

Einführung in wissensbasierende Systeme

Zusammenfassung des Skriptums
SS 03

Florian Bauer (www.flobauer.net)

Teile übernommen von:
„Eristoff“ (siehe www.informatik-forum.at) &
Szabolcs Rozsnyai (<http://www.szabolcs.at.tf/>)

1 Einführung

Was ist ein wissensbasiertes System(Expertensystem, intelligentes System)?

Computersystem, das Probleme lösen kann, für die spezielles Wissen erforderlich ist.
genauer: Weltwissen wird in deklarativer Form gespeichert(Wissensbasis) und verarbeitet
(deklarativ: beschreibend in einer geeigneten Sprache)

generisches Wissen = allgemeines Hintergrundwissen Funktionsweise d. Kopierers
Fallspez. Wissen = Wissen für Bearbeitung eines konkreten Problems Zustand v. Kopierer

Warum wissensbasierte Systeme?

In einer Wissensbasis werden über Fakten hinaus auch Zusammenhänge von Fakten in expliziter Form dargestellt. Daraus kann durch Wissensverarbeitung(Interferenz) neues Wissen gewonnen werden. Zusammenhänge von Fakten werden in Regeln beschrieben.
Vorteile: Flexibilität, Änderbarkeit & Erweiterbarkeit, Transparenz

Anforderungen an ein wissensbasiertes System:

- Lösung von Problemen die einen gewissen Anspruch haben
- Trennung von Problemwissen und Verarbeitung
- Speicherung und Organisation von Wissen
- Verständliche Wissensanzeige
- Transparenz der Lösungsfindung
- Erweiterbarkeit/Änderbarkeit für neues Wissen

Probleme:

- Komplexität der Aufgabenstellung wird oft unterschätzt
- formale Grundlagen sind zum verstehen der Funktionsweise nötig
- Schließen mit Hausverstand muss möglich sein
- Tiefenwissen muss vorhanden sein präzise und umfassende Modellierung

Ansätze der Wissensverarbeitung:

- **Kognitive AI:** Wissensverarbeitung erfolgt nach einem Muster von Intelligenz. Setzt eine Theorie der Intelligenz voraus.
- **Rationale AI:** ereignisbasierte Wissensverarbeitung. Es wird keine direkte Entsprechung zu einem Denkmuster hergestellt, aber die Wissensverarbeitung muss nachvollziehbar und erklärbar sein. Ziel ist es, ein System zu entwickeln, dass ein rationales Verhalten/Agieren zeigt.

Für wissensbasierte Systeme ist der Ansatz der Rationalen AI maßgebend.

Systemansätze innerhalb der AI:

- **Symbolverarbeitende Systeme:**
Basieren auf der Manipulation von Information und Wissen, das in symbolischer Form repräsentiert wird.
- basiert auf Logik

- Wissen wird in deklarativer Form repräsentiert
- Verarbeitung durch Inferenzprozesse auf der Symbolebene
- WBSysteme basieren darauf
- **Subsymbolische Systeme:**
 Basieren auf Signalen. System ist mit Sensoren und Aktoren mit der realen Welt verbunden.
 Dateneingabe z.B. über Tastatur wird abgelehnt.
 - Signale als wesentliche Element
 - System ist durch Sensoren/Aktoren mit der realen Welt verbunden
 - Einklang mit evolutionären Theorie der Intelligenz => Neuronale Netze

Architektur von wissensbasierten Systemen:

Wissensbasis:

Wissen ist hier in deklarativer Form abgelegt, Wissen umfasst Fakten und Regeln, generisches (Problemlösungsfall) bzw. fallspezifisches (Problembereich) Wissen

Interferenzkomponente:

Das Wissen von der Wissensbasis wird hier verarbeitet, dabei werden neue Fakten und Regeln abgeleitet, je größer die Wissensbasis desto komplexer

- * Verarbeitet Wissen aus der Wissensbasis.
- * Leitet Wissen aus Fakten und Regeln ab

Wissenserwerbskomponente:

Zwei Möglichkeiten der Erweiterung:

Manuell: Wissensingenieur über entsprechende Schnittstelle (nicht gleich Endbenutzer-Schnittstelle)

Automatisch: System lernt aus Problemlösungen selbstständig neues Wissen und erweitert die Wissensbasis

User-Interface:

Schnittstelle zum Endbenutzer und Wissensingenieur,

Erklärungskomponente:

Erteilt dem Benutzer Auskunft über die Lösungsfindung, damit sie der Benutzer nachvollziehen kann. 2 Funktionen:

How-Funktion: wie wird die Lösung gefunden (Schritte)

Why-Funktion: warum werden bestimmte Schritte unternommen bzw. bestimmte Informationen verlangt wurden

2 Problemlösen und Suchen

Suchproblem ist definiert durch:

1. einen Startzustand
2. eine nichtleere Menge von Zielzuständen(, die alle den Zieltest bestehen)
3. eine nichtleere Menge von Operatoren, wobei durch Anwendungen von Operatoren auf einen gegebenen Zustand Nachfolgezustände generiert werden. Die dabei entstehenden Kanten haben positive Kosten > 0

4. eine Kostenfunktion für Pfade, die angibt, wie sich die Kosten eines Pfades aus den Kosten der Operatorenanwendungen ergeben

Ein *Zustandsraum* wird durch Anfangszustand und Operatoren definiert.

Die *Lösung* eines Problems ist ein Pfad vom Startzustand zu einem Zielzustand

Minimale Lösung ist eine Lösung mit minimalen Pfadkosten.

Eigenschaften von Suchverfahren:

- Vollständigkeit (werden alle Lösungen gefunden)
- Zeitkomplexität (Zeitverhalten bei wachsender Größe der Problemistanz)
- Speicherkomplexität (Speicherbedarfswachstum bei wachsender Größe der Problemistanz)
- Optimalität: (wird die beste Lösung gefunden, bez. der Kostenfunktion)

Uniformierte Suche

Es werden keine problemspezifischen Informationen ausgenutzt, um während der Suche Teile des Suchraums anderen Teilen vorzuziehen.

