



Intelligente Systeme

- Problemlösen –

Suchen, Spielen, Planen

Hochschule für Angewandte Wissenschaften Hamburg
Department Informatik

Dr.-Ing. Sabine Schumann



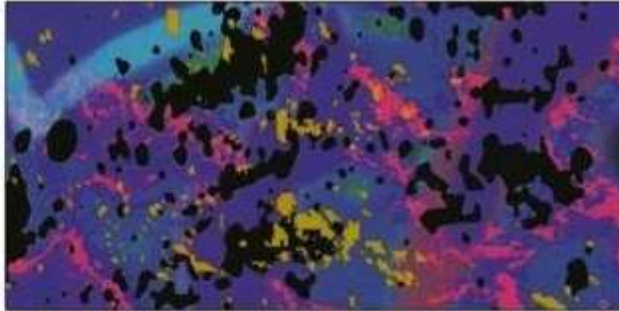
1. Einleitung
2. Logik: Aussagenlogik, Prädikatenlogik erster Stufe
3. Prolog
- (4. DCG)
5. Problemlösen: Suchen, Spielen, Planen
6. Constraints
7. Soft Computing
8. Neuronale Netze
9. Semantische Netze & Frames



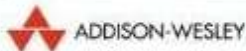
Problemlösen und Suche

- Modellierung, Wissensrepräsentation
- Suche
 - uninformierte,
 - heuristische,
 - Optimierungsprobleme.
- Spiele
 - MiniMax, α/β Schnitt
- Regelbasierte Systeme
- **Planungsprobleme**

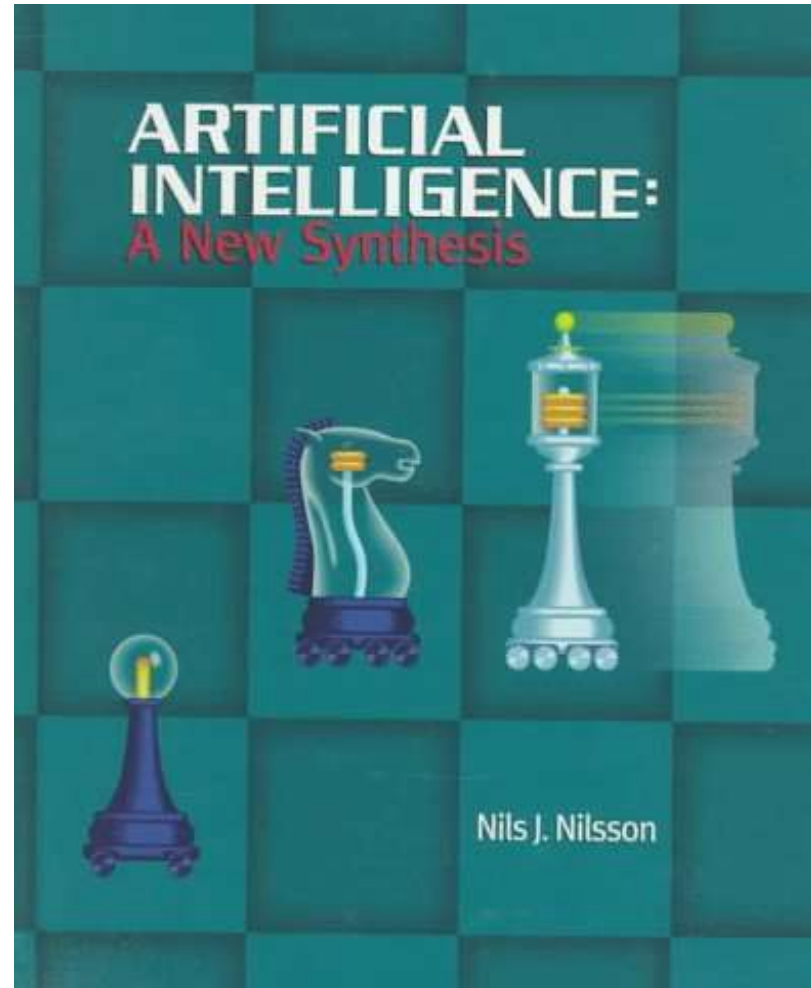
Günther Görz (Hrsg.)



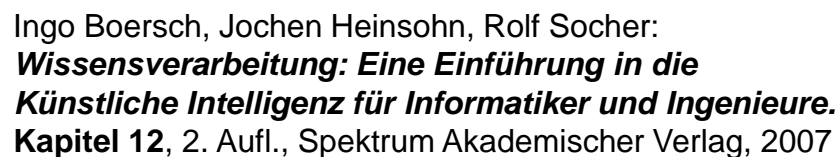
Einführung in die künstliche Intelligenz



Günther Görz (Hrsg.):
Einführung in die Künstliche Intelligenz.
Kapitel 7, Addison-Wesley, 2. Aufl. 1995



Nils J. Nilsson:
Artificial Intelligence: a new synthesis.
Morgan Kaufmann Publisher, 1998.



*Wenn ein Faden ein Ende hat,
so hat er auch noch ein anderes.*

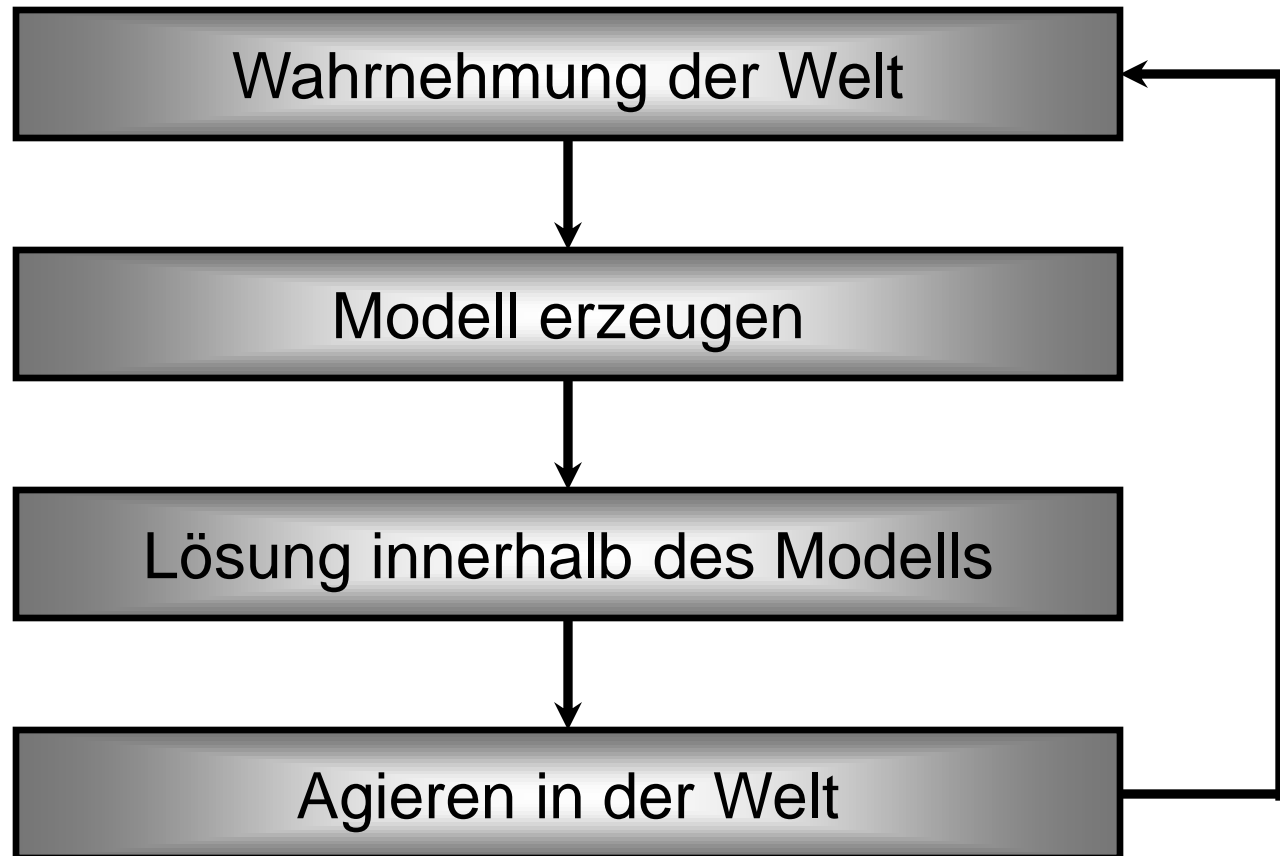
*Miksch's Gesetz,
Arthur Bloch: Murphy's Gesetze in einem Band, 1985*



