

Einführung in die KI

Prof. Dr. sc. Hans-Dieter Burkhard
Vorlesung Winter-Semester 2004/05

Planen

Repräsentation zeitlicher Veränderungen
Problemlösen/Planen

Begriffsbestimmung

Ablaufplan, Absatzplan, Alternativplan, Anlageplan, Balkenplan, Bauplan, Dienstplan, Einsatzplan, Fahrplan, Finanzplan, Flugplan, Flussplan, Fünfjahrplan, Gesamtplan, Gewinnplan, Hallenplan, Haushaltsplan, Investitionsplan, Kaplan, Kontenplan, Krisenplan, Lageplan, Lebensplan, Lehrplan, Lieferplan, Logikplan, Marktplan, Netzplan, Prüfplan, Racheplan, Rahmenplan, Reiseplan, Rohrplan, Schaltplan, Schlachtplan, Sendepan, Sozialplan, Sparplan, Speiseplan, Spielplan, Staatsplan, Stadtplan, Studienplan, Stufenplan, Tagesplan, Taktplan, Teilplan, Terminplan, Tischplan, Umweltplan, Unterrichtsplan, Volkswirtschaftsplan, Werbeplan, Zeitplan, ...

blueprint, concept, diagram, disposition, idea, intention, layout, plan, plot, program, project, proposition, schedule, scheme,...

Plan

Aktionsplan vs. Lageplan

Plan (lat->frz.) - im allgemeinsten Sinne jede der Realisierung von Vorhaben (zumindest potentiell) dienende Darstellung zusammenhängender Positionen, wobei es diese Darstellung ermöglicht, bestimmte der dargestellten Positionen auf der Grundlage der ebenfalls dargestellten Positionen zu erreichen. . . . „Am Ende des Prozesses kommt ein Resultat heraus, das beim Beginn desselben schon in der Vorstellung des Arbeiters, also schon ideell vorhanden war...“ (Marx) . . .

Philos. Wörterbuch (Leipzig 1969)

(Bedingter) Plan

Strategie

„Wir wollen uns ... vorstellen, daß jeder Spieler ... die Entscheidung über jeden Zug nicht erst dann trifft, wenn die Notwendigkeit dafür vorliegt, sondern daß er sich über sein Vorgehen bei allen möglichen Situationen vorher schlüssig wird. ... einem Plan, der angibt, welche Wahl er zu treffen hat in allen nur möglichen Situationen, für jede nur mögliche wirkliche Information, die er in diesem Augenblick ... besitzen kann. Einen derartigen Plan nennen wir eine Strategie.“

John von Neumann (Zitat nach Philos. Wörterb. Leipzig 1969)

Plan

- Handlungsanweisung, Programm
- (Vorherbestimmte) Aktionen in zeitlicher Abfolge evtl. mit Alternativen/Parametern gemäß Bedingungen

Varianten:

- Detaillierung: Grob vs. Fein, „Skelettplan“
- Zeithorizont des Plans: kurzfristig vs. langfristig
- Determiniertheit, Verzweigung (ND, bedingte Handlungen, ...)
- Ressourcen
- Ziel (einmaliges Ziel vs. Dauerhafter Zustand)
- Ausführung: einzeln/koordiniert/kooperativ

Planung

„Vorbereitung zukünftigen Handelns auf der Grundlage von Informationsgewinnung und -verarbeitung über Entwicklung und gegenwärtigen Zustand des Planobjekts. ...“

Meyers großes Taschenlexikon in 24 Bänden, 1992

Varianten:

- Zeithorizont (kurz-/mittel-/langfristig)
- Inkrementelle Planung bzgl.
 - Fortsetzung
 - Verfeinerung
- Subjekt/Objekt: Eigene vs. fremde Planung
- Alleinige vs. kooperative Planung

Planen vs. Konfigurieren

- Konfiguration/Entwurf/Design:
 - Anordnung, Zusammenstellung von Bestandteilen

- Konfigurieren:
 - Zweckorientierte Auswahl und Komposition von Bestandteilen
 - Syntheseaufgabe unter gegebenen Anforderungen

Planen vs. Konfigurieren

- Konfigurieren
Computerunterstützung:
 - Synthese einer Beschreibung, die allen Anforderungen genügt.
 - Beschreibung, Analyse, Berechnung, ...
 - Methoden:
 - Regelsysteme (z.B. XCON auf Basis OPS-5)
 - Framebasierte Systeme
 - Constraint-Systeme, Suchverfahren

Ablaufplanung/Scheduling

- **zeitliche** Zuordnung von **Aktivitäten** zu limitierten **Ressourcen** mit unterschiedlichen **Nebenbedingungen**
- zur Erreichung eines **Ziels**
- und/oder zur Optimierung bestimmter Kriterien

Unterscheidung von Planung:

- auszuführende Aktionen bekannt
- nur Anordnung gesucht

• Beispiele:

- Produktionsablaufplanung
- Stundenplan
- Einsatz-Planung

Beispiel: Produktionsablaufplanung

- Gegeben: Maschinen, Rohstoffe und Personal als variabel einsetzbare Ressourcen
- Nebenbedingungen, z.B.:
 - Herstellvorschriften
 - Verbot von Doppelbelegungen.
- Betriebswirtschaftliche Zielsetzungen, z.B.:
 - Termine
 - Durchsatz
 - Kostenoptimierung

Gesucht: **Ablaufplan** (terminliche Zuordnung von Aktionen zu Ressourcen), der gegebene Nebenbedingungen und Zielsetzungen erfüllt.

Probleme:

Stabilität, Fortsetzbarkeit („Maintenance-Problem“)

Beispiel: Produktionsablaufplanung

prädiktiv: Vorausschauend für statische Planungsumgebung

reaktiv: Anpassung an neue Situationen, alten Plan möglichst erhalten

- dynamischer Problembereich
- großer Suchraum
- unsicheres Wissen

Klassische Methoden:

- Operations Research: meist NP-hart
Optimierung einzelner Zielfunktionen, Netzplantechnik, Prioritätsregeln
- Theoretische Informatik (Komplexitätsuntersuchungen)

Beispiel: Produktionsablaufplanung

KI-Verfahren:

- Suche (auch in Operations Research)
- Frame-basiert, Regelsysteme
- Constraint-Systeme, Constraint-Logik
- Soft Computing, iterative Verbesserung
- Multi-Agenten-Systeme, Verhandlung

- explizite Darstellung.
- Verarbeitung von anwendungsspezifischem Problemlösungswissen.
- interaktive Steuerung für manuelle Vorgaben/Änderungen.

Sozionische Systeme

Handlungsplanung

Erzeugung eines zielgerichteten Verhaltens für „Agenten“
(Roboter, Menschen, Technische Systeme, Anlagen)

Spezielle Probleme:

- dynamische Umwelt
- unsichere Daten
- unsichere Aktionen

Methodisch:

- Formale Beschreibung von Zielen/Aktionen eines Agenten.
- Algorithmen, die Verhalten erzeugen („planen“).

