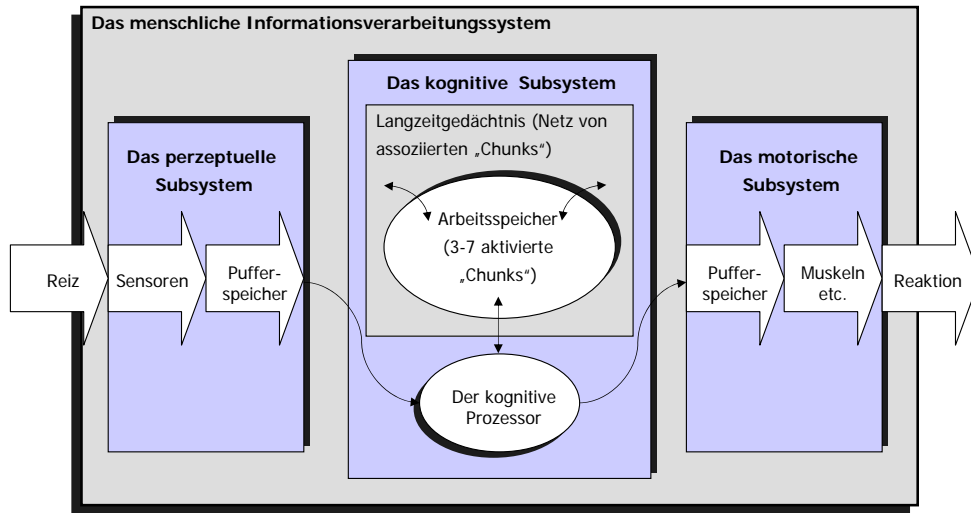


4. Wissensbasierte Anwendungssysteme

- 4.1 Einführung in wissensbasierte Systeme
- 4.2 Wissensrepräsentation
- 4.3 Exkurs: Die Programmiersprache Prolog
- 4.4 Lösungssuche und Inferenz
- 4.5 Fallstudie: Realisierung eines wissensbasierten AwS

Menschliches Problemlösungsverhalten

Die Umwelt



WS 2008/09

Prof. Dr. E. J. Sinz, Universität Bamberg

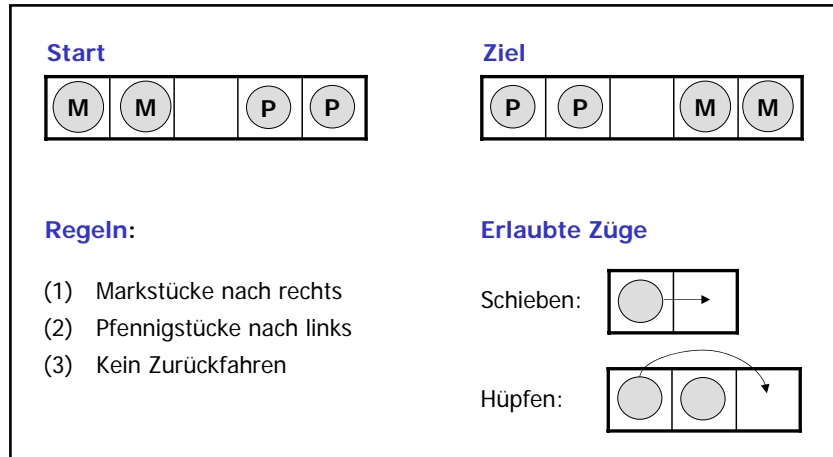
Seite 53

Ein einfaches Modell der Informationsverarbeitung des Menschen

Komponenten:

- **Perzeptuelles System** (Wahrnehmungssystem) mit Kurzzeitgedächtnis als Pufferspeicher.
- **Kognitives System** (Erkennungssystem) mit Langzeitgedächtnis, bestehend aus einer sehr großen Menge von gespeicherten Symbolen. Diese sind untereinander verknüpft und in Form von *Scripts* oder in Form von *Chunks* organisiert.
- **Motorisches System** (Bewegungssystem).

Lösungssuche und Suchstrategien

**Beispiel:**

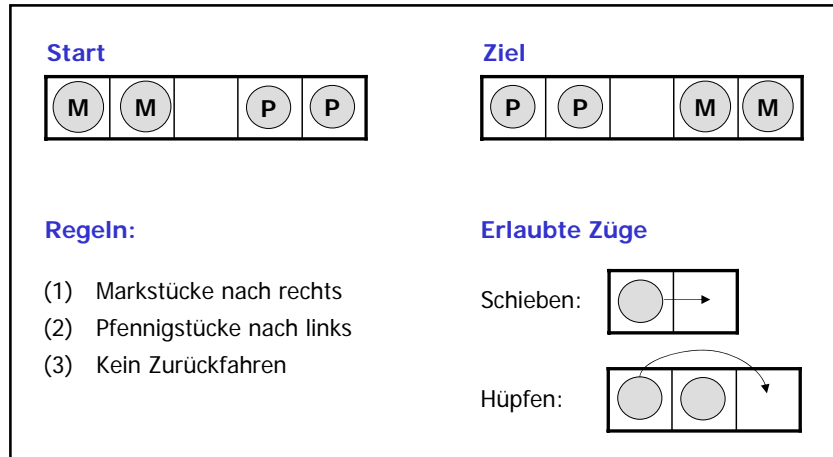
Mark-Pfennig-Problem mit Start- und Zielzustand sowie Zustandsübergängen, die bestimmten Regeln genügen müssen.

Vorgehensweise:

Modellierung des **Untersuchungsobjekts** gemäß Zustandsraummodell:

- Festlegung einer geeigneten Menge von Symbolen zur Beschreibung des Untersuchungsobjekts.
- Beschreibung der möglichen Zustände des Untersuchungsobjekts durch Muster von Symbolen.
- Beschreibung Übergänge oder Operationen des Untersuchungsobjekts durch Beziehungen zwischen Zuständen.