Breitensuche:

Ebenenweise Expansion der Zustände. Zustand der Tiefe d wird erst expandiert, wenn alle Zustände der Tiefe $d-1$ expandiert wurden

Vollständig, optimal, Speicherbedarf & Zeitkomplexität: b^d ($b = \#$ d. Nachfolger, d =Tiefe der Lösung)

Uniform Cost Search:

Modifikation der Breitensuche; Zustände die niedrige Kosten verursachen werden bevorzugt; Zuerst wird der Knoten mit den geringsten Kosten expandiert; Falls Lösung gefunden wird, werden die anderen Lösungspfade nur mehr solange verfolgt, bis die Kosten des Pfades die Kosten der günstigsten gefundenen Lösung übersteigen. Falls günstigere Lösung gefunden, wird sie weiter verglichen

Vollständig, optimal, Speicherbedarf & Zeitkomplexität: b^d ($b = \#$ d. Nachfolger, d =Tiefe der Lösung)

Tiefensuche:

Generiert immer die Nachfolger eines tiefsten Zustands, wenn alle Zustände der Tiefe d keine Nachfolger mehr besitzen, dann werden alternative Zustände der Tiefe $d-1$ expandiert. (Achtung: unendlich absteigende Pfade – terminiert die Suche nicht)

Nicht vollständig, nicht optimal, Zeitbedarf b^m , Speicherkomplexität $b \cdot m$ ($m =$ maximale Pfadlänge). Wenig Speicher!

Tiefensuche mit wachsender Tiefenschranke:

Wie Tiefensuche; es wird aber eine globale Tiefenschranke eingeführt, falls die Tiefenschranke erreicht wird, wird zurückgesetzt; damit terminiert die Tiefensuche immer; falls Schranke zu gering gewählt, wird keine Lösung gefunden!!

Vollständig, optimal, Speicherbedarf: b^d ($b = \#$ d. Nachfolger), Zeitkomplexität: $b \cdot d$

Vereint Vorteile von dfs (wenig Speicher) mit Breitensuche (optimal, vollständig).

Vergleich der Suchverfahren:

(b = Verzweigungsgrad, d = Tiefe, m =maximale Pfadlänge)

	<i>Bfs</i>	<i>Ucs</i>	<i>Dfs</i>	<i>Dfid</i>
Zeit	b^d	b^d	b^m	b^d
Speicher	b^d	b^d	bm	bd
Optimal?	Ja	Ja	nein	Ja
Vollständig?	Ja	Ja	nein	Ja

Heuristische Suche

Diese Verfahren treffen ihre Entscheidungen aufgrund von heuristischer Informationen. Dabei werden die Kosten eines Knotens zu seinem Zielknoten abgeschätzt.

Best-first Verfahren:

Grundidee der best-first Verfahren ist das expandieren des bestbewerteten Zustandes zuerst, wobei die Bewertung auf Heuristiken basiert.

Greedy Search:

Charakteristisch für Greedy Search ist das Schätzen der Kosten vom aktuellen Knoten n zu einem Zielknoten ohne Berücksichtigung bisher angefallener Kosten.

- Greedy Search ist nicht optimal und nicht vollständig.
- Speicher und Zeit: b^m .
- Entstehung von Zyklen
- in einem lokalen Minimum kann es stecken bleiben od. zw. zwei hin und her wechseln

A* Suche:

Die tatsächlichen Kosten vom Start zum aktuellen Knoten gehen mit in die Entscheidung ein. Dadurch wird erreicht, dass Knoten nicht unendlich lange von der Expansion ausgeschlossen werden.

Wird als best-first bezeichnet.

A* Suchen haben folgende Eigenschaften:

- Der Zustandsraum besitzt einen endlichen Verzweigungsgrad.
- Die Kosten JEDER Kante sind positiv und > 0 .

A* Suche ist optimal und vollständig.

Speicher und Zeit: exponentiell.

3. Wissensrepräsentation:

Anforderungen an die Wissensrepräsentation:

- Ausdrucksstärke:
Der Formalismus muss ausreichend mächtig sein, damit das relevante Wissen in ihm überhaupt dargestellt werden kann.
- Verarbeitbarkeit:
Das gespeicherte Wissen muss verarbeitet werden können.

- **Flexibilität:**
Der Ansatz zur Wissensrepräsentation soll allgemeiner Natur sein, sodass darin Wissen aus verschiedenen Anwendungsbereichen (Domänen) dargestellt werden kann.
- **Modularität:**
Veränderung des Wissens muss leicht möglich sein.
- **Verständlichkeit:**
Wissen muss in verständlicher Form in der Wissensbasis abgelegt werden.
- **Unvollständige Information:**
Die „Welt“ ist dem Agenten so bekannt, dass er alle wahren und falschen Fakten bzw Zusammenhänge kennt. Oft muss aber unvollständige Information verarbeitet werden (zb Sam: männlich oder weiblich?)
- **Erfassung unsicheren Wissens:**
Eventuell ist bestimmtes Wissen nicht gespeichert, dabei können auftreten:
 - Vage Informationen: Fakten sind ungenau
 - Unsichere Informationen: Fakten können mit einer Unsicherheit behaftet sein. (Wahrscheinlichkeit)

Arten von Wissen:

- **Faktenwissen:**
Unter Fakten versteht man die Beschreibung von elementaren Sachverhalten, die in der Welt auftreten
- **Zusammenhänge (Beziehungen):**
Fakten können untereinander in Beziehung stehen. Dabei gibt es zeitliche (Entwicklung von Eigenschaften) und kausale (erfassen Ursache – Wirkung) Zusammenhänge.
- **Methodisches Wissen:**
Gibt Auskunft darüber, wie zur Problemlösung vorgegangen werden kann.
- **Meta Wissen:**
Wissen über die Wissensbasis selbst. dh. Aufbau, Struktur und Verwendbarkeit des Wissens.

Methoden der Wissensrepräsentation:

- Logik basierte Methoden
- Prozedurale Methoden
- Objekt orientierte Methoden

Prozedurale Methoden:

Hierbei wird Wissen in Form von Prozeduren dargestellt. Dabei ist sich die Inferenzkomponente des in der Prozedur gespeicherten Wissens bewusst und der Aufruf dieser Prozedur unterliegt ihrer expliziten Kontrolle.

Die Inferenzkomponente ist mit Wissen über die Problemlösungsmöglichkeiten ausgestattet. Die in der Prozedur enthaltene Wissen sollte transparent und verständlich sein.

Vor/Nachteile:

- +) Bei der Formulierung der Prozedur kann auf die Abfolge der Suche eingegriffen werden.
-) Es können eventuell Lösungen vergessen oder übersehen werden
-) Mangelnde Modularität
-) eingeschränkte Flexibilität

Logikbasierte Methoden:

Logik: Aus einer Formulierung bestimmter Sachverhalte durch Aussagen bzw Sätze in einem logischen System, kann man auf das Zutreffen weiterer Sachverhalte, die wahr oder falsch sind, schließen.

Komponenten:

Syntax und Semantik.