- Problembereich: Zustandsraum und Operatoren
- Planungsproblem: Ausgangs- und Zielzustand
- Lösung: Folge von Operatoren zur Überführung des Ausgangszustandes in den Zielzustand



sieht im Grundansatz aus wie ein klassisches Suchproblem





planende Agenten

\approx

Problemlösung innerhalb des Modells

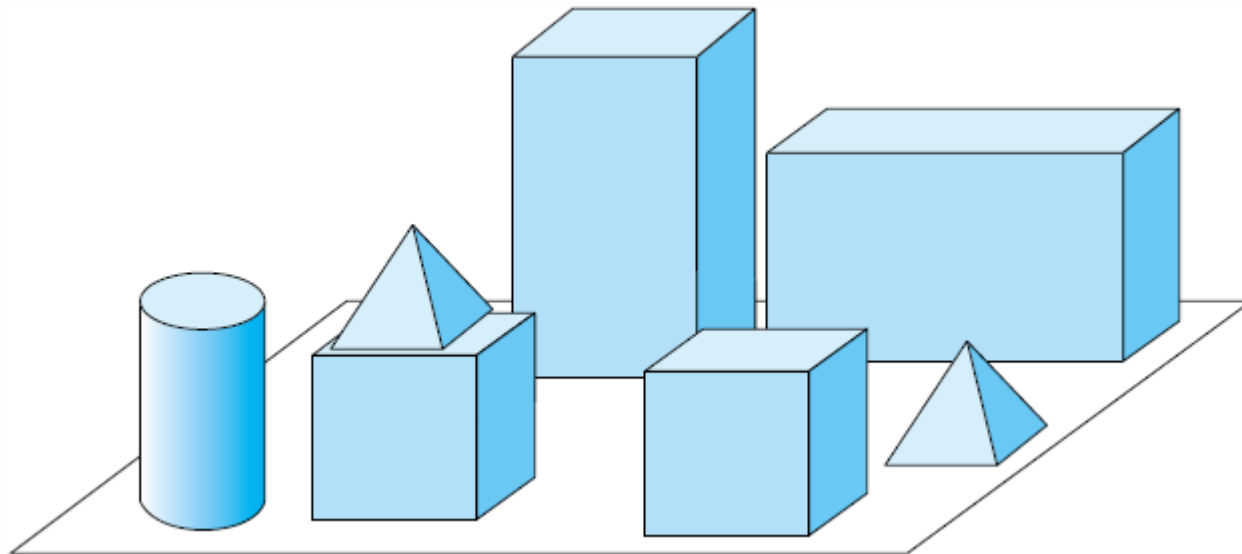
\approx

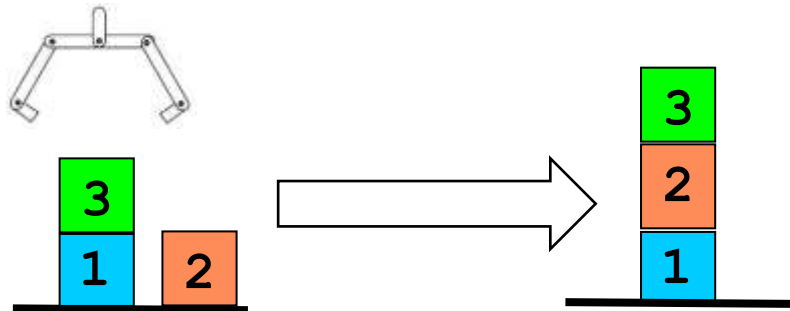
Suche innerhalb eines Zustandsgraphen



Terry Allen Winograd: SHRDLU (1972)

(aus Luger: „Künstliche Intelligenz – Strategien zur Lösung komplexer Probleme“)





Gegeben:

Anordnung von Blöcken, Zielanordnung und mögliche Roboteraktionen.

Gesucht:

Aktionenfolge zur Herstellung der Zielanordnung.

Operatoren:

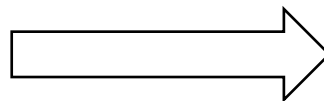
Bewege Block: move/2

Zustandsbeschreibungen:

liegt auf: on/2

ist frei: clear/1

on(b11, table)
on(b12, table)
on(b13, b11)
clear(b13)
clear(b12)



on(b11, table)
on(b12, b11)
on(b13, b12)
clear(b13)



Stundenplanerstellung

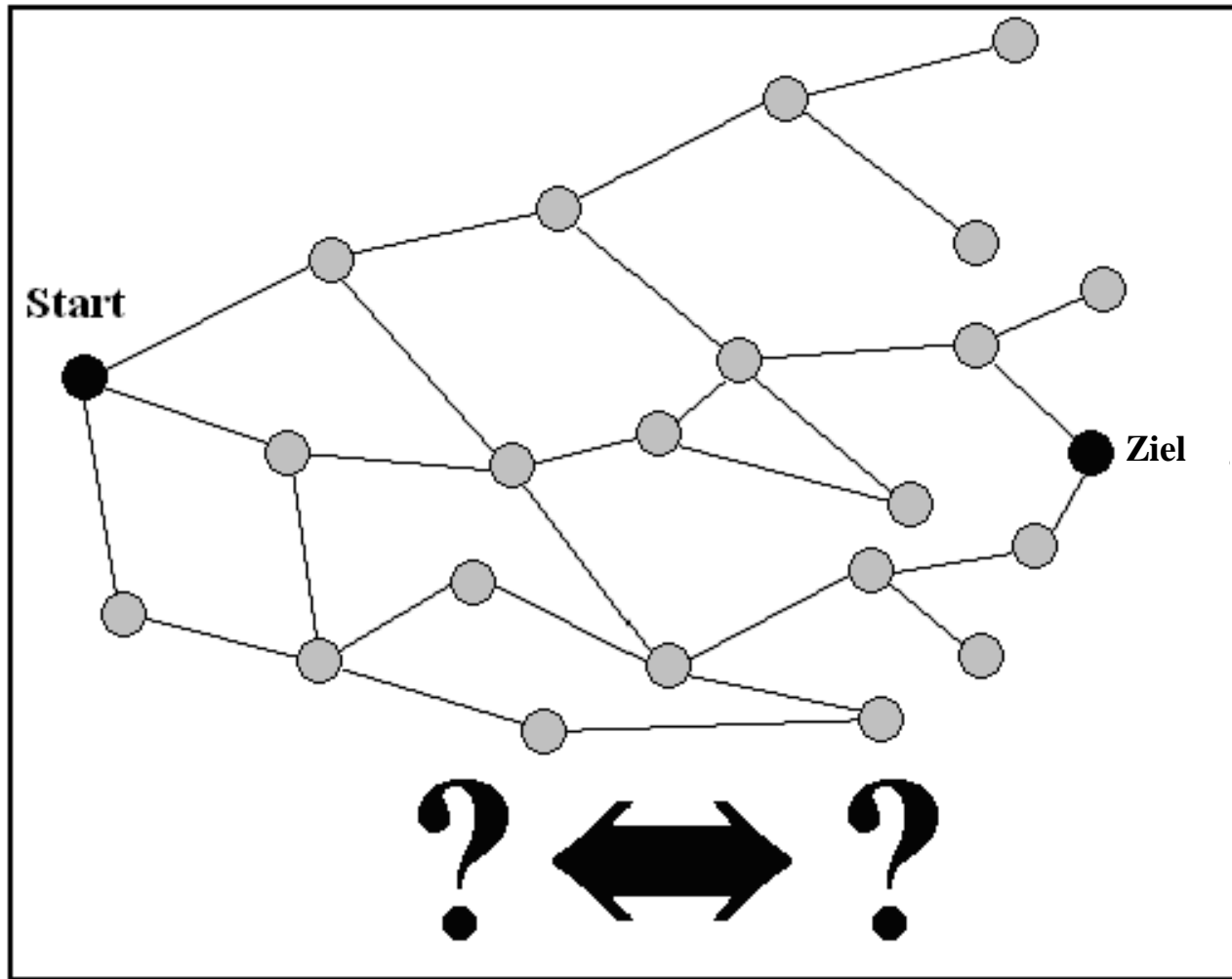
- Gegeben: Räume, Lehrer, Schulklassen, Fächer, Zuordnungen
- Gesucht: Stundenplan