Lösungssuche und Suchstrategien

**Untersuchungsproblem = Untersuchungsobjekt + Untersuchungsziel**

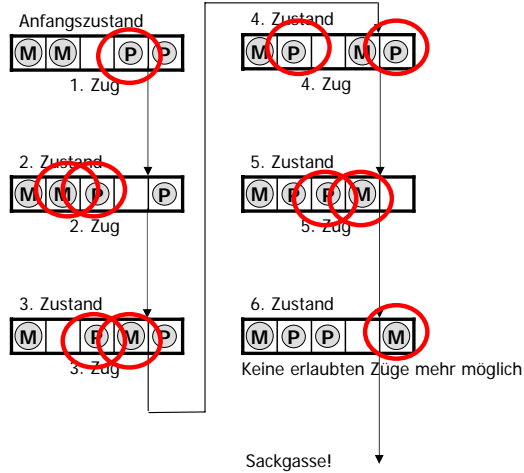
- Das Untersuchungsziel besteht im Erreichen von Zielzuständen ausgehend von vorliegenden Startzuständen.
- Zustände und Übergänge beschreiben zusammen den Problemraum des Untersuchungsproblems.

Untersuchungssituation = Untersuchungsproblem + Lösungsverfahren

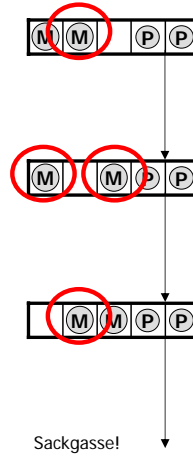
- Das Lösungsverfahren besteht in einer bestimmten Suchstrategie, welche auf den Problemraum angewandt wird.
- Eine Problemlösung ist eine Folge von Übergängen / Operationen, die ausgehend von einem Startzustand zu einem Zielzustand führt.

Entwickeln des Problemraumes

1. Versuch



2. Versuch



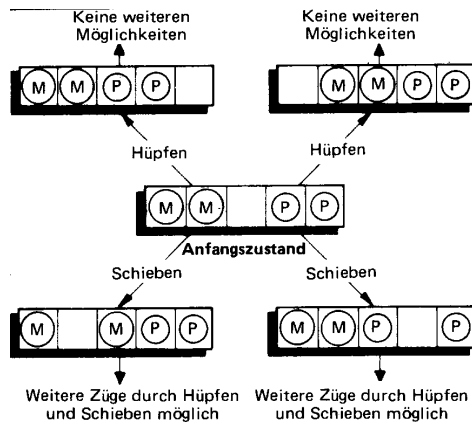
Das Entwickeln des Problemraums erfolgt durch *Generieren-und-Testen* (*generate and test*). Ausgehend von einem Startzustand werden zulässige Übergänge/Operationen durchgeführt bis

- ein Zielzustand erreicht ist,
- kein weiterer Übergang / keine weitere Operation möglich ist (Sackgasse) oder
- ein bereits untersuchter Zustand erneut erreicht wird (Zyklus).

Beispiel: 2 willkürlich herausgegriffene Versuche des Generieren-und-Testen im Mark-Pfenning-Problem

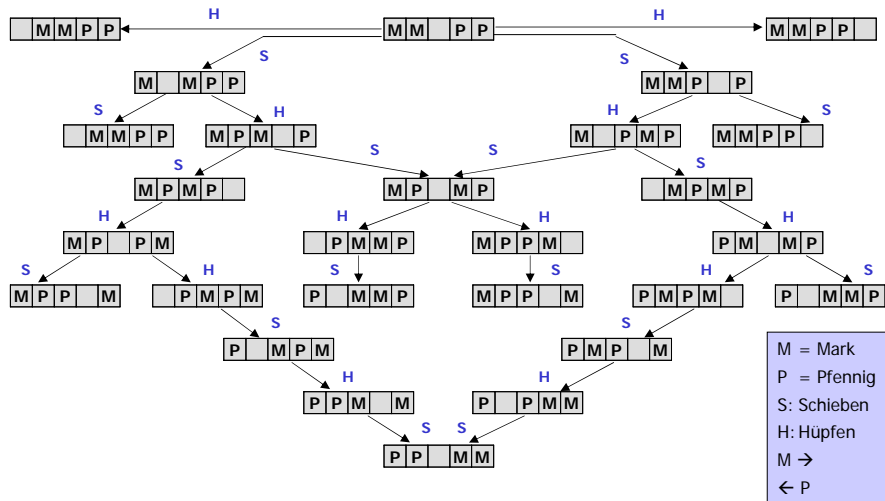
Entwickeln des Problemraumes

Beispiel: Unmittelbare Folgezustände des Startzustands im Mark-Pfenning-Problem

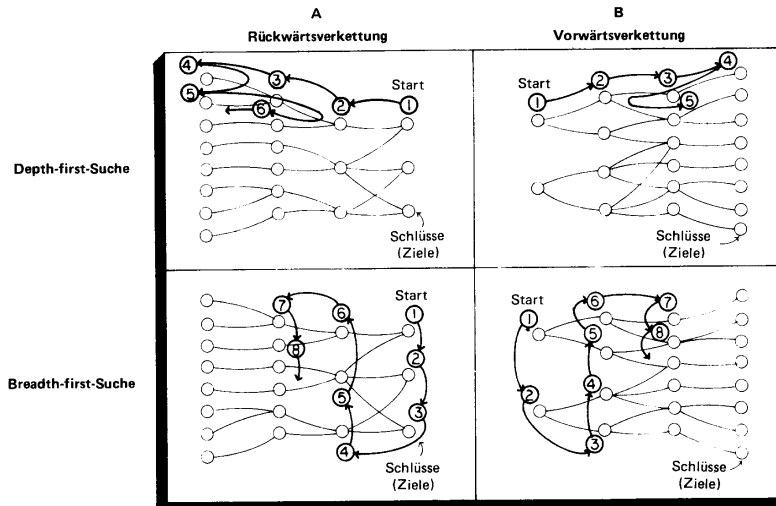


Der vollständige Problemraum ist dadurch gekennzeichnet, dass er zu jedem Zustand alle zugehörigen, durch Übergänge / Operationen erreichbaren Folgezustände enthält.