Syntax: Regeln wie man Sätze baut

Semantik: Bedeutung der Sätze

Wissen kann in Form von logischen Sätzen dargestellt werden => neues Wissen kann abgeleitet

Prädikatenlogik 1.Stufe:

Stellung der Prädikatenlogik:

Vorteile:

- der wichtigste Formalismus der formalen Logik, über den es eine Fülle von theoretischen Resultaten gibt.
- Ist sehr ausdrucksstark.
- Hat eine einfache und natürliche Syntax und eine intuitive Semantik.

Nachteile:

- es können auch nicht berechenbare Funktionen spezifiziert werden.
- Selbst für entscheidbare Fragmente kann die Rechenzeit sehr hoch sein.

Aussagenlogik:

tritt als Subformalismus der PL auf. Gegenstand der AL sind in sich geschlossene Aussagen, die mit wahr oder falsch bewertet werden können und deren logische Beziehungen zueinander.

Bsp: A1: der Mond ist aus grünem Käse, A2: es regnet, A3: es ist nass

Diese Aussagen werden durch logische Verknüpfungen zu komplexeren Aussagen:

Bsp: C1: Es regnet und der Mond ist aus grünem Käse

Syntax Bsp:

$\neg a \vee b \wedge c \rightarrow d \vee a$

Semantik:

Die Semantik der Formeln der Aussagenlogik wird durch Interpretationen und Modelle festgelegt. In der Abstraktion kann dabei jede atomare Aussage a entweder den Wert wahr oder falsch annehmen.

Modell:

Eine Interpretation I ist ein Modell für eine Aussage ϕ wenn $I(\phi) = t$

Folgende Arten von Aussagen können unterschieden werden:

- Tautologie: jede Interpretation ist ein Modell von $I(\phi \vee \neg \phi, \phi \wedge \neg \phi, \phi \rightarrow \phi, \phi \leftrightarrow \phi)$

- Kontradiktion: keine Interpretation ist ein Modell der Aussage ($a \wedge \neg a$)
- Erfüllbare Aussage: die Aussage besitzt ein Modell

Erfüllbarkeitsproblem:

Die Erfüllbarkeit AL Formeln werden durch Algorithmen entschieden. Im schlimmsten Fall müssen dabei 2^n Interpretationen getestet werden. (n = Anzahl der Atome) Es sind heute keine Algorithmen bekannt, die dieses Problem in polynomineller Laufzeit lösen.

Prädikatenlogik:

Die PL stellt wesentlich stärkere Ausdrucksmittel zur Verfügung als die AL. Gegenstand des Interesses sind hierbei Objekte und deren Eigenschaften sowie Beziehungen zueinander.

Syntax:

Ausgangspunkt für die Bildung von Formeln in der PL ist ein Vokabular $f+r$ Objekte, Eigenschaften und Beziehungen, sowie ein Satz logischer Grundzeichen.

Üblicherweise vorhanden:

- Konstantensymbole: damit wird ein bestimmtes Objekt angesprochen
- Funktionssymbole: steht für eine Funktion
- Variablensymbole: steht für ein Objekt, das für die Auswertung einer Formel festgelegt werden muss
- Prädikatensymbole: steht für ein n -äres Prädikat. $N=1$: Eigenschaft des Objektes, $n>1$ Beziehung zwischen Objekten
- Terme: Terme bezeichnen Objekte und sind definiert durch:
 - Jedes Konstantensymbol und jedes Variablensymbol ist ein Term
 - Wenn t_1, \dots, t_n bereits enthaltene Terme sind und f ein n -stelliges Funktionssymbol, dann ist auch $f(t_1, \dots, t_n)$ ein Term.
- Atomare Formeln: Wenn P ein n -stelliges Prädikatensymbol ist und t_1, \dots, t_n Terme sind, dann ist $P(t_1, \dots, t_n)$ eine atomare Formel
- Formeln
- freie Variable: x ist frei, wenn x durch keinen Quantor gebunden ist.
- Satz: ist eine Formel ohne freie Variablen.
- Allquantifizierter Satz: ein Satz der Form $\forall x_1 \forall x_2 \dots \forall x_n \phi$ heist allquantifiziert.
- Literal: ist eine Atomformel oder deren Negation.
- Klausel: ist eine Disjunktion von Literalen.
- Horn Klausel: ist eine Klausel, in der höchstens ein pos. Literal vorkommt.

Semantik:

Formal hat eine Interpretation I für ein Vokabular V zwei Komponenten:

- einen beliebigen nichtleeren Wertebereich D von Objekten.
- Zuordnungen von Objekten, Funktionen und Relationen zu über D zu allen Konstanten- Funktions- und Prädikatensymbolen.

Erfüllbarkeitsproblem:

Im Gegensatz zur AL sind die Probleme der Erfüllbarkeit und der Gültigkeit eines Satzes der PL im allgemeinen nicht entscheidbar.

Die PL ist immer semi-entscheidbar, dh. Es gibt einen Algorithmus, der genau dann terminiert, wenn die Eingabe ein gültiger Satz ist. Allerdings kann nicht vorhergesagt werden, ob der Algorithmus für eine Eingabe terminiert, geschweige denn welche Rechenzeit er benötigt.

Aspekte der Wissensmodellierung:

Wie können Wissensinhalte dargestellt werden:

- Axiome, Fakten und Regeln:
Die Beschreibung des Problembereiches erfolgt über eine Menge von Sätzen, die wahre Sachverhalte ausdrücken (domain theory). Wird erweitert um fallspezifisches Wissen (Fakten).
- Universe of discourse:
Der Wertebereich in der Wissensrepräsentation ist in der Regel eindeutig festgelegt und wird Universe of discourse genannt
- Unique Names Assumption (UNA):
Um sicherzustellen, dass verschiedene Konstantensymbole auch wirklich verschiedene Objekte bezeichnen, kann durch hinzufügen von Literalen (zb: $c1 \neq c2$) dies sichergestellt werden. Dies wird als Unique Names Assumption bezeichnet.
- Domain Closure Axiom (DCA):
Der Wertebereich kann durch folgendes Axiom abgeschlossen werden:
 $? x ? x = c1 ? x = c2 ? \dots ? x = cn ?$

Objektorientierte Methoden:

In Prozeduralen und Logikbasierten Methoden werden Eigenschaften und Beziehungen durch Sprachmittel beschrieben dabei geht die syntaktische Struktur verloren.

Bei OO Methoden wird aber die Struktur der Objekte beibehalten!

Charakteristische Merkmale:

- Zentrale Beschreibung:
Die Objektstruktur wird zentral beschrieben.
- Deklarative Strukturbeschreibung:
Strukturen und deren Zusammenhänge werden in deklarativer Form erfasst. Dies gilt auf verschiedenen Ebenen der Beschreibung:
- Objektstruktur:
konkrete Objekte: Eigenschaften werden explizit durch Attributwerte beschrieben.
- Klassenstruktur:
Gleichartige Objekte sollen durch abstrahierende Beschreibungen erfasst werden und die Ähnlichkeit der Objekte soll aus deren Beschreibung hervorgehen.