Handlungsplanung

Klassische
Einteilung:

reaktives Verhalten:

Stimulus-Response (Behaviorismus)
„einfache“ Verhaltensmuster

deliberatives Verhalten:

zielgerichtetes/planvolles Verhalten
„komplexe“ Verhaltensweisen

hybrid:

zielgerichtete Auswahl/Benutzung reaktiver
Verhaltensmuster („behavior“)

Zusätzlich:

Interne Zustände (Weltmodell, Ziele, ...) betrachten.

Handlungsplanung

Least Commitment:

Entscheidungen zeitnah treffen.

Verzögerung von Festlegungen

- bei der Planung und/oder
- Durchführung des Plans

„Plan“ in Form eines „Skripts“:

- Alternativen bis zum letzten Moment offen lassen.
- Parameter erst in der konkreten Situation festlegen.

„Klassische“ Planungsverfahren

- Gegeben: Ausgangszustand, Ziel, Menge von Aktionen
- Gesucht: Plan als Abfolge von Aktionen, die vom Ausgangszustand einen Zielzustand erreicht

Unterschied zu Scheduling :
Aktionen erst noch auswählen.

Allgemein beschreibbar als

- Zustandsraumsuche
- Problemzerlegung

Konkrete Verfahren nutzen Besonderheiten
zur Reduktion des Suchraums

Aktionen/Ereignisse, Zustände

- Ereignisse finden statt.
- Aktionen werden durch Agenten initiiert.

Einfacher Plan eines Agenten:

Folge von Aktionen des Agenten,
verändern den Zustand der „Welt“

Zustand (Situation):

- Schnappschuss der „Welt“ zu einem Zeitpunkt
- Nur partiell beschreibbar

Unterschiedliche
Formalismen

Dualität:

- Ereignisse/Aktionen verändern Zustände
- Zustände begrenzen Ereignisse/Aktionen

Repräsentation zeitlicher Veränderungen

Zeit und Situationen: *Was gilt wann*

Planung:

Vorausschauend Handlungen beurteilen,
geeignete Aktionen auswählen.

„Vorstellungsvermögen“ des Agenten:

Ereignisse, Abläufe intern repräsentieren

- bzgl. Vergangenheit/Gegenwart: „Umweltmodell“
- bzgl. Zukunft: „Ziele“, „Pläne“

Zustände des Agenten

Repräsentation von Zeit

Eng verwandt mit Repräsentation von „Raum“.
Übertragung räumlicher (sinnlich erfahbarer Konzepte)
auf Zeit:

- Abgrenzung
- Ausdehnung
- Anordnung, Beziehungen

Repräsentation durch

- Punkte
- Intervalle
- Beziehungen („in“, „zwischen“, „vor“, „nach“, ...)

Repräsentation in Raum oder Zeit

Zeitliche Referenzen:

Peter ging in die Vorlesung.
Dann setzte er sich.
Er schlief während des Beweises ein.

vs. Räumliche Referenzen:

Peter ging in den Hörsaal.
Dort setzte er sich.
Er schlief auf seiner Bank ein.

In der Prüfung konnte er den Beweis erraten.
Während der Prüfung konnte er den Beweis erraten.
(Referenz auf ein Ereignis)

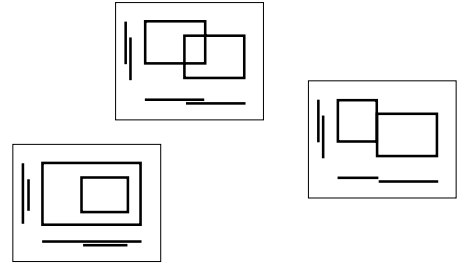
Intervall-Kalkül für Zeit (Allen-Kalkül)

	x y	STARTS(x,y)
	x y	FINISHES(x,y)
	x y	DURING(x,y)
	x y	BEFORE(x,y)
	x y	OVERLAPS(x,y)
	x y	MEETS(x,y)
	x y	EQUALS(x,y)

13 Relationen
einschließlich der
jeweiligen Inversen

Intervall-Kalkül für Raum

Darstellung mithilfe der eindimensionalen Beziehungen



Unterschiede

Raum	Zeit
3-dimensional	1-dimensional
isotrop (Richtungen gleichartig)	anisotrop (ausgezeichnete Zukunftsrichtung)
Mehrere Referenzsysteme (links/rechts, vorn/hinten: standortabhängig)	Einheitl. Referenzsystem
primäre Verankerung: Objekte	primäre Verankerung: Ereignisse
Sekundäre Verankerungen: Objekte in der Zeit: "fluents"	Sekundäre Verankerungen: Ereignisse im Raum
sinnlich wahrnehmbar	keine direkte Wahrnehmung

Repräsentation zeitlicher Veränderungen

Alternative Ausgangspunkte:

- Abfolge von Zuständen
 - z.B. zeitbezogene Gültigkeit von Fakten
- Abfolge von Ereignissen

Unterschiedliche
Formalismen

- Ereignisse/Aktionen verändern Zustände
- Zustände begrenzen Ereignisse/Aktionen

Aktion ausführbar, wenn Vorbedingungen im Zustand gelten
Zustandsänderung durch Ausführung der Aktion

Beschränkung:

interessierende/zugängliche/beschreibbare
Zustände und Bedingungen

Repräsentation zeitlicher Veränderungen

2 Aspekte bei (zustandsorientierter) Systembeschreibung:

- „statische Beschreibung“:
Was beschreibt einen globalen Zustand (Situation)
Evtl. Komposition aus lokalen Zuständen
- „dynamische Beschreibung“:
 - Wodurch werden Zustandsübergänge beschrieben (Transitionen)
 - Wie erfolgen die Zustandsänderungen

Repräsentation zeitlicher Veränderungen

Allgemeine Beschreibungs-Probleme:

- Abgrenzung
- Verlauf, Dauerhaftigkeit

– Überlappende Ereignisse, Fakten

– Fahrt von Berlin nach Adlershof:

12.28 - 12.58 (aber nicht 12.45 oder 12.21 - 12.31)

– Student sein

– Diplomarbeit schreiben

Repräsentation zeitlicher Veränderungen

Unterschiedliche Zeitstrukturen

- Verzweigte vs. lineare Zeit
- Zyklisch vs. azyklisch
- Zeitpunkte vs. Intervalle
- Diskret vs. dicht vs. kontinuierlich
- Zukunft unbeschränkt vs. beschränkt
- Vergangenheit unbeschränkt vs. beschränkt

Repräsentation zeitlicher Veränderungen

Formalisten

– Systeme:

Transitionssysteme,
Automaten/Zustandsmaschinen,
Petri-Netze, ...

– Logiken:

Temporale Logiken
Mehrsortige Logiken
(Zeit, Fakten, Ereignisse, Aktionen, Objekte ...)
Intervallkalkül (Allen-Kalkül), ...

