Inhaltsverzeichnis

Kapitel 1

Einleitung

Diese Arbeit findet statt im Rahmen eines Projektes zur Anwendung maschineller Lernverfahren in der Robotik (B-Learn II, ESPRIT P7274). Innerhalb dieses Projektes soll die Kommunikation des Menschen mit dem Roboter durch Handlungsanweisungen in der Art „Gehe durch die Tür, dann links, dann durch die zweite Tür rechts“ geschehen. Damit soll eine Navigation des Roboters ermöglicht werden. Mobile Roboter können so flexibel auf neue Umgebungen vorbereitet werden, sowohl von exakten Koordinaten als auch von bestimmten Orten wird abstrahiert. So gibt es zum Beispiel für „Tür“ ein Symbol, das für verschiedene Türen mit unterschiedlichen Abmessungen steht. Gegenwärtige Versuche finden in Büroumgebungen statt.

In einer Lernphase werden einzelne Handlungsbegriffe wie „Gehe durch die Tür“, von dem Roboter gelernt. Dazu müssen die Handlungsbegriffe aus den Bewegungsdaten des Roboters und seinen Wahrnehmungen abgeleitet werden. Da die Kluft zwischen diesen beiden Ebenen zu groß ist, um sie in einem Schritt zu überbrücken, wird eine Hierarchie von Begriffen gelernt, die die Messungen der externen Sensorik und die elementaren Bewegungsbefehle des Roboters mit den vom Menschen gegebenen Handlungsanweisungen verbindet. Auf den verschiedenen Stufen der Hierarchie werden verschiedene Arten von Begriffen gebildet. Für eine umfassende Darstellung sei [Klingspor et al., 1996] empfohlen.

- Aufbauend auf den Sensordaten, die während der Aktionen des Roboters aufgenommen werden, liefern Wahrnehmungs-Merkmale in mehreren Stufen Merkmale für die Objekterkennung.

- Die elementaren Bewegungungen des Roboters, Fahren, Drehen und

Stehen, werden mit Hilfe von Ausführungs- oder Basis-Handlungs-Merkmalen beschrieben.

- Objekte werden mit einer zugehörigen Handlung assoziiert. Zur Erkennung dessen, was der Roboter in welcher Umgebung getan hat, ist es deshalb sinnvoll, obige Merkmale zu integrieren in Merkmale zur Verbindung von Wahrnehmung und Handlung. Diese Abstraktionsstufe in der Hierarchie wird wahrnehmungsintegrierende-Handlungs-Merkmale genannt.

- Operationale Merkmale an der Spitze der Hierarchie bilden die Schnittstelle zu den von einem Menschen gegebenen Handlungsanweisungen.

Verschiedenste Stufen dieser Hierarchie wurden bereits in anderen Arbeiten behandelt, so daß an den Schnittstellen Ergebnisse genutzt, beziehungsweise weitergereicht werden. Bei den Wahrnehmungs-Merkmalen wird unterschieden zwischen Wahrnehmungen, die zu einem einzelnen Zeitpunkt stattgefunden haben, und solchen, die während einer Bewegung des Roboters eingetreten sind, sich also über ein Wahrnehmungszeitintervall erstrecken. Die zeitpunktorientierten Wahrnehmungen beschreiben Situationen, in denen der Roboter relativ zu einem Objekt steht. Beispiele hierfür sind „vor_der_tür“ und „in_der_Ecke“. Die Zeitintervallwahrnehmungen sind grundsätzlich mit einer Handlung verbunden. Begriffe wie „durch_die_tür“ oder „entlang_einer_tür“ beinhalten, daß der Roboter während der Wahrnehmung der Tür durch sie hindurch, beziehungsweise an ihr entlang gefahren ist. Die einzelnen Abstraktionsstufen der Zeitintervallwahrnehmungen sind Thema des nächsten Abschnittes.

1.1 Aufgabenbeschreibung

Das Thema dieser Arbeit enthält zwei Aspekte: zum einen die allgemeine Aufgabenstellung des Lernens von relevanten Merkmalen, welche ein Teilgebiet der Forschung auf dem Gebiet des maschinellen Lernens darstellt, zum anderen die Anwendungsdomäne, Verarbeitung von Sensordaten eines mobilen Roboters. Beide Aspekte werden im Folgenden näher erläutert.

Der in dem Projekt eingesetzte Roboter[1] besitzt vierundzwanzig rundum angeordnete Ultraschallsensoren. In der Richtung, in der sie jeweils ihren

[1] Der Roboter PRIAMOS wurde and der Universität Karlsruhe entwickelt [Dillmann et al., 1993]

Ultraschallkegel absenden, messen die Sensoren die Distanz zu dem nächstgelegenen Objekt, sofern es innerhalb einer Reichweite von fünf Metern liegt. Die Distanz berechnet sich dabei aus der Zeit zwischen dem Senden bis zum Eintreffen des von dem Objekt reflektierten Signals. Jeder Reihummessung aller 24 Sensoren wird ein Zeitpunkt zugeordnet. Man erhält zu jedem Zeitpunkt vierundzwanzig Messungen.[2]

In einem ersten Abstraktionsschritt werden die numerischen Distanzwerte in eine symbolische Repräsentation überführt. Dazu wird ein Repräsentationsformalismus eingesetzt, der in der gesamten Begriffshierarchie beibehalten wird. Es handelt sich dabei um Hornklausellogik, einer Einschränkung der Prädikatenlogik erster Stufe. Klauseln werden aus Mengen von Literalen gebildet. Ein Literal ist ein Atom oder ein negiertes Atom. Ein Atom besteht in der hier benutzten Form aus einem Prädikatssymbol und einer Menge von Termen, die Konstanten oder Variablen sind. Auf der semantischen Ebene setzt ein so gebildetes Prädikat die Terme, eine bestimmte Anzahl von Objekten, in eine durch das Prädikatssymbol beschriebene Relation:

$$prädikatssymbol(Objekt_1,...,Objekt_n)^3$$

ist ein Prädikat der Stelligkeit n. Das Prädikatssymbol wird dazu eingesetzt, Begriffe der Begriffshierarchie zu bezeichnen. Objekte besitzen eine Sorte.

Beispiele für Objekte sind solche von der Sorte „Sensor“ oder „Zeitpunkt“. Konstanten bezeichnen jeweils ein bestimmtes Objekt, beispielsweise den Sensor *s6* (kleingeschrieben), Variablen bezeichnen jeweils ein beliebiges Objekt der entsprechenden Sorte, beispielsweise einen Zeitpunkt *T1* (im Folgenden großgeschrieben).

Literale stehen entweder im Klauselkopf, der aus einem positiven Literal besteht, oder im Klauselkörper, der aus der Disjunktion mehrerer negativer Literale besteht. Klauselkopf und/oder Klauselkörper können auch leer sein. In dem Fall, in dem nur der Klauselkopf vorhanden ist, ist das ein gegebenes Fakt. Eine Klausel mit Q als positivem und P_i als negativen Literalen kann als Regel geschrieben werden,

$$Q \leftarrow P_1, ..., P_n.$$

mit deren Hilfe aus den Literalen im Klauselkörper das Literal im Klauselkopf gefolgert werden kann. Prädikate, die für Begriffe auf den unteren

[2]Tatsächlich messen die Sensoren reihum nacheinander, damit nicht ein Sensor die Signale eines anderen aufnimmt.

[3]abgekürzt auch *prädikatssymbol/n*

Ebenen der Begriffshierarchie stehen, werden im Klauselkörper, der Prämisse einer Regel, kombiniert, die daraus Begriffe einer höhergelegenen Ebene der Hierarchie folgert. Die Regeln bilden so eine inferenzielle Hierarchie für die Begriffe.

Wie die Regeln aussehen ist die Fragestellung des Lernens, insbesondere welche Begriffe repräsentiert werden sollen (Merkmalsextraktion) und wie sie aus den Begriffen niedrigerer Hierarchiestufen gefolgert werden können. Die vorliegende Arbeit versucht diese Lernaufgabe für einen Ausschnitt aus den Zeitintervallwahrnehmungen zu lösen. Gelernt werden sollen Begriffe zur Charakterisierung der Wahrnehmungen einzelner Sensoren, sogenannte Sensor-Merkmale. Zur Einordnung werden in den folgenden Abschnitten auch Lernergebnisse der darüber- und darunterliegenden Abstraktionsstufen der Zeitintervallwahrnehmungen der Reihe nach vorgestellt. Die inferenzielle Hierarchie von Begriffen ist erdacht und beschrieben worden in [Morik/Rieger, 1993] und in [Klingspor et al., 1996].

1.1.1 Basis-Merkmale

Ausgehend von den Messungen werden Basis-Merkmale gebildet. Aufeinanderfolgende Messungen, die hinreichend ähnlich sind, werden zu Intervallen zusammengefaßt, so daß die Schwankungen der Messwerte in den Intervallen möglichst gering sind. Es werden Begriffe gebildet, die die Eigenschaften der in einem Intervall enthaltenen Messungen widerspiegeln. Folgende Begriffe werden gegeben:
keine_messung, aufsteigend, absteigend, konstant, aufsteigend_sprung, absteigend_sprung, keine_bewegung, direkt_darauf_zu, direkt_davon_weg und etwas_passiert.
Damit werden sechsstellige Fakten gebildet, deren Prädikatssymbol die oben genannten Begriffe sind:

basis-merkmal[4] *(Tr,Or,S,T1,T2,Grad).*

Die Sorten der Variablen sind die Nummer einer bestimmten Roboterfahrt (bei Tr), die Orientierung des Sensors (Or), die Sensornummer (S), Anfangs- und Endzeitpunkt (T1/T2) des Zeitintervalls sowie ein Gradient (Grad), der den Begriff näher charakterisiert. Die so gebildeten Merkmale bilden die unterste Stufe der Zeitintervallwahrnehungen und werden Basis-

[4] abgekürzt „bf“

(Wahrnehmungs)-Merkmale[5] genannt. Mit Hilfe der enthaltenen Zeitpunkte lassen sich die Basis-Merkmale über die Zeit verknüpfen, so daß die Wahrnehmungen eines Sensors während einer bestimmten Fahrt durch eine Sequenz von Basis-Merkmalen repräsentiert werden.

Eine vollständige Ausarbeitung zum Thema der Basis-Merkmale, insbesondere die Darstellung einer Lernschleife ist zu finden in [Wessel, 1995]. Von den Basis-Merkmalen hängen die Merkmale der höheren Ebenen ab.

1.1.2 Sensor-Merkmale

Sensor-Merkmale sind Merkmale für die Wahrnehmungen einzelner Sensoren. Aus den Sequenzen von Basis-Merkmalen werden solche herausgesucht, die Objekte der Umwelt beschreiben. So kann man sich zum Beispiel vorstellen, daß für bestimmte Wand- oder Kantenkonstellationen, die Sequenzen von Basis-Merkmalen immer dieselben sind. In Abbildung 1.1 ist eine Kon-

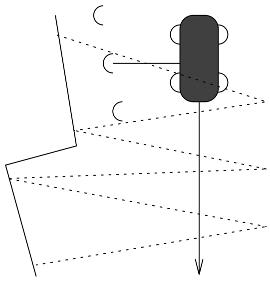


Abbildung 1.1: Sensor-Merkmale

stellation von zwei Kanten gezeigt: Nimmt ein Sensor aufeinanderfolgend die Basis-Merkmale *fallend/6-aufsteigend_sprung/6-fallend/6* auf, so ist er diagonal an zwei Wänden entlanggefahren, die in der Konstellation „sprung“ zueinander stehen. In der Abbildung ist dieses durch eine Regel ausgedrückt. Das Sensor-Merkmal *s_sprung* wurde während der Fahrt mit Nummer (Tr) vom Sensor (S) im Zeitintervall von (T1) bis (T4) wahrgenommen, wobei T1 und T4 der Anfangs- beziehungsweise Endzeitpunkt der Basis-Merkmal-Sequenz sind. Das fünfte Argument (RM) macht eine Aussage darüber, wie sich der Roboter relativ zu dem beobachteten Objekt bewegt, diagonal oder parallel. Einige weitere Kantenkonstellationen sind denkbar. Aller-

[5] Zu unterscheiden von Basis-Handlungs-Merkmalen, die hier aber nur am Rande auftreten und entsprechend gekennzeichnet sind.

dings könnten auch ganz andere Muster anstelle der Kantenkonstellationen durch die wahrgenommenen Basis-Merkmal-Sequenzen beschrieben werden. Um solche Muster beim Lernen nicht vorgeben zu müssen, besteht der in dieser Arbeit verfolgte Ansatz darin, unüberwacht Sensor-Merkmale in den Basis-Merkmal-Sequenzen zu entdecken.

1.1.3 Sensorgruppen- und handlungsorientierte Wahrnehmungs-Merkmale

Wenn mehrere benachbarte Sensoren innerhalb eines kleinen Zeitintervalles das gleiche Merkmal erkennen, wird dies zu einem Sensorgruppen-Merkmal zusammengefaßt. Die Information, welche Sensoren als benachbart gelten, ist in einem Fakt

sclass(Tr,S,T1,T2,SCL)

enthalten, wobei (Tr) wieder eine bestimmte Roboterfahrt bezeichnet und (T1/T2) das Zeitintervall, in dem der Sensor (S) zu der Sensorgruppe (SCL) gehört. Die Zugehörigkeit eines Sensors zu einer Sensorgruppe ist auch abhängig von der Bewegungsrichtung des Roboters, weshalb die sclass-Fakten nach jeder Drehung für den entsprechenden Zeitabschnitt berechnet werden. Ein Beispiel für eine Regel mit zwei Sensoren und dem Merkmal *s_sprung*[6]:

```
s_sprung(Tr,S1,T1,T2,diagonal) & sclass(Tr,S1,T0,T6,SCL) &
s_sprung(Tr,S2,T3,T4,diagonal) & sclass(Tr,S2,T0,T6,SCL) &
folgt(T1,T3) & T0<T1 & T4<T6
--> sg_sprung(Tr,SCL,T1,T4,diagonal).
```

Auf der obersten Stufe der Zeitintervallwahrnehmungen werden die Merkmale *durch_tür, entlang_tür* und *entlang_schrank* aufgrund der Sensorgruppen-Merkmale inferiert. Diese Merkmale werden handlungsorientierte Wahrnehmungs-Merkmale genannt. Ein handlungsorientiertes Wahrnehmungs-Merkmal,

hwm(Tr,T1,T2,RM),

wird festgelegt durch die Fahrtnummer (Tr), das Zeitintervall (T1/T2), in dem es gilt, und die relative Bewegung (RM) zu dem wahrgenommenen Objekt. An dieser Stelle stellt sich heraus, welche Sensor-Merkmale und welche

[6] Das Prädikat *folgt/2* erzwingt, daß die Wahrnehmungen der beiden Sensoren nur um wenige Zeitpunkte verschoben sein dürfen

Sensorgruppen für die Beschreibung handlungsorientierter Wahrnehmungs-Merkmale interessant sind. In einem Beispiel für das Merkmal *durch_tür* sind dies *s_sprung* und die Sensorgruppen rechte und linke Seite:

```
sg_sprung(Tr,rechts,T1,T2,diagonal) &
sg_sprung(Tr,links,T3,T4,diagonal) & folgt(T1,T3)
--> durch_tuer(Tr,T1,T4,diagonal)
```

Damit ist der komplette Rahmen der Aufgabe gegeben; der nun folgende Abschnitt skizziert die weiteren Schritte.

1.2 Beschreibung des Ansatzes

Zur Lösung der gestellten Aufgabe, des Lernens von Sensor-Merkmalen, sind grundsätzlich zwei verschiedene Vorgehensweisen möglich. Bereits untersucht worden ist die Möglichkeit, Merkmale als Beispiele vorzugeben und damit ein Verfahren für überwachtes Lernen zu verwenden. Diese Versuche wurden in erster Linie mit den Lernverfahren RDT/GRDT gemacht.[7] Ein Vergleich mit den Ergebnissen eines anderen Lernverfahrens, GRENDEL, ist in [Klingspor et al., 1996] zu finden.

Die vorgegebenen Merkmale sind die eingangs erwähnten Kantenkonstellationen. Neben dem Wandvorsprung wurden konvexe und konkave Ecken, sowie eine glatte Wand als Sensor-Merkmale verwandt. Gebildet wurden sie aufgrund hypothetischer Überlegungen mit der Fragestellung, was bei der gegebenen Anwendung an Merkmalen auftreten könnte. Da die Ergebnisse beim Lernen von Merkmalen höherer Abstraktionsstufen mit den Kantenmerkalen nicht zufriedenstellend sind, ergeben sich Zweifel, ob die vorgegebenen Kantenmerkmale relevant sind.

Der deswegen in dieser Arbeit verfolgte Ansatz besteht darin, Sensor-Merkmale aus den gegebenen Sequenzen von Basis-Merkmalen zu lernen, ohne explizite Muster vorzugeben. Stattdessen wird jedes der vorhandenen handlungsorientierten Wahrnehmungs-Merkmale als Oberbegriff für ein gleichnamiges Sensor-Merkmal benutzt, das den zeitlichen Rahmen (Anfangs- und Endzeitpunkt der Basis-Merkmal-Sequenz) vorgibt, innerhalb dessen das Sensor-Merkmal liegen muß.

Die Basis-Merkmal-Sequenzen zu einem Oberbegriff werden als Beobachtungen aufgefaßt, in denen nach Gemeinsamkeiten gesucht wird. Praktisch

[7] zu RDT siehe [Kietz/Wrobel, 1991], zu GRDT sei [Klingspor, 1994] empfohlen

bedeutet das eine Suche nach Clustern[8] in den Basis-Merkmalen-Sequenzen. In dem Bereich des maschinellen Lernens wird das Lernen von Begriffen aus Beobachtungen als „Conceptual-Clustering“ bezeichnet. Ein wesentlicher Bestandteil dieser Arbeit liegt also in der Anwendung und im Experimentieren mit solchen Techniken.

Die Vorgehensweise der Vorgabe handlungsorientierter Wahrnehmungs-Merkmale als Oberbegriffe soll im nächsten Abschnitt näher erläutert werden.

1.2.1 Vorgehensweise

Die Problematik verschiedener Sensor-Merkmale wird anhand der in Abbildung 1.2 gezeigten Fahrt des Roboters diskutiert. Der Roboter fährt dabei aus dem Flur (wobei die Enden des Flurs von den Sensoren nicht mehr erfaßt werden) in einen Raum hinein.

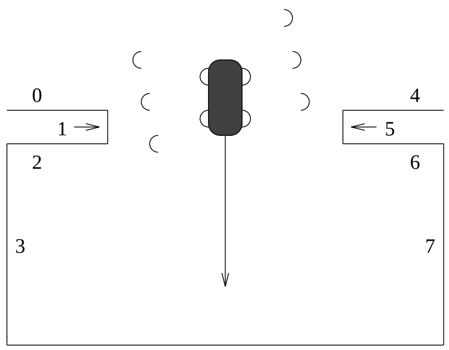


Abbildung 1.2: Fahrt durch Tür

Auf der Ebene der handlungsorientierten Wahrnehmungs-Merkmale entspricht dieses Beispiel dem *durch_tür* Begriff. Benutzt man Kantenmuster als Sensor-Merkmale, so werden zum Beispiel für die Seitensensoren die Basis-Merkmal-Sequenzen der Zeitintervalle, in denen sie die Kanten 5 und 7 zur linken, beziehungwsweise die Kanten 1 und 3 zur rechten Seite des Roboters erfassen, als Beschreibung eines *s_sprung(...,parallel)* Merkmales gelernt. Anstelle der Kantenmuster lassen sich andere Merkmale denken. Zum Beispiel kann man die während der gesamten Fahrt wahrgenommene Sequenz als Beschreibung eines Merkmales, das die ganze Tür und damit weitere Kanten umfaßt, zu lernen versuchen. Das ist dann der entgegengesetzte Extremfall.

[8]Der englische Begriff *cluster* bedeutet eine Ansammlung ähnlicher Objekte.

In dem hier verfolgten Ansatz werden hingegen beim Lernen der Sensor-Merkmale nicht Basis-Merkmal-Sequenzen, deren Anfangs- und Endzeitpunkte mit denen vorgegebener Beispiele übereinstimmen, in Betracht gezogen, sondern alle, die in dem durch den Oberbegriff gegebenen Zeitrahmen liegen.

Die zu lernenden Sensor-Merkmale werden demnach eng an die handlungsorientierten Wahrnehmungs-Merkmale gekoppelt. Dies sind die Wahrnehmungen bei der Fahrt des Roboters durch eine Tür (Merkmal *durch_tür*), entlang einer Tür (*entlang_tür*) und entlang eines Schrankes (*entlang_schrank*). Ein Lehrer wählt aus den Roboterfahrten die Bewegungsabschnitte aus, in denen diese Merkmale vorkommen. Das ist die gleiche Vorgehensweise wie bei der Auswahl von Beispielen zum überwachten Lernen der handlungsorientierten Wahrnehmungs-Merkmale. Es werden Fakten in der Form

hwm(Tr,T1,T2,RM)

vorgegeben. Dem Lehrer müssen an dieser Stelle die Fahrtstrecken und Umgebungen bekannt sein, um die Merkmale und auch die relative Bewegung zu den Objekten angeben zu können.

Anhaltspunkte für die zurückgelegten Wegstrecken sind dabei Positionsangaben, die der Roboter durch Fortschreibung der von ihm ausgeführten Bewegungen erhält. Die Positionsangaben werden vom Roboter zusammen mit den Sensormessungen jeweils zu den entsprechenden Zeitpunkten ausgegeben. Von der Umgebung sollte eine Zeichnung vorhanden sein. Um festzustellen, wann der Roboter wo gewesen ist, wird die Zeichnung der Umgebung mit der Wegstrecke und den darin eingetragenen Zeitpunkten verglichen.

Zur Vorgabe der Zeitpunkte (T1) und (T2) können die in den Basis-Handlungs-Merkmalen[9] bekannten Anfangs- und Endzeitpunkte eines Bewegungsabschnittes zur Hilfe genommen werden. Diese Zeitpunkte bilden die untere und obere Schranke, innerhalb derer handlungsorientierte Merkmale wahrgenommen werden können, da die Handlung, an der sich diese Merkmale orientieren, eben die Fahrbewegung des Roboters ist. Innerhalb eines Bewegungsabschnittes, der einem Basis-Handlungsmerkmal entspricht, kommen keine Richtungsänderungen des Roboters vor. Gibt es mehrere handlungsorientierte Wahrnehmungs-Merkmale in einem Bewegungsabschnitt, zum Beispiel wenn der Roboter auf einem Flur geradeaus an mehreren Türen vorbeigefahren ist, dann muß das Zeitintervall dieses Abschnittes

[9] Siehe hierzu die Ausführungen zu den Basis-Handlungs-Merkmalen in [Sklorz, 1995].

von dem Lehrer auf die entsprechenden Merkmale aufgeteilt werden. Die Zeitpunkte an denen getrennt werden soll, werden durch erneute Betrachtung der Fahrtstrecke ermittelt.

Die dabei festgelgten Zeitintervalle bilden den Rahmen, innerhalb dessen die Sensor-Merkmale wahrgenommen werden können. Hierin besteht die Kopplung der Sensor- an die handlungsorientierten Merkmale: die Zeitintervalle, in denen Sensor-Merkmale wahrgenommen werden, liegen innerhalb der Intervalle der handlungsorientierten Merkmale. Damit liegen auch die Basis-Merkmal-Sequenzen, welche die Sensor-Merkmale beschreiben, zeitlich innerhalb der durch die handlungsorientierten Merkmale vorgegebenen Grenzen.

Die einzelnen Sensor-Merkmale werden getrennt gelernt. Auf diese Weise werden die Sequenzen gefunden, die speziell für ein bestimmtes Sensor-Merkmal relevant sind.

Diese Form des Lernens ist insofern unüberwacht, daß zwar zielgerichtet die zugehörige Wahrnehmung eines handlungsorientierten Wahrnehmungs-Merkmales vorgegeben wird, hierdurch jedoch nur der äußere Rahmen gegeben ist, innerhalb dessen nach Clustern gesucht wird. Dabei wird ausgenutzt, daß die Sensor-Merkmale in eine Hierarchie von Wahrnehmungs-Merkmalen integriert sind. Das hat den Vorteil, daß der Suchraum für den Begriff Sensor-Merkmal durch den entsprechenden Oberbegriff eingeschränkt wird.

1.2.2 Evaluierung

In diesem Abschnitt werden die Kriterien vorgestellt, anhand derer in den Experimenten die Relevanz der gelernten Regeln für Sensor-Merkmale bewertet werden wird. Dieses kann auf zwei Ebenen geschehen, einmal nach Durchlaufen der Regelhierarchie bis zu den handlungsorientierten Wahrnehmungs-Merkmalen und einmal direkt auf der Ebene der Sensor-Merkmale.

Auf der oberen Ebene gelten ganz allgemein Regeln für Sensor-Merkmale als relevant, wenn sie Sensor-Merkmale so ableiten, daß mit ihnen Sensorgruppen- und handlungsorientierte Wahrnehmungs-Merkmale gelernt werden können.

Als Grundlage für ein Bewertungskriterium auf der Ebene der Sensor-Merkmale können die für Regelmengen geltenden Begriffe *generell* und *speziell* verwendet werden. Mit ihnen wird beim überwachten Lernen die Qualität von Regeln, und hier die derer, die aus Basis-Merkmalen Sensor-Merkmale ableiten, bewertet.

- Eine Regelmenge ist hinreichend *generell*, wenn alle positiven Beispiele abgedeckt werden, das heißt, wenn die Regelmenge aus dem Hintergrundwissen die positiven Beispiele ableitet. Die vorgegebenen Sensor-Merkmale sind die positiven Beispiele, die Basis-Merkmale bilden das Hintergrundwissen.

