg at(flat, axle)$ ,
5.  $\neg at(spare, axle)$ ,
6.  $at(flat, ground)$ , 7.  $\neg at(flat, ground)$
8.  $at(spare, ground)$ , 9.  $\neg at(spare, ground)$

Diese Fakten sind *mutual exclusive*, falls das eine die Negation des anderen ist oder falls jedes paar von Aktionen, das zu diesem Fakt führt, *mutual exclusive* ist. Im einzelnen ist dies hier:

Fakten	1	2	3	4	5	6	7	8
1	/	/	/	/	/	/	/	/
2	N	/	/	/	/	/	/	/
3			/	/	/	/	/	/
4			N	/	/	/	/	/
5					/	/	/	/
6			X			/	/	/
7						N	/	/
8	X							/
9								N

Zur Erläuterung:

- N: Diese folgenden Fakten sind mutual exclusive, weil sie Negationen voneinander sind.
- $at(spare, trunk)$  kann nur durch Aktion 5 erreicht werden. Aktion 5 ist mutex mit Aktion 1 und 3. Das heißt, wir müssen nur die Effekte von 1 und 3 mit  $at(spare, trunk)$  auf mutex testen:
  - $at(spare, ground)$  ist nur durch Aktion 1 erreichbar. Da Aktionen 1 und 5 mutex sind, sind es auch die Fakten  $at(spare, trunk)$  und  $at(spare, ground)$ .
  - $\neg at(spare, ground)$  ist erreichbar durch die Aktionen 3 und 7. Da Aktion 5 und 7 nicht mutex sind, gibt es ein paar von Aktionen, so daß parallel  $at(spare, trunk)$  und  $\neg at(spare, ground)$  erreicht werden können, deshalb sind die Fakten 1 und 9 *nicht* mutex.
  - usw.

Abschließend wird getestet, ob unsere Zielfakten komplett und nicht-mutex in  $S_1$  enthalten sind. Dies ist nicht der Fall, deshalb muß eine weitere Ebene aufgebaut werden.

Nun wird wiederum berechnet, welche Aktionen aus dem Level  $S_1$  möglich sind, die wiederum zu folgenden Fakten führen:

Vorbedingungen	Aktion	Nachbedingungen
$at(spare, trunk)$	9 $remove(spare, trunk)$	$at(spare, ground),$ $\neg at(spare, trunk)$
$at(flat, axle)$	10 $remove(flat, axle)$	$at(flat, ground),$ $\neg at(flat, axle)$
	11 $leaveovernight$	$\neg at(spare, ground),$ $\neg at(spare, axle),$ $\neg at(spare, trunk),$ $\neg at(flat, ground),$ $\neg at(flat, axle)$
$at(spare, ground),$ $\neg at(flat, axle)$	12 $puton(spare, axle)$	$at(spare, axle),$ $\neg at(spare, ground)$
$at(spare, trunk)$	13 $nop_6$	$at(spare, trunk)$
$\neg at(spare, trunk)$	14 $nop_7$	$\neg at(spare, trunk)$
$at(flat, axle)$	15 $nop_8$	$at(flat, axle)$
$\neg at(flat, axle)$	16 $nop_9$	$\neg at(flat, axle)$
$\neg at(spare, axle)$	17 $nop_{10}$	$\neg at(spare, axle)$
$at(flat, ground)$	18 $nop_{11}$	$at(flat, ground)$
$\neg at(flat, ground)$	19 $nop_{12}$	$\neg at(flat, ground)$
$at(spare, ground)$	20 $nop_{13}$	$at(spare, ground)$
$\neg at(spare, ground)$	21 $nop_{14}$	$\neg at(spare, ground)$

Damit ergeben sich folgende *Mutual Exclusions*:

Aktionen	9	10	11	12	13	14	15	16	17	18	19	20
9	/	/	/	/	/	/	/	/	/	/	/	/
10		/	/	/	/	/	/	/	/	/	/	/
11	X	X	/	/	/	/	/	/	/	/	/	/
12	X	X	X	/	/	/	/	/	/	/	/	/
13	X		X		/	/	/	/	/	/	/	/
14	X				X	/	/	/	/	/	/	/
15		X	X	X			/	/	/	/	/	/
16	X	X					X	/	/	/	/	/
17				X					/	/	/	/
18			X							/	/	/
19		X								X	/	/
20	X		X	X								/
21	X			X								X

Anmerkung: Hier kommen jetzt zum ersten Mal Competing Needs ins Spiel. Da beispielsweise die Vorbedingung  $at(spare, trunk)$  mutex zu  $at(spare, ground)$  ist, haben die Aktionen  $remove(spare, trunk)$  und  $puton(spare, axle)$  competing needs.