Situationskalkül (McCarthy, Hayes, 1969)

Welt mittels **Situationen** und **Aktionen** beschreiben

Situation

- Zustand der „Welt“ zu einer bestimmten Zeit („Schnappschuss“)
- Menge von Fakten $S = \{ \dots, \text{ist_in}(\text{Peter}, \text{Berlin}), \dots \}$

Ereignisse/Aktionen

- Übergänge zwischen Situationen mit
 - Vorbedingungen:
Voraussetzungen für Aktion/Ereignis
 - Nachbedingungen (Effekte):
Zustandsänderungen durch Aktion/Ereignis

Situationskalkül: Situationen

Situation

- Menge (Konjunktion) von positiven Fakten (Grundatome: ohne Variable)
- Closed World Assumption
Nicht aufgeführte Fakten gelten nicht.

$$S = \{ \text{At}(\text{Home}) \wedge \text{Have}(\text{Milk}) \wedge \text{Have}(\text{Beer}) \}$$
$$\Rightarrow \text{nach CWA: } \neg \text{Have}(\text{Money})$$

Situationskalkül: Fluents

Fluents

- Objekte mit zeitlich veränderlichem Wert
RussischerPräsident(Gorbatschow, S_1)

Beschreibung mit Fluents:

statt $S_1 = \{ \text{At}(\text{Home}) \wedge \text{Have}(\text{Milk}) \wedge \text{Have}(\text{Beer}) \}$

jetzt

$$\begin{aligned} &\text{At}(\text{Home}, S_1) \\ &\text{Have}(\text{Milk}, S_1) \\ &\text{Have}(\text{Beer}, S_1) \end{aligned}$$

Situationskalkül: Ereignisse

Ereignisse/Aktionen

- Übergänge zwischen Situationen mit
 - Vorbedingungen:
Voraussetzungen für Aktion/Ereignis
 - Nachbedingungen (Effekte):
Zustandsänderungen durch Aktion/Ereignis

Bedingungen dürfen Variable enthalten

$$\begin{aligned} a &= \text{Buy}(X, Y) \text{ mit} \\ V &= \{ \text{At_Shop}(X) \wedge \text{Has_Shop}(X, Y) \wedge \text{Have}(\text{Money}) \} \\ N &= \{ \neg \text{Have}(\text{Money}) \wedge \text{Have}(Y) \} \end{aligned}$$

Situationskalkül: Situationsübergänge

Situation

$S1 = \{ At_Shop(TanteEmma) \wedge Has_Shop(TanteEmma, Wein) \wedge Have(Money) \}$

Ausführbare Aktion: $a = Buy(X, Y)$ mit
 $V = \{ At_Shop(X) \wedge Has_Shop(X, Y) \wedge Have(Money) \}$
 $N = \{ \neg Have(Money) \wedge Have(Y) \}$

Führt zur neuen Situation:

$S2 = result(Buy(TanteEmma, Wein), S1)$
 $= \{ At_Shop(TanteEmma) \wedge Has_Shop(TanteEmma, Wein) \wedge Have(Wein) \}$

nach CWA:
 $\neg Have(Money)$

Situationskalkül

- keine explizite Zeit, nur Abfolge
Nicht: am 29. Februar
- keine simultanen Ereignissen (Parallelität)
Nicht: fahren + Radio hören
- keine Trennbarkeit wiederholter Ereignisse
Nicht: drei Runden laufen
- keine kontinuierliche Veränderung
Nicht: den Krug leeren

Temporallogik für Planung (McDermott, 1982)

Zustände

- Zustand der Welt zu einer bestimmten Zeit („Schnappschuss“)
- Menge von Fakten
- dicht
- vorwärtsverzweigend:
 - partiell geordnet bzgl. Aueinanderfolge:
 $s_1 \leq s_2$
 - Verzweigungspunkte: alternative Fortsetzungen (Aktionen/Ereignisse mit unterschiedlichen Konsequenzen)

Temporallogik für Planung (McDermott, 1982)

Zeitpunkte

- total geordnet (reelle Zahlen)

Chronik

- total geordnete Teilmenge der Zustände (mögliche Abfolge)

Datumsfunktion

- Abbildung einer Chronik auf Zeitpunkte

Temporallogik für Planung (McDermott, 1982)

Zeitlich veränderliche Gültigkeit von Fakten:

- Zuordnung aller Zustände, in denen ein Fakt wahr ist (abgeleiteter Begriff)

Fakt p gilt im Zustand s : $T(s, p)$

Fakt p gilt für alle s mit $s_1 \leq s \leq s_2$,

d.h. im Zustands-Intervall $[s_1, s_2]$: $TT(s_1, s_2, p)$

Temporallogik für Planung (McDermott, 1982)

Ereignisse als abgeleiteter Begriff:

Ereignis e formal beschrieben durch Zustands-Intervall $[s_1, s_2]$

(bzw. Zeit-Intervall mittels Datumsfunktion),

in dem e stattfindet

- keine Unterscheidung zeitgleicher Ereignisse möglich

Temporallogik für Planung (McDermott, 1982)

Schließen über
Aktionen/Ereignisse:

Aktion:

$do(agent, act)$

Ereignis:

$do(act)$

Relationen, z.B.:

$Not_Occur_lf(e_1, e_2)$

„ e_2 wird durch e_1 verhindert“
ist wahr, falls es keine Chronik gibt,
in der e_2 nach e_1 stattfindet

Zeit- und Handlungslogik von Allen (1984)

- Zeitintervalle gemäß Allen-Kalkül

- Mehrsortige Prädikatenlogik mit Termen für

- Zeitintervalle
- Eigenschaften
- Ereignisse
- Prozesse
- Handlungen
- Agenten
- Objekte usw.

Zeit- und Handlungslogik von Allen (1984)

HOLDS(p, t)	Fakt p gilt im Intervall t
OCCUR(e, t)	Ereignis e findet (genau) im Intervall t statt
OCCURING(π , t)	Prozess π findet (fortlaufend) während t statt
ECAUSE(e1, t1, e2, t2)	Ereignis e1 in t1 bewirkt e2 in t2
ACAUSE(a, o)	Agent a bewirkt Eintreten von Ereignis/Prozess o durch seine Aktion

OCCURING: "umherlaufen",

OCCUR: "in den Hörsaal gehen"

OCCUR(e, t) \wedge IN(t, t') \rightarrow \neg OCCUR(e, t')

OCCURING(π , t) \rightarrow \exists t': IN(t', t) \wedge OCCURING(π , t')

(mit $IN(t_1, t_2) := DURING(t_1, t_2) \vee STARTS(t_1, t_2) \vee FINISHES(t_1, t_2)$)

Zeit- und Handlungslogik von Allen (1984)

Weitere Axiome z.B.:

HOLDS(p, t) $\leftrightarrow \forall$ t' : IN(t', t) \rightarrow HOLDS(p, t')