Vererbung:

Es können 2 Arten von Vererbung unterschieden werden:

1. Subkonzept/Superkonzept Vererbung:
Objekte ähnlicher Typen können oft als Vereinfachung allgemeiner Typen aufgefasst werden. So sind zb PKW und Motorrad ein Subkonzept von KFZ, dieses wiederum ist ein Superkonzept von PKW und Motorrad.
 2. Instanzierungs-vererbung:
Wenn ein Objekt eine Instanz eines Typs ist, so erbt dieses Objekt alle Eigenschaften, die ein prototypisches Objekt dieses Typs hat. (my_bike erbt die Eigenschaften von Bike)
- Wichtige Aspekte der Vererbung sind Überdeckung und Mehrfachvererbung:
Überdeckung:
Eigenschaften werden vom allgemeinen Fall auf den speziellen übertragen. Falls für einen speziellen Fall diese Eigenschaft nicht zutrifft, so wird sie explizit Redefiniert. => Überdeckung

- Mehrfachvererbung:
Es ist allgemein Zulässig, dass ein Konzept mehrerer Superkonzepte hat. In diesem Fall erbt das Konzept die Eigenschaften von jedem Subkonzept.

Frames:

Framesysteme basieren auf einem objektorientierten Ansatz zur Wissensrepräsentation. Dabei werden Stereotype Situationen durch ein Gerüst (Frame) beschrieben, der die üblichen, möglichen Eigenschaften der Situation beinhaltet und so eine abstrakte, generische Beschreibung darstellt, die ein Prototyp für ein Objekt ist.

Durch festlegen der konkreten Eigenschaften wird aus der stereotypen Situation eine spezielle Situation, die eine Instanz des generischen Konzeptes ist.

Trigger

Man kann Prozeduren in Framestrukturen einhängen die nach einem Aktivierungsmuster (=Trigger) ausgeführt werden

Einfache Framesysteme:

Ein Frame ist ein Objekt, das durch eine Objekt-ID angesprochen werden kann und durch eine Liste von Attributen beschrieben ist, die Werte aus einem Wertebereich annehmen können. Die Unterscheidung zwischen generischem und Instanzframe kann durch das spezielle boolsche Attribut GENERIC erfolgen.

Die Erfassung der Instanzierungs und Sup/Superkonzept- Beziehungen geschieht durch die Attribute: AKO (a kind of) und INSTANCES.

Bsp:

```
Frame = RECORD OF
    FRAME ID: object-id;
    GENERIC: boolean;
    AKO: list of frame-id's;
    INSTANCES: list of frame-id's;
    A1: W1;
    ...;
    Am : Wm;
END;
```

Instanzframes: ovale umrundung der Frame-ID, generische Frames: rechteckige Umrundung.

Beispiele für Framesysteme:

- FRL (Frame Representation Language): vom MIT in den 70ern entwickelt
- RLL (Representation Language Language)
- KEE (Knowledge Engeneering Environment)
- FrameWork

Semantische Netze:

In einem semantischen Netz, wird Wissen in Form eines gerichteten Graphen repräsentiert. Die Knoten des Graphen werden als Konzepte bezeichnet und stellen Objekte, Klassen, Behauptungen und Ereignisse dar. Beziehungen werden durch Kanten modelliert, diese werden als konzeptionelle Relationen bezeichnet.

Unterschied zu OO Methoden: in semantischen Netzen werden Eigenschaften von Objekten als gleichwertige Objekte dargestellt und in Beziehung zu diesen gesetzt. OO: Eigenschaft eines Objektes wird dem Objekt untergeordnet. => OO: Tendenz zu Datenkapselung, semantische Netze: Offenlegung der Beziehungen

Unterschiede semantische Netze – logik basiertem Ansatz:

- Darstellung der Wissensinhalte in SN besser erfassbar und überschaubar
- Wissen um Vererbung wird in SN auf eine modulare Art erfasst.
- Struktur der SN erlaubt eine direkte Verarbeitung durch Algorithmen die auf diesem operieren
- SN haben keine rein deklarative Semantik, sondern Interpretation hängt vom Algorithmus ab.

Bsp für SN:

- Conceptual Graphs
- SnePs
- NETL
- KL-ONE

Anwendung: natürliche Sprachen

Typen von semantischen Netzanformatismen:

- Relationale Graphen:
In den einfachsten semantischen Netzen sind Knoten atomare Konzepte. Eine Kante drückt eine Beziehung zwischen Konzepten aus, mit deren Art die Kante markiert ist.
Aus logischer Sicht kann in einem relationalen Graphen ein Sachverhalt dargestellt werden, der in einem Fragment der PPL 1 .Stufe beschrieben werden kann. Knoten wird als Zusicherung aufgefasst, dass ein Objekt vom Typ des Konzepts existiert, Kanten werden als Prädikate interpretiert. Damit können lediglich Existenz und Konjunktion (oder) ausgedrückt werden.
Allquantor: Kann durch spezielle Knotenmarkierungen dargestellt werden, aber Wirkungsbereich bleibt unklar.
- Propositionale Netze:
Etwas ausdrucksstärker als relationale Graphen. => Komplexere Sachverhalte können dargestellt werden (durch Verknüpfung bzw Verbindung von Aussagen. Es besteht die Möglichkeit Netze zu schachteln. Dabei stellt der mit einem PROPOSITION -Rahmen umgebene Graph einen abgeschlossenen Sachverhalt dar.

Logische Formel mit Quantoren:

Oft treten logische Quantoren auf (zb Allquantor). Es gibt verschiedene Möglichkeiten diese darzustellen:

- Verwendung von Knoten zur Darstellung von Variablen oder Quantoren
- Verwendung spezieller Kanten für die Quantoren

Vererbung:

Wird dargestellt durch ISA- Kanten. Damit wird sowohl die Instanzierungsbeziehung als auch die Sub-Superkonzept Beziehung ausgedrückt.

4. Logik und Inferenz

Inferenz in Regelorientierten Repräsentationen

Die Inferenzstrategien lassen sich in 2 Gruppen aufteilen:

- Vorwärtsverkettung:
Verarbeitung findet datengetrieben statt ausgehend von den Daten wird nach Wissen gesucht.
- Rückwärtsverkettung:
Verarbeitung findet zielorientiert statt, ausgehend vom Gesamtziel wird nach Wissen gesucht, das anwendbar ist und das das Gesamtziel in eine oder mehrere einfach Unterziele zerlegt.