- Eine Regelmenge ist hinreichend *speziell*, wenn kein negatives Beispiel abgedeckt wird. Als negative Beispiele für ein Sensor-Merkmal kann man alle Zeitintervalle ansehen, für die man kein positives Beispiel angeben kann. Um nicht so viele negative Beispiele zu haben, kann man sie aber auch explizit vorgeben. Als negative Beispiele eines Merkmals X eignen sich dazu die positiven Beispiele der jeweils anderen Sensor-Merkmale.

In unserem Fall handelt es sich allerdings um unüberwachtes Lernen, so daß die Abdeckung von Beispielen nicht getestet werden kann. Was allerdings getestet werden kann, ist die Abdeckung des Oberbegriffes „handlungsorientierte Wahrnehmungen", für den die Beispiele bekannt sind. Ein Sensor-Merkmal wird nur dann **korrekt** abgeleitet, wenn für das passende handlungsorientierte Wahrnehmungs-Merkmal ein Beispiel vorhanden ist, in dessen Zeitintervall das Sensor-Merkmal liegt. So kann die Abdeckung getestet werden ohne die Regeln, die aus Sensor-Merkmalen Sensorgruppen-Merkmale und daraus handlungsorientierte Wahrnehmungsmerkmale ableiten, kennen zu müssen, indem gezählt wird

1. wieviele Sensor-Merkmale korrekt abgeleitet werden,
2. wieviele Sensor-Merkmale fälschlicherweise abgeleitet werden, und
3. zu wievielen Beispielen von handlungsorientierten Wahrnehmungs-Merkmalen keine Sensor-Merkmale abgeleitet werden.

Mit den Punkten 1. und 3. wird die **Vollständigkeit** getestet; unter Punkt 2. die **Korrektheit**.

Es soll auch gestestet werden, ob die gelernten Regeln allgemein genug sind und auch in Fahrten, die beim Lernen nicht benutzt wurden, Sensor-Merkmale ableiten. Dazu werden die Fahrten des Roboters in Lernset und Testset aufgeteilt. Zum Lernen werden nur die Basis-Merkmal-Sequenzen des Lernsets verwandt. Beim Testen der Punkte 1. und 3. werden die Basis-Merkmal-Sequenzen des Testsets hinzugenommen. Punkt 2. wird anhand von Basis-Merkmal-Sequenzen aus Fahrten, in denen das Sensor-Merkmal nicht vorkommen darf, getestet.

1.3 Datenformat

In diesem Abschnitt wird beschrieben, wie die Daten zum Lernen der Sensor-Merkmale zusammengestellt werden. Hierfür wird eine Ausgangsrepräsentation benutzt, auf die in den Kapiteln 3, 4 und 5, in denen Experimente durchgeführt werden, zurückverwiesen werden wird.

Für die in dieser Arbeit verwendete Ausgangsrepräsentation werden die von [Rieger, 1996b] entwickelten *Cases* benutzt. Ein *Case* ist dort eine Liste, deren Kopfelement als *Zielprädikat*, und deren Rest als *beschreibende Prädikate* bezeichnet wird.

Hier sind die *Zielprädikate* Rahmen für Sensor-Merkmale *s_hwm_dummy/6*, die aus den als Oberbegriffen gegebenen *hwm/6* Fakten gebildet werden. Im Namen eines Zielprädikates ist ein handlungsorientiertes Wahrnehmungs-Merkmal angegeben. Der Platzhalter „dummy" soll klarmachen, daß es sich nicht um ein Merkmal handelt, sondern nur um den äußeren Rahmen.

Die *beschreibenden Prädikate* sind Basis-Merkmale, die als zeitlich geordnete Sequenz über einen Abgleich der Zeitargumente und der Fahrt- und Sensornummer in *bf/6* Fakten der Basis-Merkmals-Daten und den *s_hwm_dummy/6* Zielprädikaten, in die *Cases* geschrieben werden. Das geschieht mit Hilfe der *case_selection* in dem ebenfalls in [Rieger, 1995b] beschriebenen Datenvorbereitungstool.

Im folgenden wird beschrieben, wie zu den von einem Lehrer vorgegebenen *hwm(Tr,T1,T2,RM)* Fakten Rahmen für Sensor-Merkmale, *s_hwm_dummy(Tr,S,T1,T2,RM,SCL)* gebildet werden.

Zu einem *hwm(Tr,T1,T2,RM)* Fakt werden mit Hilfe der Basis-Handlungs-Merkmale *sclass(Tr,S,T1,T2,SCL)* Fakten gebildet, die angeben, welche Sensoren (S) zu welcher Sensorgruppe(SCL) gehören, in dem Fahrtabschnitt zwischen T1 und T2. Jedes der *sclass/6* Fakten liefert den Sensor und die Sensorgruppe eines *s_hwm_dummy(Tr,S,T1,T2,RM,SCL)* Faktes, dessen Name und restlichen Argumente von dem entsprechenden *hwm(Tr,T1,T2,RM)* stammen.

Hierin besteht ein weiterer Unterschied zu den Kantenmerkmalen. Die Sensor-Merkmale werden hier unter Vorgabe der Sensorgruppen gelernt. Die obigen Kantenmerkmale werden jeweils nur für die Sensoren gebildet, die von den betroffenen Kanten auch ein Echo erhalten haben. Hier ist dagegen ein Sensor-Merkmal an keine bestimmte Kante gebunden.

Alle Sensoren nehmen bei der Fahrt etwas wahr, das charakteristisch für die Wahrnehmung des Merkmales sein kann. Da die Wahrnehmungen abhängig von der Position der Sensoren am Roboter allerdings verschieden sind, soll direkt auf dieser Ebene unterschieden werden, welche der gelernten Regeln für welche Sensoren gelten. Insbesondere werden hier die drei überlappenden Sensorgruppen „seite“, „vorne_groß“ und „hinten_groß“ gebildet. Die Bildung der Sensorgruppen ist im Anhang A.2 gezeigt.

Die endgültige Form der *Cases* ist somit die folgende:

$$[s\_hwm\_dummy(Tr,S,T1,T2,RM,SCL),\ bf_1/6, ..., bf_n/6].$$

Cases mit dem gleichen *Zielprädikat*[10] bilden die Eingabe zum Lernen eines entsprechenden *s_hwm/6* Sensor-Merkmales.

Die s_hwm_dummy/6 Fakten von Fahrten die die Zielprädikate eines Lernsets bilden sollen, werden in dem Werkzeug zur Datenverwaltung von [Zitzler, 1996] über ein *synt_select* aus der *Variante* „sfmgd_traces30-150“, die alle s_hwm_dummy/6 Fakten enthält, in ein *Sample* geladen. Das Kommando *synt_select(variant(D,N),C,S)* selektiert die Daten einer *Variante* (spezifiziert durch die Datenklasse (D) und den Namen (N)) gemäß den Bedingungen in den Constraints (C) in ein *Sample* (S) :

synt_select(variant(sensor_features,sfmgd_traces30-150),
[item(predname,[„Name des s_hwm_dummy“]),
item(trace, [„Fahrten des Lernsets“]),
item(sensorclasss, [„Name der Sensorgruppe (SCL)“]),
item(rmovement, [„Name der relativen Bewegung (RM)“])],
S).

Mit Hilfe eines *case_select(S,R,SB,Ca)*[11] Kommandos werden zur Erstellung eines Lernsets *Cases* (Ca) gebildet. Die Auswahl der Basis-Merkmale im Sample (SB) zu den Zielprädikaten des Samples (S) geschieht über die Relationenliste (R).[12]

Mit den *Cases* ist das Ausgangsformat gegeben. Für die Eingabe verschiedener Verfahren in den Kapiteln 3, 4 und 5 werden Umrepräsentationen vorgenommen.

Nach den Umrepräsentationen werden einige beim Lernen nicht verwendete Argumente der Basis-Merkmale nicht weiter enthalten sein. Nicht als Lernergebnis verwendet werden sollen die Orientierung, der Gradient und die

[10] genau: gleiches Prädikatssymbol und gleiche RM und SCL Konstanten

[11] zurückgreifend auf *case_selection/4* bei Anke Rieger

[12] `R = [[trace,'=',trace],[sensor,'=',sensor],[time1,'=<',time1],[time1,'=<',time2],[time2,'>=',time1],[time2,'>=',time2]]`

absolute Dauer des Wahrnehmungszeitraumes eines Basis-Merkmales. Erstere ist als Lernergebnis unbrauchbar. Von der Dauer des Wahrnehmungszeitraumes eines Basis-Merkmales soll abstrahiert werden, so daß die genauen Werte der Zeitpunkte ebenfalls nicht verwendet werden. Der Gradient ist eine Verfeinerung des Basis-Merkmals. Seine weitere Verwendung in der inferenziellen Hierarchie ist nicht vorgesehen.

1.4 Gliederung der Arbeit

Im zweiten Kapitel werden an Beispielen aus der Literatur Verfahren zum Conceptual-Clustering gezeigt, einige in diesem Bereich verwendete Begriffe eingeführt und grundlegende Methoden offengelegt. Im dritten Kapitel wird ein für das Anwendungsgebiet relevantes Verfahren aus der Literatur beschrieben und eingesetzt. Das vierte Kapitel behandelt alternativ dazu ein eigenes entwickeltes Verfahren, das die Methoden des zweiten Kapitels und die Erkenntnisse aus den Ergebnissen im dritten Kapitel integriert. Das fünfte Kapitel präsentiert einige weitere Experimente mit dem neuen Verfahren sowie einen Vergleich mit unüberwachtem Lernen. Am Schluß wird das Ergebnis der Arbeit zusammengefaßt und ein kleiner Ausblick gegeben.

Kapitel 2

Conceptual-Clustering

In diesem Teil der Arbeit werden Conceptual-Clustering-Techniken aus der Literatur verglichen. Es wird eine Einordnung in das Gebiet des maschinellen Lernens vorgenommen und zusätzlich Kriterien zur Taxonomie der später eingesetzten Verfahren angegeben.

2.1 Statistische Verfahren

Ausgangspunkt für das Conceptual-Clustering sind Methoden der Cluster-Analyse aus der Statistik. Sie werden auch Verfahren zur numerischen Taxonomie genannt. Sämtliche der hier zu diesem Thema zusammengestellten Fakten beziehen sich auf [Everitt, 1980]. Für eine Menge von Beobachtungen wird ein Schema gesucht, nach dem diese in verschiedene Klassen[1] gruppiert werden können, so daß die Beobachtungen innerhalb einer Klasse untereinander auf eine gewisse Weise ähnlich sind und sich von Beobachtungen anderer Klassen unterscheiden. Anders ausgedrückt: das Ziel ist eine Klasseneinteilung, eine „Clusterung“ der Beobachtungen bei großer Ähnlichkeit der Beobachtungen innerhalb einer Klasse (hohe „intra-class-similarity“) und geringer Ähnlichkeit der Beobachtungen über Klassengrenzen hinweg (niedrige „inter-class-similarity“).

Bei dieser Aufgabenstellung ist klar, daß die Verfahren ein Maß für die Ähnlichkeit von Beobachtungen benötigen. Beschrieben werden die Beobachtungen bei den statistischen Verfahren durch eine Anzahl meist numerischer, aber auch nominaler und ordinaler Attribute. Das Maß für die Ähnlichkeit zweier Beobachtungen wird errechnet auf der Basis eines Vergleichs

[1] Im Folgenden auch als Cluster bezeichnet

der einzelnen Attributwerte. Es nimmt Werte zwischen 0 und 1 an, 0 bei völliger Ungleichheit, 1 bei Gleichheit.

Beispiele: Für zwei numerische Werte X_k und Y_k des Attributes k kann man die Ähnlichkeit s_k zum Beispiel durch die relative Differenz der beiden Werte ausdrücken: $s_k = 1 - \frac{|X_k - Y_k|}{ko - ku}$, wenn der Wertebereich von k das Intervall [ku,ko] ist. Bei nominalen Attributen liegt es nahe, für $X_k = Y_k$, $S_k = 1$ zu nehmen und 0 bei Ungleichheit. Andere Ähnlichkeitsmaße für quantitative Attribute sind Korrelationskoeffizienten oder Transformationen von Distanzmaßen. Weiterhin existieren verschiedene Systeme zur Berechnung der Ähnlichkeit binärer Attribute, die auch auf qualitative Attribute erweitert werden können.

Eine einfache Möglichkeit, hieraus die Ähnlichkeit zweier Beobachtungen zu berechnen, besteht darin, die Ähnlichkeiten in den einzelnen Attributen der Beobachtungen aufzusummieren und durch die Anzahl der Attribute zu dividieren:

$$S_{X,Y} = \frac{\sum_k s_k}{k}$$

Darauf aufbauend gibt es verschiedene Techniken, „Clustermethoden“, um die Beobachtungen zu gruppieren. Die meisten Verfahren bilden disjunkte Cluster. Dazu gehören hierarchische Verfahren, die Cluster in einem Baum organisieren, in dem Cluster an einem Knoten des Baumes durch die Nachfolgeknoten in weitere Cluster aufgeteilt sind. Optimierungsverfahren suchen nach einer Partition der Beobachtungen, die ein numerisches Kriterium für die Güte der Clusterung maximieren. Wendet man das Verfahren rekursiv auf die erhaltenen Cluster an, läßt sich auch hiermit eine Hierarchie erstellen. Demgegenüber stehen sogenannte „clumping“ Techniken, bei denen die Cluster überlappen dürfen. Die Wahl zwischen disjunkten oder nicht disjunkten Clustern ergibt sich daraus, was in der Anwendung mehr Sinn macht. Andere Clustermethoden sind Mischformen dieser Techniken.

Da hier nicht alle Verfahren im einzelnen vorgestellt werden können, soll einmal ein hierarchisches Verfahren repräsentativ beschrieben werden:

Die resultierenden Cluster sind in einem binären Baum angeordnet, dessen Wurzelknoten ein Cluster ist, der alle Beobachtungen enthält und in dessen Blättern jeweils die einzelnen Beobachtungen stehen. Bei bottom-up Vorgehensweise, werden die Beobachtungen sukzessive zusammengruppiert. Im ersten Schritt werden die beiden Beobachtungen, die sich am ähnlichsten sind, zu einem Cluster zusammengefaßt. Die Werte hierfür werden einer Ähnlichkeitsmatrix entnommen, in der für alle Kombinationen von zwei Beobachtungen der zugehörige Ähnlichkeitswert eingetragen ist. Nach dem ersten Schritt, muß diese Matrix neu berechnet werden. Die beiden zusammengruppierten Beobachtungen werden durch einen Cluster ersetzt, der die beiden Beobachtungen enthält. Nun wird die Ähnlichkeit des Clusters zu den restlichen Beobachtungen in die neue Matrix eingetragen. An dieser Stelle benötigt man ein Maß für die Ähnlichkeit zwischen einem Cluster und einer einzelnen Beobachtung. Später, wenn mehrere Cluster entstanden sind, muß auch die Ähnlichkeit zwischen zwei Clustern berechnet werden können. In beiden Fällen kann dieselbe Methode angewandt werden. Es gibt wieder verschiedene Möglichkeiten, die einen wesentlichen Aspekt der Clustermethode ausmachen, unter anderem beispielsweise:

1. Die „Nearest-Neighbour"- und die „Furthest-Neighbour"-Methode setzen als inter-Cluster-Ähnlichkeit nur die Ähnlichkeit von zweien der enthaltenen Beobachtungen ein. Beim „Nearest Neighbour" ist das die größte gefundene Ähnlichkeit einer Beobachtung des einen und einer Beobachtung des anderen Clusters, beim „Furthest Neighbour" ist es der kleinste der Ähnlichkeitswerte.

2. Bei der sogenannten „Group-Average-Methode" werden die Ähnlichkeitswerte für alle Paare, die aus jeweils einer Beobachtung des einen und einer Beobachtung des anderen Clusters gebildet werden können, gemittelt. Die Beschreibung eines Clusters besteht bei 1. und 2. in der Aufzählung der enthaltenen Beobachtungen.

3. Im Gegensatz dazu wird bei der „Centroid-Cluster-Analyse" ein Cluster durch die Mittelwerte in den einzelnen Variablen beschrieben. Die Ähnlichkeit von Clustern wird dann genauso berechnet, wie die Ähnlichkeit zwischen zwei Beobachtungen.

In jedem folgenden Schritt wird mit Hilfe der Ähnlichkeitsmatrix die Kombination Beobachtung-Beobachtung, Cluster-Beobachtung oder Cluster-Cluster mit der größten Ähnlichkeit zu einem neuen Cluster zusammengefaßt, und die Matrix wird aktualisiert. Ein neuer Cluster wird zu einem Knoten in der Hierarchie, der auf die Cluster oder Beobachtungen verweist, aus denen er gerade entstanden ist. Es werden soviele Schritte ausgeführt, bis am Ende ein Wurzelknoten alle Beobachtungen enthält.

Zur Verdeutlichung des gesamten Vorgangs der hierarchischen Clusterbildung wird ein Beispiel mit vier Beobachtungen, beschrieben jeweils durch drei nominale Attribute, gegeben:

Beispiel: Die Beobachtungen sind jeweils dargestellt durch ein Tupel (Haar, Haut, Auge), das Farben enthält: e_1 = (braun,dunkel,blau), e_2 = (braun,dunkel,braun), e_3 = (blond,hell,blau) und e_4 = (rot,hell,braun). Zuerst werden die Ähnlichkeitswerte der Beobachtungen untereinander nach der oben angegebenen Formel berechnet und in einer Matrix eingetragen:

$$\begin{pmatrix} & e_1 & e_2 & e_3 & e_4 \\ e_1 & - & 2/3 & 1/3 & 0 \\ e_2 & & - & 0 & 1/3 \\ e_3 & & & - & 1/3 \\ e_4 & & & & - \end{pmatrix}$$

Da e_1 und e_2 den größten Ähnlichkeitswert besitzen, werden diese beiden zusammengefaßt. Damit sehen die Cluster nach dem ersten Schritt wie in Abbildung 2.1 (I) angegeben aus. Im nächsten Schritt wird die Ähnlichkeit

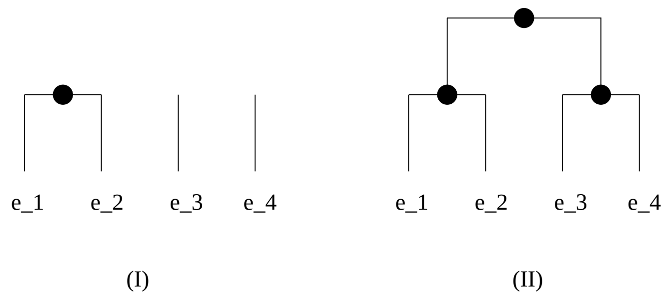


Abbildung 2.1: ein hierarchisches Verfahren

des Clusters zu den beiden anderen Beobachtungen nach der „Furthest-Neighbour“ Methode ausgewählt, die restlichen Werte bleiben stehen:

$$\begin{pmatrix} & e_{1,2} & e_3 & e_4 \\ e_{1,2} & - & 0 & 0 \\ e_3 & & - & 1/3 \\ e_4 & & & - \end{pmatrix}$$

Der größte Eintrag ist jetzt die Ähnlichkeit zwischen e_3 und e_4, die zusammen einen zweiten Cluster bilden, womit die Hierarchie für dieses kleine

Beispiel bereits fertig ist, da im nächsten Schritt nur noch zwei Cluster da sind, die unter der Wurzel zusammengefaßt werden. Abbildung 2.1 (II) zeigt das Ergebnis.

2.2 Verfahren des maschinellen Lernens

Verfahren, die eine Anzahl von Beobachtungen anhand ihrer Beobachtungsbeschreibungen in Klassen einteilen, werden im maschinellen Lernen unter dem Begriff Conceptual-Clustering zusammengefaßt. In dem folgenden Abschnitt werden die Zielsetzung und die Weiterentwicklungen des Conceptual-Clustering gegenüber den statistischen Verfahren herausgestellt. Danach werden ausgewählte Teile eines bekannten Lernverfahrens erläutert. Zum Schluß werden kurz weitere Ansätze zum Conceptual-Clustering skizziert.

Die von einem statistischen Verfahren gebildeten Hierarchien haben im wesentlichen zwei Nachteile:

1. Bei der Bildung der Cluster wird nicht darauf geachtet, welche Attribute der Beobachtungsbeschreibungen relevant sind, beziehungsweise welche im Hinblick auf sinnvolle Cluster eine Bedeutung haben.
2. Es werden keine generellen, sogenannte intensionalen Beschreibungen erzeugt, die alle in einem Cluster enthaltenen Beobachtungsbeschreibungen zusammenfassen im Hinblick auf die den Beobachtungen gemeinsamen Eigenschaften.

Im allgemeinen Fall werden bei den statistischen Verfahren alle Attribute gleichbewertet.[2] Das hat zur Folge, daß aufgrund zufällig gleicher Werte einiger unbedeutender Attribute Cluster aus Beobachtungen gebildet werden können, die sich in einem maßgeblichen Attribut unterscheiden und somit eigentlich nicht zusammengehören. Umgekehrt passiert es, daß Beobachtungen, die in einem interessanten Attribut übereinstimmen, nicht in einem Cluster zusammengefaßt werden, weil sie sich in mehreren unbedeutenden Attributen unterscheiden. Dabei ist aber das Wissen, nach welchen Attributen eine sinnvolle Einteilung der Beobachtungen in Cluster möglich ist, ein Teil des Lernzieles. Sinnvolle Cluster sind solche, die von einem Anwender interpretierbar sind, gegebenenfalls mit einem bestimmten Ziel.

[2] Ausnahme: Einige statistische Verfahren erlauben dem Benutzer, eine Gewichtung der einzelnen Attribute einzugeben. Dieses kann zum Beispiel auch aufgrund einer vorhergehenden „principal components analysis“ geschehen; diese wird mittlerweile sogar recht häufig eingesetzt.

Das erklärt auch gleichzeitig den zweiten Nachteil der statistischen Verfahren: Um interpretierbar zu sein, muß ein Cluster eine für den Anwender „lesbare“ Beschreibung besitzen. Bei den statistischen Verfahren sind dagegen die Cluster extensional beschrieben, das heißt durch Aufzählung der Beschreibungen aller enthaltenen Beobachtungen.[3] Eine generelle Beschreibung würde etwa in der Nennung der maßgeblichen, beziehungsweise der einigenden Eigenschaften der Beobachtungen eines Clusters bestehen. Damit wäre der Anwender imstande, den Cluster unter einem ihm verständlichen Oberbegriff zusammenzufassen (zum Beispiel "Cluster X ist der, dessen Beobachtungen alle die Eigenschaft Y haben“).

Zu den bisher genannten Nachteilen statistischer Verfahren kommt noch ein weiterer hinzu. Wie im vorigen Abschnitt angedeutet, gibt es eine Vielzahl verschiedener Ähnlichkeitsmaße. Das Ergebnis der Cluster-Analyse ist nun stark abängig von der Art der gewählten Ähnlichkeitsmaße und der Clustermethode. Bei der Benutzung verschiedener dieser Parameter gibt es in den resultierenden Clusterhierarchien zum Teil große Abweichungen. Eine Auswahl nach dem Kriterium, welche der so erhaltenen Cluster einen Sinn machen, bleibt dem Anwender überlassen, mit den unter Punkt zwei angeführten Schwierigkeiten bei der Interpretation der Cluster.

2.2.1 Lernen aus Beobachtungen

Aus den Problemen der statistischen Verfahren läßt sich direkt die Zielsetzung des Conceptual-Clustering ableiten. Ursprünglich definiert worden ist der Begriff Conceptual-Clustering ("konzeptuelle Klassenbildung“) in [Michalski/Stepp, 1980]. Gegeben sei eine Menge von Beobachtungsbeschreibungen, finde:

1. eine hierarchische Klasseneinteilung der Beobachtungen (**Aggregation**),
2. Konzepte in der Form intensionaler Beschreibungen der Klassen (**Charakterisierung**).

Die beiden Schritte des Conceptual-Clustering sind voneinander abhängig. Für eine aussagekräftige Konzeptbildung ist es notwendig, eine passende Klasseneinteilung gefunden zu haben. Die Qualität der Klasseneinteilung wiederum läßt sich an der Brauchbarkeit der Beschreibungen messen, die

[3] Nicht so bei der „Centroid-Cluster-Analyse“.

für diese spezielle Klasseneinteilung gefunden werden können. Gesucht wird also nach Klassen mit guten konzeptuellen Beschreibungen.

Motiviert ist diese Definition auch durch Erkenntnisse über das menschliche Lernen. Menschen versuchen Beobachtungen aus der Umwelt in Klassen einzuteilen. Das geschieht zunächst intuitiv, ohne daß ein bestimmtes Ziel dahintersteht. Durch die Klassenbildung wird das aufgenommene „Umweltwissen" organisiert. Es werden dann nicht mehr die einzelnen Beobachtungen gespeichert, sondern jeweils die besonderen Eigenschaften der Klassen zusammen mit dem Wissen, welche der Beobachtungen von welcher Klasse abgedeckt werden. Somit geschieht eine Kompression der Wahrnehmungen, wobei eine kompakte Darstellung gewonnen wird. Das läßt sich leicht nachvollziehen, wenn man bedenkt, welch ein unmöglicher Aufwand es wäre, sich jede Beobachtung als einzelne Einheit zu merken. Ebenso unmöglich ist es, jeder Beobachtung einen eigenen Namen zu geben, eine solche Vielzahl an Worten ist für eine Sprache zur menschlichen Kommunikation nicht denkbar. Damit ist die Notwendigkeit zur Klassen- und Konzeptbildung im menschlichen Bereich schon für Sprache und Erinnerung gezeigt. Weiterhin sind laut [Smith/Medin, 1981] auch Wahrnehmung und Denken davon abhängig, welche Konzepte dem Menschen zur Verfügung stehen. Die einzelnen gelernten Klassen werden in einer Hierarchie angeordnet, die auf unteren Ebenen mehr, auf oberen Ebenen weniger spezifische Konzepte beschreibt.