Computer-Konfigurierung

- Gegeben: Hauptkomponenten
- Gesucht: Layout und Hilfskomponenten

Planung von Experimenten in der Molekulargenetik

- Gegeben: Substanzen und Organismen, durchführbare Aktionen, Ziel des Experiments (z.B. Insulin herstellen)
- Gesucht: Sequenz von Aktionen und Spezifikation der benötigten Substanzen und Organismen



Repräsentation
durch Zustände
und Operatoren

- vom Start **vorwärts** zum Ziel
- vom Ziel **rückwärts** zum Start



Einfachstes Prinzip:

Wiederhole

- Bestimme **Differenz** vom aktuellen Zustand zum gesuchten,
- Beschaffe einen **Operator**, der diese Differenz am „besten verringert“,
- Wende diesen Operator an.

Das Wissen steckt dann

- im Bestimmen der Differenz und
- in der Auswahl des Operators.



Zusätzliche Terminologie:

Entscheidungsmonoton:

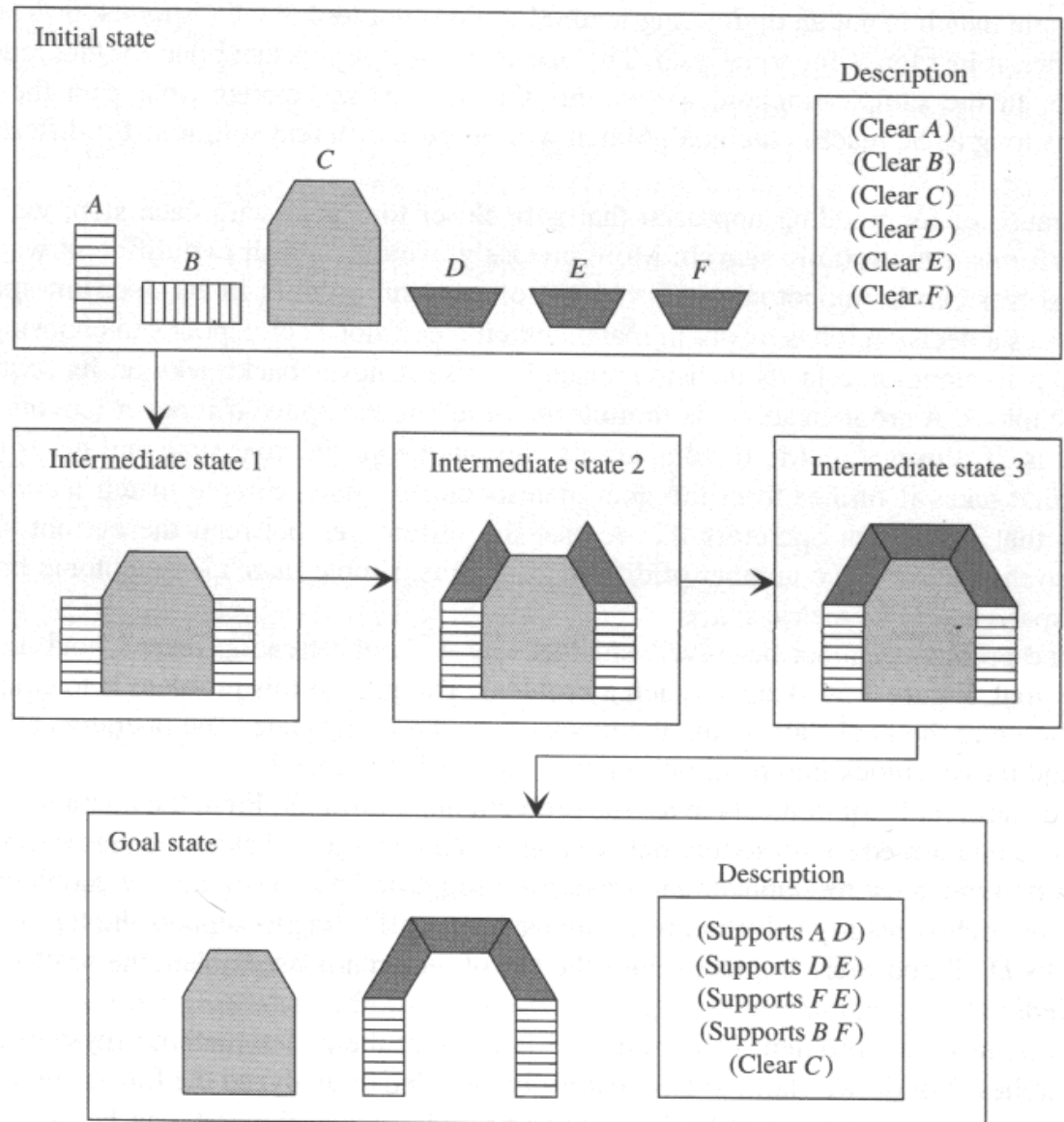
Der Algorithmus nimmt niemals eine Entscheidung über einen gewählten Operator zurück.

Metrisch monoton:

Der Algorithmus wählt nie einen Operator, der den Abstand zum Ziel vergrößern würde.

Problem mit Operatoren, die nie einen Abstand zum Ziel vergrößern.

Das Teilziel (Clear C) muss temporär verletzt werden.





Means-End Verfahren

ist das „klassische“ Rückwärtsverfahren.

Definition eines Operators:

- Anwendbarkeitsbedingung,
- Beschreibung der beseitigten Differenz,
- Beschreibung der Bewertung.

Einfaches Beispiel: von X nach Y fahren

Anwendbar:	Ich bin in X. Mein Auto ist in X.
Differenz:	Ich bin in Y. Mein Auto ist in Y.
Wirkung:	Ich bin nicht mehr in X. Mein Auto ist nicht mehr in X. Ich bin in Y. Mein Auto ist in Y.



Spezifikation von Zielen:

Sammlung von Fakten, die gelten sollen.

Nicht notwendigerweise vollständig!

Speziell (z.Bsp.): STRIPS-Operatoren

add-Liste:

Sammlung von Fakten, die nach Ausführung der Operation zusätzlich gelten.

delete-Liste:

Sammlung von Fakten, die nach Ausführung der Operation nicht mehr gelten.

Differenzen werden durch die Einträge in der **add-Liste** bzw. **delete-Liste** beschrieben, je nach dem, ob man das Fakt haben will oder nicht.



Grundidee von Means-End:

- Nimm ein beliebiges Fakt aus dem Ziel (falls bereits erfüllt umso besser).
- Suche einen Operator, der dieses Fakt herstellt.
- Stelle (rekursiv) seine Vorbedingung her.
- Führe diesen Operator aus.

Erster Schritt:

für jedes Element des Ziels.

Zweiter Schritt:

für wenigstens einen Operator.

STanford **R**esearch **I**nstitute **P**roblem **S**olver

R. E. Fikes , Nils J. Nilsson 1971

Zustände durch Menge von Formeln repräsentiert.