HOLDS(and(p, q), t) \leftrightarrow HOLDS(p, t) \wedge HOLDS(q, t)

HOLDS(not(p), t) $\leftrightarrow \forall$ t' : IN(t', t) \rightarrow \neg HOLDS(p, t')

HOLDS(or(p, q), t) $\leftrightarrow \forall$ t' : IN(t', t) \rightarrow

\exists t'' : IN(t'', t) \wedge (HOLDS(p, t'') \vee HOLDS(q, t''))

OCCUR(e1, t1) \wedge ECAUSE(e1, t1, e2, t2) \rightarrow OCCUR(e2, t2)

ECAUSE(e1, t1, e2, t2) \rightarrow BEFORE(t1, t2) \vee MEETS(t1, t2) \vee
EQUALS(t1, t2) \vee OVERLAPS(t1, t2) \vee IN(t1, t2)

OCCURING(ACAUSE(a, π), t) \rightarrow OCCURING(π , t)

OCCUR(ACAUSE(a, e), t) \rightarrow OCCUR(e, t)

Allgemeine Eigenschaften der Formalismen

- Was wird explizit repräsentiert

Ereignisse?

Zustände?

Zeit?

Dazu dann Axiome:

- Festlegungen der Zusammenhänge bei expliziter Darstellung

- Festlegung der Beschreibung bei abgeleiteter Darstellung

„Reasoning about change“

Zentrales Problem

... to reason both rigorously and efficiently about the future...
(Shoham/McDermott, 1988)

- Spezifikation der Wirkung von Aktionen
- Übertragung von "Gültigkeit" auf andere Situationen
(speziell mit logischen Mitteln -- vgl. dagegen OPS5)

Sinnvolle Annahme (aber nicht selbstverständlich)

Wenn Fakt f im Zustand z / zur Zeit t gilt,
dann gilt f auch im nächsten Zustand / zur Zeit t+1,
falls nichts anderes (z.B. durch Aktionen) festgelegt ist.

„Reasoning about change“

frame problem

(Kulissenproblem, Rahmenproblem, Trägheitsproblem)

Das Problem, in der Nachbedingung eines Operators anzugeben, was alles durch ihn nicht verändert wird.

persistence problem

(Beharrungsproblem, Fortdauerproblem, Persistenzproblem)

Das Problem, vorherzusagen, welche Fakten über eine Folge von Operatoren gültig bleiben.

Beide Probleme eng verwandt.

Beispiel: Modellierung „Türme von Hanoi“.

„Reasoning about change“

prediction problem (Vorhersageproblem)

Das Problem, vorherzusagen, welche Fakten nach einer Folge von Operatoren wahr sein werden.

Bsp. (Shoham und McDermott):

Lauf einer Billardkugel vorausberechnen.

Konkurrierende Aktionen:

- Lauf von anderen Billardkugeln
- plötzliches Herabstoßen von Adlern auf den Billardtisch usw.

Alle Aktionen sind determiniert, aber bei Verarbeitung aller Informationen wird die Komplexität zu groß.

„Reasoning about change“

qualification problem

(Qualifikationsproblem, Voraussetzungsproblem)

Das Problem, die Zukunft vorherzusagen, ohne alle denkbaren Voraussetzungen aus der Vergangenheit zu berücksichtigen.

- Vorbedingungen werden nur beschränkt aufgeführt:
IF zuenschluessel_gedreht THEN auto_springt_an
- Weitere potentielle Vorbedingungen dieser Regel:
 - keine Kartoffel im Auspuff
 - keine Möhre (Kivi usw.) im Auspuff
 - Motor vorhanden usw

„Reasoning about change“

ramification problem

(Schneeballproblem, Verzweigungsproblem)

Das Problem, vorherzusagen, welche weiteren Fakten wahr werden, wenn in Situation S ein Fakt f wahr wird.

Bsp.: Fahren mit dem Auto von A nach B.

Wer oder was befindet sich danach in B ?

Auto, FahrerIn, Motor, Taschenuhr, Tramper,...

„Reasoning about change“

Lösungsansätze für

- frame problem
- persistence problem
- prediction problem
- qualification problem
- ramification problem

- Spezielle Axiome (Frame-Axiome)

Alles was nicht aufgeführt ist, bleibt unverändert, z.B.

$\forall s, o, p (p \notin \text{Effekte}(o) \rightarrow (\text{HOLDS}(p, s) \leftrightarrow \text{HOLDS}(p, \text{result}(o, s))))$

- Nichtmonotone Logiken
- Nicht-logische Formalismen

Klassisches Problemlösen/Planen

Planungsproblem beschreiben als **Suchraum** mit

- Ausgangszustand,
- Zielzuständen (Planziel)
- Aktionen/Operatoren für Zustandsübergänge

Plan:

Geeignete Operatorenfolge
(oder: Situationenfolge)
vom Anfangszustand
zu einem Zielzustand

Klassisches Problemlösen/Planen

Einschränken des Suchraums durch Ausnutzen spezieller Eigenschaften von Planungsproblemen
(z.B. Zustände als Fakten-Mengen)

Heuristiken, um nutzlose Aktionen auszuschließen.
Alternativen gleichzeitig betrachten.

- Vorwärtsverkettung ("Progression")
- Rückwärtsverkettung ("Regression")
- Günstiger, wenn wenige Zielbedingungen zu erfüllen sind.
- Problem: verschiedene Zielbedingungen zugleich erfüllen.

Klassisches Problemlösen/Planen

Konsequenz aus Frame-Problem usw.:

Einschränkungen bzgl. Vollständigkeit/Korrektheit

- „rein logisch“
- bzgl. des betrachteten Weltausschnitts

Unsicherheiten durch
unvorhergesehene Ereignisse

- Planung unter Vorbehalt
- Möglichkeit zur Anpassung, Neuplanung
- Grobplanung/partielle Planung
- *least commitment*

⇒ Handlungsplanung

Klassisches Problemlösen/Planen

Beschreibung eines Planungsproblems

Zustände als Mengen gültiger Fakten
Operatoren verändern Fakten (z.B. Situationskalkül)
Ausgangszustand, Zielzustand: Fakten-Mengen

Plan: Folge von Operatoren (Aktionen, Tasks)
Erfolgreicher Plan: Überführt Anfangszustand in Zielzustand

Planung: Methodische Erzeugung von Plänen.
– Vollständig: Findet einen Plan, falls existent.
– Korrekt: Gefundener Plan ist erfolgreich.