Entwickeln des Problemraumes

Beispiel: Vollständiger Problemraum im Mark-Pfennig-Problem

Grundlegende Strategien zur Lösungssuche



WS 2008/09

Prof. Dr. E. J. Sinz, Universität Bamberg

Seite 59

Die Entwicklung des Problemraums wird simultan mit der Lösungssuche durchgeführt. Dabei sind grundsätzlich zwei unterschiedliche Strategien anwendbar:

- **Breitensuche (breadth first):** Zunächst werden alle Startzustände getestet. Anschließend werden der Reihe nach alle unmittelbaren Folgezustände dieser Zustände generiert und getestet. usw.
- **Tiefensuche (depth first):** Der erste Startzustand wird getestet.. Anschließend wird der erste Folgezustand dieses Zustands generiert und getestet. usw.

Hinweise:

- Breitensuche entspricht der Strategie von „Generalisten“, Tiefensuche der Strategie von „Spezialisten“.
- Die meisten existierenden Systeme wenden (standardmäßig) Tiefensuche an.
- Die Wahl der Suchstrategie wird u.a. von der (erwarteten) Struktur des Problemraums beeinflusst und davon, ob eine oder alle Lösungen zu ermitteln sind.

Grundlegende Strategien zur Lösungssuche

Algorithmus zur Breitensuche:

```
Bilde eine einelementige Liste, die aus dem Wurzelknoten besteht;  
REPEAT  
  IF Erstes Element in der Liste ist der Zielknoten THEN  
    RETURN true  
  ELSE  
    Entferne das erste Element der Liste;  
    Setze die Nachfolger des ersten Elements an das Ende der Liste;  
  END;  
UNTIL Liste leer;  
RETURN false;
```

Grundlegende Strategien zur Lösungssuche

Algorithmus zur Tiefensuche:

```
Bilde eine einelementige Liste, die aus dem Wurzelknoten besteht;  
REPEAT  
  IF Erstes Element in der Liste ist der Zielknoten THEN  
    RETURN true  
  ELSE  
    Entferne das erste Element der Liste;  
    Setze die Nachfolger des ersten Elements an das Spitze der Liste;  
  END;  
UNTIL Liste leer;  
RETURN false;
```

Grundlegende Strategien zur Lösungssuche

Verbesserungen der Breiten- und Tiefensuche

- Bergsteigen (Hillclimbing)

Bergsteigen ist eine Verbesserung der [Tiefensuche](#). Das Suchverfahren ist anwendbar, falls für die einzelnen Elemente Schätzungen über die Restentfernung bis zum Ziel vorliegen.

...

Entferne das erste Element der Liste;

Ordne die Nachfolger des ersten Elements nach der geschätzten Restentfernung und setze sie an die *Spitze* der Liste

...

Grundlegende Strategien zur Lösungssuche

Verbesserungen der Breiten- und Tiefensuche

- Strahlsuche

Strahlsuche ist eine Verbesserung der **Breitensuche**, bei der nur die besten w Elemente in jeder Ebene weiter durchsucht werden. Die anderen Elemente werden ignoriert.

- Bestensuche

Bestensuche expandiert zu jedem Zeitpunkt den besten Teilpfad.

...

Entferne das erste Element der Liste;

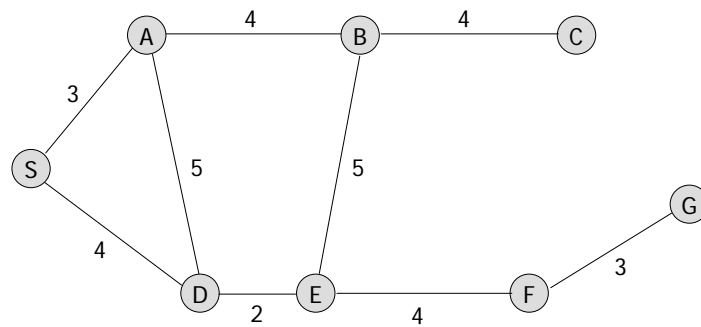
Füge die Nachfolger des ersten Elements in die Liste ein;

Sortiere die gesamte Liste nach der geschätzten Restentfernung;

...

Grundlegende Strategien zur Lösungssuche

Durchführung von Suchstrategien:

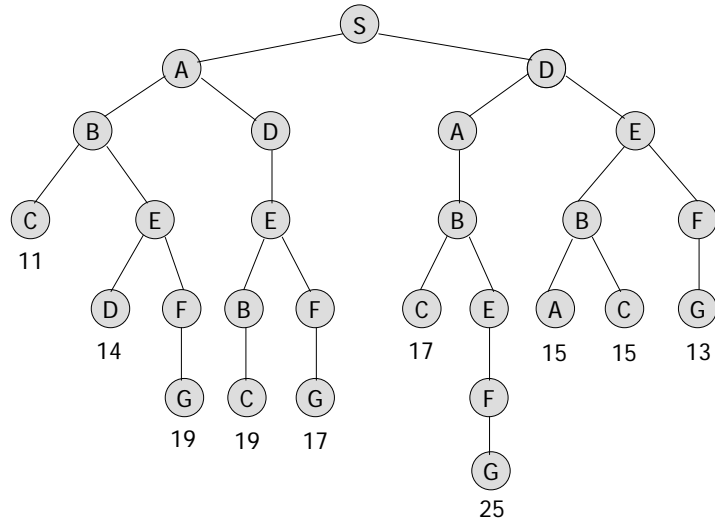
**Beispiel:**

In einem Straßennetz sei der kürzeste Weg von Startknoten S zu Zielknoten G zu finden.