Operatoren haben

- Vorbedingung
- add-Liste
- delete-Liste

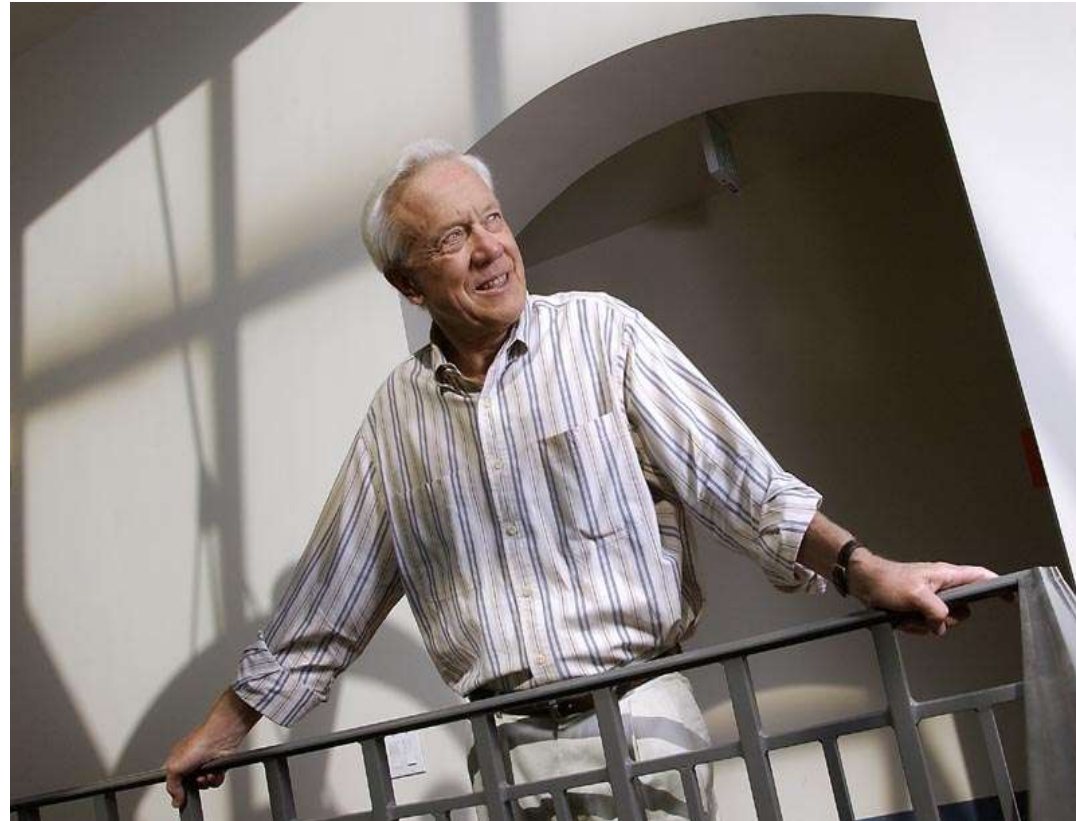
Alle nicht in delete- und add-Liste genannten Formeln des aktuellen Zustandes bleiben erhalten.
(„STRIPS Assumption“)

Co-Entwicklung

- A*-Algorithmus
- Planungssystem STRIPS

Projektsteuerung

- Mobiler Roboter SHAKEY
- Expertensystem PROSPEKTOR



Nur ein Operator zur Zeit, keine parallele Ausführung von Operatoren.

Alle Konsequenzen eines Operators sind durch delete-Liste und add-Liste erfasst.

Ausführung von Operatoren beansprucht keine Zeit, Änderungen passieren sofort.



Problem:

Glas ist leer, Wasserhahn öffnen

Glas ist voll, Wasserhahn schließen

Das allmähliche Volllaufen des Glases wird nicht modelliert.



Operator move/2:

move(x,y)

Vorbedingung:

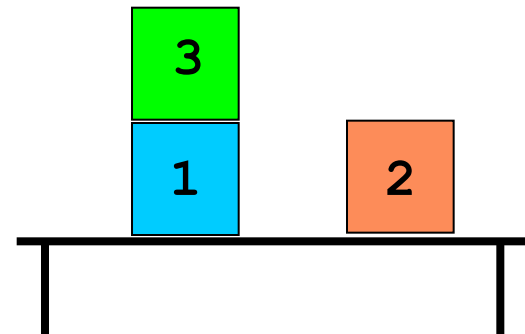
$\text{clear}(x) \wedge \text{clear}(y) \wedge \text{on}(x,z)$

add-Liste:

$\text{on}(x,y), \text{clear}(z)$

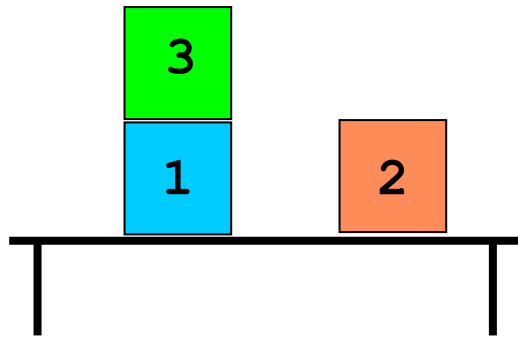
delete-Liste:

$\text{clear}(y), \text{on}(x,z)$

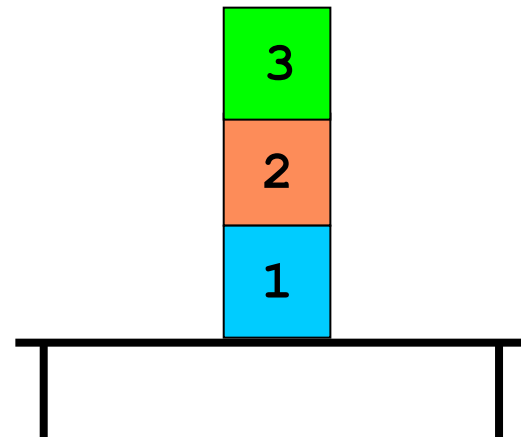




Start Situation :



Ziel: { on(bl2,bl1), on(bl3,bl2) }





Ziel zergliedert in Teilziele:

Teilziel1: ✓

`on (b12, b11)`

→ `move (b12, b11)`

→ `clear (b11)`

→ `clear (b12)`

Teilziel2: ✓

`on (b13, b12)`

→ `move (b13, b12)`

→ `clear (b12)`

→ `clear (b13)`

Teilziel3: ✓

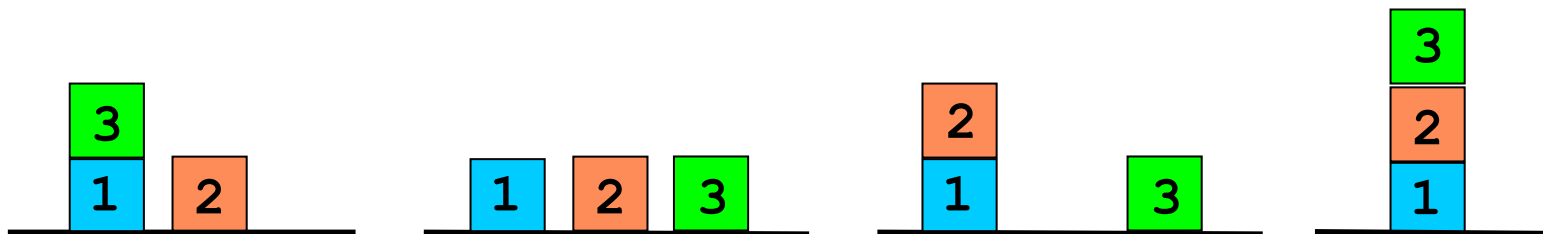
`on (b11, table)`

Teilziel4: ✓

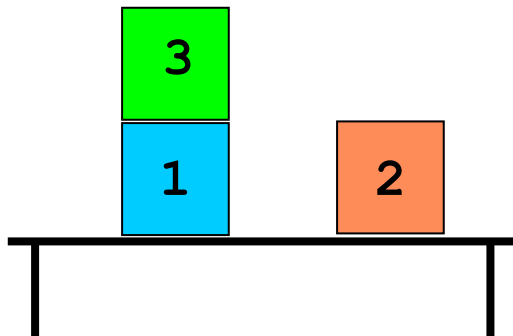
`clear (b13)`

`clear (b11)`

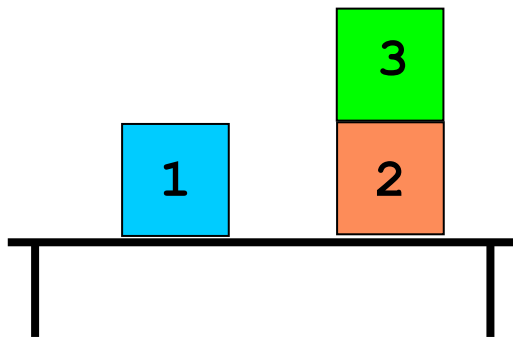
→ `move (b13, table)`



Wenn nun aber zuerst Teilziel2 ausgewählt und vollständig (vor Teilziel1) ausgeführt wird?



```
on (b13, b12)  
→ move (b13, b12)  
→ clear (b12)  
→ clear (b13)
```



```
on (b12, b11)  
→ move (b12, b11)  
→ clear (b11)  
→ clear (b12)
```

?