Klassisches Problemlösen/Planen

Deduktive Planung

- Suchraum: logische Ausdrücke
- Theorembeweiser
- (Konstruktiver) Beweis der Lösbarkeit des Planungsproblems ergibt einen Plan

Mengenbasierte Planung:

- Suchraum:
 - Zustände = *Mengen* von Fakten
 - Operatoren = Aktionen
- Suchverfahren ergibt Plan=Lösungsweg

Klassisches Problemlösen/Planen

Planbasierte Planung:

- Suchraum: Partielle Pläne als Zustände
- partieller Plan: nur partiell geordnet
 - Lösung von Teilzielen
 - Grob-Lösungen
 - Operatoren zur Spezifikation von Plänen
 - Vervollständigung
 - Verfeinerung
 - genauere Anordnung

STRIPS (Fikes et al, 1972)

= STanford Research Institute Problem Solver

Situation = Menge (Konjunktion) von gültigen Fakten

- Positive Grundlitterale
- CWA: nicht enthaltene Grundlitterale gelten als falsch

At(Home), Have(Milk), Have(Money)
gemäß CWA: $\neg(\text{Have(Beer)})$

Zielbedingungen:

Menge Z von erwünschten Fakten
(positive Litterale, Variable ggf. existentiell quantifiziert)

Plan erfolgreich, falls er Situation S mit $Z \subseteq S$ erreicht

Operatoren in STRIPS

Operator durch 3 Listen (V,D,A) beschrieben:

V: "precondition list":

Vorbedingungen für Ausführung

D: "delete list":

Bedingungen, die nach Ausführung nicht mehr gelten

A: "add list":

Bedingungen, die gültig werden durch Ausführung

- Bedingungen sind (positive) Literale.
- Falls Variable in Bedingungen („Operator-Schema“):
Jede Instanziierung (gemäß aktueller Situation) ergibt einen Operator („Operator-Instanz“, „task“)

Operatoren in STRIPS, Beispiel

Operator Go(x,y)

[V]: {At(x), Path(x,y) }

[D]: {At(x)}

[A]: {At(y)}

Alternative Schreibweise:

Liste E ("Effekte")

anstelle von D (delete list) und A(add list)

[V]: {At(x), Path(x,y) }

[E]: {¬ At(x), At(y) }

Operatoren in STRIPS

- Operator mit (V,D,A) anwendbar in Situation s,
falls Vorbedingungen erfüllt:

$$V \subseteq s$$

- Anwendung des Operators führt zu neuer Situation s':

$$s' := (s - D) \cup A$$

$s_0 = \{At(Home), Path(Home, Tante_Emma)\}$

[V]: {At(x), Path(x,y) }

[D]: {At(x)}

[A]: {At(y)}

gemäß CWA:

¬ At(Home)

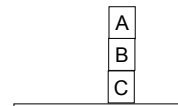
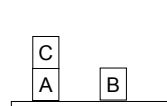
result (Go(Home, Tante_Emma), s₀)

= {At(Tante_Emma), Path(Home, Tante_Emma) }

Beispiel aus „blocks world“: Start, Ziel

auf(C,A)
frei(B)
frei(C)
auf(A, Tisch)
auf(B, Tisch)

auf(A,B)
auf(B,C)



Beispiel: Operatoren

o1 Bewege_C_von_A_auf_den_Tisch:
V: { auf(C,A), frei(C) }
D: { auf(C,A) }
A: { auf(C,Tisch), frei(A) }

o2 Bewege_B_vom_Tisch_auf_C:
V: { auf(B,Tisch), frei(B), frei(C) }
D: { auf(B,Tisch), frei(C) }
A: { auf(B,C) }

o3 Bewege_A_vom_Tisch_auf_B:
V: { auf(A,Tisch), frei(A), frei(B) }
D: { auf(A,Tisch), frei(B) }
A: { auf(A,B) }

H.D.Burkhard, HU Berlin Winter-Semester 2004/05 Vorlesung Einführung in die KI Planen 61

Vorwärtsplaner

H.D.Burkhard, HU Berlin Winter-Semester 2004/05 Vorlesung Einführung in die KI Planen 62

Problemzerlegung

Planungsaufgabe: Start in s_0 , Ziel Z

Falls Z nicht erfüllt:

1. Wähle eine Bedingung $b \in Z$, die in s_0 nicht erfüllt ist.
2. Wähle Operator o , der b erzeugt.
3. Konstruiere einen Plan P_1 für:
 - Start mit s_0 , Ziel Z' := Vorbedingungen von o
4. Konstruiere einen Plan P_2 für:
 - Start mit $s'_0 := \text{result}(s_0, P_1 \bullet o)$, Ziel $Z'' := Z - \{b\}$
 - Nebenbedingung:
 b darf von P_2 nicht zerstört werden („Protection“)
5. Lösung ist dann $P_1 \bullet o \bullet P_2$.

H.D.Burkhard, HU Berlin Winter-Semester 2004/05 Vorlesung Einführung in die KI Planen 63

Problemzerlegung

Planungsaufgabe: Start in s_0 , Ziel Z

Falls Z nicht erfüllt:

1. Wähle eine Bedingung $b \in Z$, die in s_0 nicht erfüllt ist.
2. Wähle Operator o , der b erzeugt.

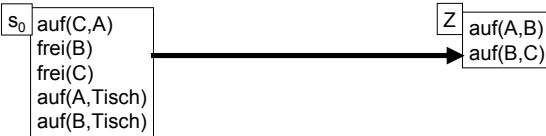
Heuristiken für Auswahl:

- „Wichtige Bedingungen“
- „Bedingungen mit wenig Alternativen für Operatoren“

General Problem-Solver (= GPS, Newell, Simon, 1963).
Means-end-analysis (MEA):
„Distanzen zum Ziel verringern“

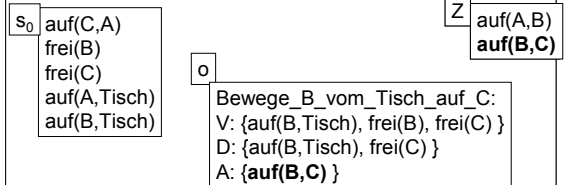
H.D.Burkhard, HU Berlin Winter-Semester 2004/05 Vorlesung Einführung in die KI Planen 64

Problemzerlegung



Problemzerlegung

1. Wähle eine Bedingung $b \in Z$, die in s_0 nicht erfüllt ist.
2. Wähle Operator o , der b erzeugt.



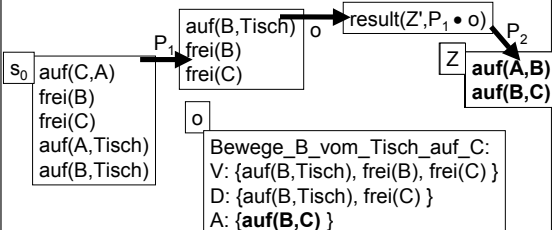
3. Konstruiere einen Plan P_1 für:

- Start mit s_0 , Ziel Z' := Vorbedingungen von o

4. Konstruiere einen Plan P_2 für:

- Start mit $s'_0 := \text{result}(s_0, P_1 \bullet o)$, Ziel $Z'' := Z - \{b\}$

Nebenbedingung: b darf von P_2 nicht zerstört werden



Problemzerlegung

Problem:

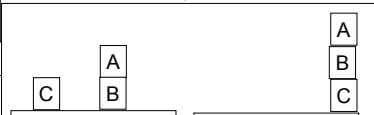
- Verschiedene Zielbedingungen zugleich erfüllen.
- Zielbedingungen können ungültig werden (speziell Zielbedingungen $b \in Z \cap D(o)$)

3. Konstruiere einen Plan P_1 für:

- Start mit s_0 , Ziel Z' := Vorbedingungen von o

4. Konstruiere einen Plan P_2 für:

- Start mit $s'_0 := \text{result}(Z', P_1 \bullet o)$, Ziel $Z'' := Z - \{b\}$



Regressions-Strategien

Spezielle Strategien:

Konsistenz bzgl. Zielbedingungen fordern: $Z \cap D(o) = \emptyset$

Weitere Bedingungen bereits durch P_1 erreichen.