Ein spezieller Lerneffekt kann beobachtet werden, wenn eine Aufgabe gelöst werden soll. Die Klassifikation dient in einer Einsatzphase als Basis für Vorhersagen über bestimmte Eigenschaften. So kann man zum Beispiel über das Objekt einer Beobachtung, das aufgrund der Eigenschaften „hat Flossen" und „hat Kiemen" in die Klasse „Fische" eingeordnet wurde, schließen, daß es im Wasser lebt. Versuche zeigen, daß Menschen aus den Eigenschaften der Wahrnehmungen nur einige wenige auswählen, um auf der Basis von Ähnlichkeiten in den Werten dieser Attribute Klassen zu bilden. Die Wahl, welche der Attribute zur Begriffsbildung herangezogen werden, ist auch gesteuert durch die Brauchbarkeit für eine spätere Anwendung, also durch die Frage, ob aufgrund dieser Attribute auf für die Anwendung interessante Eigenschaften zukünftiger Beobachtungen geschlossen werden kann.

Im Gegensatz zum Lernen aus unklassifizierten Beobachtungen gibt es noch die Möglichkeit, daß ein Lehrer zu einer Menge von Beobachtungsbeschreibungen die Klassenzugehörigkeit der Beobachtungen angibt. Da hier von außen in den Lernvorgang eingegriffen wird, bezeichnet man dies auch als überwachtes Lernen im Gegensatz zum unüberwachten Lernen im ersteren Fall. Der Lernschritt besteht in der Suche nach Charakterisierungen der vorgegebenen Klassen. Die klassifizierten Beobachtungen sind für die Klassen Beispiele, die eine Charakterisierung induzieren sollen. Eine Charakterisierung einer Klasse muß die zu ihr gehörenden Beispielinstanzen abdecken. Das Erstellen einer Charakterisierung wird als Suche im Raum aller innerhalb einer Beschreibungssprache bildbaren Charakterisierungen aufgefaßt. [Mitchell, 1982] hat dieses unter dem Begriff „Lernen im Version Space“ beschrieben. Dieser Raum ist geordnet bezüglich einer genereller-als Relation. Die Suche nach einer geeigneten Charakterisierung kann auf drei verschiedene Arten durchgeführt werden: 1. Man startet mit ganz speziellen Charakterisierungen, zum Beispiel mit einzelnen Beispielbeschreibungen und generalisiert diese, bis alle Beispiele abgedeckt werden. 2. Man startet mit einer sehr generellen Charakterisierung und spezialisiert diese, bis sie gegenüber den Beispielen für andere Klassen hinreichend diskriminiert. 3. Man startet irgendwo in der Mitte und sucht in beide Richtungen.

Auf diesem Hintergrund wird das Conceptual-Clustering als zweifache Suche angesehen. Die Suche nach einer geeigneten Klassenbeschreibung entspricht dabei dem Lernen aus Beispielen. Die Klassenbildung, also das was sonst beim Lernen aus Beispielen von einem Lehrer vorgegeben wird, ist eine Suche im Raum aller Möglichkeiten, die vorhandenen Beobachtungen in Klassen einzuteilen. Beim Conceptual-Clustering ist das aber kein reiner Vorverarbeitungsschritt, sondern Aggregations- und Charakterisierungsschritte sind miteinander verzahnt.

Die nun folgenden Teile behandeln exemplarisch zwei Verfahren, die zu diesem Thema entwickelt wurden. Beide Verfahren sind propositional, das heißt, daß Cluster und Beobachtungen in einer Attribut-Werte Repräsentation beschrieben werden.

2.2.2 CLUSTER/*Star*

CLUSTER [Michalski/Stepp, 1980] bildet aus einer Menge von Beobachtungen eine Hierarchie disjunkter Konzepte: Auf jeder Ebene der Hierarchie wird jede Beobachtung jeweils von nur einer der Klassen abgedeckt. Durch die Wahl einer Attribut-Werte Repräsentation mit nominalen Wertebereichen der Attribute wird ein endlicher „Ereignisraum“ von möglichen Beobachtungsbeschreibungen aufgespannt. Die Beobachtungen sind wahrgenommene Objekte (Ereignisse) des Ereignisraumes. Gesteuert wird der Prozeß des Conceptual-Clustering durch eine Bewertungsfunktion. Diese bevorzugt einerseits eine überschaubare Klasseneinteilung, um der Forderung nach Lesbarkeit für den Anwender gerecht zu werden; ein Maß hierfür ist die Anzahl der Klassen, die möglichst gering sein sollte. Andererseits ist die Qualität der Klasseneinteilung von Interesse, die daran gemessen wird, wie genau die Klassen die Objekte repräsentieren, indem sie im Verhältnis zur Zahl abgedeckter Beobachtungen möglichst wenige unbeobachtete Objekte des Ereignisraumes abdecken.[4] Die Hierarchiebildung geschieht top-down: angefangen mit dem gesamten Ereignisraum wird dieser in Cluster aufgeteilt. Das Clusterverfahren wird dann rekursiv auf die einzelnen Cluster angewandt, solange sich die Qualität durch weitere Verfeinerung der Klassen verbessern läßt.

Die Clusterbildung an einem Knoten der Hierarchie funktioniert in einer iterativen Schleife:

1. Es werden k Ausgangsbeobachtungen beliebig gewählt. Diese bilden k Ausgangscluster für die nachfolgenden Schritte. Der Parameter k bestimmt damit auch die Anzahl der Cluster im Ergebnis. Da diese nicht bekannt ist, wird das Clusterverfahren an jedem Knoten mit verschiedenen Werten für k gestartet. Die Auswahl unter den Ergebnissen für verschiedene k 's wird mit Hilfe der Bewertungsfunktion vorgenommen, die einen Tradeoff zwischen Überschaubarkeit und Klassenqualität darstellt, da eine große Anzahl von Clustern die Qualität der Klasseneinteilung begünstigt, andererseits aber der Überschaubarkeit abträglich ist. Deswegen reicht es aus, nur wenige, kleine Werte von k zu versuchen.

2. *Star*-Bildung: Jede der Ausgangsbeobachtungen[5] wird soweit wie möglich generalisiert, so daß jeweils nur die restlichen $k-1$ Ausgangsbeobachtungen ausgeschlossen werden. Die so entstandenen Cluster werden *Stars* genannt.

Abbildung 2.2: *Stars*

Beispiel: In Abbildung 2.2 wird der Ereignisraum durch zwei Variablen mit den Wertebereichen W(V_1) = (a,b,c) und W(V_2) = (h,i) aufgespannt.

[4] Dieses ist kein allgemein gültiges Kriterium. In [Michalski/Stepp, 1980] werden einige weitere vorgeschlagen.

[5] genauer gesagt: die Beschreibung der Ausgangsbeobachtungen

Drei Beobachtungen sind eingetragen, die Buchstaben in den Klammern stellen die Werte der Variablen V_1 und V_2 dar: e_1 = (a,i), e_2 = (b,h) und e_3 = (c,h). Seien beispielsweise e_1 und e_3 (jeweils gekennzeichnet durch ein x) als Ausgangsbeobachtungen ausgewählt worden, so werden zwei *Stars* gebildet. Die *Stars* von e_1 (linkes Bild) und e_3 (rechtes Bild) bestehen dann aus der Vereinigung zweier Rechtecke (eines davon ist hier ein Viereck).

Jeder *Star* deckt die Ausgangsbeobachtung ab, aus der er entstanden ist, und gegebenenfalls einige weitere Beobachtungen und unbeobachtete Ereignisse des Ereignisraumes. Damit sind die *Stars* bezüglich der Ausgangsbeobachtungen disjunkt, andere Beobachtungen und unbeobachtete Ereignisse können von mehreren *Stars* abgedeckt sein. Da die *Stars* maximal generell sind, decken sie viele unbeobachtete Ereignisse ab. Um deren Anahl zu verringern, werden die *Stars* eingeschränkt zu „*Reduced-Stars*“: Die *Stars* werden soweit wie möglich spezialisiert, so daß aber noch alle von einem ursprünglichen *Star* abgedeckten Beobachtungen weiterhin abgedeckt werden.

Abbildung 2.3: *Reduced-Stars*

Beispiel: Die *Reduced-Stars* von e_1 und e_3 sind in Abbildung 2.3 dargestellt. Das kleine Rechteck des *Reduced-Stars* von e_1 in dem linken Bild ist aus der Spezialisierung des Rechteckes der vorherigen Abbildung enstanden. Das andere Rechteck ist unverändert geblieben. Die beiden Rechtecke des *Stars* von e_3 sind beide zu demselben kleineren Rechteck in dem rechten Bild spezialisiert worden.

Von den *Reduced-Stars* werden weiterhin auch unbeobachtete Ereignisse abgedeckt. Der genaue Vorgang der Bildung von *Reduced-Stars* wird später exemplarisch vorgeführt werden.

3. Konstruktion einer disjunkten Abdeckung: Beobachtungen, die nicht zu den ausgewählten Ausgangsbeobachtungen gehören, können von mehreren der *Reduced-Stars* abgedeckt sein. In diesem Schritt werden die *Reduced-*

Stars weiter spezialisiert, so daß sie sich bei den Beobachtungen nicht mehr überschneiden. Die Spezialisierung geschieht auf zwei Ebenen:

a) Die Beschreibung der *Reduced-Stars* besteht aus einer Disjunktion von Teilbeschreibungen (Die Teilbeschreibungen entsprechen den Rechtecken im Beispiel). Aus jedem der *Reduced-Stars* wird jeweils nur ein Disjunkt als Beschreibung für diesen Cluster genommen. Die ausgewählten Disjunkte bilden zusammen eine Clusterung. Es sind nur Clusterungen erlaubt, die alle Beobachtungen abdecken.

b) Eine so erhaltene Clusterung wird disjunkt gemacht (wenn sie es nicht schon ist), indem einzelne Cluster weiter eingeschränkt werden.

Auf Stufe a) werden alle möglichen Kombinationen von Teilbeschreibungen erzeugt. Nachdem diese auf Stufe b) disjunkt gemacht worden sind, wird die beste Clusterung gemäß der Qualität der Klasseneinteilung ausgewählt. Auch auf Stufe b) gibt es eine Vielzahl von Möglichkeiten, die Cluster weiter einzuschränken. Um die Komplexität des Verfahrens zu begrenzen, bietet es sich an, an dieser Stelle nur eine Einschränkung zu generieren. In [Michalski/Stepp, 1980] wird eine Prozedur (NID) benutzt, die mehrere Einschränkungen probiert und mit Hilfe der Bewertungsfunktion ordnet, wodurch allerdings ein exponentieller Aufwand verursacht wird. Ergebnis von Schritt drei ist somit eine bezüglich aller Beobachtungen disjunkte Clusterung mit einem zugehörigen Wert der Klassenqualität für die Bewertungsfunktion.

4. Evaluierung der Clusterung: Die nach 3. erhaltene Clusterung wird gespeichert, wenn sie besser als die bisher Erzeugten ist, beziehungsweise wenn es die erste Iteration ist. Danach wird ein Abbruchkriterium abgefragt. Das Abbruchkriterium prüft, ob eine Mindestanzahl an Iterationen erreicht ist. Wenn ja, wird gezählt, wieviele Iterationen seit der letzten Verbesserung der Clusterung durchlaufen wurden. Ist hier ein Schwellwert überschritten, wird die bisher beste Clusterung als Ergebnis ausgegeben. Ansonsten wird mit neuen Ausgangsbeobachtungen bei Schritt 2. wieder eingesetzt. Die Wahl der neuen Ausgangsbeobachtungen geschieht nicht mehr rein zufällig. Es werden die Cluster der letzten Clusterung zugrundegelegt. Solange sich die Clusterungen aufeinanderfolgender Iterationen verbessern, wird jeweils eine Beobachtung aus dem Zentrum eines jeden Clusters genommen, das heißt, Beobachtungen, die nahe am errechneten geometrischen Mittel der Cluster liegen. Damit werden erstmal charakteristische Eigenschaften der Cluster gelernt. Um eine bessere Abgrenzung gegenüber anderen Clustern zu erreichen,

werden Beobachtungen am Rande der Cluster als Ausgangsbeobachtungen gewählt, wenn sich die Qualität der Clusterung im letzten Iterationsschritt nicht weiter verbessert hat.

Beispiel: Einige Details des Verfahrens werden an einem größeren Beispiel veranschaulicht. Es sollen Menschen nach ihren Äußerlichkeiten, den Farben von Haar, Haut und Auge, in verschiedene Klassen eingeteilt werden. Die Haarfarbe kann braun (br), blond (bl) oder rot (ro) sein. Die Hautfarbe ist hell (h) oder dunkel (d) und die Augen sind entweder blau (ba) oder braun (br). Vier Beobachtungen, dargestellt durch ein Tupel (Haar, Haut, Auge) sind gemacht worden: e_1 = (br,d,ba), e_2 = (br,d,br), e_3 = (bl,h,ba) und e_4 = (ro,h,br). In Abbildung 2.4 ist der Ereignisraum mit den Beobachtungen graphisch dargestellt.

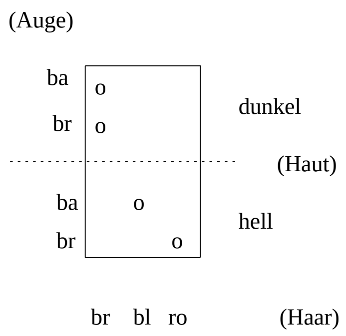


Abbildung 2.4: CLUSTER - Bsp.

Gezeigt werden soll hier schrittweise nur die oberste Stufe der Hierarchiebildung, bei der geringen Zahl an Beobachtungen dürfte eine feinere Einteilung auch nicht sinnvoll sein.

In Schritt 1 werden e_1 und e_4 als Ausgangsbeobachtungen gewählt. Eine andere Wahl anstelle von $k = 2$ soll, wieder im Hinblick auf die wenigen Beobachtungen, nicht betrachtet werden.

Schritt 2 ist die *Star*-Bildung. Der *Star* von e_1, notiert als G(e_1), entspricht in diesem Beispiel einer Menge von Disjunkten, die e_4 ausschließen, indem jeweils für eine der drei Variablen der Wert, den e_4 an dieser Stelle hat, ausgeschlossen wird:[6]

[6] Bei mehr als zwei Ausgangsbeobachtungen müßte dieses zur Berechnung von G(e_1) für alle $k-1$ Ausgangsbeobachtungen gemacht werden. Darüber würde dann das logische Produkt gebildet.

$G(e_1) = (br \vee bl, Y, Z) \vee (X, d, Z) \vee (X, Y, ba)$. Entsprechend ist $G(e_4) = (bl \vee ro, Y, Z) \vee (X, h, Z) \vee (X, Y, br)$.
Abbildung 2.5 zeigt dies graphisch. Die Spezialisierung zu *Reduced-Stars*

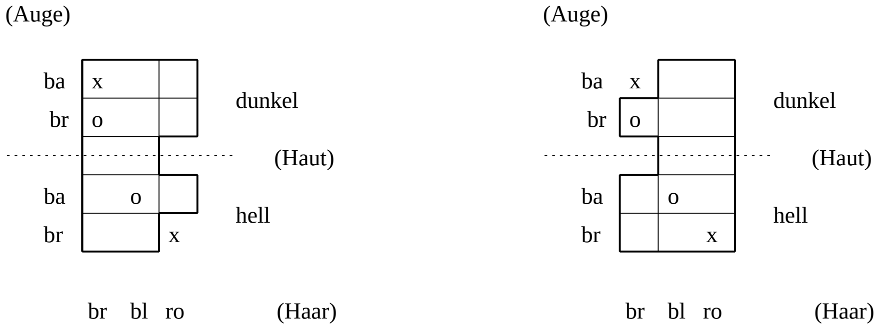


Abbildung 2.5: CLUSTER - *Stars*

geschieht, indem zu jeder Einschränkung einer der drei Variablen die beiden anderen Variablen eingeschränkt werden. Gegeben die erlaubten Werte der bereits eingeschränkten Variablen, werden an Stelle der restlichen Variablen nur Werte eingetragen, die die dann noch abgedeckten Beobachtungen haben. Der *Reduced-Star* zu e_1, notiert als RG(e_1), wird gegeben durch
$RG(e_1) = (br \vee bl, h \vee d, ba \vee br) \vee (br, d, ba \vee br) \vee (br \vee bl, h \vee d, ba)$,
derjenige zu e_4 durch
$RG(e_4) = (bl \vee ro, h, ba \vee br) \vee (br \vee ro, h \vee d, br)$,
wobei zwei Disjunkte gleich sind. Abbildung 2.6 zeigt dies wiederum graphisch.

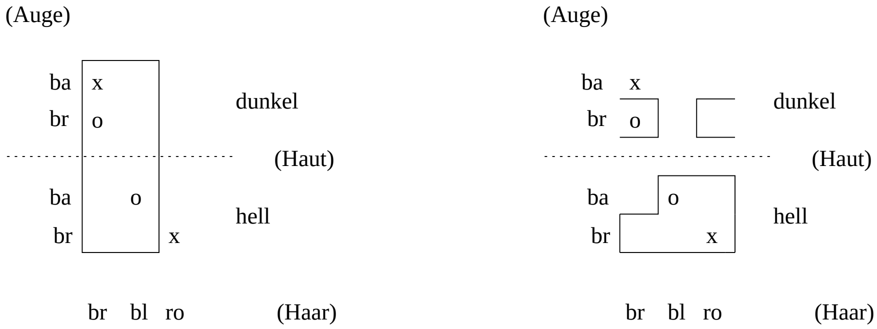


Abbildung 2.6: CLUSTER - *Reduced-Stars*

Zur Konstruktion der disjunkten Abdeckung in Schritt 3 werden die Disjunkte aus den zwei *Reduced-Stars* miteinander kombiniert. Insgesamt gibt es sechs Möglichkeiten, von denen zwei entfallen, da sie e_2 beziehungsweise e_3 nicht abdecken. Von den restlichen vier Kombinationen brauchen zwei nicht weiter eingeschränkt zu werden, da sie bereits disjunkte Abdeckungen darstellen. Es sind dies die Clusterungen

I $(br, d, ba \vee br)$, der Cluster um e_1 und $(bl \vee ro, h, ba \vee br)$, der Cluster um e_4, sowie

II $(br \vee bl, h \vee d, ba)$, als Cluster um e_1 und $(br \vee ro, h \vee d, br)$ als Cluster um e_4.

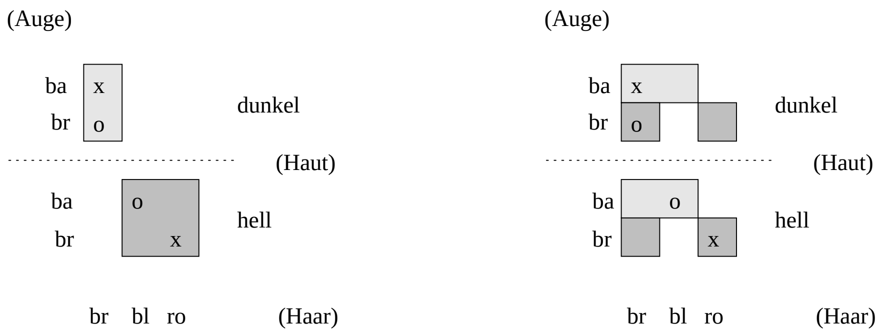


Abbildung 2.7: CLUSTER - Ergebnisse

Die Abbildung 2.7 zeigt links die Clusterung I und rechts Clusterung II. Für beide Clusterungen wird jetzt die Qualität der Klasseneinteilung berechnet.

Sie ergibt sich als Summe der Bewertungen der einzelnen Klassen, welche durch die relative Anzahl abgedeckter unbeobachteter Objekte gegeben werden. Ein geringer Wert dieser Summe sagt aus, daß die Klassen sehr speziell auf die Beobachtungen passen. Die Bewertung für den Cluster um e_1 der Clusterung I ist

$$\frac{\textit{Gesamtzahl abgedeckter Objekte} - \textit{Zahl abgedeckter Beobachtungen}}{\textit{Gesamtzahl abgedeckter Objekte}}$$

$= \frac{2-2}{2} = 0$. Der Cluster um e_4 hat die Bewertung $\frac{4-2}{4} = 1/2$, womit sich für die Clusterung I insgesamt ein Wert von $1/2$ ergibt. Die Clusterung II erhält eine Bewertung von $1/2 + 1/2 = 2$, sie ist also schlechter. Theoretisch müßten jetzt noch die anderen zwei Kombinationen zu einer disjunkten

Abdeckung eingeschränkt und anhand ihrer Klassenqualität mit den beiden vorigen verglichen werden.

Schritt 4: Es soll nur eine Interation gezeigt werden. Demnach wird hier abgebrochen. Damit ist eine Clusterung gegeben, die die Menschen anhand gleicher Werte bei der Haar- und Hautfarbe klassifiziert. Bei Clusterung II wären die Menschen dagegen hauptsächlich aufgrund ihrer Augenfarbe unterschieden worden.

Zusammenfassend soll noch einmal hervorgehoben werden, wie die beiden Ziele, Aggregation und Charakterisierung, erreicht werden. Die Einteilung der Beobachtungen in Klassen geschieht durch Generalisierung und Spezialisierung der Beschreibungen einiger Ausgangsbeobachtungen. Dadurch werden die restlichen Beobachtungen von einer der entstehenden Klassen automatisch abgedeckt. Die Auswahl unter mehreren Alternativen, die sich aus verschiedenen Spezialisierungen ergeben, ist stark abhängig von der gewählten Qualitätsfunktion. Die Charakterisierung wird gleichzeitig mit Generalisierung und Spezialisierung erreicht, da die *Star*-Bildung und die Bildung von *Reduced-Stars*, sowie deren weitere Einschränkung, Prozeduren sind, die direkt auf den Beschreibungen arbeiten.

2.2.3 Inkrementelle Verfahren

Eine Variante des Conceptual-Clustering ist die Aufgabe des „Concept Formation". Die Ziele sind dieselben, wie die unter der Überschrift „Lernen aus Beobachtungen definierten", mit dem Zusatz, daß die Konzeptbildung inkrementell geschieht. Das heißt, daß die Beobachtungen nacheinander verarbeitet werden und nicht alle von Beginn an dem Verfahren bekannt sind. Zukünftige Beobachtungen können erst zum Zeitpunkt ihres Auftretens zum Lernen benutzt werden, alte Beobachtungen werden im allgemeinen nicht zu einem späteren Zeitpunkt neu verarbeitet.

Das ist näher an der Realität des menschlichen Lernens, da Menschen aus Beobachtungen lernen, die in einer zeitlichen Abfolge von Interaktionen mit der Umwelt auftreten. Der inkrementelle Charakter bedingt auch, daß neue Konzepte auf Basis der bisher bekannten Konzepte entwickelt werden, als Verfeinerungen oder Abgrenzungen gegenüber den alten. Das Lernen wird dazu mit der Anwendung der gelernten Konzepte verknüpft.

Bei den hierarchischen Verfahren wird die bereits aufgebaute Begriffshierarchie beim Lernen eingesetzt, indem versucht wird, neue Beobachtungen aufgrund der bisher gelernten Begriffe zu klassifizieren. Während der Klassifikation der neuen Beobachtung wird die Begriffshierarchie modifiziert, damit die Beobachtung so eingeordnet werden kann, daß ein ausreichendes Maß an Klassenqualität erhalten bleibt.

Bekannte hierarchische Verfahren sind [Gennari et al., 1989] EPAM, UNIMEM und COBWEB. UNIMEM [Lebowitz, 1987] wird hier kurz vorgestellt, da es erlaubt darzustellen, wie die Methoden des Conceptual-Clustering bei einem inkrementellen Verfahren einfach realisiert sind. Die von UNIMEM gebildeten Klassen sind nicht disjunkt. Vielmehr kann es vorkommen, daß Beobachtungen an mehreren Stellen der Hierarchie eingeordnet werden. Konzeptbeschreibungen sind jeweils eine Liste von Attribut-Wert-Paaren, in der diejenigen Attribute vorkommen, in denen die zu der Klasse gehörenden Beobachtungen übereinstimmen. Eine Beobachtung wird vom Wurzelknoten aus den möglichst speziellsten Knoten in dem Baum zugeordnet. Je niedriger ein Konzept in der Hierarchie liegt, umso spezieller ist es, - es sind mehr Attributwerte festgelegt, da die Beschreibungen von übergeordneten Konzepten geerbt werden.

In UNIMEM dient die Prozedur SEARCH(Knoten, Attribute) zur Einordnung einer Beobachtung in den Hierarchiebaum. Sie wird zuerst mit dem Wurzelknoten aufgerufen:

1. Vergleiche die Attribute, deren Werte in dem Knoten vorkommen, mit den entsprechenden Attributwerten der Beobachtung. Unterscheiden sich die Werte zu sehr, wird beim Nachbarknoten weitergemacht: Ergebnis ist dann die leere Liste. (Ausnahme ist der Wurzelknoten; da er keine Attributwerte enthält, wird direkt zu Schritt 2. gesprungen.)

2. Die Attribute die in dem Knoten von Schritt 1 vorkommen werden zu festgelegten Attributen der Beobachtung. Es wird rekursiv bei allen Folgeknoten weitergesucht, indem die dort angegebenen Attributwerte mit den restlichen der Beobachtung verglichen werden: Es wird SEARCH(Folgeknoten, noch nicht festgelegte Attribute) für alle Folgeknoten aufgerufen.

3. Wenn SEARCH für die Folgeknoten aus 2. eine Liste von passenden Knoten erhält, ist das Ergebnis die Vereinigung dieser Listen.