Ein grundlegendes Suchproblem. Vom Startknoten S bis zum Zielknoten G ist ein Pfad zu finden. Suchprozeduren erkunden Netze wie diese und erhalten beim Vordringen Informationen über Verbindungen und Entfernungen.

Grundlegende Strategien zur Lösungssuche

Entwicklung des Problemraumes in Form eines Suchbaumes:



WS 2008/09

Prof. Dr. E. J. Sinz, Universität Bamberg

Seite 65

Aus einem Netz entsteht ein Baum, indem man ausgehend von einem Startknoten (= Wurzelknoten) alle Wege verfolgt, bis man einen bereits besuchten Knoten erreicht.

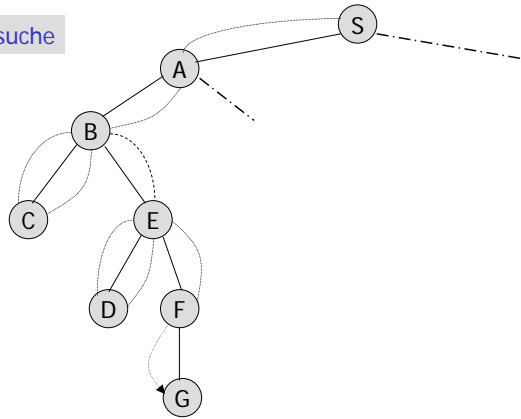
Im Bild sind alle Endknoten des Baumes mit den kumulierten Entfernungen bewertet.

Anmerkung: Der Suchbaum wird normalerweise während der Durchführung des Suchverfahrens sukzessive entwickelt.

Grundlegende Strategien zur Lösungssuche

Anwendung der Suchstrategien:

Tiefensuche

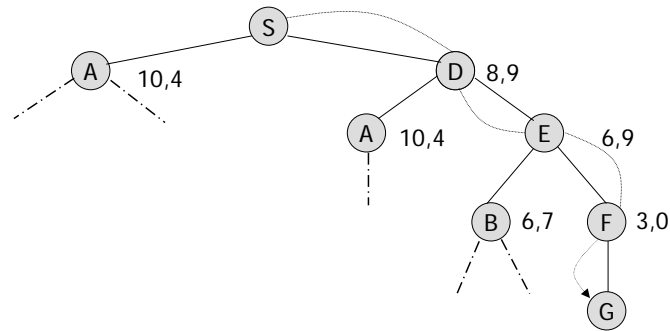


In Abhängigkeit von der (angenommenen) Struktur des Problemraums kann es bei der Implementierung der Tiefensuche notwendig werden, die maximale Suchtiefe zu begrenzen. Bei Erreichen dieser maximalen Suchtiefe wird die Suche abgebrochen und an der nächsten, noch nicht untersuchten Verzweigung fortgesetzt.

Grundlegende Strategien zur Lösungssuche

Anwendung der Suchstrategien:

Bergsteigen (Hillclimbing)



Die Bewertung der Knoten erfolgt mit den „Luftlinien-Entfernungen“ bis zum Zielknoten

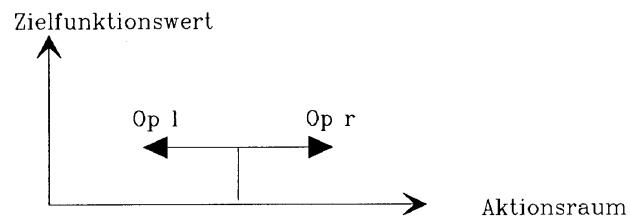
- Ein Beispiel für Bergsteigen. Bergsteigen ist Tiefensuche heuristischer Messung, die die Entscheidungen bei der Expansion eines Knotens ordnet. Die Zahlen neben den Knoten sind Luftlinien-Entfernungen zu den Zielknoten.

Beim Hillclimbing wird versucht, durch *lokale* Bewertungen der einzelnen Knoten das *globale* Ziel effizienter zu finden.

Grundlegende Strategien zur Lösungssuche

Anwendung der Suchstrategien:

Bergsteigen (Hillclimbing)



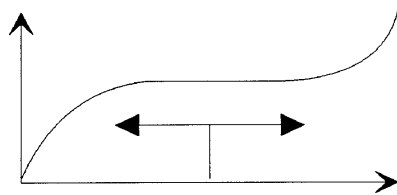
Beispiel: Suche des Maximums einer Funktion. Als Operatoren stehen Links- und Rechtsbewegungen mit einer bestimmten Schrittweite zur Verfügung.

Grundlegende Strategien zur Lösungssuche

Anwendung der Suchstrategien:

Bergsteigen (Hillclimbing)

- a) **Plateauproblem:** In keiner Richtung erhöht sich der Zielfunktionswert.



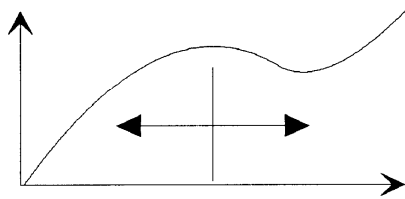
Lösung :
Schrittweite gleitend erhöhen

Grundlegende Strategien zur Lösungssuche

Anwendung der Suchstrategien:

Bergsteigen (Hillclimbing)

b) **Lokale Maxima:** Der Suchalgorithmus „fängt sich“ in einem lokalen Maximum der Funktion.



nur, falls kein natürlicher Grenzwert der Bewertungsfunktion existiert.