Planungsaufgabe: Start in s_0 , Ziel Z

Falls Z nicht erfüllt:

1. Wähle eine Bedingung $b \in Z$, die in s_0 nicht erfüllt ist.
2. Wähle Operator o , der b erzeugt und konsistent ist.
3. Konstruiere einen Plan P_1 für:
 - Start mit s_0 ,
 - Ziel $Z' := (Z - \{b\}) \cup$ Vorbedingungen von o
4. Lösung ist dann $P_1 \bullet o$.

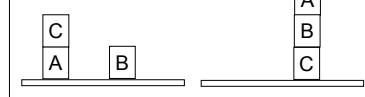
Problemzerlegung

Unvollständigkeit von STRIPS-Planern

Problem:

- Manche Planungsprobleme sind nicht durch einfache Verkettung der Teilpläne lösbar.
- Stattdessen: Verzahnung der Teilpläne notwendig.

„Sussman-Anomalie“:



5. Lösung ist dann $P_1 \bullet o \bullet P_2$.

Hierarchische Planer

In mehreren Ebenen planen und verfeinern.

ABSTRIPS:

- Literale entsprechend ihrer Wichtigkeit anordnen.
- Zunächst vollständigen Plan nur für "wichtige Bedingungen".
- In weiteren Schritten Verfeinerung unter Berücksichtigung von weniger wichtigen Bedingungen (evtl. Neuplanung erforderlich).

Planbasierte Planung

Idee:

- Teilpläne geeignet
 - Zusammenfügen
 - Verfeinern

Teilplan:

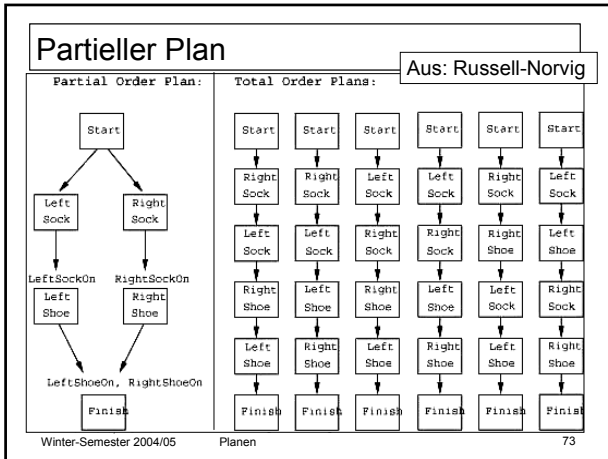
- Partiiell (!) geordnete Menge von Aktionen.
- Realisiert gewisse Zustandsübergänge („Zwischenziele“).
- Kann durch mengenbasiertes Planen erzeugt worden sein.

Anordnung zunächst nur soweit notwendig (gemäß Bedingungen für Aktionen)

- Least Commitment bzgl. Anordnungen
- Verkleinert Suchraum

Suchraum:

- Teilpläne als Zustände
- Operatoren:
 - Verfeinerung,
 - Verkettung bzw. Verzahnung,
 - Modifikation



POP = Partial Ordering Planning

$P=(T, Succ, Depend)$ *Partiell geordneter Plan* mit

T : Menge von Tasks t (Plan-Schritte)
 Jede Task t hat Mengen von Vor- bzw. Nachbedingungen.

$Succ \subseteq T \times T$ Ordnungsconstraints
 $t \rightarrow t'$ bedeutet: t' folgt (irgendwann) auf t

$Depend \subseteq T \times \text{Fakten} \times T$ t erzeugt f für t'
 Menge von Abhängigkeiten („causal links“)
 $\langle t, f, t' \rangle$ bedeutet:
 Fakt f ist Nachbedingung von Task t (Produzent) und Vorbedingung von Task t' (Konsument)

H.D.Burkhard, HU Berlin Winter-Semester 2004/05 Vorlesung Einführung in die KI Planen 74

POP = Partial Ordering Planning

$P=(T, Succ, Depend)$ sei ein partiell geordneter Plan.

Konsistenz:
 Plan P (d.h. $Succ \subseteq T \times T$) ist *konsistent*, falls Vervollständigung zu totaler Ordnung möglich ist.

Konflikt:
 Abhängigkeit $\langle t, f, t' \rangle \in Depend \subseteq T \times \text{Fakten} \times T$ steht in *Konflikt* mit Task $t_1 \in T$, falls $Succ \cup \{t \rightarrow t_1 \rightarrow t'\}$ ist konsistent, aber t_1 hat $\neg f$ als Nachbedingung.

H.D.Burkhard, HU Berlin Winter-Semester 2004/05 Vorlesung Einführung in die KI Planen 75

Rekursive Definition von POP

$POP((T, Succ, Depend), Agenda)$ liefert zu einem partiellen Plan $P=(T, Succ, Depend)$ und einer Liste offener Aufgaben $Agenda \subseteq \text{Fakten} \times T$ einen verfeinerten Plan $P'=(T', Succ', Depend')$, und eine neue Liste offener Aufgaben $Agenda'$.