4. Wenn SEARCH für alle Folgeknoten aus 2. die leere Liste erhält, wird die Beobachtung dem gerade aktuellen Knoten zu-

geordnet: Das Ergebnis von SEARCH(Knoten, Attribute) ist der Aufrufknoten selbst.

Die Beobachtung muß in die Knoten, die mit SEARCH gefunden wurden, eingeordnet werden:

1. Definiere *neue-Attribute* als die restlichen Attribute der Beobachtung, die nicht durch den Knoten oder die Vorgängerknoten festgelegt sind.

2. Vergleiche die Werte von *neue-Attribute* der Beobachtung mit denen aller unter dem Knoten gespeicherten Beobachtungen. Unterscheiden sich die Werte zu sehr, wird die Beobachtung unter dem Knoten gespeichert.

3. Ansonsten wird für jede Beobachtung, die hinreichend viele Attributwerte mit der neuen Beobachtung gemeinsam hat, ein neuer Knoten gebildet. Der Knoten erhält als Beschreibung die Attribute, in denen die beiden Beobachtungen übereinstimmen. Der neue Knoten wird ein Folgeknoten des aktuellen Knotens; unter ihm werden die beiden Beobachtungen neu gespeichert.

Damit werden in UNIMEM neue Konzepte auf der Basis von nur zwei Beobachtungen gebildet. Das führt zu sehr speziellen Konzepten. Um die Konzepte weiterhin zu generalisieren, werden im weiteren Verlauf die beschreibenden Attribute bewertet, um so diejenigen, die das Konzept zu speziell machen, löschen zu können.

Die folgende Bewertungsfunktion wird vor dem ersten Schritt der Einordnungsprozedur eingefügt. Dadurch werden genau die Beschreibungen der für die neue Beobachtung potentiell relevanten Knoten bewertet.

Bewertungsfunktion vor Schritt 1. von SEARCH: Vergleiche die Attribute, deren Werte in dem Knoten vorkommen, mit den entsprechenden Attibutwerten der Beobachtung. Haben der Knoten und die Beobachtung in einem Attribut den gleichen Wert, wird die Verläßlichkeit des Attributes bei dem Knoten erhöht, ansonsten wird sie gesenkt. Fällt dabei die Verläßlichkeit eines Attributes unter einen Schwellwert, wird das Attribut aus der Beschreibung des Knotens gelöscht. Steigt dagegen die Verläßlichkeit eines Attributes über einen Schwellwert, so wird das Attribut gesichert und kann nicht mehr gelöscht werden.

Knoten, die nur wenige, anfängliche Beobachtungen repräsentieren, werden wieder gelöscht, wenn ihre Beschreibungen nicht mehr genügend Attribute enthalten und damit zu generell geworden sind.

Um einen Vergleich mit CLUSTER zu haben, wird das dort benutzte Beispiel hier einmal aufgegriffen. Dazu werden nur die Ergebnisse der Clusterung und nicht die Bewertungsschritte gezeigt, auch weil in dem Beispiel zu wenige Beobachtungen vorkommen, um damit alle Eigenschaften von UNIMEM zu erklären.

Beispiel: Gegeben sind wieder die vier Beobachtungen aus dem letzten Beispiel: e_1 = (br,d,ba), e_2 = (br,d,br), e_3 = (bl,h,ba) und e_4 = (ro,h,br). Sie werden in der Reihenfolge ihrer Numerierung eingegeben. Die Hierarchie

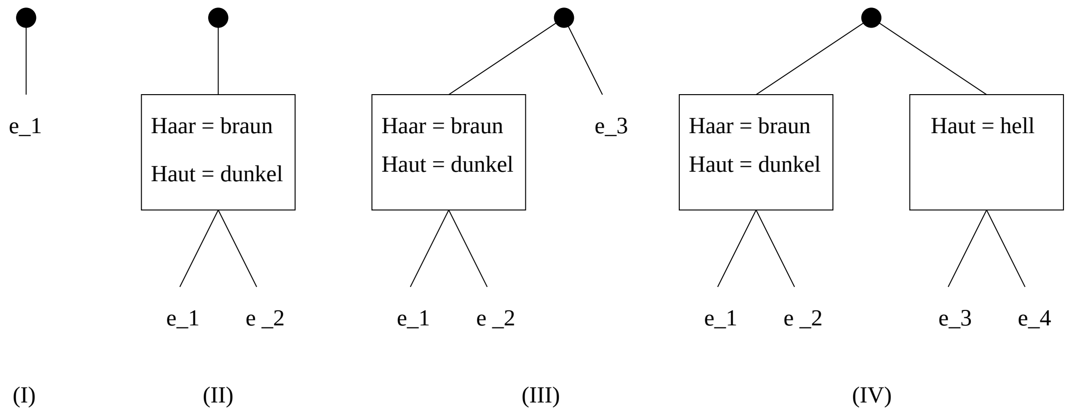


Abbildung 2.8: UNIMEM - Clusterungen

wie sie jeweils nach Bearbeitung der Beobachtung aussieht ist in Abbildung 2.8 gezeigt. Die Beobachtungen e_1 (I) und danach e_2 werden mit der Prozedur SEARCH erst einmal dem Wurzelknoten zugeordnet. Dann wird in der Einordnungsprozedur e_2 mit e_1 verglichen und festgestellt, daß beide gleiche Haut- und Haarfarben besitzen. Mit diesen Attributen wird ein Knoten erzeugt, der unter dem Wurzelknoten angesiedelt wird (II). Danach wird e_3 der Prozedur SEARCH übergeben. Diese versucht e_3 dem neuen Knoten zuzuordnen. Da aber weder die Haut-, noch die Haarfarbe mit der Knotenbeschreibung übereinstimmt, wird e_3 unter dem Wurzelknoten gespeichert (III). Dasselbe gilt für e_4. Nun kann die Einordnungsprozedur wieder versuchen, die zwei Beobachtungen zu einem neuen Knoten unter dem Wurzelknoten zu generalisieren. Die beiden haben die helle Hautfarbe gemeinsam.

Bei entsprechender Parametersetzung könnte dies ausreichen um insgesamt eine Clusterung, wie in (IV) gezeigt, zu erhalten.

Auch für UNIMEM sei hier eine Zusammenfassung der wesentlichen Aspekte im Hinblick auf die Ziele des Conceptual-Clustering gegeben. Der Aggregationsschritt, die Einordnung der Beobachtungen in die Knoten der Hierarchie, ist die Grundlage des Verfahrens. Hierbei wird ein Ähnlichkeitsmaß eingesetzt: In Schritt 1. der Prozedur SEARCH wird die Summe der Distanzen der Attributwerte errechnet. Die Distanzen zwischen einzelnen Attributwerten können je nach dem Typ des Attributes errechnet werden. Ein Parameter legt den Schwellwert fest, unterhalb dessen eine Beobachtung einem Knoten hinreichend ähnlich ist. Die Charakterisierung der Klassen geschieht auf Basis der gefundenen Ähnlichkeiten. Es wird generalisiert, indem nur die jeweils gleichen Attribute zweier Beobachtungen eines Knotens in die Beschreibung einer neuen Klasse aufgenommen werden. Beim Einordnen weiterer Beobachtungen wird weiter generalisiert. Es werden solche Attribute aus der Klassenbeschreibung herausgenommen, die nicht bei genügend Beobachtungen für die die Klasse relevant ist[7], gleich sind (so daß die Verläßlichkeit unter die Schranke fällt).

UNIMEM geht also von den Beobachtungen aus, die sukzessive generalisiert werden, um Konzepte zu erhalten. Im Gegensatz dazu geht CLUSTER von Konzepten aus, die den Beobachtungen angepaßt werden.

In diesem Kapitel wurden grundlegende Mechanismen des Conceptual-Clustering anhand seiner Ursprünge in den Verfahren zur numerischen Taxonomie und anhand zweier bekannter Standardverfahren offengelegt. Neuere Entwicklungen auf diesem Gebiet befassen sich mit der Einbindung von Hintergrundwissen in das Lernen und mit der Erweiterung der Repräsentationssprache auf eingeschränkte Prädikatenlogik erster Stufe. Einen Schritt in diese Richtung leistet das Verfahren KBG, das im nächsten Kapitel für die Roboter-Anwendung eingesetzt wird.

[7] Die von der Prozedur SEARCH beim Einorden einer Beobachtung besuchten Knoten sind relevant

Kapitel 3

KBG

Die im zweiten Kapitel skizzierten Conceptual-Clustering-Verfahren können in der im ersten Kapitel beschriebenen Anwendung, das unüberwachte Lernen von Sensor-Merkmalen aus Basis-Merkmal-Sequenzen, nicht eingesetzt werden. Die Ursache dafür liegt in der Repräsentationssprache, die es im Falle von CLUSTER und UNIMEM nicht ermöglicht, Basis-Merkmale über ihre (Zeit-)Argumente zu verketten.

Erst mit einer relationalen Repräsentationssprache läßt sich dieses modellieren. In der Repräsentationshierarchie des Roboters wird hierzu Hornklausellogik verwendet.[1] Da KBG („Knowledge Based Generalizer") in [Bisson, 1992a] unter dem Aspekt des „Conceptual-Clustering in Prädikatenlogik erster Stufe" eingeführt wird, liegt es nahe, hiermit die Basis-Merkmal-Sequenzen in Cluster zu gruppieren. In KBG werden Beobachtungen und Konzepte in einer auf die Berechnung eines Ähnlichkeitsmaßes abgestimmten Form der Einschränkung von Prädikatenlogik erster Stufe repräsentiert.

Im zweiten Teil dieses Kapitels wird gezeigt, wie Basis-Merkmal-Sequenzen in KBG repräsentiert werden können. Danach folgt die Darstellung der praktischen Anwendung auf die Daten. Insgesamt werden zwei verschiedene Repräsentationen und die damit erzielten Ergebnisse verglichen. Die erste Repräsentation modelliert die Gleichheitsrelation über die Zeitargumente in den Basis-Merkmalen. Die zweite Repräsentation führt zusätzlich die „Kleiner-als"-Relation über die Zeitargumente ein. Zuerst aber wird die Clustermethode von KBG beschrieben.

[1] siehe Einleitung

3.1 Das Verfahren

Das Akronym KBG steht übersetzt für „Wissensbasiertes Generalisieren“: Von den Beobachtungen ausgehend, werden bottom-up Cluster gebildet, deren Charakterisierung eine Generalisierung der Beschreibungen aller Beobachtungen jeweils eines Clusters sind. Aus den nicht-disjunkten Clustern wird eine Hierarchie aufgebaut. Die Clustermethode basiert auf einem Ähnlichkeitsmaß, das speziell für die verwendete Repräsentationssprache definiert wird. Im Folgenden werden hauptsächlich die Teile der Funktionalität des Verfahrens beschrieben, die bei der gegebenen Anwendung relevant sind. Weitere Eigenschaften von KBG, das Einbinden von Hintergrundwissen, die automatische Generierung einer Hierarchie von Klassifikationsregeln aus den gelernten Clustern, sowie die Gewichtung einzelner Prädikate, können der Literatur entnommen werden [Bisson, 1992a; Bisson, 1992b].

3.1.1 Repräsentation von Beobachtungen

Die Repräsentationssprache der Beobachtungen basiert in KBG auf Klauseln, die einen leeren Klauselkopf besitzen, und nur aus einer Disjunktion negativer Literale im Klauselkörper bestehen. Eine Beobachtung wird in KBG repräsentiert durch einen Namen, der in Regelschreibweise gleichgesetzt wird mit einer Konjunktion grundinstantiierter, also variablenfreier, Literale:

$$Name = P_1, ..., P_n.$$

Um quantitative Information in numerischer oder anderer Form bei der Clusterung in spezieller Weise behandeln zu können, werden zwei grundsätzlich verschiedene Arten von Sorten eingeführt: Objekte und Werte. Ein Literal ist dann in KBG von der Form

$$\textit{prädikatsname}(Objekt_1,...,Objekt_k,Wert_1,...,Wert_l),$$

mit $k, l \geq 0$. Das Prädikatssymbol beschreibt die Eigenschaft eines Objektes, oder die Relation zwischen mehreren Objekten. Die Werte geben ein quantitatives Maß für die Eigenschaft/Relation an.

Beispiel: In der Beobachtung
E = entfernung(peter,paul,820), fahrtdauer(peter,6);
wird der Wert der Entfernung zwischen den beiden Objekten Peter und Paul

mit 820 Kilometern angegeben, wenn Peter Paul besuchen möchte, muß er 6 Stunden fahren.

Der Unterschied zu Literalen in Hornklausellogik besteht darin, daß die Werte die durch das Prädikat angegebene Eigenschaft/Relation der Objekte nur quantitativ bewerten und **nicht**, wie Objekte, zwei Literale miteinander *verbinden*[2]. Das heißt, wenn in zwei Literalen einer Beobachtungsbeschreibung gleiche Werte stehen, werden sie nicht als dasselbe „Objekt" betrachtet.

Beispiel: In der Beobachtung
E = entfernung(peter,paul,820), fahrtdauer(820,6);
in der die 820 einen Wert darstellt, wird das Prädikat *fahrtdauer/2* **nicht** zur Beschreibung der Entfernung von Peter und Paul ausgewertet. Diese Semantik wird erst wie oben durch die *Verbindung* über eines der beiden Objekte hergestellt:
E = entfernung(peter,paul,820), fahrtdauer(peter,6);

In einer Prädikatsdefinition wird zu jedem Prädikat eine Sortendeklaration erstellt, die angibt, welche Argumente als Objekte und welche als Werte zu interpretieren sind. Den Werten wird vom Benutzer eine Untersorte zugewiesen, für die ein Datentyp inklusive Wertebereich angegeben wird. Bei der Berechnung der Ähnlichkeit und bei der Generalisierung mehrerer Werte werden diese Untersorten benutzt. Um Objekte als solche zu kennzeichnen, werden in der Prädikatsdefinition die Argumentstellen, an denen Objekte auftreten, mit der Sorte „Objekt" angegeben.

Beispiel: Die Prädikatsdefinition für *entfernung/3* kennzeichnet die ersten beiden Argumente als Objekte, das dritte Argument als nominalen Wert. Die Definition für *fahrtdauer/2* ist ähnlich:
Prädikate: *entfernung(name,name,distanz); fahrtdauer(name,zeit);*
Sorten: *name = Objekt, distanz = Ganzzahlig im Intervall I = [1,1000], zeit = Ganzzahlig im Intervall J = [1,24].*

[2]Definition hierzu folgt.

3.1.2 Das Ähnlichkeitsmaß

Das Maß zur Bewertung der Ähnlichkeit zweier Beobachtungen ist der Mittelwert über Ähnlichkeiten der in den Prädikaten der Beobachtungsbeschreibungen enthaltenen Objekte. Die Ähnlichkeit von Objekten baut im Prinzip auf dem gemeinsamen Vorkommen in den Prädikaten der Beobachtungsbeschreibungen auf:
Für jedes Paar von Objekten aus den zwei Beobachtungen wird gezählt, in wievielen Fällen die Objekte in dem gleichen Prädikat an derselben Argumentposition auftreten, relativ zur Gesamtanzahl der Vorkommen der Objekte. Dieser Wert steht für die Ähnlichkeit der beiden Objekte (1).
Die Repräsentationssprache erlaubt es, weitere Informationen einzubeziehen, um diesen Wert besser festzulegen.

Erstens gehören dazu Relationen zu anderen Objekten. Sind zwei Objekte, deren Ähnlichkeit berechnet werden soll, über mehrstellige Prädikate mit weiteren Objekten *verbunden*, so soll auch deren Ähnlichkeit in die Berechnung einfließen, mit einer Gewichtung proportional zum Grad der Verbundenheit.

> **Definition der Verbundenheit:** Ein Objekt X ist mit einem Objekt Y direkt (durch einen Pfad der Länge 1) *verbunden*, wenn es ein Prädikat gibt, das X und Y in Relation setzt. Ist Y mit einem Objekt Z über einen Pfad der Länge n *verbunden*, so sind X und Z ebenfalls *verbunden*, über einen Pfad der Länge n+1. Die Länge des Pfades drückt den Grad der Verbundenheit aus. Dieser ist umso größer, je kürzer der Pfad ist.

Zweitens werden zur Ähnlichkeitsberechnung die Werte, die Eigenschaften und Relationen quantitativ gewichten, berücksichtigt. Im Folgenden werden vier verschiedene Arten von Ähnlichkeiten benutzt:

1. Die Ähnlichkeit zweier Beobachtungen.
2. Die Ähnlichkeit zweier Objekte insgesamt: *SIM(X,Y)*.
3. Die Ähnlichkeit zweier Objekte für ein gemeinsames Prädikat T, in dem beide Objekte an derselben Position R stehen: *T_SIM*(T,R).
4. Die Ähnlichkeit von Werten w1, w2 einer Untersorte: V_SIM_{Typ}(w1,w2).

Mit dem Ähnlichkeitswert *T_SIM* wird die Einbeziehung der Ähnlichkeit verbundener Objekte und Werte erreicht. Anstatt ein gemeinsames Prädikat zweier Objekte mit einem Ähnlichkeitswert 1 zu zählen[3], wird an jeder Argumentposition des Prädikates eine Ähnlichkeit ermittelt. Die Ähnlichkeiten an den Argumentpositionen, an denen Objekte stehen, werden aufsummiert und durch die Anzahl der Positionen dividiert. Gleiches wird mit den Argumentpositionen, an denen Werte stehen, durchgeführt. Die letztere Summe wird zur Gewichtung der Ersteren mit dieser multipliziert.

Seien T und T' die beiden Prädikate, bei denen die Objekte X und Y an derselben Argumentposition R auftreten:
$T = p(A_1, ..., X, ...A_n, U_1, ..., U_m)$
$T' = p(B_1, ..., Y, ...B_n, V_1, ..., V_m)$,
wobei A_i, B_i Objekte darstellen, U_j, V_j Werte und $n, m \geq 0$ sind. Dann ist[4]

$$T\_SIM(p, R) = \frac{1}{n} \sum_{i=1}^{n} si(A_i, B_i) * \frac{1}{m} \sum_{j=1}^{m} si(U_j, V_j),$$

si(X, Y) = 1, an der Stelle R,
si(A_i, B_i)=$SIM(A_i, B_i)$, wenn i $\neq$ R,
si(U_j, V_j)= V$_SIM_{Typ}(U_j, V_j)$.

Die Ähnlichkeit von Werten (V$_SIM_{Typ}$) wird abhängig von der Untersorte der Werte berechnet. Der Benutzer kann eigens auf Werte der Anwendung angepaßte Funktionen zur Berechnung der Ähnlichkeit angeben. Die Ähnlichkeit von Werten entspricht den Ähnlichkeitsmaßen statistischer Cluster-Analyse-Verfahren. In KBG sind unter anderem V$_SIM_{Typ}$ für ganzzahlige und reelle Werte, sowie für nominale Wertemengen vordefiniert.

Der erste Schritt zur Berechnung der Ähnlichkeit zweier Beobachtungen besteht also darin, für alle Prädikate T, die in beiden Beobachtungsbeschreibungen vorkommen, die Formel für den Wert *T_SIM(T,R)* aufzustellen, und zwar an allen Stellen R, an denen Objekte stehen. Ein *T_SIM(T,R)* für ein bestimmtes Prädikat T an der Stelle R betrifft damit immer zwei Objekte, je eines aus den zu vergleichenden Beobachtungen.

Für ein bestimmtes Paar (X,Y) von Objekten sind mehrere *T_SIM(T,R)*-Werte vorhanden, wenn es mehrere Prädikate T gibt, in denen

[3] vgl. (1).

[4] Diese Definition soll auch den Fall abdecken, in dem ein Prädikat keine Objekte ($n = 0$), oder keine Werte ($m = 0$) enthält. Der entsprechende Ausdruck wird dann gleich 1 gesetzt.

genau dieses Paar an derselben Stelle R auftritt. Die folgende Festlegung ist eine Verfeinerung der Aussage von (1):
Die Ähnlichkeit *SIM(X,Y)* zweier Objekte X und Y entspricht der Summe ihrer *T_SIM*-Ähnlichkeiten, dividiert durch die Gesamtanzahl der Vorkommen von X oder Y in der entsprechenden Beobachtungsbeschreibung. Maßgeblich ist dabei das häufigere der beiden, falls X oder Y noch an weiteren Stellen auftritt. (2)
Sei *co* die Anzahl der Prädikate, in denen X und Y an gleicher Stelle auftreten, *occ(X)* die Anzahl der Prädikate einer Beobachtung, in denen das Objekt X enthalten ist. Dann ist

$$SIM(X,Y) = \frac{\sum_{i=1}^{co} T\_SIM(T_i, R_i)}{max(occ(X), occ(Y))}.$$

Die *SIM*-Ähnlichkeit zweier Objekte wird für alle Paare von Objekten[5] errechnet, die sich durch gemeinsames Vorkommen der Objekte bilden lassen. Zur Berechnung des *SIM*-Wertes eines Objektpaares werden in manchen Fällen also die *SIM*-Werte anderer Objektpaare benötigt. Dieses führt auf ein lineares Gleichungssystem zur Berechnung aller *SIM*-Werte. Die Anzahl der Gleichungen entspricht der Anzahl der Objektpaare, für die der *SIM*-Wert berechnet werden muß, also der Anzahl der Unbekannten.

Beispiel zur Berechnung der Arten von Ähnlichkeiten 2. 3. und 4.:
Gegeben sind die beiden Beobachtungen
E = entfernung(a,b,groß), fahrtdauer(a,mittel);
E' = entfernung(a',b',groß), fahrtdauer(a',lang);[6]
Die Prädikatsdefinition für *entfernung/3* kennzeichnet die ersten beiden Argumente als Objekte, das dritte Argument ist diesmal ein nominaler Wert. Auch bei *fahrtdauer/2* wird ein nominaler Wert verwendet.
Prädikate:*entfernung(name,name,distanz); fahrtdauer(name,zeit);*
Sorten: *name = Objekt, distanz = [klein,mittel,groß], zeit = [kurz,mittel,lang].*

Als erstes werden alle möglichen Ähnlichkeiten zweier Objekte für ein gemeinsames Prädikat (*T_SIM*(T,R)) nach der oben angegebenen Formel errechnet. Es sind zwei Objektpaare vorhanden, das erste tritt an der Stelle R=1 von *entfernung/3* und von *fahrtdauer/2* auf, das zweite an der Stelle

[5] Es gibt auch einen Wert *SIM*(0,0), der *T_SIM*-Werte der Prädikate zusammenfaßt, in denen keine Objekte vorkommen

[6] a benuzt das Flugzeug und schafft somit auch große Entfernungen in mittlerer Zeit

R=2 von *entfernung/3.*

für a,a':
T_SIM(entfernung,1)= 1/2(1+*SIM*(b,b')) * V_$SIM_{distanz}$(groß,groß)
T_SIM(fahrtdauer,1)= 1 * V_SIM_{zeit}(mittel,lang)
für b,b':
T_SIM(entfernung,2)= 1/2(*SIM*(a,a')+1) * V_$SIM_{distanz}$(groß,groß).

V_SIM_{zeit}(u,u') und V_$SIM_{distanz}$(u,u') sind so definiert, daß bei u=u' die Ähnlichkeit 1 beträgt, und sonst 0. Aus den *T_SIM* Ähnlichkeiten wird das lineare Gleichungssystem für die Ähnlichkeit der beiden Objektpaare a,a' und b,b' aufgestellt und gelöst:

$SIM(a,a')= \frac{T\_SIM(entfernung,1)+T\_SIM(fahrtdauer,1)}{2}$
$SIM(b,b')= T\_SIM(entfernung,2)$
$\Leftrightarrow$
$SIM(a,a')= \frac{1/2(1+SIM(b,b'))+0}{2}$
$SIM(b,b')= 1/2(SIM(a,a')+1)$
$\Leftrightarrow$
$SIM(a,a') - 1/4 SIM(b,b') = 1/4$
$-1/2 SIM(a,a') + SIM(b,b') = 1/2$
$\Leftrightarrow$
$SIM(a,a')=3/7=0.43, SIM(b,b')=5/7=0.71.$

Die Ähnlichkeit zweier Beobachtungen (Ähnlichkeitsart Nr. 1), das Maß aufgrund dessen die Cluster gebildet werden, ist der Mittelwert von Objektähnlichkeiten (Ähnlichkeitsart Nr. 2) der in den Beobachtungsbeschreibungen vorkommenden Objekte. Es werden aber nicht alle Objektänlichkeiten verwendet, die bei der Lösung des Gleichungssystemes errechnet worden sind. Vielmehr wird versucht, jedes Objekt der einen Beobachtung einem Objekt der anderen zuzuordnen. Es werden die Objektähnlichkeiten der einander zugeordneten Objekte gemittelt. Es gibt verschiedene Möglichkeiten, Objekte der einen Beobachtung den Objekten der anderen zuzuordnen. Objektpaare, für die kein Ähnlichkeitswert existiert, scheiden von vornherein aus. Aus den verbleibenden möglichen Zuordnungen der Objekte wird die ausgewählt, die zu einem maximalen Mittelwert der Objektähnlichkeiten führt.

Beispiel: In obigem Beispiel ist die Zuordnung der Objekte klar, a entspricht a', b entspricht b', da nur die beiden Objektähnlichkeiten *SIM*(a,a') und *SIM*(b,b') errechnet wurden. Die Ähnlichkeit der Beobachtungen E und

E' beträgt somit 1/2(*SIM*(a,a')+*SIM*(b,b')). Eine Auswahl unter mehreren Zuordnungen wird zum Beispiel bei der Ähnlichkeitsberechnung dieser beiden Beobachtungen nötig:
E1= entfernung(a,b,groß), fahrtdauer(a,mittel), fahrtdauer(b,mittel)
E1' = entfernung(a',b',groß), fahrtdauer(a',lang), fahrtdauer(b',mittel);
Da nun auch „b" **und** „b'" in dem Prädikat „fahrtdauer" an erster Argumentposition stehen, gibt es folgende Objektähnlichkeiten:
SIM(a,a')=0.46, *SIM*(b,b')=0.86, *SIM*(a,b')=0.5, *SIM*(b,a')=0.0
Es wäre jetzt denkbar, a und b' einander zuzuordnen. Mit einer Objektähnlichkeit von 0.5 sind a und b' ähnlicher als a und a' (0.46). Dann müßte aber b, a' zugeordnet werden, und der Mittelwert betrüge nur $0.5 + 0$. Der maximale Ähnlichkeitswert von $\frac{1}{2} * (0.46 + 0.86)$ läßt sich wieder mit der Zuordnung a-a' und b-b' erreichen.