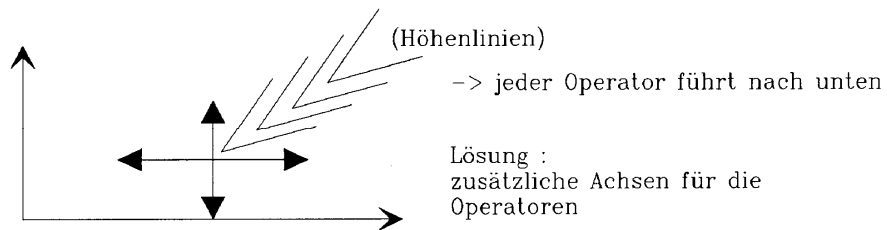
Lösung :
Alternative Startpunkte wählen

Grundlegende Strategien zur Lösungssuche

Anwendung der Suchstrategien:

Bergsteigen (Hillclimbing)

c) **Schräger Grat:** Sowohl in X- als auch in Y-Richtung nimmt der Wert der Zielfunktion ab. In X°Y-Richtung nimmt er aber zu.

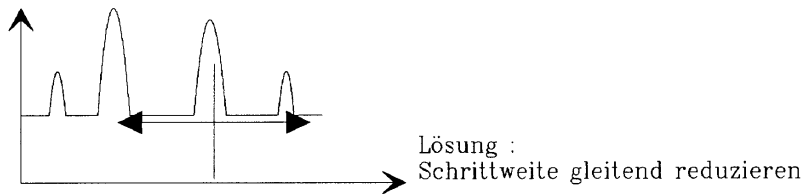


Grundlegende Strategien zur Lösungssuche

Anwendung der Suchstrategien:

Bergsteigen (Hillclimbing)

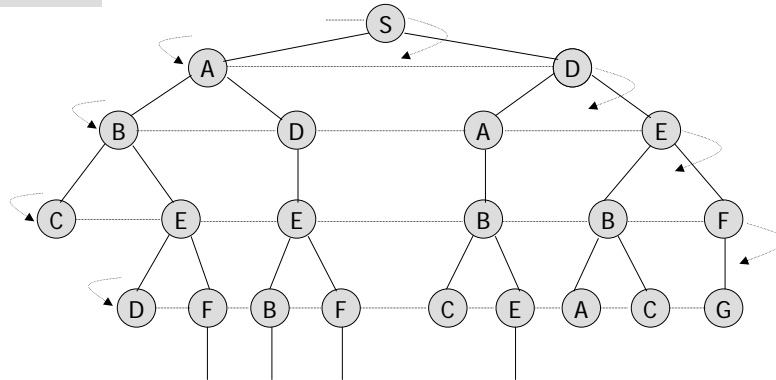
d) **Nadelkissenproblem:** Die Schrittweite ist zu grob, um die Maxima zu erreichen.



Grundlegende Strategien zur Lösungssuche

Anwendung der Suchstrategien:

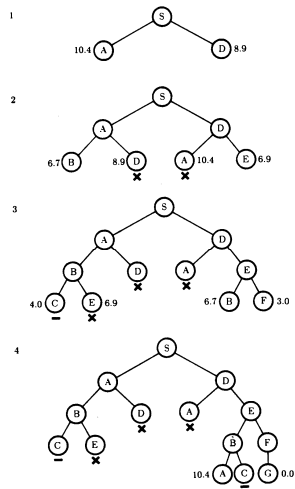
Breitensuche



Grundlegende Strategien zur Lösungssuche

Anwendung der Suchstrategien:

Strahlsuche



Im Gegensatz zur Breitensuche werden nur die besten w Knoten einer Ebene weiter untersucht ($w = 2$; „X“ bedeutet ausgeschieden; „-“ bedeutet Suchbaum zu Ende).

Grundlegende Strategien zur Lösungssuche

Anwendung der Suchstrategien:

Bestensuche

Expandiere den zum jeweiligen Zeitpunkt besten Knoten.

Ableitung von Wissen durch Schließen

Vorwärts- und Rückwärtsverkettung

Produktionsregeln:

Produktionsregeln (siehe relationsorientierte Wissensrepräsentation) sind als Aktionen interpretierbar, die mit einer Vorbedingung versehen sind. Die Aktion ist ausführbar, wenn die Vorbedingung erfüllt ist.

Durch die Aktion wird insbesondere die Faktenbasis verändert, d.h. der Faktenbasis wird ein neuer Fakt hinzugefügt.

Eine Menge von Produktionsregeln, eine zugehörige Faktenbasis sowie ein zugehöriger Interpretationsprozess werden zusammen als **Produktionssystem** bezeichnet.

Betriebsarten für Interpreter:

Interpreter für Produktionssysteme lassen sich im Modus der **Vorwärtsverkettung** oder im Modus der **Rückwärtsverkettung** betreiben.

Ableitung von Wissen durch Schließen

Vorwärts- und Rückwärtsverkettung

Betriebsarten für Interpreter:

Vorwärtsverkettung:

(forward chaining, forward reasoning, synthetisches Schließen, datengetriebene Schlussfolgerung)

Ausgehend von vorliegenden Fakten (Startknoten) wird im Problemraum nach Goals (Zielknoten) gesucht. Dazu wird zyklisch nachstehende Schrittfolge ausgeführt:

1. Feststellen, welche Produktionsregeln ausführbar sind (Konfliktmenge).
2. Auswählen einer ausführbaren Produktionsregel (Konfliktauflösung).
3. Ausführen der ausgewählten Produktionsregel (Aktionsteil wird auf der Faktenbasis durchgeführt).

Abbruch, wenn Ziel erreicht, oder wenn keine Produktionsregeln mehr ausführbar ist.

Ableitung von Wissen durch Schließen

Vorwärts- und Rückwärtsverkettung

Betriebsarten für Interpreter:

Rückwärtsverkettung:

(backward chaining, backward reasoning, analytisches Schließen, zielgetriebene Schlussfolgerung)

Ausgehend von einem Goal (Startknoten) wird der Problemraum sukzessive nach Fakten (Zielknoten) durchsucht, mit Hilfe derer das Goal bewiesen werden kann.

Bei der Rückwärtsverkettung wird somit in umgekehrter Richtung, d.h. vom Ziel ausgegangen. Dazu wird eine Regel ausgewählt, deren Aktionsteil dem Ziel entspricht, und dieser Aktionsteil auf der Faktenbasis durchgeführt. Der Fakt wird also „probeweise“ in die Faktenbasis eingefügt.