Start mit $P=(\{t_{start}, t_{ziel}\}, \{t_{start} \rightarrow t_{ziel}\}, \emptyset)$ und $Agenda = \{[f_1, t_{ziel}], \dots, [f_n, t_{ziel}]\}$ für Zielfakten f_1, \dots, f_n

t_{start} hat keine Vorbedingungen und die Startfakten als Nachbedingungen
 t_{ziel} hat keine Nachbedingungen und die Zielfakten als Vorbedingungen

H.D.Burkhard, HU Berlin Winter-Semester 2004/05 Vorlesung Einführung in die KI Planen 76

POP((T, Succ, Depend), Agenda)

- 1) Falls $Agenda = \emptyset$: EXIT mit $P = (T, Succ, Depend)$.
- 2) Wähle $(f, t) \in Agenda$.
- 3) Wähle $t' \in T$ oder neue Task t' , die f erzeugt.
 Falls nicht möglich: Backtracking (falls keine Alternativen: Misserfolg).
 $Depend' := Depend \cup \{ \langle t', f, t \rangle \}$, $Succ' := Succ \cup \{ t' \rightarrow t \}$.
 Falls $t' \in T$: $T' := T$.
 sonst: $T' := T \cup \{ t' \}$, $Succ' := Succ \cup \{ t_{start} \rightarrow t', t' \rightarrow t_{ziel} \}$.
- 4) $Agenda' := Agenda - \{ (f, t) \}$.
 Falls $t' \notin T$: $Agenda' := Agenda' \cup \{ (f'_1, t'), \dots, (f'_m, t') \}$
 für die Vorbedingungen f'_1, \dots, f'_m von t' .
- 5) Für alle $t_i \in T'$ und alle $\langle t, f, t' \rangle \in Depend'$, die in Konflikt stehen:
 $Succ' := Succ' \cup \{ t_i \rightarrow t \}$ (Demotion), falls Ergebnis konsistent.
 bzw. $Succ' := Succ' \cup \{ t' \rightarrow t_i \}$ (Promotion), falls Ergebnis konsistent.
 Falls beides nicht möglich: Backtracking.
- 6) Weiter mit POP((T', Succ', Depend'), Agenda')

POP = Partial Ordering Planning

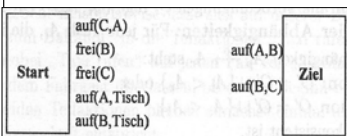
Nicht-Determinismus bei 2), 3), 5):

- Die Reihenfolge in 2) hat nur Einfluss auf Effizienz.
- Alternativen in 3) bzw. 5) führen evtl. zu Backtracking.
- EXIT mit Misserfolg, falls alle Alternativen verbraucht.

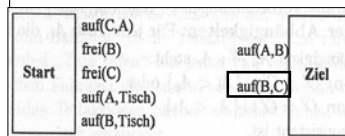
- Das Verfahren ist vollständig und korrekt.
- Jede totale Ordnung über dem Resultat ergibt einen vollständig geordneten korrekten Plan

- Least Commitment:
Verzögerte Festlegung der Reihenfolge.

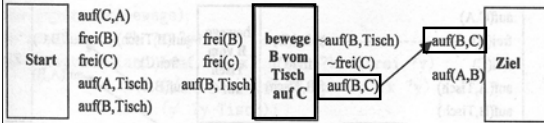
Beispiel (aus Görtz et al.: Handbuch der KI)



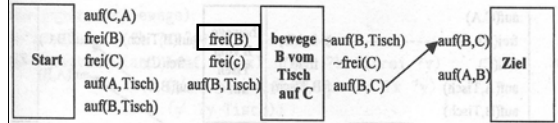
Beispiel (aus Görtz et al.: Handbuch der KI)



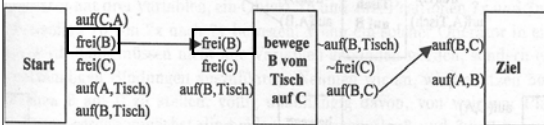
Beispiel (aus Görtz et al.: Handbuch der KI)



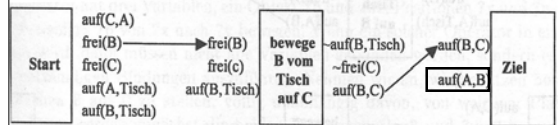
Beispiel (aus Görtz et al.: Handbuch der KI)



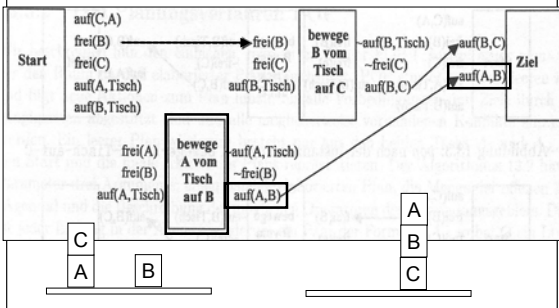
Beispiel (aus Görtz et al.: Handbuch der KI)



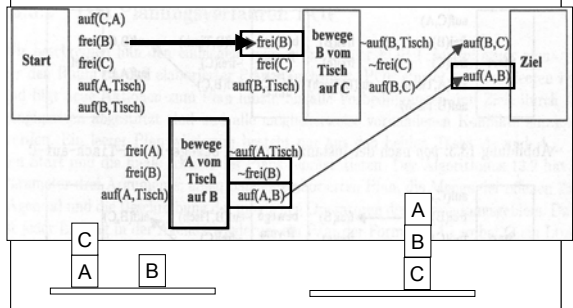
Beispiel (aus Görtz et al.: Handbuch der KI)



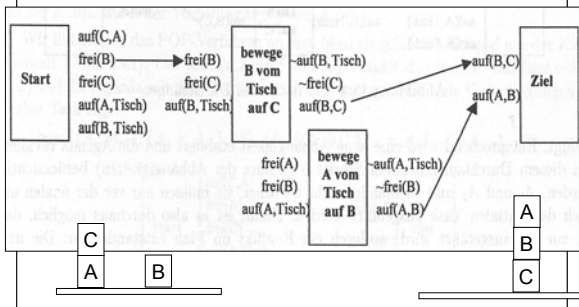
Beispiel (aus Görtz et al.: Handbuch der KI)



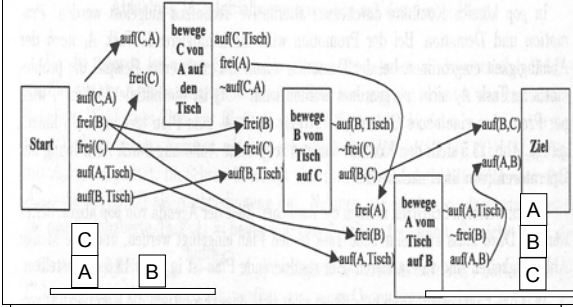
Beispiel (aus Görtz et al.: Handbuch der KI)



Beispiel (aus Görtz et al.: Handbuch der KI)



Beispiel (aus Görtz et al.: Handbuch der KI)



POP = Partial Ordering Planning

- Erweiterungen
 - für Variable in Tasks.
- Erfordert u.a.:
 - Bindungen an Objekte verwalten,
 - Unifikation bei Schritt 3),
 - weitere Konfliktlösungsmechanismen.
- für Allquantifizierung in Nachbedingungen.

Graphbasiertes Planen

Kompakte Darstellung des Suchraums als Graph:
Möglicherweise parallel ausführbare Aktionen in einer Schicht.

- Markierung von Konflikten bzgl. paralleler Ausführbarkeit.

Kombinierte Vorwärts- und Rückwärtssuche im Zustandsraum.