Vorwärtsverkettende Systeme

Verarbeitung findet datengetrieben statt ausgehen von den Daten wird nach Wissen gesucht

Repräsentation des Wissens in Form von Regeln

Regel- und Produktionssysteme bestehen auf 2 Elementen:

- Arbeitsspeicher
- Regelspeicher

Elemente des Arbeitsspeichers:

Das WM besteht aus einer Menge von Typen sowie deren Instanzen => WME (working memory element)

verschiedene Sichtweisen: WME entspricht:

- den „record“ Deklarationen in prozeduralen Programmiersprachen
- einer frameartige Repräsentation mit Namen und Wertfacette
- einem Tupel einer Datenbankrelation

es gibt 3 Operationen auf WME's:

make, modify, remove

Elemente des Regelspeichers:

Regeln kodieren zumeist Expertenwissen über zusammenhänge in der Form:

wenn Bedingung dann Aktion

LHS RHS

mehrere Bedingungen werden konjunktiv verknüpft.

Regelspeicher Abarbeitung:

- Starten des Systems => es wird eine Regel mit leerer LHS ausgeführt => RHS Teil hat Regeln die erforderliche Initialisierungen durchführen.
- Musterung => Regeln werden gegen das WM gematcht => Resultat ist eine Konfliktmenge die eine Regelinstanz enthält (Datenstruktur)
- Regelauswahl => Regelinstanz wird aus der Konfliktmenge ausgewählt zur Verarbeitung. Dabei gibt es mehrere Merkmale die zur Regelauswahl herangezogen werden:
 - Verhinderung mehrfacher Ausführung gleicher Regeln
 - Auswahl aufgrund zeitlicher Bedingungen
 - Auswahl aufgrund syntaktischer Struktur
 - Bevorzugte Regeln aufgrund von Meta Wissen.
 - Zufällig Auswahl

- Aktion der oben ausgewählten Regel wird ausgeführt => Änderung des WM => Zyklus beginnt von vorne

Durch die Musterung wird das Wissen, das in Form von Regeln kodiert ist, auf Relevanz geprüft. Relevantes Wissen wird in der Konfliktmenge zur heuristischen Bewertung bereitgestellt => Es wird in der Aktionsphase angewendet und es entsteht eine neue Situation
 Ende entweder durch halt (bei Erreichen des Zielzustandes) od. wenn CS nach der Musterung leer ist.

Rückwärtsverkettende Systeme

Unifikation

Mechanismus f. Instanziierung

Forderung an Kalküle (dienen zur Automatisierung und Mechanisierung der Inferenz) ist die zielgerichtete Generierung von Instanzen. Bei der Vorwärtsverkettung wurden Regeln durch Musterung instanziiert. Der Mechanismus bei rückwärtsverketteten Systemen ist hierbei die Unifikation

Substitution

ist eine Abbildung $\theta = V \text{ tend to } T$

Komposition von Substitutionen:

Seien α und β zwei Substitutionen. Die Komposition von α und β ist erklärt durch:

$$x \alpha \beta = \beta \alpha x$$

Unifikator:

Zwei Terme oder Literale E_1 und E_2 sind unifizierbar, wenn eine Substitution θ existiert, sodass $E_1 \theta = E_2 \theta$. Diese Substitution heißt dann Unifikator von E_1 und E_2 . Sie heißt allgemeinsten Unifikator, wenn sie allgemeiner ist als jeder andere Unifikator.

Hornklauseln

Dies sind Klauseln mit höchstens einem Literal.

$$\neg x_1 \dots \neg x_n \vee (p_1 \vee \dots \vee p_n \vee q)$$

Allgemeine Resolution:

Das Resolutionskalkül arbeitet auf einer syntaktischen Unterklasse der Formeln der PL erster Stufe, also den Klauseln. Durch die Verwendung von Klauselformen wird sowohl die Struktur als auch die Anzahl der vorkommenden Konnektive und Quantoren eingeschränkt, was zu einem sehr einfachen Kalkül führt.

Das Resolutionskalkül besteht aus nur 2 Regeln.

Das Resolutionskalkül gibt es sowohl in der AL als auch in der PL.

Lineare Resolution:

Ein Merkmal der linearen Resolution ist die Einschränkung, dass in jedem Resolutionsschritt mindestens eine Elternklausel eine Klausel aus der Eingabeklauselmengen ist.

5. Spezielle Aspekte der Wissensrepräsentation

In bestimmten Situationen wäre es praktisch Annahmen zu treffen die nicht logisch aus einer Wissensbasis folgen.

- Wenn man nicht genug Fakten hat will man trotzdem was draus schließen
- Logik hat Probleme damit weil man in der Logik eigentlich alle Infos schon implizit hat
- Klassische Logik besitzt die monotonie Eigenschaft
- Man benötigt Formalismen die flexibel sind und inkorrekte Schlüsse erlauben
- Herleitung aus Wissensbasis die plausibel aber nicht notwendigerweise logisch aus T folgen.

2 Klassen von Formalismen, die für das Schließen bei unvollständiger Information entwickelt wurden:

- quantitative:
Basieren auf numerischen Verfahren und verwenden Begriffe der Wahrscheinlichkeits-theorie.
Die wichtigsten sind:
 - Bays'sche Netze
 - Überzeugungstheorien
 - Fuzzy Mengen
- qualitative:
Unsicherheit und Unwissen ohne Bezug zu numerischen Methodiken.
Werden als nichtmonotone Logiken bezeichnet.

Qualifikationsproblem

Begriffe sind Änderungen unterworfen

Bsp.: Prädikatenlogik

Aussage: alle Vögel fliegen

Man hat ein Problem weil Pinguine nicht fliegen und tote Vögel, Neugeborene,...

=> Qualifikationsproblem

Man kann aber „temporäre“ Annahmen, die geändert werden können, treffen

=> Default Annahme

Frame Problem

Welche Eigenschaften ändern sich beim Ausführen einer Aktion? Welche bleiben gleich?

Typeischerweise beeinflusst eine Aktion nur einen geringen Teil der Eigenschaften

Bsp: Würfel der bemalt wird bleibt ein Würfel ... 4 Ecken, usw.....