3.1.3 Die Clustermethode

Nach der Definition eines Ähnlichkeitsmaßes können für den Aggregationsschritt des Conceptual-Clustering Clustermethoden aus der Statistik verwandt werden. Das Ähnlichkeitsmaß von KBG wird anstelle der dort bekannten Ähnlichkeiten benutzt. Die Ähnlichkeiten zwischen allen Beobachtungen werden in die Ähnlichkeitsmatrix eingetragen. Wie bei den statistischen Verfahren auch, wird ein Maß für die Ähnlichkeit von Clustern benötigt. Vergleichbar mit der „Centroid Cluster-Analyse", gibt es in KBG zu jedem Cluster eine Beschreibung in der Beschreibungssprache der Beobachtungen, so daß die Ähnlichkeit zwischen zwei Clustern wieder genau so berechnet werden kann, wie die zwischen zwei Beobachtungen. Diese Beschreibung ergibt sich aus einer Generalisierung der in dem Cluster enthaltenen Beobachtungsbeschreibungen. Die Generalisierung soll die strukturelle Gleichheit der Beobachtungen aufdecken: Die Struktur besteht in der *Verbundenheit* unter den Objekten in einer Beobachtungsbeschreibung. In der generalisierten Beschreibung stehen die Prädikate, die in allen Beobachtungen des Clusters vorkommen. Die Objekte an den entsprechenden Argumentstellen, werden konsistent mit der während der Ähnlichkeitsberechnung gewählten Zuordnung der Objekte untereinander verbunden. Das heißt, daß die Verbundenheit zweier Prädikate bei der Generalisierung erhalten bleibt, wenn sie in beiden Beobachtungen durch Objekte verbunden werden, die einander zugeordnet werden konnten. An den Argumentstellen die mit Werten belegt sind, stehen Generalisierungen der in den Beobachtungen vorkommenden Werte. Für jeden Datentyp gibt es dazu eine eigene

Berechnungsvorschrift V-GEN_{Typ}.

Beispiel: Bei Mengentypen bietet es sich an, die Vereinigung aller Werte als Generalisierung zu nehmen. Aus der Generalisierung der Beobachtungen E und E',
E = entfernung(a,b,groß), fahrtdauer(a,mittel);
E'= entfernung(a',b',groß), fahrtdauer(a',lang);
ergibt sich[7] folgende Beschreibung des Clusters C:
C = entfernung(x,y,groß), fahrtdauer(x,[mittel,lang]).

KBG bildet bottom-up eine Clusterhierarchie in Form eines gerichteten Wurzelgraphen. Solange nicht alle Beobachtungen auf oberster Ebene der Hierarchie in einem Wurzelknoten zusammengeführt sind, werden folgende Schritte auf der Menge von Beobachtungen durchgeführt:

1. Die Ähnlichkeitsmatrix wird berechnet.
2. Die Beobachtungen, deren relative Ähnlichkeit untereinander einen Schwellwert übersteigt, werden in Cluster gruppiert.
3. Die Cluster werden generalisiert und ihre Beschreibung in die Menge der Beobachtungen aufgenommen. Die generalisierten Beobachtungen werden aus der Menge gelöscht.

Die Vorgehensweise unter 2. fürht dazu, daß an dieser Stelle Cluster entstehen können, die überlappen. Das Ergebnis ist also nicht, wie bei rein hierarchischen Clustermethoden, ein Baum. Stattdessen entsteht ein gerichteter Graph, dessen Kanten eine „Genereller-als"-Relation zwischen Vorgänger- und Nachfolgecluster ausdrücken. Auf der untersten Ebene der Hierarchie stehen die einzelnen Beobachtungen.

KBG läßt sich gut mit UNIMEM vergleichen. Bei beiden besteht die Charakterisierung in einer sukzessiven Generalisierung der Beobachtungen. Hier wie dort ist der Aggregationsschritt von einem Ähnlichkeitsmaß bestimmt. Charakterisierungs- und Aggregationschritte werden in mehreren Iterationen abwechselnd durchgeführt und beeinflussen sich gegenseitig. Der wesentliche Unterschied zu UNIMEM besteht in dem bedingt durch die Repräsentationssprache erweiterten Ähnlichkeitsmaß und darin, daß KBG nicht inkrementell ist.

[7] Die Generalisierung entält Objektvariablen. Um daraus eine Beobachtungsbeschreibung zu erhalten, werden die Variablen beliebig instanziiert.

3.2 Darstellung der Anwendung

In der nun folgenden Anwendung werden Basis-Merkmal-Sequenzen von KBG in einer Clusterhierarchie organisiert. Das Ziel der Anwendung von KBG ist das Auffinden von Clustern in den beobachteten Basis-Merkmal-Sequenzen, so daß

> die Beschreibungen der Cluster es erlauben, zusammenhängende Sequenzen zu finden, die Sensor-Merkmale beschreiben.

In diesem Teil des Kapitels wird gezeigt, wie die Roboterdaten von KBG verarbeitet werden können. Dazu ist es erstens notwendig, eine angemessene Repräsentation der Daten in dem von KBG auswertbaren Format zu finden. Um die bei der Clusterbildung auftretenden Ergebnisse zu erklären, wird gezeigt, wie sich die Ähnlichkeitsberechnung im Falle dieser speziellen Eingabe verhält. Schließlich muß zur Erreichung des Zieles die Clusterhierarchie ausgewertet werden, um zu klären, welche der Cluster überhaupt verwertbar sind.

3.2.1 Umrepräsentation der Daten

Die Umrepräsentation der Eingabedaten, die in der Form von *Cases* vorliegen,[8] muß drei Dinge leisten:

1. Die Daten müssen in die Syntax von Beobachtungen in der Repräsentationssprache von KBG überführt werden.
2. Die zeitliche Reihenfolge der Basis-Merkmale muß repräsentiert werden.
3. Die Struktur der Beobachtungen muß unter den Cluster- und Generalisierungsoperationen von KBG erhalten bleiben, so daß auch die entstehenden Clusterbeschreibungen Sequenzen von Basis-Merkmalen enthalten.

Nach folgender Formel wird die Umrepräsentation der *Cases* in KBG-Beobachtungen vorgenommen:

$$[s\_hwm\_dummy(Tr, S, T_1, T_{n+1}, RM, SCL), bf_1(Tr, O, S, T_1, T_2, G_1),$$

[8] siehe Kapitel 1.3

$$bf_2(Tr, O, S, T_2, T_3, G_2), ..., bf_n(Tr, O, S, T_n, T_{n+1}, G_n)]$$

wird zu

$$TrS = bf_1(T_1, T_2), bf_2(T_2, T_3), ..., bf_n(T_n, T_{n+1}).$$

Dazu ein **Beispiel:**
$[s\_durch\_tür\_dummy(t50, s6, 15, 41, parallel, seite),$
$konstant(t50, 0, s6, 15, 17, -2), steigend(t50, 0, s6, 17, 18, 87),$
$konstant(t50, 0, s6, 18, 41, 0)]$

wird zu

t50s6 = konstant(t1,t2),steigend(t2,t3),konstant(t3,t4).

Fahrt- und Sensornummer bilden zusammen den Namen, der zur Unterscheidung der Beobachtungen dient. Die in einem Lernlauf betrachteten *Cases*, unterscheiden sich durch Fahrt- und Sensornummer. Demnach wird der Name einer Beobachtung aus diesen beiden zusammengesetzt. Die Prädikate der Beobachtungsbeschreibung entsprechen den beschreibenden Prädikaten des *Cases*. Dabei wird in der Prädikatsdefinition für die Beobachtungsbeschreibung die Anzahl der Argumente auf das nötigste reduziert, so daß genau die Basis-Merkmalsnamen und ihre zeitliche Verkettung repräsentiert werden.[9] Da eine Beobachtungsbeschreibung durch genau eine Kombination von Fahrt- und Sensornummer identifiziert wird, müssen diese nicht in die beschreibenden Prädikate übernommen werden.

Übrig bleiben nur die Zeitpunkte, so daß zweistellige Prädikate verwandt werden, die den Anfangs- und den Endzeitpunkt der Basis-Merkmale aus den *Cases* enthalten. Eine entscheidende Festlegung wird in der Sortendeklaration der Zeitpunkte getroffen. Nur wenn die Zeitpunkte von der Sorte „Objekt“ sind, verbindet ein Zeitpunkt zwei Basis-Merkmale zu einer zeitlichen Abfolge, indem er in einem Basis-Merkmal als Endzeitpunkt, in einem anderen als Anfangszeitpunkt vorkommt. Anstelle der Zeitpunkte stehen demnach beliebige Objektkonstanten, die keinerlei ganzzahlige Eigenschaften besitzen. Eine andere Möglichkeit gibt es für die Sorte der Zeitpunkte nicht. Wählt man ganzzahlige Werte als Sorte für die Zeitpunkte, erkennt KBG gleiche Werte nicht als das gleiche Objekt. Zwei Basis-Merkmale, die diesen Wert als Anfangs-, beziehungsweise Endzeitpunkt besitzten sind nicht *verbunden*.

[9] Siehe Kapitel 1.3: Die Orientierung, der Gradient und die Dauer der Zeitintervalle werden beim Lernen nicht berücksichtigt.

Eine Kennzeichnung der Beobachtung mit ihrem Oberbegriff (dem handlungsorientierten Wahrnehmungsmerkmal mit der Relativbewegung und der Sensorklasse) ist nicht nötig, da die Lernläufe nach Oberbegriffen getrennt durchgeführt werden.

Ob die so erhaltene Repräsentation der Beobachtungen von den obigen Forderungen auch Teil 3. erfüllt, hängt von der Ähnlichkeitsberechnung ab: Die Generalisierung von zwei Sequenzen wird die Basis-Merkmal-Prädikate enthalten, die in beiden Sequenzen vorkommen. Entscheidend dafür, daß die Generalisierung wieder zusammenhängende Sequenzen enthält, ist die richtige Verbindung der Prädikate durch die Zeitpunkte. Da die Zeitpunkte (als Objekte) aufgrund der während der Ähnlichkeitsberechnung gewählten Zuordnung untereinander *verbunden* werden, soll dieser Schritt an einigen ausgewählten Basis-Merkmal-Sequenzen untersucht werden.

3.2.2 Ähnlichkeiten von Beobachtungen der Anwendung

Einige Beispiele zur Ähnlichkeitsberechnung zweier Basis-Merkmal-Sequenzen sollen die Hypothese unterstützen, daß Basis-Merkmal-Sequenzen, die gleich oder ähnlich sind, auch von KBG als ähnlich angesehen werden, indem sich bei der Berechnung der Ähnlichkeit hohe Werte ergeben. Zu diesem Zweck werden

a) zwei gleiche Sequenzen,

b) zwei Sequenzen, von denen eine ein Präfix der anderen ist,

c) zwei Sequenzen, die sich in der Mitte um ein Basis-Merkmal unterscheiden,

verglichen. Daß die dabei errechneten Ähnlichkeiten den Erwartungen entsprechen, wird verifiziert, indem die Clustermethode auf die beteiligten Sequenzen angewandt, und gezeigt wird, daß KBG daraus eine sinnvolle Clusterhierarchie bildet, deren Cluster Basis-Merkmal-Sequenzen enthalten.

a) Zwei gleiche Sequenzen. Der einfache Fall liegt vor, wenn zwei gleiche Sequenzen verglichen werden (a,b und c sind Abkürzungen für Basis-Merkmale):
S1 = a(v1,v2),b(v2,v3),c(v3,v4);
S1' = a(u1,u2),b(u2,u3),c(u3,u4);

Zuerst werden auch hier alle möglichen Ähnlichkeiten zweier Objekte für ein gemeinsames Prädikat (*T_SIM*(T,R)) errechnet. In diesem Fall gibt es vier Objektpaare - v1/u1, v2/u2, v3/u3 und v4/u5 - in den ersten und zweiten Argumenten von „a“, „b“ und „c“ .

für v1/u1:
T_SIM(a,1)= 1/2(1+SIM(v2,u2)) * 1
für v2/u2:
T_SIM(a,2)= 1/2(1+SIM(v1,u1)) * 1
T_SIM(b,1)= 1/2(1+SIM(v3,u3)) * 1
für v3/u3:
T_SIM(b,2)= 1/2(1+SIM(v2,u2)) * 1
T_SIM(c,1)= 1/2(1+SIM(v4,u4)) * 1
für v4/u4:
T_SIM(c,2)= 1/2(1+SIM(v3,u3)) * 1

Aus den T_SIM Ähnlichkeiten wird das lineare Gleichungssystem für die Ähnlichkeit der vier Objektpaare aufgestellt und gelöst:

SIM(v1,u1)= 1/2(1+SIM(v2,u2))
SIM(v2,u2)= $\frac{1/2(1+SIM(v1,u1))+1/2(1+SIM(v3,u3))}{2}$
SIM(v3,u3)= $\frac{1/2(1+SIM(v2,u2))+1/2(1+SIM(v4,u4))}{2}$
SIM(v4,u4)= 1/2(1+SIM(v3,u3))
$\Leftrightarrow$
SIM(v1,u1)= 1/2(1+SIM(v2,u2))
SIM(v2,u2)= 1/2+1/4(1/2+1/2*SIM(v2,u2))+1/4*SIM(v3,u3)
SIM(v3,u3)= 1/2+1/4(1/2+1/2*SIM(v3,u3))+1/4*SIM(v2,u2)
SIM(v4,u4)= 1/2(1+SIM(v3,u3))
$\Leftrightarrow$
SIM(v1,u1)= 1/2(1+SIM(v2,u2))
7/8*SIM(v2,u2)= 5/8+1/4*SIM(v3,u3)
7/8*SIM(v3,u3)= 5/8+1/4*SIM(v2,u2)
SIM(v4,u4)= 1/2(1+SIM(v3,u3))
$\Leftrightarrow$
SIM(v1,u1)= 1/2(1+SIM(v2,u2))=1
SIM(v2,u2)= 1
SIM(v3,u3)= 1
SIM(v4,u4)= 1/2(1+SIM(v3,u3))=1

Zur Berechnung der Ähnlichkeit von S1 und S1’ muß eine Zuordnung der Objekte von S1 (v1,v2,v3,v4) und von S1’ (u1,u2,u3,u4) gefunden werden.

In diesem Fall ist sie durch die vorhandenen *SIM*-Werte bereits festgelegt: v1 wird u1 zugeordnet, die beiden besitzen einen Ähnlichkeitswert von 1, gleiches gilt für v2/u2, v3/u3 und v4/u4. Aufgrund dieser Zuordnung wird bei der Generalisierung von S1 und S1' versucht, die Prädikate *a/2 b/2* und *c/2* zu *verbinden*. Da in den Sequenzen *a/2* und *b/2* durch v2 beziehungsweise u2, sowie *b/2* und *c/2* durch v3, u3 *verbunden* sind, behält auch die Generalisierung diese Eigenschaft. Die Ähnlichkeit von S1 und S1', der Mittelwert der vier *SIM*-Werte, ist 1, die beiden Sequenzen werden, wie erwartet, als gleich angesehen.

b) Zwei Sequenzen, von denen eine ein Präfix der anderen ist. Nun soll betrachtet werden, ob trotz eines weiteren Prädikates am Ende einer der Sequenzen, die beiden Beobachtungen wieder zu einer Sequenz generalisiert werden können.
S1 = a(v1,v2),b(v2,v3),c(v3,v4);
S2 = a(u1,u2),b(u2,u3),c(u3,u4),a(u4,u5);

Die oben aufgestellten *T_SIM* Formeln für Objektpaare aus S1 und S2 bleiben dieselben. Zusätzlich müssen zwei weitere Formeln aufgestellt werden, da in S2 das Prädikat *a/2* doppelt vorkommt.

für v1/u4:
T_SIM(a,a,1)= 1/2(1+SIM(v2,u5)) * 1
für v2/u5:
T_SIM(a,a,2)= 1/2(1+SIM(v1,u4)) * 1

In dem hieraus resultierenden Gleichungssystem für die Ähnlichkeit der sechs Objektpaare, sind die Formeln für die Paare v1/u1, v2/u2 und v3/u3 gleich geblieben. Diejenige für das Paar v4/u4 wird durch zwei dividiert, da u4 noch in einem weiteren Prädikat von S2 vorkommt. Außerdem kommen Ähnlichkeiten für die Paare v1/u4 und v2/u5 hinzu:

$SIM(\mathrm{v1,u1}) = 1/2(1+SIM(\mathrm{v2,u2}))$
$SIM(\mathrm{v2,u2}) = \frac{1/2(1+SIM(v1,u1))+1/2(1+SIM(v3,u3))}{2}$
$SIM(\mathrm{v3,u3}) = \frac{1/2(1+SIM(v2,u2))+1/2(1+SIM(v4,u4))}{2}$
$SIM(\mathrm{v4,u4}) = \frac{1/2(1+SIM(v3,u3))}{2}$
$SIM(\mathrm{v1,u4}) = \frac{1/2(1+SIM(v2,u5))}{2}$
$SIM(\mathrm{v2,u5}) = \frac{1/2(1+SIM(v1,u4))}{2}$

Nach der Auflösung des Gleichungssystems ergeben sich folgende Werte: *SIM*(v1,u1)= 0.98 , *SIM*(v2,u2)= 0.96, *SIM*(v3,u3)= 0.86, *SIM*(v4,u4)= 0.46, *SIM*(v1,u4)= 0.33 und *SIM*(v2,u5)= 0.33.

Nun müssen, aufgrund dieser Werte, von den fünf Objekten in S2 vier ausgewählt, und den vier Objekten in S1 zugeordnet werden. Eine maximale Gesamtähnlichkeit von 0.81 ergibt sich, wenn u5 nicht zugeordnet wird, und u4 weiterhin v4, und nicht etwa v1 entspricht. Die übrigen Zuordnungen bleiben wie im ersten Beispiel. Damit besteht auch die generalisierte Beschreibung von S1 und S2 aus der Sequenz „a(t1,t2),b(t2,t3),c(t3,t4)“. Das zweite *a/2* Prädikat von S2 wird nicht aufgenommen, da das Objekt u5 keinem Objekt in S1 zugeordnet wurde.

c) Zwei Sequenzen, die sich in der Mitte um ein Basis-Merkmal unterscheiden. In dem nächsten Beispiel werden Phänomene betrachtet, die auftreten, wenn in der Mitte zweier Sequenzen unterschiedliche Prädikate stehen.

S3 = a(v1,v2),b(v2,v3),c(v3,v4),d(v4,v5);
S4 = a(u1,u2),b(u2,u3),e(u3,u4),d(u4,u5);

Wieder werden alle möglichen Ähnlichkeiten zweier Objekte für ein gemeinsames Prädikat (*T_SIM*(T,R)) errechnet, wobei es für die Paare v3/u3 und v4/u4 nur einen Wert gibt, weil *c/2* und *e/2* unterschiedliche Prädikate sind:

für v1/u1:
T_SIM(a,1)= 1/2(1+*SIM*(v2,u2)) * 1
für v2/u2:
T_SIM(a,2)= 1/2(1+*SIM*(v1,u1)) * 1
T_SIM(b,1)= 1/2(1+*SIM*(v3,u3)) * 1
für v3/u3:
T_SIM(b,2)= 1/2(1+*SIM*(v2,u2)) * 1
für v4/u4:
T_SIM(d,1)= 1/2(1+*SIM*(v5,u5)) * 1
für v5/u5:
T_SIM(d,2)= 1/2(1+*SIM*(v4,u4)) * 1

Daraus ergeben sich fünf Gleichungen für die Objektähnlichkeiten (*SIM*). Da die Objekte der Paare v3/u3 und v4/u4 jeweils zusätzlich in verschiedenen Prädikaten vorkommen, werden die *SIM*-Werte durch zwei dividiert, und keine weiteren Ähnlichkeiten hinzuaddiert.

SIM(v1,u1)= 1/2(1+SIM(v2,u2))
SIM(v2,u2)= $\frac{1/2(1+SIM(v1,u1))+1/2(1+SIM(v3,u3))}{2}$
SIM(v3,u3)= $\frac{1/2(1+SIM(v2,u2))}{2}$
SIM(v4,u4)= $\frac{1/2(1+SIM(v5,u5))}{2}$
SIM(v5,u5)= 1/2(1+SIM(v4,u4))

Die Auflösung des Gleichungssystems liefert folgendes:
SIM(v1,u1)= 0.92, SIM(v2,u2)= 0.85, SIM(v3,u3)= 0.46, SIM(v4,u4)= 0.43 und SIM(v5,u5)= 0.71. Da hier jeweils fünf Objekte einander zugeordnet werden sollen, werden die Zuordnungen entsprechend der Objektpaare in den vorhandenen fünf *SIM*-Ähnlichkeiten gewählt. Der Mittelwert dieser Werte, welcher die Ähnlichkeit der beiden Sequenzen beschreibt, beträgt nur noch 0.67, obwohl alle Objekte zugeordnet werden konnten. Interessant ist, wie die Generalisierung von S3 und S4 aussieht. Sie enthält erst einmal die Prädikate *a/2*, *b/2* und *d/2*, wohingegen *c/2* und *e/2* nicht enthalten sind, da sie nicht in beiden Sequenzen vorkommen. Durch die Objekte u2 beziehungsweise v2 sind *a/2* und *b/2* miteinander *verbunden*. Das Prädikat *d/2* wird nicht mit dieser Anfangssequenz verbunden. Man hat sich die generalisierte Beschreibung demnach so vorzustellen: „a(t1,t2),b(t2,t3), d(t10,t11)“.

Am Schluß dieses Abschnittes wird die von KBG aus den Sequenzen S1, S2, S3 und S4 gebildete Clusterhierarchie gezeigt. Wie in Abschnitt 3.1.3 beschrieben, werden in einem ersten Schritt die Ähnlichkeitswerte der Sequenzen in einer Matrix zusammengefaßt:[10]

$$\begin{pmatrix} & S1 & S2 & S3 & S4 \\ S1 & - & 0.81 & 0.81 & 0.56 \\ S2 & & - & 0.81 & 0.56 \\ S3 & & & - & 0.67 \\ S4 & & & & - \end{pmatrix}$$

In Schritt 2. werden aus den Sequenzen, deren relative Ähnlichkeit untereinander einen Schwellwert übersteigt, Cluster gebildet. Angenommen, der Schwellwert liegt so, daß für die Clusterbildung ein Ähnlichkeitswert von mindestens 0.67 erreicht werden muß. Dann werden zwei Cluster gebildet, C1 aus den Sequenzen S1, S2 und S3, sowie C2 aus den Sequenzen S3 und S4. In Abbildung 3.1 werden für die Cluster die oben hergeleiteten Beschreibungen eingesetzt. Sie bilden den Ausgangspunkt für die, hier nicht

[10] Der Wert 0.56 als Ähnlichkeit von S1,S2 und S4 wurde hier nicht hergeleitet.

gezeigte nächste Iteration der Clusterbildung. Sie führt die beiden Cluster in einem Startpunkt C3 zusammen, dessen Beschreibung der allen vier Sequenzen gemeinsame Präfix „a(t1,t2),b(t2,t3)“ ist. Der gleiche Präfix ist

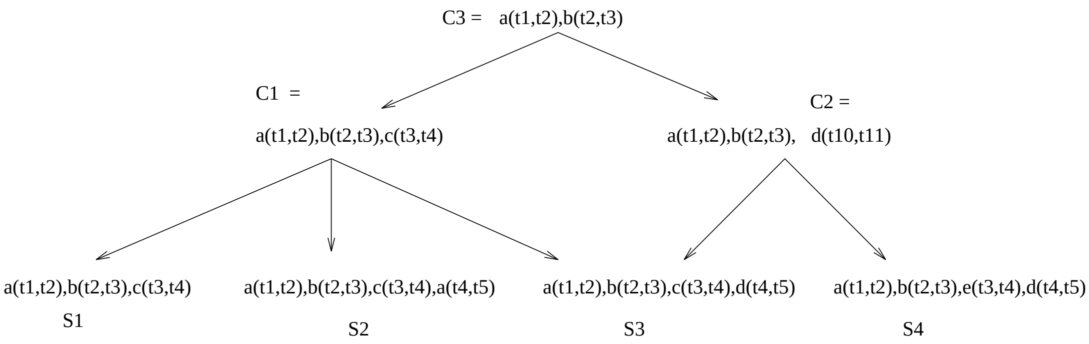


Abbildung 3.1: Sequenzen in KBG

auch in C2 enthalten, so daß C2 und C3, wenn nur Sequenzen in den Cluster-Beschreibungen interessant sind, unter diesem Aspekt gleich sind. Bei einer geschickteren Wahl des oben genannten Schwellertes,[11] wird C2 gar nicht gebildet und die Sequenz S4 wird direkt unter C3 gehängt.

Das Ziel der obigen Beispielrechnungen ist es, einige Effekte der Clusterung von Basis-Merkmal-Sequenzen mit KBG zu zeigen, die Möglichkeiten der nun folgenden Auswertung zu motivieren und ihre Ergebnisse plausibel zu machen. Auch sei festgehalten, daß die auf Basis des eingesetzten Ähnlichkeitsmaßes errechneten Werte durchaus mit intuitiven Erwartungen, welche Paare von Sequenzen als ähnlich anzusehen sind, vereinbar sind.