Diese Vorgehensweise wird sukzessive für die Vorbedingungen dieser Regel(n) fortgesetzt.

Ableitung von Wissen durch Schließen

Vorwärts- und Rückwärtsverkettung

Betriebsarten für Interpreter:

Rückwärtsverkettung:

Im Verlauf des Interpretationsprozesses kann bei Rückwärtsverkettung ein (partieller) Zurücksetzen des Ableitungsvorgangs notwendig werden.

Beispiel:

Gegeben seien folgende Implikationen:

Regel 1: $E \leftarrow A \wedge C$
Regel 2: $H \leftarrow F \wedge C$
Regel 3: $H \leftarrow B \wedge E$
Regel 4: $C \leftarrow B$
Regel 5: $X \leftarrow H$

Folgende Aussagen gelten als wahr: A, B

Zu beweisen ist:

X

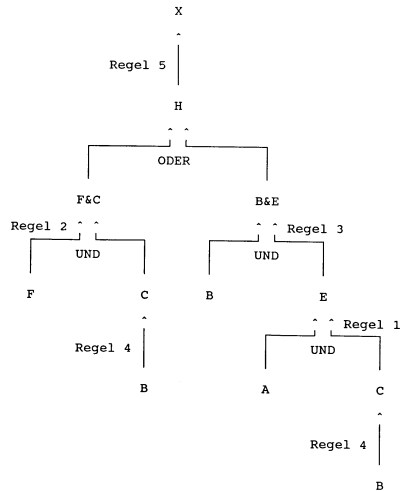
Als Schlussregel wird der Modus ponens verwendet:

Aus $B \leftarrow A$ und A folgt B

Für dieses Beispiel lässt sich folgender Suchbaum entwickeln:

Ableitung von Wissen durch Schließen

Vorwärts- und Rückwärtsverkettung



Vorwärtsverkettung:

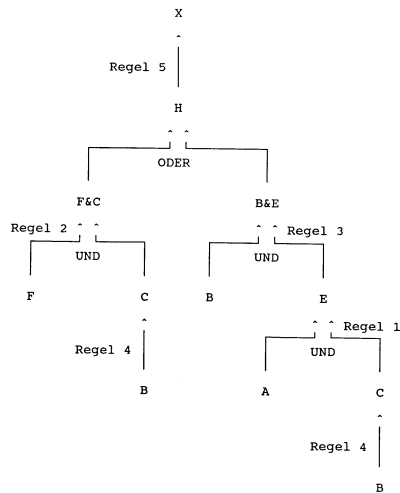
- Mit B kann C bewiesen werden, C wird in die Menge der wahren Aussagen aufgenommen.
- Mit A und C kann E bewiesen werden, E wird in die Menge der wahren Aussagen aufgenommen.
- Mit H kann X bewiesen werden, X wird in die Menge der wahren Aussagen aufgenommen.

Rückwärtsverkettung:

- X ist beweisbar, wenn H beweisbar ist.
- H ist beweisbar, wenn F und C beweisbar sind.
- F ist nicht beweisbar.
- H ist beweisbar, wenn B und E beweisbar sind.
- B ist wahr.
- E ist beweisbar, wenn A und C beweisbar sind.
- A ist wahr.
- C ist beweisbar, wenn B beweisbar ist.
- B ist wahr.

Ableitung von Wissen durch Schließen

Vorwärts- und Rückwärtsverkettung



Hinweise:

- Vorwärts- und Rückwärtsverkettung lassen sich sowohl mit Breitensuche als auch mit Tiefensuche kombinieren.
- Eine Kombination von Vorwärts- und Rückwärtsverkettung ist die **bidirektionale Suche**. Sie verfolgt das Ziel, den exponentiell wachsenden Suchbaum durch zwei (oder mehrere) kleinere Suchbäume zu ersetzen. Ein Suchbaum wird ausgehend vom Goal, die anderen Suchbäume ausgehend von den Fakten entwickelt. Die Suche ist beendet, wenn sich der ausgehend vom Goal entwickelte Suchbaum mit einem der anderen Suchbäume schneidet.

Ableitung von Wissen durch Schließen

Vorwärts- und Rückwärtsverkettung

Lösungssuche in Prolog:

Zusammenfassend lässt sich die Lösungssuche in Prolog wie folgt beschreiben:

- **Wissensrepräsentation:** Produktionsregeln, Fakten.
- **Verkettung:** Rückwärtsverkettung (Analytisches Schließen, Theorembeweiser).
- **Suchstrategie:** Tiefensuche (leftmost depth first).

Konfliktauflösung:

- Datenbasis von oben nach unten.
- Subgoals von links nach rechts.
- Redefinition eines Prädikats, sobald die Vorgängerdefinition endgültig scheitert.

Inferenzregel:

- Modus ponens

Ableitung von Wissen durch Schließen

Wissensableitung bei objektorientierter Wissensrepräsentation

Operationen auf Semantischen Netzen:

Die Schlussfolgerung auf Semantischen Netzen beruht insbesondere auf der Transitivität der Beziehungen *is_a* und *hat_teil*.

Bei Semantischen Netzen können zwei Arten von Vererbung unterschieden werden:

- **Vererbung zur Anfragezeit:** Das ableitbare Wissen wird nicht im Semantischen Netz gespeichert, sondern erst zum Zeitpunkt der Anfrage generiert.
- **Vererbung zur Änderungszeit:** Sobald neues Wissen in die Wissensbasis eingestellt wird, erfolgt die Generierung des ableitbaren Wissens, welches explizit gespeichert wird.

Bei der Wissensrepräsentation ist zwischen Integritätsbedingungen und Inferenzregeln zu unterscheiden. Integritätsbedingungen legen zulässige Repräsentationen fest. Inferenzregeln bestimmen, welche Schlussfolgerungen auf den vorhandenen Repräsentationen unter Beachtung der Integritätsbedingungen gezogen werden können.