Sussman Anomalie:

- kein Teilziel kann vollständig ausgeführt werden ohne die Ausführung der anderen Teilziele unmöglich zu machen,
- erstmals beschrieben in der Doktorarbeit von Gerald Sussman,
- zeigt die Schwäche von linearen Planern.

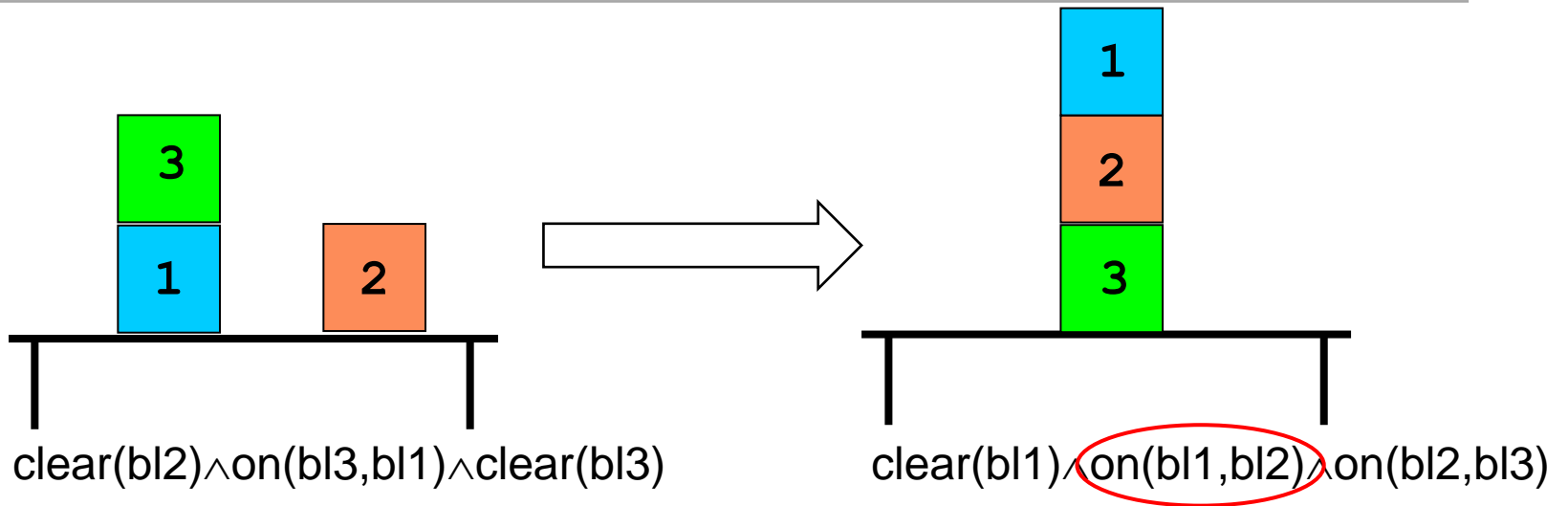


Gerald Jay Sussman
(*1947)

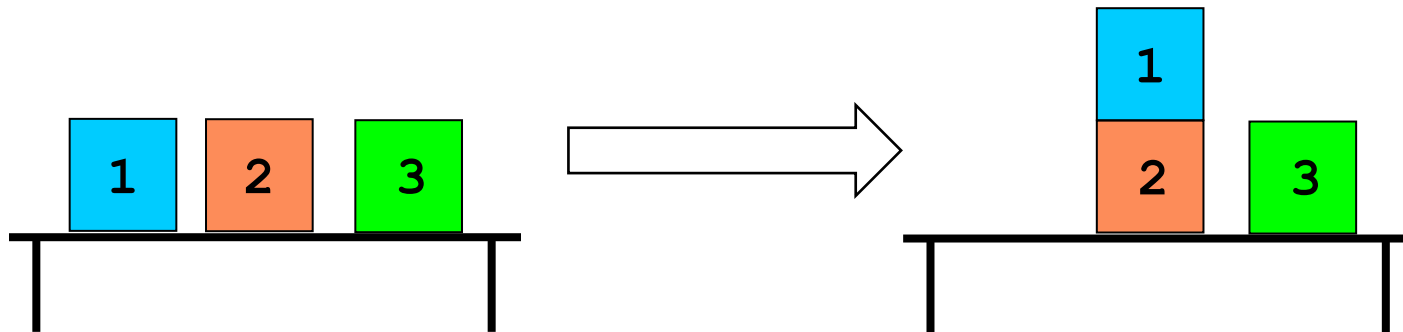
Im vorherigen Beispiel musste mit Teilziel1 begonnen werden, damit das Gesamtziel erreicht werden konnte. Es gab also eine Möglichkeit von einem Teilziel ausgehend, die Gesamtlösung zu erreichen.

Nachfolgend Beispiel für die Sussman Anomalie.

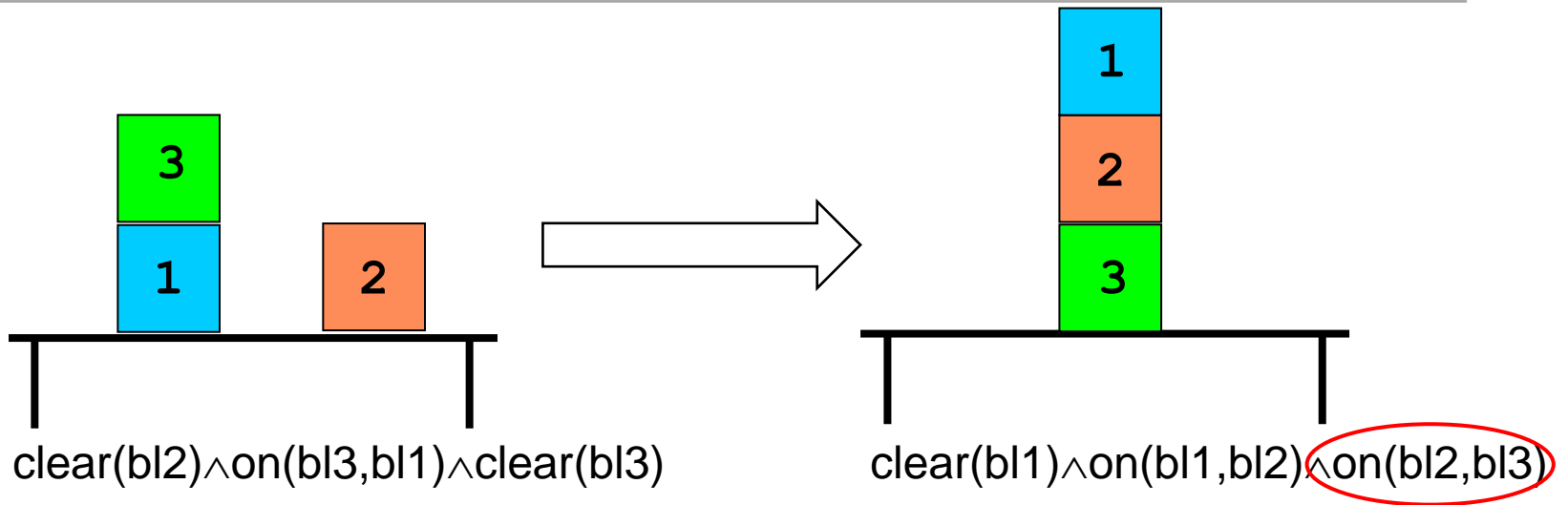
Sussman Anomalie (2)