3.3 Experiment und Auswertung

In diesem Abschnitt soll gezeigt werden, daß mit der gewählten Repräsentation Sensor-Merkmale mit KBG gelernt werden können.

Wie bereits erwähnt, liefert KBG aus der Eingabe, einer Menge von Basis-Merkmal-Sequenzen, die alle dasselbe Sensor-Merkmal beschreiben, eine Hierarchie von Clustern, die Teilsequenzen enthalten. Um die Sequenzen zu finden, mit denen das Sensor-Merkmal beschrieben werden kann, muß eine Auswahl unter den Clustern vorgenommen werden. Eine Chance von KBG

[11] Er kann in KBG als Parameter angegeben werden.

kann darin bestehen, daß hierfür die Struktur der Clusterung ein bestimmtes Vorgehen nahelegt.

Ein Auswahlkriterium nutzt aus, daß durch die gerichteten Kanten im Ergebnisgraphen eine „Genereller-als“-Beziehung etabliert wird. Generellere Cluster decken mehr Beobachtungen ab als ihre Nachfolgecluster. Allerdings haben diese Beobachtungen weniger Gemeinsamkeiten, so daß ihre Generalisierung nur in sehr kurzen, unzusammenhängenden Sequenzteilen besteht. Der Cluster C3 in Abbildung 3.2 beispielsweise deckt zwar alle Beobachtungen ab, stellt aber keine Sequenz mehr dar. Gemäß des Kriteriums

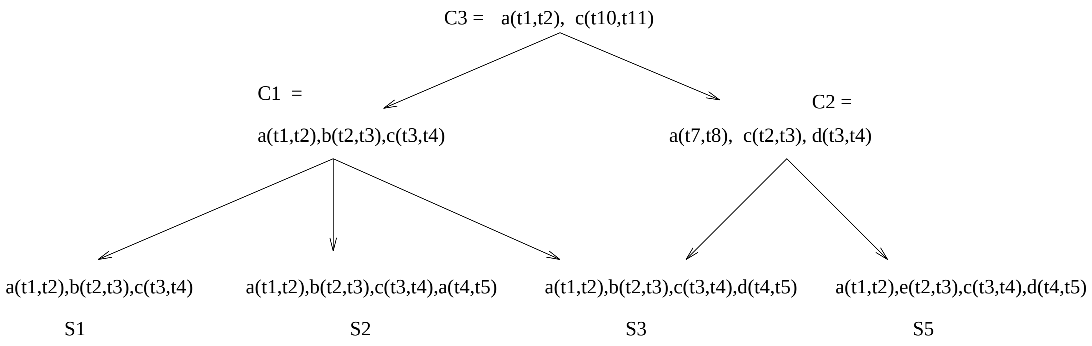


Abbildung 3.2: Cluster und Sequenzen

werden Cluster ausgewählt, die maximal generell sind, deren Beschreibung aber noch zusammenhängende Sequenzen einer bestimmten Mindestlänge enthält. Angenommen, der Graph in Abbildung 3.2 sei das Ergebnis zu einem bestimmten Sensor-Merkmal. Dann sind C1 und C2 die maximal generellen Cluster, die zusammenhängende Sequenzen der Mindestlänge zwei enthalten.

In der Praxis hat sich gezeigt, daß nur Cluster, die direkt aus den *Case*-Sequenzen der Eingabe oder einer Cluster-Ebene darüber bestehen, diese Eigenschaft besitzen.[12] Die Mindestlänge der Sequenzen wird auf vier Basis-Merkmale festgelegt. Tests mit kürzeren Sequenzen haben gezeigt, daß sie nicht ausreichend gegenüber anderen Merkmalen diskriminieren.

[12]Im vorigen Abschnitt ist gezeigt worden, daß die Erhaltung der Sequenzeigenschaft bei der Generalisierung theoretisch möglich ist. Hier schlägt dagegen der Effekt durch, daß bereits ein unterschiedliches Basis-Merkmal innerhalb zweier Sequenzen die generalisierte Sequenz in zwei Teile spaltet, wie ebenfalls gezeigt wurde.

Ergebnisse In diesem Abschnitt sind Daten aus einem Simulator verwendet worden. Die Lernläufe wurden mit Daten aus Traces-4 durchgeführt.[13] Gelernt werden sollte das Sensor-Merkmal s_durch_tür mit relativer Bewegungsrichtung parallel-, und diagonal-entlang. Tabelle 3.1 zeigt die Aufteilung der Fahrten in Lernset und Testset. In Anhang A.1 ist ein Ergebnis-

Sensor-Merkmal	RM	Lernset	Testset
s_durch_tür	parallel	t50,t51,t60,t61,t70,t71	t80,t81,t90,t91
s_durch_tür	diagonal	t52,t53,t62,t63,t72,t73	t30,t82,t83,t92,t93

Tabelle 3.1: Lernset und Testset

graph gezeigt. Die Evaluierung geschieht entsprechend den Ausführungen in Kapitel 1.2.2. Die Abdeckung wird anhand der Oberbegriffe von Lernset, Testset, sowie zum Test der Korrektheit außerdem von 22 Fahrten, die nicht durch eine Tür gingen, zuzüglich der Lern- und Testfahrten des jeweils anderen Merkmals, getestet. In Tabelle 3.2 sind die Ergebnisse der Sensor-Merkmale jeweils getrennt nach Sensorgruppen angegeben.

Die Prozentzahlen in der Spalte „abgedeckt" errechnen sich aus der Anzahl korrekt abgedeckter Oberbegriffe bezogen auf die in der Spalte „gegeben" (geg.) angegebene Gesamtzahl der Oberbegriffe zu dem Merkmal. Damit ist auch ausgesagt, wieviel Prozent fehlen. Die Anzahl der unkorrekt abgedeckten Oberbegriffe wird in der Spalte „negativ" (neg.) bezogen auf die Gesamtzahl der Oberbegriffe, die nicht zu dem Merkmal passen. Die Spalte ist somit definiert als: Anzahl der Fehler dividiert durch die Anzahl möglicher Fehler. Bei paralleler Fahrt stammen die nicht abgedeckten Ober-

s_durch_tür-parallel								
	seite_links		seite_rechts		vorne_groß		hinten_groß	
geg.	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.
10	80%	0%	100%	0%	100%	6%	70%	15%
s_durch_tür-diagonal								
	seite_links		seite_rechts		vorne_groß		hinten_groß	
geg.	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.
11	64%	0%	64%	0%	91%	0%	100%	6%

Tabelle 3.2: Testergebnisse

begriffe alle aus dem Testset, so daß die Abdeckung auf dem Lernset in allen Sensorgruppen vollständig ist. Zu beobachten ist, daß die Ergebnisse

[13] *Varianten* „sfmgd_traces30-150 und „bpf_traces30-150_1"

des Merkmales bei diagonaler Fahrt weniger vollständig sind. Die bei diesem Merkmal wahrgenommenen Sequenzen bestehen aus wesentlich mehr Basis-Merkmalen. Dadurch gibt es mehr Stellen, an denen sich die Wahrnehmungen unterscheiden, was sich in der größeren Zahl von Regeln wiederspiegelt, die bei diesem Merkmal gelernt wurden. Trotzdem sind die Ergebnisse als gut zu bewerten, vor allem die niedrigen Fehlerraten sind positiv. Es ist somit möglich, Sensor-Merkmale mit KBG in der gezeigten Weise zu lernen. Verbesserungen sind im Bereich der Vollständigkeit erwünscht.

In Tabelle 3.3 sind Laufzeitangaben von KBG je Sensor-Merkmal als Durchschnitt der Lernzeiten in den einzelnen Sensorgruppen gegeben. Die Laufzeiten liegen bei den Seitensensoren um einiges höher als bei den anderen beiden Sensorgruppen, obwohl die in Spalte drei angegebene Zahl der Beobachtungen bei den Seitensensoren nur wenig höher liegt. Das liegt daran, daß die von den Seitensensoren wahrgenommenen Basis-Merkmal-Sequenzen im Durchschnitt länger sind (3. Spalte). Aus diesem Grunde sind auch die Laufzeiten bei diagonaler Fahrt größer.

Sensor-Merkmal	RM	Beobachtungen seite, vorne/hinten	Sequenz-Länge l	KBG-Laufzeit
s_durch_tür	parallel	60, 54	4.0, 3.6	36 Sek.
s_durch_tür	diagonal	60, 54	6.8, 4,4	51 Sek.

Tabelle 3.3: Laufzeiten

Der Aufwand besteht je Berechnung einer Ähnlichkeitsmatrix darin, bei einer Anzahl von k Beobachtungen, $k\frac{(k-1)}{2}$ Gleichungssysteme zu lösen. Im schlimmsten Fall müssen $k-1$ Ähnlichkeitsmatrizen berechnet werden, allerdings jeweils zu einer sinkenden Anzahl von Beobachtungen, womit insgesamt die Anzahl der zu lösenden Gleichungssysteme nach oben hin mit $\sum_{i=1}^{k} \frac{k(k-1)}{2} = O(k^3)$ abgeschätzt werden kann. Die Zahl n der Unbekannten eines der Gleichungssysteme besteht bei einer Länge l der zu vergleichenden Sequenzen aus den $n = l + 1$ Zeitobjekten.

Zur Performanz des Verfahrens ist anzumerken, daß KBG speicherplatzbedingt nur eine begrenzte Anzahl von Sequenzen verarbeiten kann. Je nach durchschnittlicher Länge der Sequenzen sind das 30 bis 100.[14]

[14] Bei Verwendung von „Sun Common Lisp 4.0“ und einer Speicherplatzhöchstgrenze von 32 MB.

3.4 Eine zweite Repräsentation

Bei der Auswertung der Cluster im letzten Abschnitt sind von den Clusterergebnissen nur zusammenhängende Sequenzen mit einer bestimmten Mindestlänge verwendet worden. Alle anderen Ergebnisse wurden verworfen. So würde zum Beispiel aus dem Cluster C2 in der Abbildung 3.2 nur die Sequenz „c(t2,t3),d(t3,t4)“ ausgewählt. Aus der Beschreibung des Clusters läßt sich aber noch entnehmen, daß irgendwann vor oder nach dieser Sequenz das Prädikat *a/2* auftritt. In diesem Abschnitt soll die Hypothese untersucht werden, daß sich die Ergebnisse beim Lernen der Sensor-Merkmale verbessern lassen, wenn die Repräsentation der Beobachtungen in KBG so geändert wird, daß zusätzlich Sensor-Merkmale der Form „nach einem Sequenzteil X folgt irgendwann ein Sequenzteil Y“ gelernt werden können.

Die Information, in welcher Reihenfolge die Sequenzteile auftreten, ist in den Clustern noch nicht enthalten. Daß man, wiederum in dem Beispiel, nicht aus „7 ist größer als 4“ darauf schließen kann, daß das Prädikat *a/2* (das mit t7 beginnt) nach der Sequenz *c/2, d/2* (die in t4 endet) folgt, liegt daran, daß die Zeitpkunkte nicht als numerische Werte betrachtet werden - eine Folge der Verwendung der Objektsorte für die Zeitpunkte. Dadurch geht die „Kleiner-als“-Relation auf den Zeitpunkten verloren, nur die Gleichheitsrelation bleibt erhalten.

Durch die Aufnahme weiterer Literale in die Beobachtungsbeschreibungen kann die „Kleiner-als“-Relation wiederhergestellt werden. Das Prädikat *folgt(T1,T2)*, mit den Zeitpunkten T1 und T2 wiederum von der Objektsorte, stellt die „Kleiner-als“-Relation für T1 und T2 dar. Enthält demgemäß die Beschreibung des Clusters C2 im Beispiel ein Prädikat *folgt(t8,t2)*, wird dadurch ausgesagt, daß der Cluster Beobachtungen enthält, in denen *a/2* und irgendwann später die Sequenz *c/2, d/2* auftritt, was ja auch von S3 und S5 erfüllt wird. Damit das *folgt(t8,t2)* Prädikat als Teil der Beschreibung von C2 gelernt werden kann, muß in den Beschreibungen von S3 und S5 die „Kleiner-als“-Relation für alle enthaltenen Zeitpunkte durch *folgt/2* Prädikate festgelegt werden.

Ausgehend von einem *Case* wird die Umrepräsentation einer Beobachtung in eine zweite KBG-Repräsentation nach folgender Formel vorgenommen:

$$[s\_hwm\_dummy(Tr,S,T_1,T_{n+1},RM,SCL), bf_1(Tr,O,S,T_1,T_2,G_1),$$
$$bf_2(Tr,O,S,T_2,T_3,G_2), ..., bf_n(Tr,O,S,T_n,T_{n+1},G_n)]$$

wird zu

$$TrS = bf_1(T_1, T_2), bf_2(T_2, T_3), ..., bf_n(T_n, T_{n+1}),$$
$$folgt(T_1, T_2), folgt(T_1, T_3), ..., folgt(T_1, T_n),$$
$$folgt(T_2, T_3), ..., folgt(T_2, T_n),$$
$$...$$
$$folgt(T_{n-1}, T_n).$$

Ergebnisse Für die Lernläufe wurden die gleichen Daten verwendet, wie im vorigen Abschnitt. Tabelle 3.4 zeigt noch einmal die Aufteilung in Lernset und Testset.[15] Bei der Auswertung wurden aus den Clusterbeschreibungen des Ergebnisgraphen alle Teile ausgewählt, die:

Sensor-Merkmal	RM	Lernset	Testset
s_durch_tür	parallel	t50,t51,t60,t61,t70,t71	t80,t81,t90,t91
s_durch_tür	diagonal	t52,t53,t62,t63,(t72,t73)	t30,t82,t83,t92,t93

Tabelle 3.4: Lernset und Testset

a eine Sequenz der Länge vier oder mehr darstellen.

b aus zwei Sequenzteilen X und Y bestehen, die durch ein *folgt/2* Prädikat miteinander *verbunden* sind und deren Gesamtlänge (Länge von X plus Länge von Y) vier oder mehr beträgt.

In Anhang A.3 sind einige Regeln abgedruckt, die aus den ausgewählten Clusterbeschreibungen gebildet wurden. Die mit den so gewonnenen Sensor-Merkmalen erzielten Testergebnisse sind in Tabelle 3.5 gezeigt. Für das

s_durch_tür-parallel

	seite_links		seite_rechts		vorne_groß		hinten_groß	
geg.	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.
10	90%	3%	100%	0%	100%	15%	90%	15%

s_durch_tür-diagonal

	seite_links		seite_rechts		vorne_groß		hinten_groß	
geg.	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.
11	64%	0%	64%	0%	36%	0%	91%	9%

Tabelle 3.5: Testergebnisse, 2. Repräsentation

[15] Bei diagonaler Fahrt wurde der Lernset aufgrund von Speicherplatzproblemen in KBG aufgespalten. Die Fahrten t72 und t73 wurden in einem extra Lauf betrachtet.

Sensor-Merkmal s_durch_tür-parallel sind die Ergebnisse geringfügig besser als die im vorigen Abschnitt, was die Vollständigkeit anbetrifft. Dies geht allerdings ein wenig zu Lasten der Korrektheit. Bei diagonaler Fahrt sind die Ergebnisse eindeutig schlechter. Insgesamt wird die Hypothese, daß mit der Einführung eines *folgt/2* Prädikates in die Repräsentation und entsprechender Auswertung die Sensor-Merkmale besser zu lernen sind, nicht unterstützt.

Durch die geänderte Repräsentation ändern sich nicht nur die Beschreibungen der Cluster im Ergebnisgraphen, sondern auch ihre Zusammensetzung. Das Ähnlichkeitsmaß bezieht jetzt nicht nur die Basis-Merkmale und ihre Reihenfolge ein, sondern auch die *folgt/2* Literale und insbesondere deren Anzahl. Die Anzahl der *folgt/2* Literale in einer Beobachtung steigt quadratisch (genaueres siehe unten) mit der Länge l der Basis-Merkmal-Sequenz, so daß sich obiger Effekt besonders bei diagonaler Fahrt auswirkt.

Dadurch kommt ein anderes Ergebnis zustande, in dem auf der einen Seite einige vollständige Sequenzen fehlen, auf der anderen Seite zusätzliche Regeln nach dem neuen Muster gelernt werden.

Zu erwähnen ist auch, daß einige Regeln nach dem neuen Muster nicht gebildet werden konnten, weil zu zwei Teilsequenzen X und Y in der generalisierten Clusterbeschreibung gerade das verbindende *folgt/2* Prädikat nicht enthalten war.

In Tabelle 3.6 sind die Laufzeiten des Verfahrens bei geändeter Repräsentation eingetragen. Die höheren Laufzeiten ergeben sich durch die

Sensor-Merkmal	RM	Beobachtungen seite,vorne/hinten	Sequenz-Länge l	KBG-Laufzeit
s_durch_tür	parallel	60, 54	4.0, 3.6	83 Sek.
s_durch_tür	diagonal	40, 36	6.8, 4.4	160 Sek.

Tabelle 3.6: Laufzeiten

Vielzahl der *folgt/2* Literale in den Beobachtungsbeschreibungen, die bei einer Anzahl von l Basis-Merkmal-Literalen einer Beobachtung $l\frac{(l-1)}{2}$ beträgt. Das erklärt auch, warum das Verfahren für den kleineren Lernset bei diagonaler Fahrt trotzdem mehr Zeit benötigt.

Kapitel 4

Clustern von Sequenzen

Alternativ zu KBG wird eine speziell für die Anwendung entwickelte Verfahrensweise des Clustering gewählt. Hierbei werden alle möglichen Teilsequenzen von Basis-Merkmal-Sequenzen der Sensoren während einer Roboterfahrt als Beobachtungen aufgefaßt, um daraus Cluster zu bilden. Die Cluster repräsentieren Sequenzen von Basis-Merkmalen, die Objekte oder Teile von Objekten beim daran Vorbeifahren charakterisieren. Das Auffinden der Cluster soll derart angegangen werden, daß alle Teilsequenzen gezählt werden.

Die Zielsetzung zur Entdeckung der Sensor-Merkmale wird folgendermaßen festgelegt:

> Gegeben sind Basis-Merkmal-Sequenzen von 24 Sensoren aus mehreren Fahrten des Roboters. Sie liegen in der Form von *Cases* vor.
>
> Gesucht sind Teilsequenzen,
>
> - die bei der Vorbeifahrt des Roboters an gleichen Objekten immer wieder auftreten und
> - die bei der Vorbeifahrt an anderen Objekten möglichst nicht oder nur selten auftreten.

Mit dem zweiten Punkt ist ein Aspekt angesprochen, der hinterher so getestet werden kann: Auf der Ebene der übergeordneten handlungsorientierten Wahrnehmungs-Merkmale wird später gelernt werden, welche Kombinationen von Sensor-Merkmalen, aggregiert zu Sensorgruppen-Merkmalen, ein handlungsorientiertes Merkmal beschreiben. Damit das gelernt werden kann, sollte nicht dieselbe Sequenz verschiedene Sensor-Merkmale beschreiben. Sie ist dann nicht speziell genug, weil anhand dieser Sequenz nicht zwischen den verschiedenen handlungsorientierten Wahrnehmungs-Merkmalen unterschieden werden kann. Hierzu können die Regelmengen zu verschiedenen Sensor-Merkmalen daraufhin untersucht werden, ob sie disjunkt sind.

Zu Punkt 1 muß sichergestellt werden, daß die gefundenen Sequenzen insgesamt generell genug sind. Schließlich soll in der Anwendungsphase jedesmal eine Zeitintervallwahrnehmung bei ihrer Ausführung erkannt werden. Dazu müssen von den gefundenen Sequenzen in jeder der entsprechenden Fahrten des Roboters einige vorkommen. Dieser Test ist im Test der Abdeckung des Oberbegriffes[1] enthalten.

Zur Erreichung der oben angegebenen Zielsetzung wird folgende Annahme gemacht:

> Die für die Bildung von Sensor-Merkmalen relevanten Teilsequenzen sind *häufig* in den Basis-Merkmal-Sequenzen enthalten.

Die Zielsetzung ist eine Heuristik für die Auswahl der Teilsequenzen aus den Basis-Merkmal Sequenzen. Die Richtigkeit der Annahme soll im Experiment nachgewiesen werden.

Der Aggregationsschritt der Clusterung besteht in der Zusammenfassung häufig vorkommender, gleicher Teilsequenzen. Die resultierenden Cluster haben die Besonderheit, daß die in ihnen enthaltenen Objekte (die Sequenzen) gleich sind. Ein besonderer Schritt zur Bildung einer Charakterisierung ist nicht nötig: Die Beschreibung eines Clusters ist die Sequenz, die dieser Cluster repräsentiert.

Es geht also darum, in einer Menge von Sequenzen Teilsequenzen zu finden, die mehrfach auftreten. Dazu müssen alle vorkommenden Teilsequenzen jeder Länge gezählt werden.

4.1 Zählen von Teilsequenzen

Zu Beginn dieses Abschnittes wird festgelegt, was gezählt wird:

> Gegeben seien m Sequenzen: Für jede Teilsequenz, die in einer der m Sequenzen enthalten ist, finde die Anzahl der Sequenzen, die diese Teilsequenz enthalten.

Diese Zählweise legt fest, daß im Ergebnis steht, wie viele der Eingabesequenzen eine bestimmte Teilsequenz enthalten.[2] Anders ausgedrückt: Zähle

[1] siehe Kap. 1.2.2

[2] zu unterscheiden von „wie oft kommt die Teilsequenz in den Eingabesequenzen vor“

für alle in den Sequenzen vorkommenden Teilsequenzen X die Sequenzen, die dem regulären Ausdruck **X** genügen. Sequenzen der Form **X*X*...*, in denen das Pattern X mehrfach vorkommt und die damit ebenfalls unter die Ausdrücke der Form **X** fallen, werden mitgezählt, aber nur einfach und nicht nach Anzahl der Vorkommen von X.

Später wird bei der Auswahl der Teilsequenzen, die ein Sensor-Merkmal bilden sollen, das Zähl-Ergebnis ausgewertet. Hierbei werden Teilsequenzen, die in bestimmter Mindestanzahl vorkommen, ausgewählt werden. Zusätzlich wird sich noch herausstellen, daß die Länge der Teilsequenzen eine Rolle spielt und ein weiteres Kriterium bei der Auswahl der Teilsequenzen bildet. Zwei Verfahren, die das Zählen der Teilsequenzen leisten sollen, werden in den folgenden Unterabschnitten diskutiert.

4.1.1 Präfixbäume

Als erstes wird nun der Präfixbaum wie im Original in [Rieger, 1996d] vorgestellt beschrieben. Danach werden die Modifikationen beschrieben, die notwendig sind, um das oben definierte Ergebnis zu erhalten.

Die Präfixbäume wurden für das Anwendungsgebiet des Lernens von Sensor-Merkmalen von Anke Rieger entwickelt. Dort werden für überwachtes Lernen Basis-Merkmal-Sequenzen, die ein Sensor-Merkmal darstellen sollen, als Beispiele direkt vorgegeben. Genauer gesagt, ein Sensor-Merkmal-Fakt vorgegeben wird, unter anderem inklusive der Fahrtnummer, des Zeitintervalls der Wahrnehmung und des Sensors, von dem es wahrgenommen wird. Die entsprechende Sequenz wird dann über einen Abgleich der Argumente aus den Basis-Merkmals-Daten herausgenommen. Das geschieht mit Hilfe des Datenvorbereitungstools, das zu der Beispielinstanz Basis-Merkmale mit der gleichen Fahrt- und Sensornummer heraussucht, die innerhalb des angegebenen Zeitintervalls liegen, und diese nach ihrem zeitlichen Auftreten ordnet. Das Beispiel und die beschreibende Basis-Merkmal-Sequenz bilden zusammen einen „*Case*". In diesem Zusammenhang wird das Beispiel für das Sensor-Merkmal als *Zielprädikat*, die Basis-Merkmal-Sequenz als *beschreibende Prädikate* bezeichnet.

Mit Hilfe des Präfixbaumes wird dann gezählt, welche Sequenzen wie oft in den *Cases* vorkommen. Sequenzen, die in *Cases* mit demselben Sensor-Merkmal oft genug vorkommen, werden als Beschreibung für dieses Sensor-Merkmal genommen. Zusätzlich werden die Vorkommen aller Präfixe der Seqenzen in den *Cases* gezählt.

Aufbau des Präfixbaumes: Jede Kante von einem Knoten zu seinem Nachfolger ist mit einem Basis-Merkmal-Prädikat versehen. Die Argumente der Prädikate sind durch Variablen ersetzt.[3] An dieser Stelle gehen die Bindungen der Argumente aus den *Cases* verloren, was einem Repräsentationswechsel gleichkommt.

Der wesentliche Aspekt der vorher vorhandenen relationalen Repräsentation der Basis-Merkmale ist die Verknüpfung der Wahrnehmungen über die Zeitargumente. Die Relation „zeitliche Abfolge“ der Wahrnehmungen spiegelt sich aber auch implizit in der Reihenfolge der Basis-Merkmale als Listenelemente in den *Cases* wieder. Diese Reihenfolge bleibt auch beim Ersetzen der Bindungen durch Variablen erhalten und wird vom Präfixbaum ausgenutzt.

Ein Knoten repräsentiert die Sequenz, die von den Basis-Merkmalen der Kanten gebildet wird, die auf dem Pfad vom Wurzelknoten zu dem Knoten verfolgt werden. Diese Sequenz ist ein Präfix aller Sequenzen der Folgeknoten, die von hier aus erreicht werden können. Alle Basis-Merkmal-Sequenzen, die sich mit der von dem Knoten repräsentierten Sequenz unifizieren lassen, werden von dem Knoten abgedeckt. Jeder Knoten besitzt zwei Zähler. Der erste Zähler („CC“) enthält die Anzahl der Sequenzen, die komplett von ihm abgedeckt werden. Der zweite Zähler („SC“) enthält die Anzahl der Sequenzen, deren Präfix er abdeckt. Das sind Sequenzen, die von einem der Folgeknoten abgedeckt werden. Weiterhin sind in jedem Knoten die *Cases* der ganz oder teilweise abgedeckten Sequenzen gespeichert.