Um diese Invarianzeigenschaften darzustellen benötigt man eine große Anzahl an speziellen Zusatzaxiomen, sog. Frame Axiome. (ineffizient)

=> Lösung: Frame-Axiome durch nicht monotone Regeln ersetzen

Ramifikationsproblem

beschreibt implizite Folgerungen aus einer durchgeführten Aktion.

man hebt einen Stein und hebt damit auch Staub => man will aber den Staub nicht extra spezifizieren sonder aus der Bewegung des Steins herleiten

=> Lösung: nicht monotone Regeln

Minimierungsmethoden

Closed World Assumption

ist eine Methode um negative Fakten effizienter zu repräsentieren. (da die Anzahl negativer Fakten meist weit größer ist als die der positiven)

=> jeder atomare Satz, der aus einer gegebenen Menge von Fakten nicht hergeleitet werden kann ist falsch

Bsp: Zugfahrpläne: alle Züge die nicht angezeigt werden gibt es nicht (dh Verbindung existiert nicht)

T... Theorie

P... geschl. Atomformel

T ist vollständig wenn P od. $\leftarrow P$ in T liegt

CWA(T) ... vollständige Theorie wenn f. jede P, die nicht in CWA(T) liegt

gilt: $\leftarrow P \sum CWA(T)$

Prädikatenvervollständigung

Oft kann man mit Hilfe einer einzigen logischen Formel zeigen, dass jene Objekte von denen man zeigen kann, dass sie ein gewisses Prädikat erfüllen, die einzigen Objekte sind die dieses Prädikat erfüllen.

T... besteht aus einer Formel $p(a)$

p ist logisch äquivalent $\exists x \exists x = a \text{ tendtot } p(x)$

Annahme dass keine weiteren Objekte p erfüllen – Umkehrung d. Formel

$\exists x \exists p(x) \text{ tendtot } x = a$

Diese Formel nennt man Vervollständigungsformel. In diesem Fall liefert die Prädikatenvervollständigung dasselbe wie die CWA.

Semantisch gesehen führt Prädikatenvervollständigung von P zur Interpretation von P dass es in einem Modell minimiert wird.

Solitär:

Menge v. Klauseln ist solitär in P falls jede Klausel mit einem positiven Vorkommenis von P höchstens ein Vorkommenis von P hat.

Obwohl in den einfachsten Fällen die CWA und Prädikatenvervollständigung dasselbe Ergebnis haben, so sind sie im allgemeinen doch verschieden.

Default Logik

nichtmonotone Schlüsse mittels spez. Inferenzregeln, den sog. Defaults, realisiert.

$(\phi: v_1 \dots v_n) / \gamma$

ϕ ... Vorbedingung

$v_1 \dots v_n$... Rechtfertigung

zusammen: Prädikatenlogische Formel

γ ... Konsequenz

wenn γ schon bewiesen wurde und $v_1 \dots v_n$ konsistent zum bisherigen Wissen sind dann kann man auf γ schließen

Bsp.: $\text{Vogel}(x):\text{fliegt}(x) / \text{fliegt}(x)$

Default-Theorie

Wissen wird in Form von Default Theorien repräsentiert

geordnete Paare $\langle W, D \rangle$

W... Menge von geschlossenen Formeln, Hintergrundwissen

D... Menge von Defaults

6. Unsicherheit:

Elemente der Wahrscheinlichkeitstheorie:

Grundbegriffe:

Man hat Zufallsvariablen V_i die aus verschd. Domänen kommen kann:

- Boolesche Werte: $V_i = T$, oder F
- Numerische Werte, V_i stammt aus physikalischen Messungen
- Kategorische Werte = $V_i = \text{Kategorien}$ (Farbe, Buchstabe, Studienrichtung, ...)

Bedingte (a posteriori) Wahrscheinlichkeit

Man sucht Informationen über die Wahrscheinlichkeit gewisser Größen unter der Voraussetzung, dass andere Größen bestimmte Werte annehmen. Dies ist die bedingte Wahrscheinlichkeit.

Berechnungen unter der bedingten Wahrscheinlichkeit fasst man unter dem Begriff „probabilistisches Schließen“ zusammen.

Kettenregel = nützlich um bedingte Wahrscheinlichkeiten zu berechnen. Sie erlaubt es, die Wahrscheinlichkeitsverteilung nach bedingten Wahrscheinlichkeiten hin zu entwickeln.

Bedingte Unabhängigkeit:

Bedingte Unabhängigkeit $I(V, V_i | E = e)$ bedeutet, dass es bei zusätzlicher Information V_i bei gegebener Evidenz $E = e$ kein zusätzliches Wissen in bezug auf V gibt.

Bayesche Netze:

Dies ist ein gerichteter, azyklischer Graph (DAG) dessen Knoten mit Zufallsvariablen markiert sind und der in graphischer Weise gewisse Unabhängigkeitseigenschaften von Variablen repräsentiert.

- Zwei Knoten V_1, V_2 sind mit einem Pfeil verbunden, wenn V_1 einen direkten Einfluss auf V_2 hat.
- Zu jedem Knoten V ist die bedingte Wahrscheinlichkeit $P(V|V_x)$ bekannt, wo V_x die unmittelbaren Vorgänger von V sind.

Vor und Nachteile:

- große Datenmengen benötigt um die marginalen und bedingten Wahrscheinlichkeiten angeben zu können
- Schwer zum sagen ob Variablen bedingt unabhängig sind
- kann kein Nichtwissen darstellen – nicht monotone Logiken schon
- Widersprüche werden nicht entdeckt und pflanzen sich fort

Arten des Schließens in Bayes'schen Netzen

Kausales Schließen

man interessiert sich für die Berechnung von Wahrscheinlichkeiten von Ursachen und Wirkungen

Diagnostisches Schließen

Umkehrung des kausalen Schließens. Dh. Wahrscheinlichkeit von Effekten zu Ursachen.

Interkausales Schließen

Mischform aus den zwei vorigen Schlussarten. Dh. Schlüsse zwischen Ursachen eines gemeinsamen Effektes.

D-Selektion:

Probabilistisches Schließen in einfachen Verbund Netzen:

Ein einfach verbundenes Netz ist ein DAG, in dem es nur genau einen ungerichteten Pfad zwischen 2 Knoten gibt.

Weitere Formalismen quantitativen Schließens:

Certainty Factors:

Ein Ansatz um unsicheres Schließen mittels quantitativen Methoden in regelbasierten Systemen zu ermöglichen.

Dempster-Shafer Theorie:

Diese Theorie umgeht das Problem, dass bei der Anwendung der Wahrscheinlichkeitstheorie zur Modellierung unsicheren Wissens, alle relevante marginale und bedingte Wahrscheinlichkeiten bekannt sein müssen. Dabei wird mit einem unvollständigen probabilistischem Modell gearbeitet, das keine definiten Wahrscheinlichkeitswerte berechnet, sondern Intervalle angibt, wie nahe die angegebene Evidenz zur Bestimmung eines Wahrheitswertes ist

Es handelt sich also um Methoden, um mit dem Unterschied von Unsicherheit zu Unwissen umzugehen.

Fuzzy Logik

Formalismus, der nicht Unsicherheit, sondern Vagheit beschreibt.