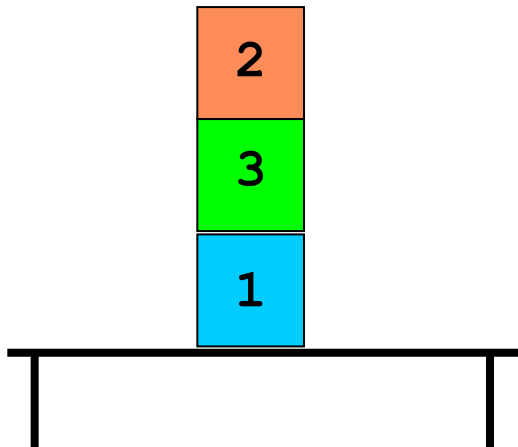
Umsetzung Teilziel1:



Sussman Anomalie (3)



Umsetzung Teilziel2:





Least Commitment Strategie

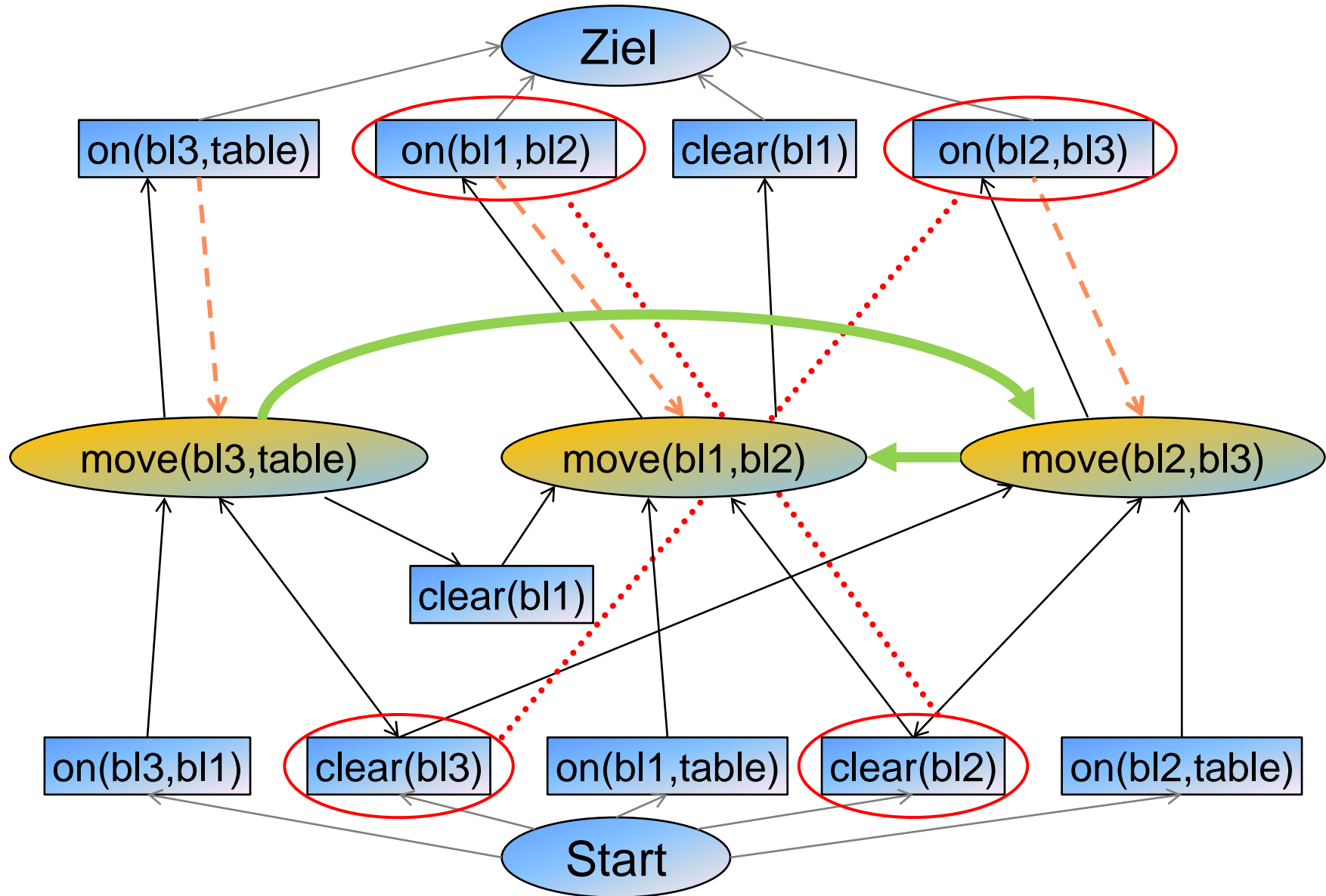
- Einschränkungen bezüglich zukünftiger Entscheidungen vermeiden
 - Vermeiden von Annahmen
- Reihenfolge von Aktionen
- Instantiierungen von Variablen

Nicht-lineares Planen

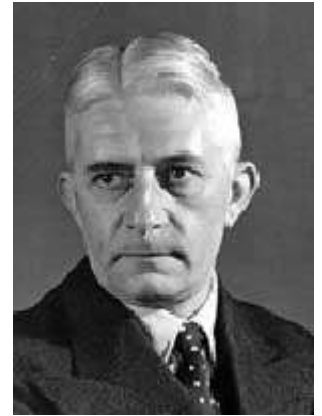
- Partielle Ordnung von Operatoren
- Festlegung von Reihenfolge entsprechend Least-Commitment Strategie

Hierarchisches Planen

- Zuerst Grobplan, dann Verfeinerung
- Beispiel Reiseplanung: Zuerst die Flugverbindungen



Wolfgang Köhler gilt mit Max Wertheimer und Kurt Koffka als einer der Begründer der Gestaltpsychologie bzw. der Gestalttheorie.



Wolfgang Köhler
(1887-1967)



„Eine Banane wird oben ins Drahtgitterdach des Spielplatzes gehängt, viel zu hoch, um selbst im Sprung von einem Schimpansen erreicht zu werden. Einige Meter von dieser Stelle entfernt befindet sich eine Kiste von beträchtlicher Größe. Hier zögerte Sultan niemals; er schleppte die Kiste so weit, bis sie gerade unter der Banane stand, kletterte herauf, sprang von hier aus in die Höhe und erreichte die Banane ohne die geringste Mühe.“

Wolfgang Köhler:
Die Aufgabe der Gestaltpsychologie, Berlin 1971, S. 117

Fotos: http://wkprc.eva.mpg.de/deutsch/files/wolfgang_koehler.htm

Abgewandelte Aufgabenstellung:

In einem Raum befindet sich ein Affe (am Fenster), eine Kiste (an der Tür) und eine Banane hängt von der Decke, jedoch vom Affen selbst im Sprung nicht zu erreichen.

Beschreiben Sie die Zustände und Operatoren für dieses Planungsproblem.

Lösen Sie das Problem mittels nichtlinearer Planung.





Zustände:

posH(Wer, WoH)

Wer {Affe, Kiste, Banane}, WoH {Fenster, Tür, Mitte}

posV(Wer, WoV)

Wer {Affe, Banane}, WoV {Boden, Decke, Kiste}

inHand(Was)

Was {leer, Banane}

Operatoren:

affe_geht(Von, Nach)

Vorbedingung: $\text{posV}(\text{Affe}, \text{Boden}), \text{posH}(\text{Affe}, \text{Von}), \text{Von} \neq \text{Nach}$

add-Liste: $\text{posH}(\text{Affe}, \text{Nach})$

delete-Liste: $\text{posH}(\text{Affe}, \text{Von})$



Operatoren (Fortsetzung):

affe_schiebt_kiste(Von, Nach)