Die Sequenzen werden nacheinander abgearbeitet. Man beginnt jedesmal an dem Wurzelknoten. Beginnend mit dem ersten Basis-Merkmal wird eine Kante gesucht, die mit diesem Merkmal unifizierbar ist. Ist keine solche Kante vorhanden, wird sie erzeugt. Von dem über diese Kante erreichten Folgeknoten ausgehend, wird mit den weiteren Basis-Merkmalen der Sequenz auf die gleiche Weise verfahren. Der „CC“-Zähler des Knotens, der auf diese Weise nach dem letzten Merkmal der Sequenz erreicht wird, wird um eins hochgezählt. Danach wird mit der nächsten Sequenz fortgefahren.
Beispiel: Aus den Sequenzen „a/6,b/6,c/6“, „b/6,c/6“ und „a/6,b/6“ wird der in Abbildung 4.1 gezeigte Baum erzeugt.

Um in dem Präfixbaum zu zählen, wie oft alle in den Sequenzen enthaltenen Präfixe vorkommen, werden die „SC“-Zähler aller Knoten erhöht, die bei der Abarbeitung einer Sequenz auf dem Weg vom Wurzelknoten aus besucht oder neu erzeugt werden. Ausgenommen davon ist der zuletzt er-

[3] im folgenden dargestellt durch die Kombination Prädikatssymbol/Stelligkeit

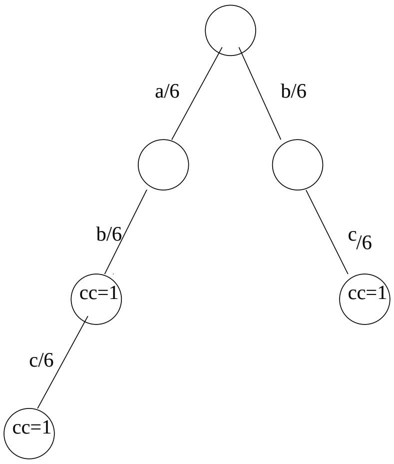


Abbildung 4.1: Präfixbaum, CC-Werte

reichte Knoten. Neu erzeugte Knoten werden mit dem Zählerstand Null für „CC“ und „SC“ initialisiert.
Beispiel: Der Baum aus Abbildung 4.1 sieht dann wie in Abbildung 4.2 gezeigt aus: Die Sequenz „a/6,b/6“ ist somit einmal komplett und einmal als Präfix von „a/6,b/6,c/6“ enthalten. Der Präfix „a/6“ ist zweifach enthalten, „a/6,b/6,c/6“ ist einmal enthalten.

4.1.2 Modifikationen

Mit dem Präfixbaum, wie er bis hierhin beschrieben ist, kann das zu Beginn des Abschnittes definierte Problem noch nicht gelöst werden. Zwei Eigenschaften wurden dazu von mir modifiziert.

> **1. Modifikation:** Mit dem Präfixbaum von Anke Rieger werden komplette Sequenzen gezählt. Die 1. Modifikation bewirkt, daß die Anzahl aller Teilsequenzen gezählt wird. Dabei wird ausgenutzt, daß bereits Zähler für die Präfixe der Sequenzen eingerichtet sind.

Um Teilsequenzen zu zählen, werden erst einmal die „CC“- und die „SC“-Zähler eines Knotens zu einem Zähler („CSC“) addiert. Der „CSC“-Zähler entspricht dann der Anzahl der Präfixe (einschließlich des nullten Präfix), die die von dem Knoten repräsentierte Sequenz enthalten. Es muß noch die Anzahl aller Suffixe und aller ihrer Präfixe gezählt werden. Dazu wird

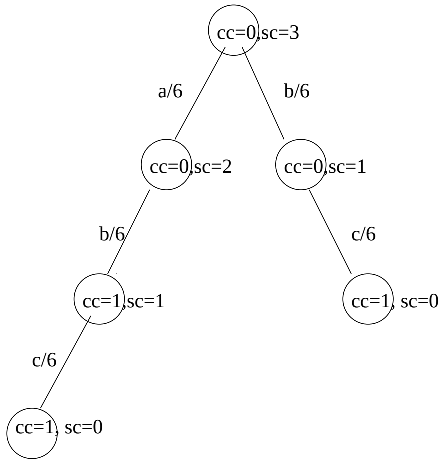


Abbildung 4.2: Präfixbaum

die Eingabe des Präfixbaumes modifiziert: Zu jedem der ursprünglich eingegebenen *Cases* werden *Cases* mit allen Suffixen der enthaltenen Sequenz generiert. Alle *Cases* erhalten als Zielprädikat das des ursprünglich eingegebenen *Cases*. Diese *Cases* werden bei der Bildung des Präfixbaumes zur Eingabe hinzugenommen.

Die Knoten des so entstandenen Baumes repräsentieren alle Präfixe aller Suffixe der Sequenzen, also alle Teilsequenzen. Der „CSC"-Zähler eines Knotens enthält die Häufigkeit der von ihm repräsentierten Teilsequenz (die Anzahl aller Sequenzen, inklusive der aus den Suffixen gebildeten, die die von dem Knoten repräsentierte Sequenz, als Präfix haben).
Im **Beispiel** wird der Präfixbaum zu den obigen Sequenzen zuzüglich der Sequenzen „b/6,c", „c/6" (das sind die Suffixe von „a/6,b/6,c/6"), „c/6" und „b/6" gebildet (Abb. 4.3). An dem Ergebnis kann man ablesen, daß die häufigste Teilsequenz das „b/6" ist, und daß „a/6,b/6", „b/6,c/6" und „c/6" jeweils zweimal vorkommen.

> **2. Modifikation:** Mit dem Präfixbaum von Anke Rieger werden Teilsequenzen, die innerhalb einer Sequenz mehrfach vorkommen, entsprechend oft gezählt. Die 2. Modifikation bewirkt, daß gemäß der Festlegung zu Beginn von 4.1 im Ergebnis steht, wie viele der Sequenzen eine bestimmte Teilsequenz enthalten.

Dazu wird auf die in den Knoten gespeicherten *Cases* zurückgegriffen. In dem Fall, in dem die von einem Knoten repräsentierte Teilsequenzg mehr-

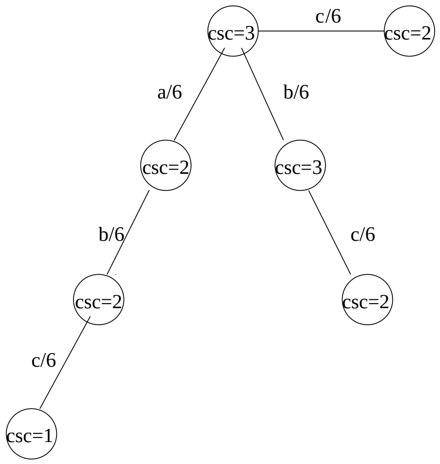


Abbildung 4.3: Präfixbaum mit allen Suffixen

fach in derselben Sequenz vorkommt, deckt der Knoten mehrere *Cases* mit demselben Zielprädikat ab. Diese *Cases* besitzen dasselbe Zielprädikat, da sie Suffixe derselben Sequenz sind. Die „CSC"-Zähler der Knoten werden deshalb um die Anzahl der mehrfach in den gespeicherten *Cases* vorhandenen Zielprädikate vermindert.

Beispiel: Die zuletzt dargestellte Problematik sei anhand eines Beispiels noch einmal aufgezeigt: In Abbildung 4.4 ist die Eingabesequenz „a/6,b/6,c/6,a/6,b/6" und der aus ihr entstandene Präfixbaum gezeigt. In den Knoten sind ihre jeweiligen „CSC" Zählerstände angegeben. In dem rechten Baum ist das Mehrfachzählen gleicher Teilsequenzen innerhalb einer Eingabesequenz nicht mit Hilfe der *Cases* verhindert worden. Aus dem linken Baum geht hervor, daß die Teilsequenzen „a/6,b/6" und „b/6" zweimal in der Eingabe vorkommen. Der rechte Präfixbaum liefert das gewünschte Ergebnis. Dort sind alle vorkommenden Teilsequenzen einfach gezählt, denn in dem Beispiel sind alle bisher abgearbeiteten Zeichenketten Suffixe ein und derselben Sequenz.

4.1.3 Suffixbäume

Die Repräsentation der Basis-Merkmal-Sequenzen kann, wie gezeigt, reduziert werden. Anstelle der relationalen Darstellung reicht es aus, nur die Prädikatssymbole der Basis-Merkmale zu betrachten und als Zeichenkette zu interpretieren. Die Prädikatssymbole werden als Konstanten angesehen,

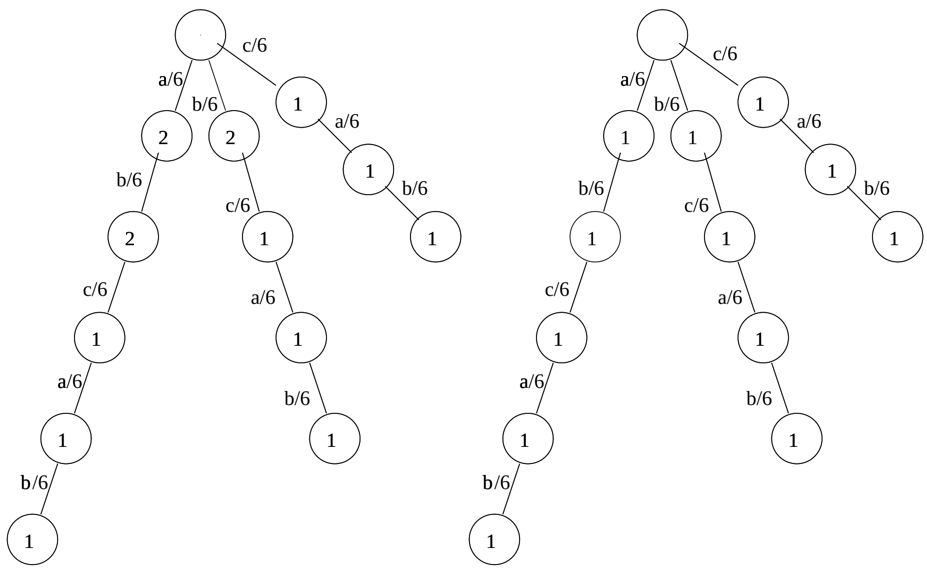


Abbildung 4.4: 2. Modifikation, links: vorher, rechts: nacher

so daß es sich, wenn man diese entsprechend der Sequenz der Basis-Merkmale aneinanderreiht, dann um Zeichenketten handelt. Das entspricht einer Umrepräsentation der *Cases* nach folgender Formel:

$$[s\_hwm\_dummy(Tr, S, T_1, T_{n+1}, RM, SCL), bf_1(Tr, O, S, T_1, T_2, G_1),$$
$$bf_2(Tr, O, S, T_2, T_3, G_2), ..., bf_n(Tr, O, S, T_n, T_{n+1}, G_n)]$$

wird zu

$$TrS = bf_1, bf_2, ..., bf_n.$$

Dazu ein **Beispiel**:
$[s\_durch\_tür\_dummy(t50, s6, 15, 41, parallel, seite),$
$konstant(t50, 0, s6, 15, 17, -2), steigend(t50, 0, s6, 17, 18, 87),$
$konstant(t50, 0, s6, 18, 41, 0)]$

wird zu

t50s6 = konstant,steigend,konstant.
Der Zeichenkette wird ein *Label*, bestehend aus Fahrt- und Sensornummer, hinzugefügt, entsprechend dem Namen der Beobachtungen.

Mit der Interpretation der Basis-Merkmal-Sequenzen als Zeichenketten lassen sich Algorithmen aus dem Bereich des String Matching anwenden.

Die Problemstellung des Zählens von Teilsequenzen, wie oben festgelegt, ist aus dem Bereich des String Matching als „All Patterns"-Problem bekannt:

> Gegeben seien m *Zeichenketten*: Für jede Teilsequenz, die in einer der m *Zeichenketten* enthalten ist, finde die Anzahl der Zeichenketten, die die Teilsequenz enthalten.

In der Literatur findet sich zur Lösung des „All Patterns"-Problems ein Verfahren, das auf dem „Generalized Suffix Tree" GST aufbaut [Hui, 1992]. Für einen Alternativenvergleich wird dieses Verfahren hier vorgestellt. Der GST wird dabei mit dem oben beschriebenen Präfixbaum verglichen, danach wird beschrieben, wie in dem GST gezählt wird. Abschließend wird eine Bewertung vorgenommen, wie gut die gegebene Aufgabenstellung mit den Verfahren behandelt werden kann.

Der GST ist eine Erweiterung des Suffixbaumes. Eine Beschreibung der Konstruktion des Suffixbaumes findet sich in [McCreight, 1976]. Der Suffixbaum wird aus allen Suffixen einer einzelnen Eingabesequenz gebildet, im GST sind die Suffixe aller Eingabesequenzen repräsentiert. Der Suffixbaum ist damit vergleichbar mit dem Präfixbaum, der aus einer einzelnen Eingabesequenz und aller ihrer Suffixe gebildet wird. Im Gegensatz zum Präfixbaum ist eine Kante des Suffixbaumes mit mindestens einem Zeichen beschriftet, es kommen aber auch Kantenbeschriftungen mit Zeichenketten vor. Ein Knoten repräsentiert in diesem Falle die Aneinanderreihung der Kantenbeschriftungen, die auf dem Pfad vom Wurzelknoten zu dem Knoten verfolgt werden. Diese Sequenz ist wie beim Präfixbaum ein Präfix aller Sequenzen der Folgeknoten, die von hier aus erreicht werden können.

Aufbau des Suffixbaumes: Alle Suffixe einer Sequenz werden nacheinander abgearbeitet. Beginnend mit der gesamten Sequenz, werden danach ihre Suffixe in der Reihenfolge ihrer Länge (die längeren zuerst) verarbeitet. Angefangen wird jedesmal an dem Wurzelknoten. Nun wird versucht, den gerade bearbeiteten Suffix soweit wie möglich auf einen Pfad in dem Baum abzubilden. Das heißt: beginnend mit dem ersten Zeichen wird versucht, zeichenweise einem Pfad in dem Baum zu folgen, soweit es möglich ist. Zeichen, die nicht auf diesen Pfad abgebildet werden, bilden den **Restsuffix**. Kann der ganze Suffix auf einen Pfad abgebildet werden, so ist der **Restsuffix** leer.

Es gibt zwei Möglichkeiten, wie der Pfad enden kann. **Fall 1:** Der Pfad endet an einem Knoten, dann gibt es keine davon ausgehende Kante, deren

Beschriftung mit dem nächsten Zeichen beginnt. **Fall 2:** Der Pfad endet auf einer Kante (innnerhalb der Zeichenkette, mit der die Kante beschriftet ist), dann stimmt das nächste Zeichen nicht mit dem nun folgenden Zeichen der Kantenbeschriftung überein.

Am Ende des Pfades wird eine Kante eingerichtet, die den **Restsuffix** als Beschriftung erhält. Dadurch entsteht bei **Fall 2** ein neuer Knoten an der Stelle der gerade verfolgten Kante, an der die neue Kante eingerichtet wird. Das Ende der neuen Kante bildet in beiden Fällen einen Knoten. In **Fall 2** enststeht auch dann ein neuer Knoten auf der gerade verfolgten Kante, wenn der **Restsuffix** leer ist, so als ob eine Nullkante eingefügt würde.

Am Beispiel der Sequenz „a,b,c,a,b" ist in Abbildung 4.5 der zugehörige Suffixbaum dargestellt. Alle Suffixe der Sequenz sind durch jeweils einen der Knoten repräsentiert. In den Knoten, die einen kompletten Suffix re-

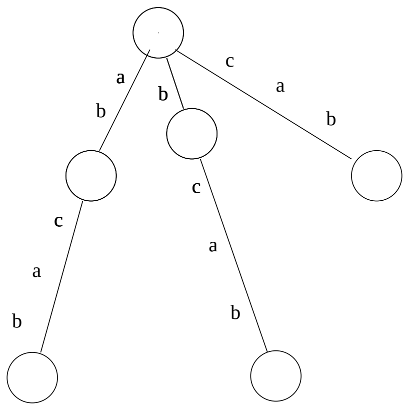


Abbildung 4.5: Suffixbaum

präsentieren, ist zusätzlich vermerkt, zu welcher Eingabesequenz der Suffix gehört, indem dort das oben beschriebene *Label* gespeichert wird. Die *Label* werden benötigt, wenn aus dem Suffixbaum durch Verarbeitung weiterer Eingabesequenzen der GST gebildet wird.

Um in dem GST die Anzahl der Vorkommen einzelner Teilsequenzen in den Eingabesequenzen zu zählen, wird für jeden Knoten die Anzahl verschiedener *Label*, die in dem von dem Knoten ausgehenden Teilbaum liegen, ermittelt. In Abbildung 4.6 sind GST und Präfixbaum zu den Sequenzen T/S-1 = „a,b,c,a,b", T/S-2 = „a,b,a,b" nebeneinandergestellt. Unter den Knoten sind die *Label* der Sequenzen angegeben, die die von dem Knoten repräsentierte Sequenz enthalten. Aus Gründen der Übersichtlichkeit sind

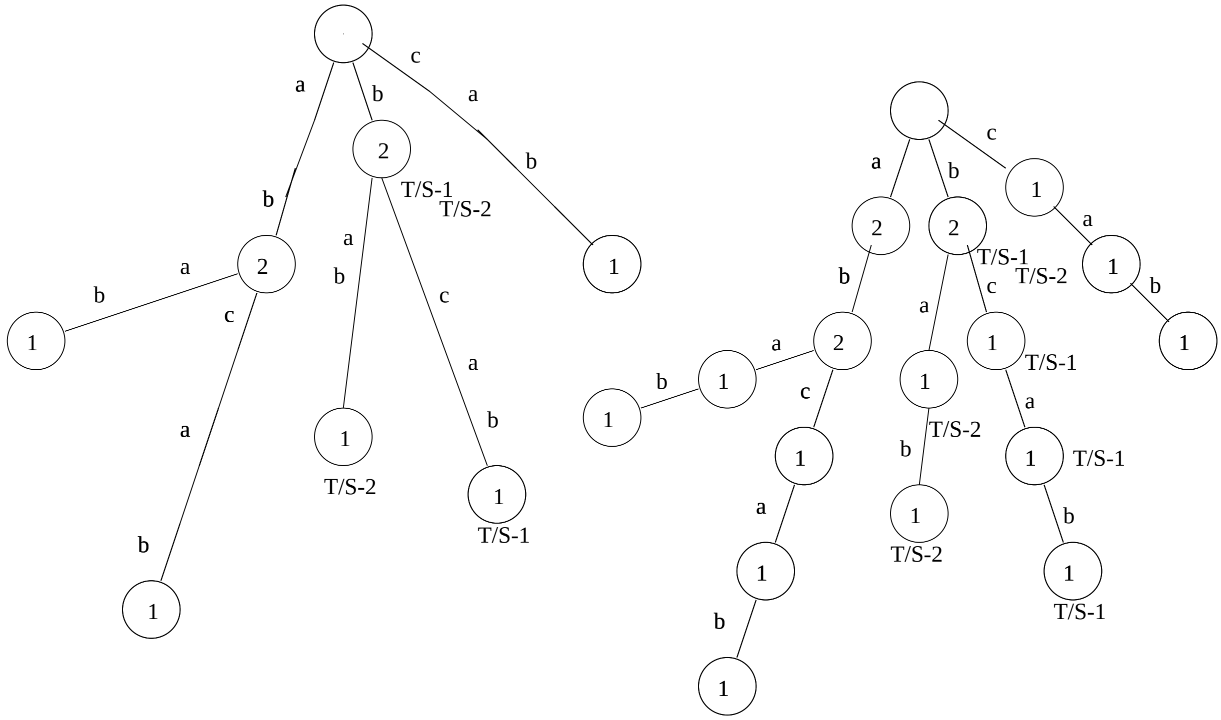


Abbildung 4.6: links GST, rechts modifizierter Präfixbaum

die *Label* nur für den mittleren Teilbaum gezeigt. Der Unterschied zwischen den beiden besteht darin, daß im GST nicht alle Teilsequenzen durch einen Knoten repräsentiert werden, hier zum Beispiel das „a“. Ein Knoten entsteht immer dann nicht, wenn zu einer Teilsequenz keine zweite Teilsequenz mit unterschiedlichem Suffix existiert, der nach obiger Beschreibung zur Einrichtung einer neuen Kante führen würde (auf „a“ folgt jedesmal ein „b“, gäbe es zusätzlich in den Sequenzen irgendwo ein „a,c“, so würde nach dem „a“ ein Knoten eine Verzweigung zwischen „a,b“ und „a,c“ bilden). Somit sind im GST einige Präfixe nicht durch Knoten repräsentiert. In [Hui, 1992] wird vorgeschlagen, als Lösung des “All Patterns“-Problems die von den Knoten des GST repräsentierten Teilsequenzen und die Zählerstände[4] der Knoten zu nehmen. Weil einige Präfixe aber nicht durch Knoten in dem GST repräsentiert sind, ist diese Lösung unvollständig.

[4] Die Zählerstände werden dort, anders als hier beschrieben, durch Lösung des „Color Set Size“-Problems im GST ermittelt.

Aus diesem Grunde wird der modifizierte Präfixbaum gewählt, um die Aufgabe des Zählens von Teilsequenzen nach 4.1 zu leisten. Der modifizierte Präfixbaum bildet somit einen wesentlichen Bestandteil des Verfahrens zum Lernen von Sensor-Merkmalen durch Clustering von Basis-Merkmal-Sequenzen. Aus diesem Grunde zeigt Abbildung 4.7 im Vergleich die Verfahren sowie die Ein- und Ausgaben bei der Benutzung des Präfixbaumes zum Lernen aus Beispielen (oberer Bildteil), und des modifizierten Präfixbaumes zum Lernen aus Beobachtungen (unterer Bildteil). Letzteres Ver-

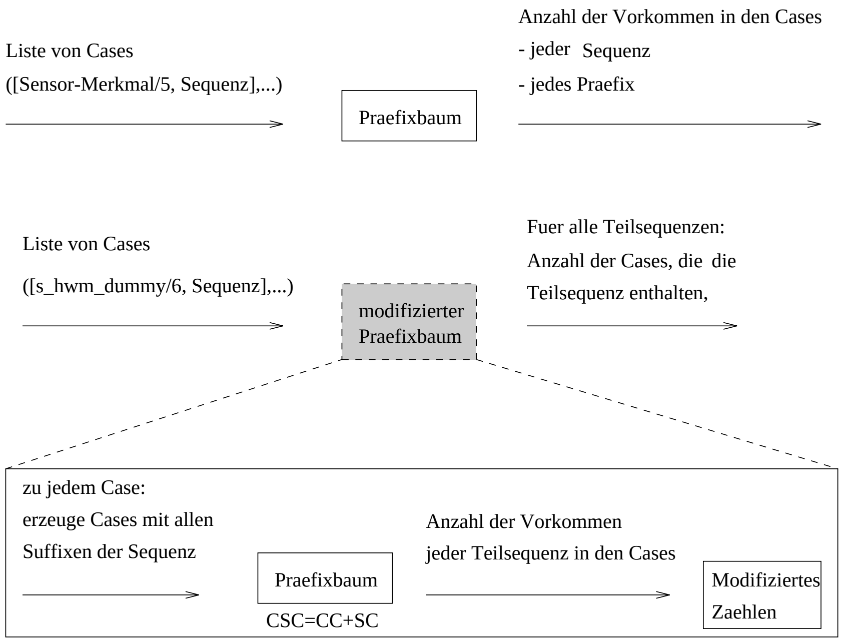


Abbildung 4.7: Zählen mit dem Präfixbaum

fahren wird vervollständigt durch die Auswertung der Zählergebnisse, die im nächsten Abschnitt direkt im Zusammenhang mit den Experimenten beschrieben wird.

4.2 Experiment und Auswertung

Mit dem folgenden Experiment soll gezeigt werden, daß sich die Vollständigkeit der Sensor-Merkmale im Vergleich zu den Ergebnissen aus KBG durch die Anwendung des in diesem Kapitel entwickelten Verfahrens verbessern läßt.

Das Ergebnis eines Zähllaufes mit den *Cases* zu einem der *s_hwm_dummy(_,_,_,_,RM=x,SCL=y)* Zielprädikate soll Regeln für das entsprechende Sensor-Merkmal mit relativer Bewegung RM und der Sensorgruppe SCL ergeben. Die gefundenen Teilsequenzen werden zu Prämissen der Regeln. Eine weitere Prämisse prüft die Sensorklasse. Das s_hwm_dummy-Prädikat ohne den Platzhalter bildet das Zielprädikat dieser Regeln. Der Platzhalter wird jetzt weggelassen, da die so erhaltene Regelmenge ein Sensor-Merkmal, *s_hwm/6*, beschreibt.

Die zu beantwortende Frage ist, wie das Zählen der Teilsequenzen so verwertet wird, daß die daraus gebildete Regelmenge das Sensor-Merkmal beschreibt. Für eine Auswahl der Teilseqenzen werden zwei Parameter eingesetzt, entsprechend der zu Beginn des Kapitels festgelegten Zielsetzung. Diese beiden Parameter sind

a) die Mindestanzahl der *Cases*, die eine Teilsequenz enthalten, und

b) die Mindestlänge einer Teilsequenz.