Die eigentliche Anfrage an ein Semantisches Netz besteht in einer Form des Pattern Matching unter Verwendung von **Anfragenetzen**. Falls das Anfragenetz als Teilnetz des Semantischen Netzes bestimmt werden kann, ist die Anfrage erfüllt.

Ableitung von Wissen durch Schließen

Wissensableitung bei objektorientierter Wissensrepräsentation

Operationen auf Frames:

Die Operationen auf Frames stellen eine Erweiterung der Operationen auf Semantischen Netzen dar. Zusätzlich zu diesen sind verfügbar:

- **Terminologische Ersetzungen** (bei semantischen Beziehungen zwischen Konzepten).
- **Strukturabgleich von Frames** (Anfragen an Frames werden in Form von Frames formuliert).

Ableitung von Wissen durch Schließen

Schlüsse über unsicheres Wissen

Kombination von Sicherheitsfaktoren:

Die Darstellung erfolgt am Beispiel von Sicherheitsfaktoren des Intervalls $[0,1]$:

- Ist einer Aussage a der Sicherheitsfaktor w zugeordnet, dann besitzt die Aussage $\neg a$ den Sicherheitsfaktor $(1-w)$.
- Bei Regeln wird der Sicherheitsfaktor der Vorbedingung multiplikativ mit dem Sicherheitsfaktor der Regel verknüpft.
- Bei mehreren Evidenzen für ein Faktum kann der Sicherheitsfaktor nach folgender Regel kombiniert werden:

Ableitung von Wissen durch Schließen

Schlüsse über unsicheres Wissen

Kombination von Sicherheitsfaktoren:

$$SF_{neu} = SF_{alt} + SF_r SF_v (1 - SF_{alt})$$

SF_{neu} = neuer Sicherheitsfaktor

SF_{alt} = alter Sicherheitsfaktor

SF_r = Sicherheitsfaktor der Regel

SF_v = Sicherheitsfaktor der Vorbedingung

Ableitung von Wissen durch Schließen

Schlüsse über unsicheres Wissen

Kombination von Sicherheitsfaktoren:

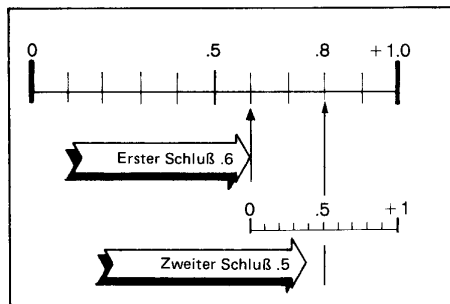
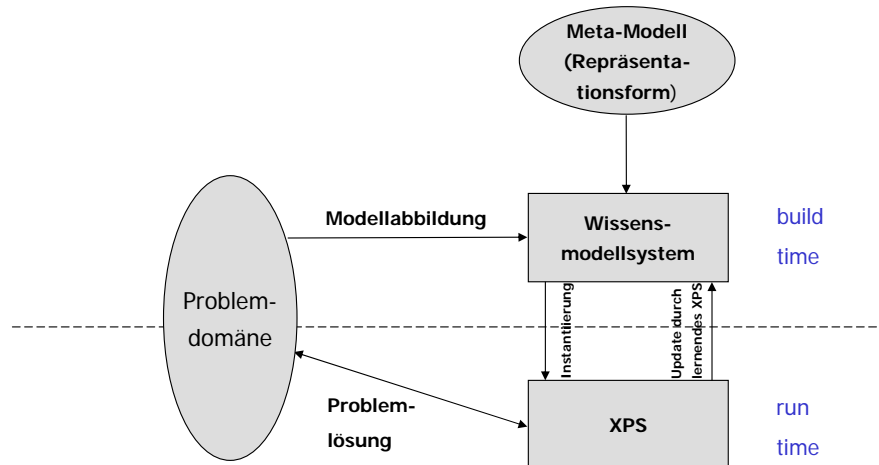


Abbildung 5.3 Grafische Darstellung der Kombination von zwei Konfidenzwerten. Der erste Schluß ist .6 sicher, der zweite .5, so daß der kombinierte Konfidenzfaktor um 40% näher an die absolute Gewißheit herankommt.

Wissensmodellierung

Beziehung zwischen Problemdomäne und XPS



WS 2008/09

Prof. Dr. E. J. Sinz, Universität Bamberg

Seite 88

Grundlagen der Wissensmodellierung

Wissensmodellierung umfasst die Identifikation (Erfassung), die Formalisierung (in Form eines Modellsystems) und die Wartung (Pflege) des Wissen, das für die Wissensbasis eines XPS benötigt wird.

Für die Wissensmodellierung sind drei Formen bekannt:

- **Indirekte Wissensmodellierung:** Wissensingenieur (knowledge engineer) befragt Experten und formalisiert das Wissen zu einem Wissensmodellsystem.
- **Direkte Wissensmodellierung:** Experte formalisiert selbst sein Wissen zu einem Wissensmodellsystem.
- **Automatischer Wissenserwerb:** XPS extrahiert sein Wissen selbst aus Falldaten und anderen Quellen. Lernendes XPS.