Vorbedingung: $\text{posV}(\text{Affe}, \text{Boden}), \text{posH}(\text{Affe}, \text{Von}),$
 $\text{posH}(\text{Kiste}, \text{Von}), \text{Von} \neq \text{Nach}$

add-Liste: $\text{posH}(\text{Affe}, \text{Nach}), \text{posH}(\text{Kiste}, \text{Nach})$

delete-Liste: $\text{posH}(\text{Affe}, \text{Von}), \text{posH}(\text{Kiste}, \text{Von})$

affe_steigt_auf_kiste

Vorbedingung: $\text{posV}(\text{Affe}, \text{Boden}), \text{posH}(\text{Affe}, \text{Ort}), \text{posH}(\text{Kiste}, \text{Ort})$

add-Liste: $\text{posV}(\text{Affe}, \text{Kiste})$

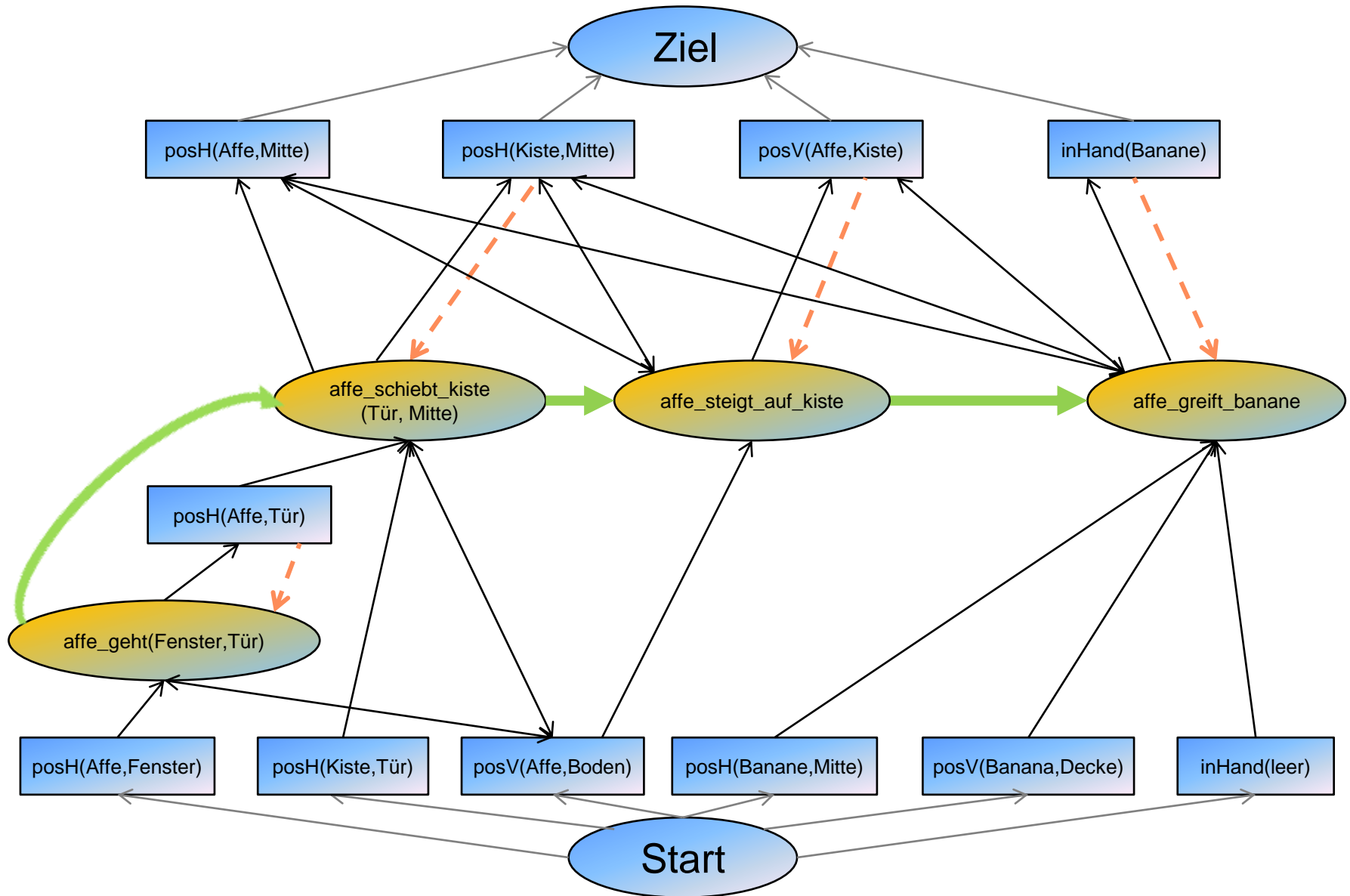
delete-Liste: $\text{posV}(\text{Affe}, \text{Boden})$