7. Konfiguration

Unter Konfiguration versteht man die Aufgabe, aus einer fixen Menge von Komponenten gewisse Elemente so auszuwählen und anzuordnen, dass sie eine gegebene Spezifikation erfüllen. (zb Küchenplaner)

Systeme:

- XCON
- M1
- VT
- Cossack
- Cocos

Konfiguration / Planen / Sheduling:

- Konfiguration:
Zeit spielt eine nominelle Rolle, da nur garantiert sein muss, dass es einen gewissen Zeitpunkt geben muss, zu dem das Resultat die gewünschten Erfordernisse erfüllen soll.
- Planen:
Zeit spielt eine ordinale Rolle, da es dabei nur auf die richtige Reihenfolge der Aktionen, jedoch nicht auf die Dauer ankommt.
- Sheduling:
Dauer der Abfolge von Aktionen ist von bedeutung. Zeit wird in Form von Intervallen berücksichtigt.

Bestandteile eines Konfigurationsproblems:

1. fixe, wohldefinierte Menge von Komponenten, beschrieben durch:
 - * Menge von Eigenschaften
 - * Menge von Verbindungsstellen (Ports)
 - * Randbedingungen, beschreiben welche Komponenten an die jeweiligen Ports angeschlossen werden können
2. Beschreibung der gewünschten Konfiguration
3. optionale Kriterien zur Optimierung der Lösung.

Constraint Satisfaction Probleme (CSP):

hat folgende Bestimmungsstücke:

- endliche Menge von Variablen
- jede Variable hat einen Wertebereich
- eine Menge von Constraints. Ein Constraint legt gültige Werte für die Variablen fest, auf denen das Constraint definiert ist und beschreibt eine Teilmenge des kartesischen Produktes der Wertebereiche.

Ein Constraint kann explizit durch die Menge aller gültigen Wertebelegungen angegeben werden

Ein Constraint kann implizit durch eine Formel oder eine boolsche Funktion angegeben werden, die dann true liefert, wenn der Wert zulässig ist.

Diagnose

Soll fehlerhafte Komponenten in einem System erkennen. Dies schließt sowohl technische Fehlersuche, als auch medizinische Systeme ein, die erkennen an welcher Krankheit ein Patient leidet.

2 Ansätze:

- Regelbasierte: schlecht zu warten (Erweiterbarkeit, Korrektheit, Vollständigkeit)
- Modellbasierte: Vorteil zum heuristischen Verfahren: Unabhängigkeit des Diagnosesystems von spezifischen Bauteilen und daher leicht Erweiterbar.

Komponenten in einem Diagnosesystem

- Diagnosemodell: korrektes Verhalten wird formal beschrieben
- Diagnosealgorithmus: Es wird das tatsächliche Verhalten mit dem vorhergesagtem verglichen. Menge von fehlerhaft erkannten Bauteilen wird als Diagnose bezeichnet
- Messpunktselektion: möglichst wenig Messpunkte um eine Diagnose zu bekommen. Optimale Messpunkte durch Heuristiken/Kostenfunktionen.

Diagnoseproblem (SD, OBS, COMP) besteht aus:

- logische Theorie – Systembeschreibung SD
- Menge von Fakten – Beobachtungen OBS
- Menge von Konstantensymbolen – Bauteile COMP

Diagnose für Diagnoseproblem ist eine min. Menge von fehlerhaften Komponenten

Aufgabe eines Diagnosesystems:

- Diagnosen müssen berechnet und ausgegeben werden (alle od. wahrscheinl.)
- Messpunkte vorschlagen um best möglich zw. Diagnose differenzieren zu können. Ziel: Mit möglichst wenig Messungen auf eine Diagnose kommen.

Klassifikation:

Dabei handelt es sich um die Aufgabe, Objekte, Phänomene, Muster, Messungen ... als Teil einer bekannten Klasse einzuordnen. Klassifikationsprobleme im Allgemeinen gehen von gewissen Daten aus und identifizieren Klassen als Lösungen.

8 Planen:

Eine wichtige Aufgabe, die intelligente Systeme oftmals bewältigen sollten, ist Planen. Ein Planungsproblem besteht darin, ein gewünschtes Ziel durch Ausführung von Aktionen zu erreichen. Jede dieser Aktionen kann durch einen Stimulus aktiviert werden und tätigt einen Effekt auf die Welt des Agenten, der die Aktion ausführt.

Die Fähigkeit Planungsprobleme ad hoc zu erledigen und nicht durch in der Wissensbasis aufgelegte Pläne zu erschlagen ist erforderlich weil:

- zu viele Pläne gespeichert werden müssten
- ev. die Pläne nicht alle Aspekte berücksichtigen
- ev. Pläne für bestimmte Szenarien vergessen werden

Erforderlich sind:

- ein Modell der Welt
- ein Modell für Aktionen und deren Effekten

Suchraummodelle:

Planen involviert einen Suchraum von möglichen Zuständen der Welt. Es gibt dafür verschiedene Ansätze:

Zustandsraum Graph:

In diesem einfachen Ansatz wird der Suchraum als Suchgraph dargestellt. Die möglichen Zustände sind Knoten, die Aktionen sind markierungen von Kanten.

Voraussetzungen:

- alle möglichen Zustände müssen repräsentiert werden
- es muß eine exakte Modellierung der Aktionen zwischen Zuständen erfolgen
- die Aktionen müssen die angegebenen Effekte haben
- es wird angenommen, dass es keine Äusseren Einflüsse gibt und dass es keine Änderungen im system während der Planungsphase gibt

Feature-basierter Suchzustand:

hier ist ein Suchzustand kein vollständiges Szenario, in dem alle Einzelheiten bekannt sind, sondern er wird durch Teilaspekte eines vollständigen Szenarios (Features) beschrieben. Informell repräsentiert ein Suchzustand hier eine nichtleere Menge von vollständigen Szenarien, die mit diesem Zustand komptaibel sind.

Situationskalkül:

Dient zur repräsentation von Zuständen und zeitlichen Abfolge von Aktionen in einer Welt. Es war ein früher Ansatz für ein logik-basiertes Planungssystem.

Die Idee ist, PL 1.Stufe zur beschreibung des Systems heranzuziehen und durch geschickte Verwendung des Sprachvokabulars über logische Formeln den Sachverhalt adäquat zu repräsentieren.