- Abwechselnd:
 - Erweiterung des Graphen (Expansion, vorwärts)
 - Test auf Lösung (Extraktion, rückwärts)

Least commitment bzgl. Reihenfolge

Graphbasiertes Planen

Planungsgraph:
Bipartiter Graph

- Propositionsknoten (Bedingungen)
- Aktionsknoten (Tasks)

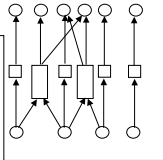
2 Sorten von Knoten,
Kanten nur zw. untersch. Sorten

Frame-Axiom !

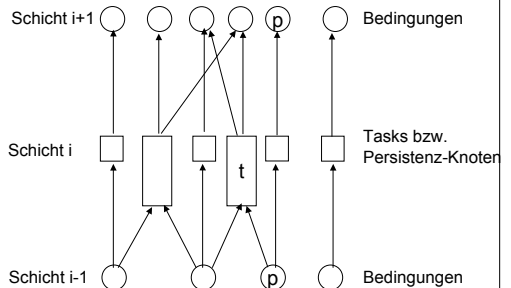
einschl. „Persistenzaktionen“ für unveränderte Bedingungen

Anordnung in Schichten

- unterste Schicht: Literale der Startsituation
- Kanten von
 - Vorbedingungen zu Tasks
 - Tasks zu Nachbedingungen



Planungsgraph



Planungsgraph

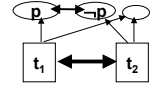
Planungsgraph beschreibt unterschiedliche Pläne:

- Tasks einer Ebene, deren Vor- und Nachbedingungen nicht in Konflikt stehen, sind in beliebiger Reihenfolge (bzw. parallel) ausführbar.
- Beschreibung von Konflikten für „nicht parallel“ ausführbare Tasks einer Ebene notwendig.

mutex (=mutual exclusive) -Relation
zwischen Tasks einer Ebene

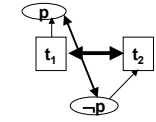
Konflikte zwischen Tasks einer Ebene

- Inkonsistente Nachbedingungen



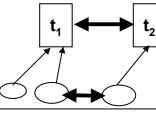
oder

- Interferenz:
Eine Nachbedingung der einen Task ist Negation einer Vorbedingung der anderen Task



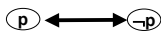
oder

- Konflikt bei Vorbedingungen
(- siehe nächste Folie)
von zwei Tasks



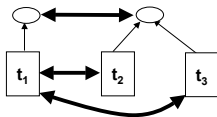
Konflikt zwischen Bedingungen einer Ebene

- Eine Bedingung ist Negation der anderen



oder

- Alle Paare erzeugender Tasks stehen in Konflikt



Graphbasiertes Planen

Abwechselnd 2 Phasen:

- Expansion des Planungsgraphen („Vorwärts“)
 - Notwendige Bedingungen für Existenz eines Planes
 - Konflikte markieren
- Extraktion der Lösung („Rückwärts“)
 - Suche im Planungsgraphen nach Lösung:
 - Verträglichkeit sichern
 - Abbruch, wenn Lösung gefunden

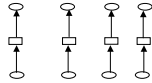
Start mit Schicht 1:

- enthält Bedingungen der Startsituation

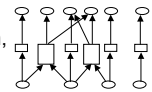
Expansion (Phase 1)

1. Neue Schicht i und i+1 erzeugen

a) Alle Bedingungen der Schicht i-1 über Persistenzknoten der neuen Schicht i in neue Schicht i+1 übernehmen.



b) Alle mit Bedingungen der Schicht i-1 ausführbaren Tasks in die Schicht i eintragen, Nachbedingungen in Schicht i+1 eintragen.



2. Konflikte markieren

Extraktion der Lösung (Phase 2)

Start: Falls alle Zielbedingungen in Schicht i+1 enthalten:
 $B :=$ Zielbedingungen in Schicht i+1. $P := \emptyset$.

(*) Iteration:

Falls $i=1$: Exit(aus P ist ein Plan erzeugbar).

Falls $i > 1$:

a) Wähle minimale Menge T (Tasks bzw. Persistenzknoten) der Schicht i, die B erzeugen.

b) Falls die Tasks aus T paarweise nicht in Konflikt:

$B :=$ Vorbedingungen von T, $P := P \cup T$, $i := i-2$.

Weiter bei (*)

Andernfalls: Backtracking

(andere T in diesem oder in vorherigen Schritten).

Falls alle Versuche erfolglos: Weitere Expansion (Phase 1).

Graphbasiertes Planen

- Anwendbar für variablenfreie Bedingungen

- Erzeugung eines Planes aus P:
Lineare Ordnung der Tasks T aus P unter Beachtung der Anordnung im Planungsgraphen.

- Der Algorithmus terminiert.
Fixpunkt: Schichten wiederholen sich.

- Der Algorithmus ist korrekt und vollständig.

- Weitere Optimierungsmöglichkeiten (siehe Literatur).

Beispiel (aus Görtz et al.: Handbuch der KI)

Kochen
V: { sHand }
N: { Dinner }

BlüVa
V: { Ruhe }
N: { Blumen }

Wischen
V: { }
N: { -Schmutz, -sHand }

Saugen
V: { }
N: { -Schmutz, -Ruhe }

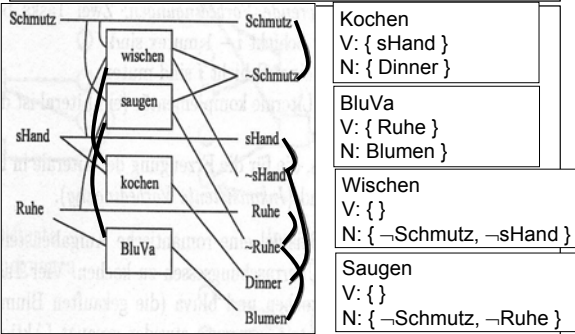
Start:

- Schmutz,
- Saubere Hand (sHand),
- Ruhe

Ziel:

- Dinner,
- Blumen,
- Schmutz

Beispiel (aus Görtz et al.: Handbuch der KI)



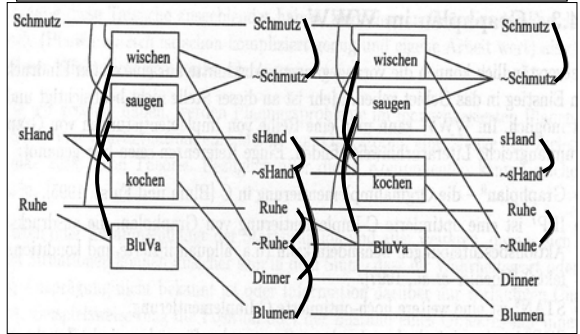
Kochen
V: { sHand }
N: { Dinner }

BluVa
V: { Ruhe }
N: Blumen }

Wischen
V: { }
N: { -Schmutz, -sHand }

Saugen
V: { }
N: { -Schmutz, -Ruhe }

Beispiel (aus Görtz et al.: Handbuch der KI)



Beispiel (aus Görtz et al.: Handbuch der KI)