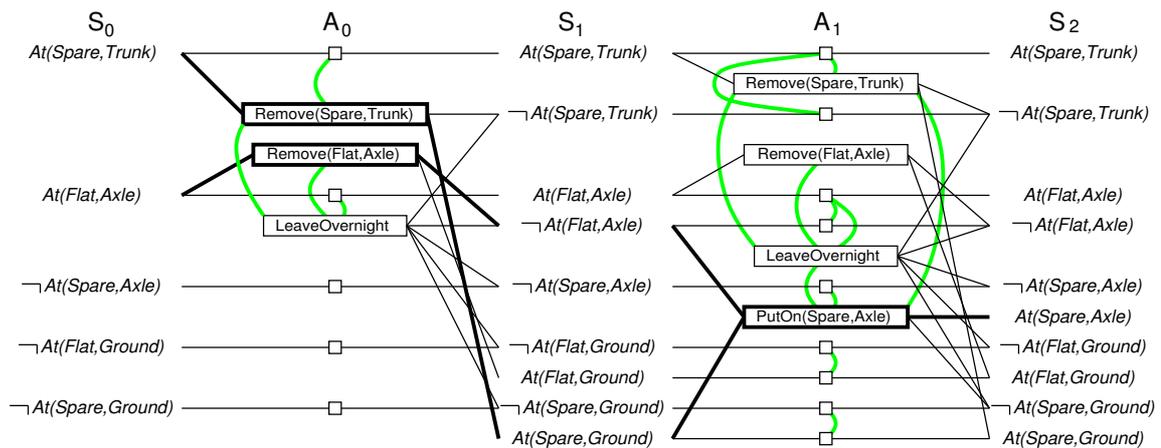
Die Menge aller Fakten, die sich durch die obigen Aktionen ergeben können, bilden das nächste Level  $S_2$ :

10.  $at(spare, trunk)$ , 11.  $\neg at(spare, trunk)$ ,  
 12.  $at(plate, axle)$ , 13.  $\neg at(plate, axle)$ ,  
 14.  $at(spare, axle)$ , 15.  $\neg at(spare, axle)$ ,  
 16.  $at(plate, ground)$ , 17.  $\neg at(plate, ground)$   
 18.  $at(spare, ground)$ , 19.  $\neg at(spare, ground)$

Als nächstes wird wieder propagiert, welche dieser Fakten mutex sind:

Fakten	10	11	12	13	14	15	16	17	18
10	/	/	/	/	/	/	/	/	/
11	N	/	/	/	/	/	/	/	/
12			/	/	/	/	/	/	/
13			N	/	/	/	/	/	/
14			X		/	/	/	/	/
15					N	/	/	/	/
16							/	/	/
17							N	/	/
18					X				/
19									N

Abschließend wird wieder getestet, ob unsere Zielfakten komplett und nicht-mutex in  $S_2$  enthalten sind. Beides ist nun der Fall. Damit ist der Aufbau des Graphen (zunächst) abgeschlossen und wir erhalten das Bild aus Russel/Norvig:



b) Wie viele Level müssen Sie erzeugen?

**Lösungsvorschlag:**

Zunächst 2, da unsere Zielfakten komplett und nicht-mutex in  $S_2$  enthalten sind. Da wir außerdem in der nächsten Teilaufgabe tatsächlich einen Plan finden, bleibt es bei den 2 Leveln.

c) Erzeugen Sie aus Graphen einen Plan.

**Lösungsvorschlag:**

Der Graph wird nun zur Rückwärtssuche des Plans benutzt.

$at(spare, axle)$  kann nur mittels Aktion 12 erreicht werden,  $at(Flat, ground)$  mittels Aktionen 10 und 18. Damit sind zunächst überhaupt nur zwei partielle Pläne möglich:

- $puton(spare, axle)$  parallel zu  $remove(Flat, axle)$ 
  - Die Aktionen 12 und 10 sind jedoch mutex, deshalb kommt dieser Plan nicht in Frage.
- $puton(spare, axle)$  parallel zu  $nop_{11}$ 
  - Die Aktionen 12 und 18 sind nicht mutex, deshalb kann die Suche hier weitergehen.
  - Als neue Teilzeile ergeben sich jetzt die Vorbedingungen der beiden Aktionen, d.h. 8.  $at(spare, ground)$ , 4.  $\neg at(Flat, axle)$ , 6.  $at(Flat, ground)$
  - Alle diese Literale (4, 6, 8) sind paarweise nicht mutex, d.h. die Suche kann fortgesetzt werden.
  - $at(spare, ground)$  kann nur mittels Aktion 1,  $at(Flat, ground)$  nur mittels Aktion 2 und  $\neg at(Flat, axle)$  mittels der Aktionen 2 und 3 erzeugt werden. Damit kommen wieder überhaupt nur 2 Teilpläne in Frage:
    - \*  $remove(spare, trunk)$  parallel zu  $remove(Flat, axle)$  parallel zu  $leaveovernight$ 
      - Da die Aktionen 2 und 3 mutex sind, kommt dieser Plan nicht in Frage.
    - \*  $remove(spare, trunk)$  parallel zu  $remove(Flat, axle)$

- Die Aktionen 1 und 2 sind nicht mutex sind, damit ist dieser Teilplan möglich.
- Wir sind nun in Level  $S_0$  angekommen und haben somit einen Plan gefunden.

Der gefundene Plan lautet: `remove(spare, trunk)` parallel zu `remove(flat, axle)` gefolgt von `puton(spare, axle)`.