Wissensmodellierung

Phasenmodell des Wissenserwerbs (nach Buchanan)

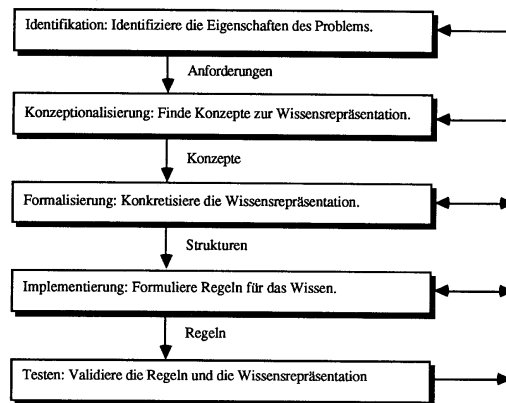


Abb. 13.1 Standardphasenmodell des Wissenserwerbs (nach Buchanan [83, S. 139])

Quelle: F. Puppe

WS 2008/09

Prof. Dr. E. J. Sinz, Universität Bamberg

Seite 89

Wissensmodellierung

Detailliertes Phasenmodell des Wissenserwerbs

PHASEN	BESCHREIBUNG	DURCHFÜHRUNG	HILFSMITTEL
<pre> graph TD A[Problemerkarakterisierung] --> B[Shell-Entwicklung] B --> C[Aufbau der Wissensbasis] C --> D[Wartung der Wissensbasis] B --> A D --> C </pre>	Identifikation der Problemlösungsstrategie und der Wissensrepräsentation	Wissensingenieur befragt Experten	Interviewtechniken und Protokolle
	Bereitstellung eines Expertensystem-Shells mit komfortabler Wissenserwerbskomponente	Auswahl oder Neuentwicklung durch den Wissensingenieur	allgemeines Expertensystem-Werkzeug
	Formalisierung des Expertenwissens	Experte, eventuell unterstützt durch Wissensingenieur	Shell
	Tuning und Weiterentwicklung, Anpassung an neue oder geänderte Anforderungen	Experte, eventuell unterstützt durch automatische Analysetechniken	Falldatenbank

Abb. 13.2 Konkretes Phasenmodell des Wissenserwerbs

Quelle: F. Puppe

WS 2008/09

Prof. Dr. E. J. Sinz, Universität Bamberg

Seite 90

Wissensmodellierung für ein einfaches Expertensystem

Typ des Expertensystems: Diagnosesystem

Expertensystem MVV (Münchner Verkehrsverbund). Quelle: Schnupp / Huu 1987

Die Aufgabe besteht in der Beratung von MVV-Nutzern hinsichtlich der Auswahl von Fahrscheinen.

a) Tarifregeln

- *ist_eine_Kinderkarte_moeglich* usw.
- Die Reihenfolge der Regeln bestimmt die Reihenfolge der Hypothesengenerierung. Dabei wird mit der jeweils günstigsten Hypothese begonnen.
- Tarifregeln unterstützen keine Erklärung.

Wissensmodellierung für ein einfaches Expertensystem

b) Fakten (Falldaten)

- *alter*, *geld* usw.
- Falldaten werden von den Tarifregeln als bekannt vorausgesetzt.
- Die Modellierung erfolgt über das Prädikat *status*:

status(geld, unbekannt), *status(alter, unbekannt)*.

Der Wert *unbekannt* wird während des Dialogs sukzessive durch konkrete Werte ersetzt.

Wissensmodellierung für ein einfaches Expertensystem

c) Dialogführung

- Eingaben: *run, hilfe, weiter, mvv* usw.
- Nutzung des XPS als reines **Auskunftssystem**, z.B.
ist_eine_Mehrfahrtenkarte_A_moeglich
- *Nutzung des XPS als **Diagnosesystem**: mvv.* Das Problem des Nutzers muss auf adäquate Weise gelöst werden; der Nutzer kennt die Problemstruktur nicht (wichtig: Reihenfolge der Hypothesengenerierung und -verwerfung).

Wissensmodellierung für ein einfaches Expertensystem

d) Abarbeitung der Regelbasis durch das Inferenzverfahren

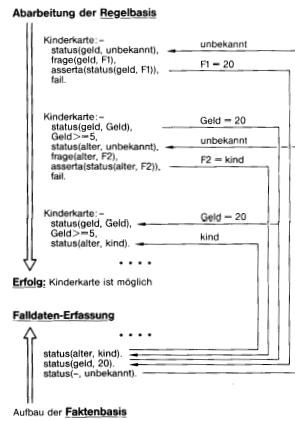
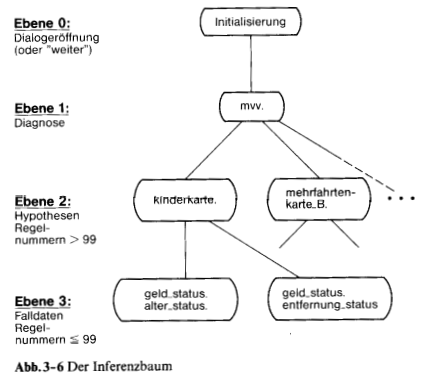


Abb. 3-4 Abarbeitung der Regeln mit schrittweiser Falldaten-Erfassung in der Faktenbasis

Wissensmodellierung für ein einfaches Expertensystem

e) Inferenzbaum und Inferenzprotokoll



Zur Modellierung werden folgende Fakten verwendet

- schritt nummeriert Schritte
- baum Trace(Schritt, Regel)
- ebene 2: Regel > 99; 3: Regel ≤ 99
- inkrement Ebeneninkrement
- status Falldaten

Wissensmodellierung für ein einfaches Expertensystem

f) Erklärungen

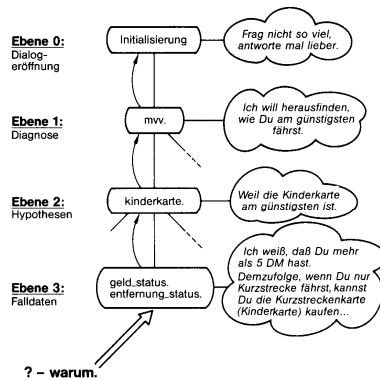


Abb.3-7 Die Erklärungshierarchie bei einer Diagnose

Auf die Frage *warum* soll eine nutzeradäquate und kontextabhängige Erklärung gegeben werden. Ein weiteres *warum* führt zu der nächst höheren Erklärungsebene.

erklaerung(Ebene, Regel): Zu jeder Regel und jeder Erklärungsebene wird ein Erklärungstext angeboten.