Ansatz:

- Formalisiere Zustände, Aktionen, Effekte von Aktionen durch OBJEKTE
- Repräsentiere Wissen über Zustände, Zustandsübergänge, ... durch Prädikate und Formeln
- Verwende ein Deduktionssystem

Elemente des Situationskalküls:

- Zustände:
Zustände eines Systems werden als eigene Objekte betrachtet und durch Objektkonstanten repräsentiert.
- Fluents:
sind Prädikate, die Aussagen über Zustände machen
- Aktionen:
werden als Funktionen modelliert, die auf Objekten ausgeführt werden.
- „do“ Funktion:
Spezielle Funktion... weist einem Paar (a,S) aus einer Aktion c und einem Zustand S einen Zustand S' zu: $do(a,S) = S'$

Planen im Situationskalkül:

1. Formalisier ein „Goal“ $g(s)$ wobei s den Zielzustand beschreibt
2. Beweise das ein s Existiert aus $g(s)$ aus der Systembeschreibung, die den Anfangszustand, die Fluents die Effekt und Frameaxiome beinhaltet
3. Extrahiere einen Plan P aus der Antwort Substitution.

Probleme des Ansatzes:

- Mangelnd Effizienz:
Beweisauwand ist sehr groß. Grund: komplexe Systembeschreibungen
- Representational Frame Problem:
Zahl der Frameaxiome bläht Systembeschreibung auf. Alle Fluents vom Momentanen Zustans müssen in den Nachfolgezustand übernommen werden, auch wenn sie sich nicht ändern.

Strips:

Einer der wesentlichen Nachteile des Situationskalküls ist die aufwendige Eliminierung des Frame Problems, das durch inhärente Eigenschaften der verwendeten klassischen PL entsteht.. Der STRIPS Ansatz vermeidet dieses Problem, indem zwar logische Formeln zur Beschreibung von Planungsszenarien verwendet werden, die Effekte der Ausführung von Aktionen aber durch operationale Definition un nicht einfach über Semantik der klassischen Logik festgelegt wird.

Elemente von Strips:

- Zustandsbeschreibung
Besteht aus einer Menge von Variablenfreien Literalen der PL (Bsp: $S = \{on(B,A), on(A,C), on(C,F), clear(B), clear(F)\}$)
- Aktion
Aktionen führen einen Zustand in einen andern über.
- Operator
ein Operator hat drei komponenten:
 - Precondition:
Eine Menge $PC(op)$ von variablenfreien Literalen, die als Vorbedingung bezeichnet wird.
 - Add List:
Eine Menge $Add(op)$ von variablenfreien Literalen, die als Add List bezeichnet wird .
 - Delete List
Eine Menge $Del(op)$ von variablenfreien Literalen, die als Delete List bezeichnet wird .

Planungsmethoden:

- Progressives Planen
- Progressives Strips Planen mit rekursiver Teilplanung
- Sussman Anomalie
- Regressives Planen

Progressives Planen:

Planen durch Suche nach einer Vorwärtsstrategie. Dabei wird ausgehend von S, ein Plan P schrittweise über die Suche einer state description S' im kenzeptuellen Suchgraph konstruiert, die das „Goal“ erfüllt.

Bei einer großen Zahl von STRIPS Regeln und großen Zustandsbeschreibungen ist dieser naive Ansatz allerdings nicht praktikabel. Grund: Mangelnde Effizienz durch zu große Verzweigungen der Möglichkeiten Operatoren anzuwenden. Eine fokussierte Suche ist heri erforderlich.

Progressives Strips Planen mit rekursiver Teilplanung:

Eine Möglichkeit die Suche zu fokussieren, ist das „Goal“ in Teile zu zerlegen und ausgehend von einer Lösung für einen Teil eine Gesamtlösung zu konstruieren. Am einfachsten ist hierbei, zunächst einen Plan P1 für das Subgoal L1 zu finden. Ist ein solcher Plan kosntruiert, so versucht man, durch weitere Aktionen das nächste Subgoal zu erfüllen, bis alle Subgoals erfüllt sind.

Der Algorithmus terminiert aber nicht immer. Es kann daher zu Zyklen kommen. Will man dies vermeiden, muß eine entsprechende Erkennung zyklischer Situationen vorgesehen werden, was sehr aufwendig ist.

Sussman Anomalie

Oben beschriebener Algorithmus terminiert nicht immer. Im Falle der Terminierung ist Minimalität des Plans in dem Sinn nicht sichergestellt, dass er keine überflüssigen Operatoren enthält.

Bei Rückwertssuche lässt sich dieses Problem vermeiden.

Regressives Planen

Ausgehend vom „goal“ wird in Richtung der anfänglichen state description S ein Plan P in umgekehrter Reihenfolge produziert. Dh: es werden die Effekte von Operatoren der Reihe nach rückgängig gemacht, bis die Erfüllbarkeit des veränderten Goals in S gegeben ist.

Wann soll man regressiv Planen:

- Prinzipiell ist der regressive Ansatz zur Lösung praktischer Probleme besser geeignet, da das Goal in der Regel klein ist und für die Erfüllung eines Subgoals wenige Operatoren vorhanden sind
- Der progressive Ansatz ist im Fall mit Variablen wesentlich einfacher. Die Behandlung von Variablen, insbesondere die Beschränkung der zulässigen Instanzierungen ist im Falle von regressiver Planung kompliziert und aufwendig.

Partial Order Planning:

Die bisherigen Planungsmethoden basieren darauf, Schritt für Schritt einen Plan durch Suche in einem Raum verschiedener Zustände bzw state descriptions zu finden. Ein anderer Ansatz ist, einen Plan in einem Raum von Plänen zu finden. Dabei wird ein unvollständiger Plan, der zu Beginn nur aus einem Rahmen besteht, durch Änderungen wie Einfügen, Entfernen oder Umordnen von Aktionen so lange modifiziert, bis ein vollständiger Plan entsteht, der das Gewünschte leistet.

Hierarisches Planen:

Eine wesentliche Technik bei der Planung komplexer Abläufe ist, logisch zusammengehörende Folgen von elementaren Aktionen zu abstrakten Aktionen zusammenzufassen, um die Planung übersichtlicher zu gestalten. Umgekehrt kann es erforderlich sein, Aktionen eines Planes in konkrete Folgen von nichtelementaren Aktionen umzusetzen, die unmittelbar ausgeführt werden können. Dies führt zum Ansatz des hierarchischen Planens, in dem abstrakte Planungsoperatoren durch hierarchische Zerlegung mit einem Plan, bestehend aus konkreten Operatoren ersetzt werden. Dieses Prinzip wird dabei mehrstufig fortgesetzt.

Erweitern eines Strips basierenden Planungsansatzes auf hierarchisches Planen:

- Elementare und nichtelementare Operatoren. Ein Plan heißt abstrakt, wenn er nichtelementare Operatoren enthält, andernfalls elementar.
- Dekomposition von nichtelementaren Operatoren, dh Realisierung durch einen Plan aus anderen Operatoren.