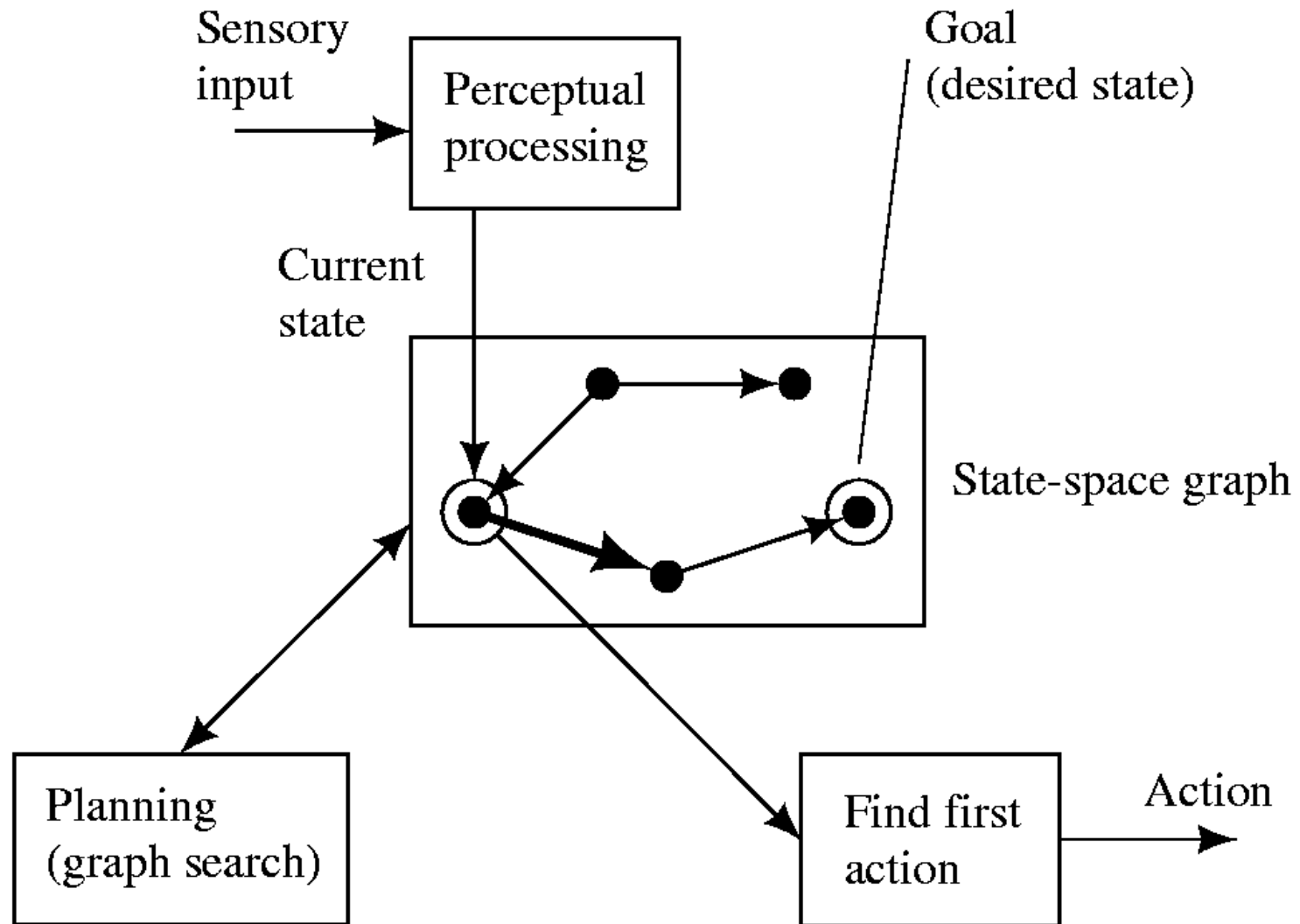
affe_greift_banane

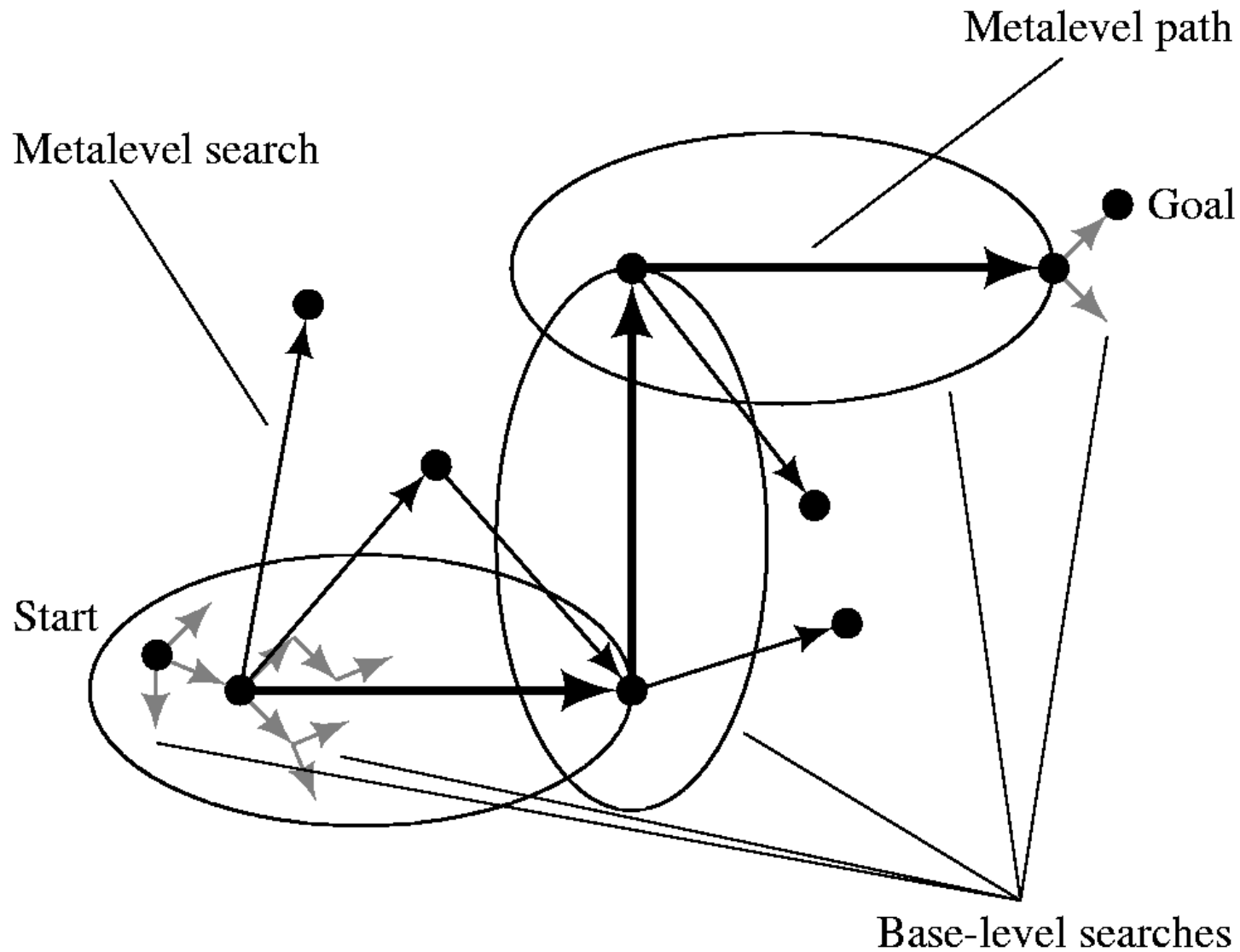
Vorbedingung: $\text{posH}(\text{Kiste}, \text{Mitte}), \text{posV}(\text{Affe}, \text{Kiste}), \text{posH}(\text{Affe}, \text{Mitte}),$
 $\text{inHand}(\text{leer}), \text{posH}(\text{Banane}, \text{Mitte}), \text{posV}(\text{Banane}, \text{Decke})$

add-Liste: $\text{inHand}(\text{Banane})$

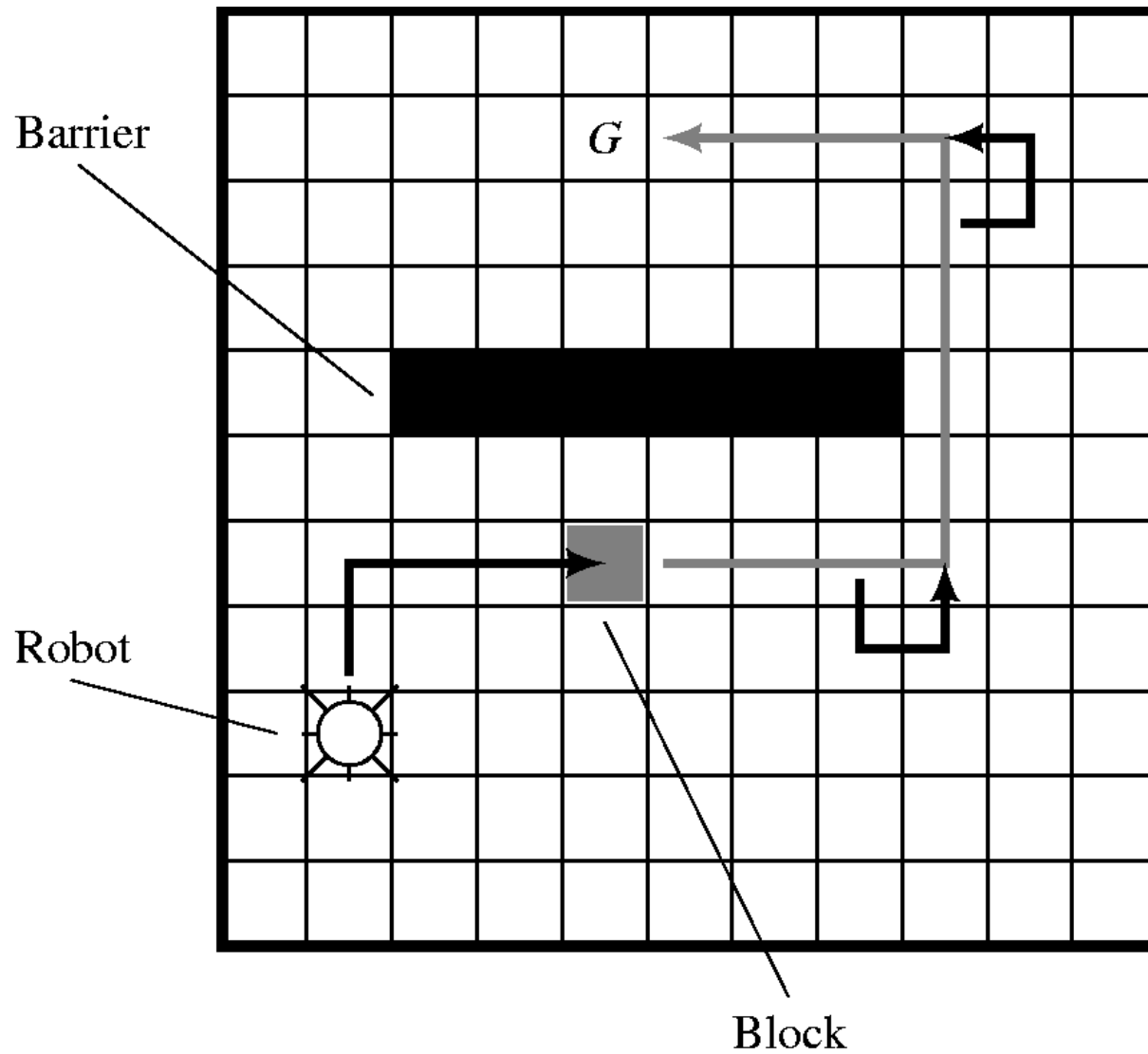
delete-Liste: $\text{inHand}(\text{leer}), \text{posV}(\text{Banane}, \text{Decke})$







Block move - Sokoban



© 1998 Morgan Kaufman Publishers



- ... Grundverständnis für Planungsprobleme.
- ... wissen, wie man ein Planungsproblem auf ein Suchproblem abbilden kann (Zustände und Operatoren).
- ... Means-End erklären können.
- ... wissen, was hierarchisches Suchen bedeutet.