Der erste Parameter entspricht direkt der Annahme bezüglich relevanter Teilsequenzen. Er spezifiziert einen Mindestwert, ab dem häufig vorkommende Teilsequenzen ausgewählt werden. Der Parameter Mindestlänge hängt indirekt mit der Zielsetzung zusammen, daß derselbe Cluster von Teilsequenzen möglichst nur bei einem Sensor-Merkmal auftreten soll, um Mehrdeutigkeiten zu vermeiden.

Das Problem der Mehrdeutigkeiten ergibt sich nicht nur, wenn gleiche Cluster auftreten. Bereits wenn die Teilsequenz eines Clusters den Präfix der Teilsequenz eines Clusters zu einem anderen Sensor-Merkmal bildet, werden in der Anwendungsphase die beiden entsprechenden Regeln angewandt. Um das zu verhindern, werden nur Teilsequenzen ab einer bestimmten Länge ausgewählt. Denn je länger eine Teilsequenz ist, desto unwahrscheinlicher ist es im allgemeinen, daß sie ein Präfix einer noch längeren, ebenfalls häufig vorkommenden Teilsequenz ist. Somit wird auch hier ein Mindestwert vorgegeben.

Ergebnisse: Die mit diesem Verfahren erhaltenen Sensor-Merkmale stammen aus exakt demselben Lernset wie es im vorigen Kapitel verwendet wurde. Die Auswertung der Ergebnisse wurde ebenfalls in derselben Weise vorgenommen. Tabelle 4.1 zeigt die entsprechenden Prozentzahlen. Der Parameter (a) Mindestanzahl wurde auf 10 gesetzt, die Mindestlänge (b) auf 4. Das Ziel, die Vollständigkeit der Sensor-Merkmale zu verbessern, ist

s_durch_tür-parallel

	seite_links		seite_rechts		vorne_groß		hinten_groß	
geg.	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.
10	100%	6%	100%	3%	100%	22%	100%	22%

s_durch_tür-diagonal

	seite_links		seite_rechts		vorne_groß		hinten_groß	
geg.	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.
11	91%	9%	100%	6%	82%	9%	100%	6%

Tabelle 4.1: Testergebnisse

erreicht worden. Bei paralleler Fahrt beträgt sie hundert Prozent in allen Sensorgruppen. Auch bei diagonaler Fahrt ist das Verfahren in dieser Hinsicht besser, als die beiden Versuche mit KBG. Das so verbesserte Ergebnis geht allerdings zu Lasten der Korrektheit. Besonders bei den vorderen und hinteren Sensoren treten höhere Fehlerraten auf.

Die höhere Vollständigkeit im Vergleich zu KBG läß sich dadurch erklären, daß mit KBG nicht alle Sequenzen gefunden werden, die beim Zählen der Teilsequenzen gelernt wurden. Das Ähnlichkeitsmaß ist nicht direkt auf das Auffinden gleicher Teilsequenzen zugeschnitten. Verschiedene Effekte werden hierbei wirksam.

Beispiel 1: Bei mehreren Sequenzen, die alle dieselbe interessante Teilsequenz enthalten, werden deren Ähnlichkeiten untereinander auch durch die restlichen enthaltenen Basis-Merkmale beeinflußt, so daß verschiedene Ähnlichkeitswerte errechnet werden.

Beispiel 2: Zwei Sequenzen, die aus zwei gemeinsamen Teilsequenzen der Länge 3 bestehen, werden als ähnlicher angesehen, als zwei Sequenzen, denen eine Teilsequenz der Länge 4 gemeinsam ist.

Die Vollständigkeit ist bei der Bewertung der Sensor-Merkmale das wichtigere Kriterium. Die Sensor-Merkmale stehen in der Hierarchie der Wahrnehmungs-Merkmale auf der zweituntersten Ebene. Auf den höheren Ebenen kann noch verhindert werden, daß aus einem falsch abgeleiteten Sensor-Merkmal das entsprechende handlungsorientierte Wahrnehmungsmerkmal abgeleitet wird. Fehlt dagegen ein Sensor-Merkmal, so kann es auch in den höheren Ebenen der Hierarchie nicht abgeleitet werden; mehr dazu in Kapitel 5.2 .

Zum Vergleich der Laufzeiten sind in Tabelle 4.2 die Zeiten für den Aufbau des Präfixbaumes gegeben, welcher den Kern des Verfahrens ausmacht.

Die Zeiten sind gemittelt über den gemessenen Zeitbedarf in den einzelnen Sensorgruppen, wobei die Gruppe der Seitensensoren wieder einiges mehr an Zeit benötigt. Die letzte Spalte zeigt die Gesamtlaufzeit des Verfahrens für ein Sensor-Merkmal mit allen drei Sensorgruppen. Hierin enthalten sind die Schritte zur Erzeugung aller Suffixe sowie die Auswahl der Sequenzen und die Bildung der Regeln, die bei KBG von Hand vorgenommen wurden.

Sensor-Merkmal	RM	Präfixbaum Laufzeit	Gesamtlaufzeit
s_durch_tür	parallel	5 Sek.	53 Sek.
s_durch_tür	diagonal	12 Sek.	113 Sek.

Tabelle 4.2: Laufzeiten

Kapitel 5

Weitere Lernergebnisse

Das Kapitel besteht aus zwei unabhängigen Teilen. Im ersten Teil werden mit dem Verfahren aus dem letzten Kapitel weitere Sensor-Merkmale gelernt. Im zweiten Teil wird ein Vergleich mit Sensor-Merkmalen aus überwachtem Lernen vorgenommen.

5.1 Alle Sensor-Merkmale

In diesem Abschnitt soll gezeigt werden, daß mit dem verwendeten Ansatz weitere Sensor-Merkmale gelernt werden können. Gelernt werden sollten die Sensor-Merkmale *s_durch_tür/6, s_entlang_tür/6* sowie *s_entlang_schrank/6.* Zu den ersten beiden Sensor-Merkmalen wurden Versuche mit relativer Bewegungsrichtung parallel- und diagonal-entlang durchgeführt, zum letzteren ausschließlich Versuche mit parallel-entlang. Die Lernläufe wurden wieder mit simulierten Daten der Roboterfahrten aus Traces-4 durchgeführt.[1] Zu jedem der Sensor-Merkmale wurden alle Fahrten verwendet, die unter Traces-4 jeweils vorhanden sind. Jede Fahrt enthält jeweils ein handlungsorientiertes Wahrnehmungs-Merkmal; Die Fahrten im Detail:

Sensor-Merkmal	RM	Fahrten
s_durch_tür	parallel	t50,t51,t60,t61,t70,t71,t80,t81,t90,t91
s_durch_tür	diagonal	t30,t52,t53,t62,t63,t72,t73,t82,t83,t92,t93
s_entlang_tür	parallel	t54,t64,t74,t84,t94,t55,t65,t75,t85,t95
s_entlang_tür	diagonal	t31,t56,t66,t76,t86,t96
s_entlang_schrank	parallel	t105,t123,t131,t132,t142,t152

[1] *Varianten* „sfmgd_traces30-150 und „bpf_traces30-150_ 1, 2 und 4“

In der folgenden Tabelle sind zu jeder Sensorklasse die Zahlen der *Cases* angegeben - die Zahl der Fahrten multiplipziert mit der Zahl der Sensoren je Sensorklasse. Von den oben genannten Fahrten bildete jeweils eine Fahrt den Testset, der Lernset setzt sich dementsprechend aus den restlichen Fahrten zusammen („leave-one-out“). Bei den „s_entlang_“ Merkmalen wurden nur Sensorwahrnehmungen der dem Merkmal zugewandten Seite des Roboters genutzt, womit sich bei der Klasse „seite“ die Zahl der Sensoren auf fünf halbiert und bei den anderen beiden Klassen vier, statt neun Sensoren beteiligt sind. Die angegebenen *Cases* bilden jeweils eine Eingabe für das Zählen mit dem modifizierten Präfixbaum.

Sensor-Merkmal	RM	seite	vorne_groß	hinten_groß
s_durch_tür	parallel	9*10	9*9	9*9
s_durch_tür	diagonal	10*10	10*9	10*9
s_entlang_tür	parallel	9*5	9*4	9*4
s_entlang_tür	diagonal	5*5	5*4	5*4
s_entlang_schrank	parallel	5*5	5*4	5*4

Verschiedene Lernvarianten Die Lernläufe zu jedem der 5 Sensor-Merkmale werden mit

- verschiedenen Basis-Merkmalen als Hintergrundwissen,
- verschiedenen Parametern für die Auswahl der Teilsequenzen,
- verschiedener Aufteilung der Fahrten in Lernset und Testset,

durchgeführt. Aus der Kombination verschiedener Festlegungen dieser drei Punkte ergeben sich insgesamt sechs Lernvarianten. Diese sind in Tabelle 5.1 aufgeführt. Die unter Testset angegebenen Fahrten sind diejenigen, die beim Lernen aus der Gesamtmenge der Fahrten herausgenommen wurden. Das ist jeweils eine für jedes der 5 Sensor-Merkmale. Alle dort nicht angegebenen Fahrten bilden den Lernset des entsprechenden Sensor-Merkmales (gemäß der Tabelle am Beginn dieses Abschnittes).

Da zur Erzeugung der Basis-Merkmale beim überwachten Lernen der Sensor-Merkmale ein Parameter für die „Empfindlichkeit“ in zwei Lernschleifen ermittelt wird, soll auch hier der Effekt des Parameters auf die Lernergebnisse untersucht werden. Je höher der Parameter, desto eher werden Ausreißer bei den Sensormeßwerten noch mit vorhergehenden Messungen in ein Basis-Merkmal aufgenommen, anstatt ein neues zu bilden. Es werden Basis-Merkmal-*Varianten* verwendet, die mit den Parametern 8 (bp8),

Nr.	Basis-Merkmale	Länge $L\geq$	Anzahl $CSC\geq$	Testset	extra Parameter
1.	bp15	4	4	t90,t92,t95,t96,t123	-
2.	bp10	4	4	t90,t92,t95,t96,t123	-
3.	bp8	4	4	t90,t92,t95,t96,t123	-
4.	bp8	4	4	t50,t52,t54,t96,t123	-
5.	bp8	5	3	t90,t92,t95,t96,t123	-
6.	bp8	4	2	t90,t92,t95,t96,t123	$2*L+CSC\geq 14$

Tabelle 5.1: 6 Lernvarianten

10 (bp10) und 15 (bp15) erzeugt wurden. Um die Ergebnisse vergleichen zu können, wurde zu den Lernläufen mit verschiedenen Basis-Merkmal-*Varianten* jeweils eine Lernvariante erstellt, bei der die restlichen Parameter übereinstimmen (Nr. 1-3).

Zu einer der Basis-Merkmal-*Varianten* wurde entlang der restlichen Parameterdimensionen variiert (Lernvariante Nr. 3-6). Es wurden sowohl andere Fahrten beim Lernen ausgelassen um den Testset zu bilden (Nr. 4), als auch weitere Werte der Parameter (a) Mindestanzahl und (b) Mindestlänge ausprobiert (Nr. 5 und 6). Die Größe des Unterschiedes im Vergleich der Ergebnisse bei den Lernvarianten Nr. 3 und Nr. 4 zeigt die Empfindlichkeit des Verfahrens gegenüber dem verwendeten Lernset. Der Parameter, der die Mindestlänge einer Basis-Merkmal-Sequenz vorschreibt, wurde empirisch ermittelt. Es hat sich herausgestellt, daß Sequenzen der Länge 3 nicht hinreichend diskriminieren. Die Mindestlänge dagegen auf fünf zu setzen, bedeutet zu wenig Sequenzen zu lernen.[2] Auch die Werte für den Parameter Mindestanzahl liegen in einem kleinen Intervall: Mehr als einmal muß eine Sequenz mindestens vorkommen, damit ein Cluster bildbar ist. Um nicht zu speziell zu werden, wird der Mindestwert noch etwas höher angesetzt. Dagegen darf die verlangte Mindestanzahl nicht zu hoch liegen, wenn sehr lange Sequenzen, die hervorragend diskriminieren, gelernt werden sollen.

In Lernvariante Nr. 6 ist deswegen aus den Parametern (a) Mindestanzahl und (b) Mindestlänge ein weiterer gebildet worden, der einen „Tradeoff" zwischen den beiden Parametern erlaubt: Für jedes Basis-Merkmal, um das eine Sequenz länger als vier ist, reduziert sich die Mindestanzahl um zwei, so daß bei einer Länge von 6 die Mindestanzahl nur noch 3 beträgt. Eine Sequenz der Länge vier muß mindestens siebenmal vorkommen.

[2] vgl. später die Ergebnisse hierzu

Nach den Lernläufen: Von den in einer Lernvariante gelernten Basis-Merkmal-Sequenzen werden die unbrauchbaren, die das nichtssagende Merkmal „keine_bewegung“ enthalten, gelöscht.

Alle anderen Sequenzen werden als Bestandteil von Metafakten zu entsprechenden Regelmodellen in MOBAL geladen. Dort werden Regeln zur Ableitung von Sensor-Merkmalen gebildet. Um diese mit den Regeln inferieren zu können, werden die Basis-Merkmal-Sequenzen[3] aller (zu Beginn des Abschnittes) angegebenen Fahrten als Hintergrundwissen hinzugeladen. Ebenfalls als Hintergrundwissen werden Fakten zu den Sensorklassen[4] der einzelnen Sensoren während aller Fahrten eingefügt. Obwohl die Sensor-Merkmale der beiden Seitenklassen „links“ und „rechts“ zusammen in einer Sensorgruppe gelernt wurden, ist in den Tests zwischen den beiden unterschieden worden, um erstens die Korrektheit bei „entlang-“Fahrten überprüfen zu können, und zweitens weil später Regelmodelle höherer Ebenen der Repräsentationshierarchie verwendet werden sollen.

Die in MOBAL inferierten Sensor-Merkmale bilden zusammen mit den Beispielen der handlungsorientierten Wahrnehmungsmerkmale die Eingabe eines Programmes, das die Abdeckung der Beispiele in der in 1.2.2 beschriebenen Weise zählt. Ein detailliertes Ergebnis ist für Lernvariante Nr. 2 in Anhang B zu finden. Alle Ergebnisse sind in den Tabellen 5.2 - 5.6 zusammengefaßt und werden nun nach Sensor-Merkmalen getrennt kommentiert. In der Zeile unter dem Strich sind die gemittelten Prozentwerte der einzelnen Sensorgruppen noch einmal zu einem Wert für die Vollständigkeit und die Korrektheit zusammengefaßt. Diese Werte erlauben eine allgemeine Aussage darüber, wie gut das Sensor-Merkmal insgesamt gelernt werden konnte.

s_durch_tür-parallel Das beste Ergebnis zu diesem Sensor-Merkmal wurde mit Lernvariante Nr. 2 gewonnen, was für den Parameter *bp10* bei den Basis-Merkmalen spricht. Was die Vollständigkeit betrifft, wird in den Fahrten t70, t71 keines für die Sensorgruppe „seite_links“ und in der Fahrt t91 kein Merkmal für „hinten_groß“ abgeleitet. Damit liegt die Vollständigkeit der vorderen und hinteren Sensor-Merkmale zusammengenommen bei 95% . Die seitlichen Sensor-Merkmale zusammengenommen sind nur zu 90% vollständig. Dafür sind die Sensor-Merkmale der Seitensensoren zu 100% korrekt. Da die Sensor-Merkmale der Sensorgruppe „vorne_groß“ in 2 von 33 Fahrten, die nicht parallel durch eine Tür gingen, abgeleitet wird, beträgt

[3] bp8, bp10 oder bp15: jeweils der Parameter der Lernvariante
[4] *Variante* sclasses_mg_t30-150

Nr.	seite_links		seite_rechts		vorne_groß		hinten_groß	
	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.
1.	60%	0%	100%	0%	100%	6%	30%	0%
2.	80%	0%	100%	0%	100%	6%	90%	15%
3.	60%	6%	100%	0%	100%	6%	80%	12%
4.	70%	18%	100%	12%	100%	0%	90%	15%
5.	40%	0%	70%	0%	80 %	0%	30%	0%
6.	80%	18%	100%	12%	100%	21%	100%	15%
ges.	65%	7%	95%	4%	97%	7%	70%	10%
ges.						abgedeckt	82%	neg. 7%

Tabelle 5.2: s_durch_tür-parallel

der Fehler hier $\frac{2}{33}$*100% ≈ 6%. Der Fehler bei den hinteren Sensoren beträgt $\frac{5}{33}$*100% ≈ 15%. Bei den Lernvarianten Nr. 1 und Nr. 3, die sich von Nr. 2 nur durch die verwendeten Basis-Merkmale unterscheiden, sind die Ergebnisse insgesamt weniger vollständig. Insbesondere die hinteren Sensoren in Lernvariante Nr. 1 leiten nur wenige Sensor-Merkmale ab. Lernvariante Nr. 3 liefert zusätzlich unkorrekte Sensor-Merkmale der Seitensensoren.

Beim Lernen mit Lernvariante Nr. 4 wurde im Vergleich zu Nr. 3 eine andere Fahrt ausgelassen. Es werden mehr Sensor-Merkmale abgeleitet, korrekte wie unkorrekte. Diese Tendenz ist bei Lernvariante Nr. 6 mit dem extra Parameter noch größer. Die Ergebnisse mit Lernvariante Nr. 5 sind, wenn auch korrekt, deutlich zu unvollständig. Die Mindestlänge der Basis-Merkmal-Sequenzen von fünf führt dazu, daß die Sensor-Merkmale der Fahrt t85 komplett fehlen.

Positiv zu vermerken ist, daß jeweils in den Fahrten des Testsets, t50 oder t90, die Sensor-Merkmale richtig vorhergesagt werden.

s_durch_tür-diagonal In Lernvariante Nr. 2 ist das Verhältnis von Vollständigkeit und Korrektheit wiederum am ausgewogensten. Die Sensor-Merkmale der Seitensensoren sind zu 94% korrekt (die der vorderen und hinteren Sensoren zu 100% und zu 97%). Die Vollständigkeit beträgt hier 100%, bei den vorderen und hinteren Merkmalen immerhin noch 82% beziehungsweise 91%.

Im Vergleich zu den parallelen Fahrten durch die Tür werden bei Diagonalfahrten längere Basis-Merkmal-Sequenzen wahrgenommen. Das erklärt die etwas besseren, aber weiterhin indiskutablen Ergebnisse mit Lernvariante Nr. 5, die eine Mindestlänge von 5 vorschreibt.

Nr.	seite_links		seite_rechts		vorne_groß		hinten_groß	
	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.
1.	73%	0%	91%	6%	82%	0%	91%	0%
2.	100%	6%	100%	6%	82%	0%	91%	3%
3.	100%	31%	100%	6%	82%	0%	91%	3%
4.	100%	31%	100%	6%	82%	0%	100%	3%
5.	64%	3%	91%	6%	45%	0%	73%	3%
6.	100%	41%	100%	12%	100%	6%	100%	6%
ges.	90%	19%	97%	7%	74%	1%	91%	3%
ges.						abgedeckt 88%		neg. 7%

Tabelle 5.3: s_durch_tür-diagonal

Auch dieses Sensor-Merkmal wird in den Fahrten des Testsets, t52 beziehungsweise t92, ebenfalls abgeleitet.

Nr.	seite_links und _rechts		vorne_groß		hinten_groß	
	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.
4.	70%	3%	80%	9%	0%	0%
5.	30%	0%	0%	0%	70%	0%
6.	70%	3%	40%	6%	80%	12%
ges.	57%	2%	40%	5%	50%	4%
ges.				abgedeckt 49%		neg. 4%

Tabelle 5.4: s_entlang_tür-parallel

s_entlang_tür-parallel Dieses ist von den untersuchten Sensor-Merkmalen mit einem Gesamtergebnis von nur 49% Vollständigkeit das am schwersten zu lernende. Zu den Lernvarianten Nr. 1 - 3 ist in mehr als der Hälfte der Fahrten überhaupt kein Sensor-Merkmal gelernt worden. Daher sind die Ergebnisse auch in der Tabelle 5.4 nicht aufgeführt.

Die Fahrt t95 die in Lernvariante Nr. 4 anstelle von t54 eingesetzt wird scheint einige wichtige Sequenzen zu enthalten. Jedenfalls konnte hier, zusammen mit Lernvariante Nr. 6 das beste Ergebnis erzielt werden.

Es gilt zu beachten, daß bei diesem und den weiteren „entlang"-Merkmalen entweder nur die linke, oder nur die rechte Seite das Merkmal wahrnehmen kann. Der Roboter ist in den Fahrten mit gerader Nummer (t54, t64 ...) mit der linken Seite, in den Fahrten mit ungerader Nummer (t55, t65 ...) mit der rechten Seite an der Tür entlanggefahren. In der Tabelle sind aus diesem Grunde die Ergebnisse der Seitensensoren zusammengefaßt, wobei darauf geachtet wurde, ob das Merkmal auf der richtigen Seite abgeleitet wurde.

Das gute Ergebnis mit Lernvariante Nr. 6 läßt sich damit erklären, daß durch den extra Parameter das Auswahlkriterium besser an die verminderte Zahl der *Cases* angepaßt werden konnte. Dementsprechend wurden mit Lernvariante Nr. 5 ebenfalls bessere Ergebnisse erzielt, als mit der vergleichbaren Lernvariante Nr. 3. In Lernvariante Nr. 5 ist die Mindesthäufigkeit um eins niedriger, was bewirkt, daß wenigstens in 3 Fahrten eine Regel gelernt wurde, die für einen seitlichen Sensor das Merkmal ableitet, obwohl die Mindestlänge um eins höher liegt.

Zu der Fahrt t95 des Testsets wird in Lernvariante Nr. 6 ein Sensor-Merkmal richtig vorhergesagt. Ein Sensor-Merkmal der Fahrt t54, die zu Lernvariante Nr. 4 den Testset bildet, wird nicht vorhergesagt.

Das entsprechende Sensor-Merkmal mit der relativen Bewegung „diagonal" konnte aufgrund der dort wahrgenommenden längeren Basis-Merkmal-Sequenzen besser gelernt werden. Deshalb wurde versucht, „s_entlang_tür-parallel" mit Sequenzen der Länge 3 zu lernen. Die Menge der erhaltenen Sequenzen war aber nicht geeignet, das Sensor-Merkmal von anderen zu unterscheiden. Hier sind Verbesserungen erwünscht.

Nr.	seite_links und _rechts		vorne_groß		hinten_groß	
	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.
1.	33%	0%	33%	0%	33%	0%
2.	83%	0%	83%	3%	50%	0%
3.	83%	8%	83%	3%	50%	0%
4.	83%	8%	83%	3%	50%	0%
5.	33%	0%	83%	0%	33%	0%
6.	83%	8%	83%	3%	50%	0%
ges.	66%	4%	75%	2%	44%	0%

ges. abgedeckt 62% neg. 3%

Tabelle 5.5: s_entlang_tür-diagonal

s_entlang_tür-diagonal In allen Fahrten liegt die Tür zur rechten des Roboters. Dort kann das Sensor-Merkmal in Lernvariante Nr. 2 von den rechten und den vorderen Sensoren in 5/6 der Fälle, von den hinteren in 3/6 der Fälle abgeleitet werden. Fahrt t86 ist ein Sonderfall, die Tür liegt dort zu nahe an der nächsten Wand. Mit (ohne) Berücksichtigung dieser Tatsache liegt damit die Vollständigkeit bei 100% (83%) beziehungsweise 60% (50%). Bis auf einen „Ausreißer" in Fahrt t53, sind alle Sensor-Merkmale in Lernvariante Nr. 2 korrekt abgeleitet.

Die Ergebnisse der weiteren Lernvarianten sind wieder etwas schlechter. Wie bei den anderen Sensor-Merkmalen auch, tendiert Lernvariante Nr. 1 zu unvollständigen, Lernvariante Nr. 3 zu unkorrekten Ergebnissen. Die Lernvarianten Nr. 4-6 liefern keine neuen Erkenntnisse, was die Variation der Parameter anbetrifft. Zu der Testfahrt t96 sind bei Lernvariante Nr. 2 entsprechende Sensor-Merkmale abgeleitet worden.

Nr.	seite_links und _rechts		vorne_groß		hinten_groß	
	abgedeckt	neg.	abgedeckt	neg.	abgedeckt	neg.
1.	100%	11%	100%	0%	100%	3%
2.	100%	3%	50%	0%	83%	0%
3.	100%	3%	100%	0%	100%	0%
4.	100%	3%	100%	0%	100%	0%
5.	100%	3%	83%	0%	83%	0%
6.	100%	3%	100%	0%	100%	14%
ges.	100%	4%	89%	0%	94%	3%
ges.					abgedeckt 94%	neg. 2%

Tabelle 5.6: s_entlang_schrank-parallel

s_entlang_schrank-parallel Trotz der geringen Anzahl Fahrten in den Lernvarianten konnte das Merkmal gut gelernt werden. In allen Fahrten liegt die Tür links vom Roboter. Bei den Lernvarianten Nr. 3 und 4 werden alle Sensor-Merkmale vollständig abgeleitet, viele mehrmals, auch in der Testfahrt t123. Ebenfalls bei Lernvariante Nr. 3 und 4 sind die Merkmale korrekt. Die in der Tabelle dort als unkorrekt vermerkten Sensor-Merkmale stammen aus der Fahrt t95 und sind leicht erklärbar. Direkt neben der Tür, an der der Roboter in Fahrt t95 parallel entlangfährt, steht auch ein Schrank.

Folgerungen: Zusammenfassend wird festgestellt, daß der verwendete Ansatz zum Lernen von Sensor-Merkmalen auf verschiedene handlungsorientierte Wahrnehmungen angewendet werden kann. Die Qualität der Ergebnisse ist abhängig von den gewählten Parametern, der Größe der Eingabe und besonders von der Art der durch das jeweilige Objekt hervorgerufenen Basis-Merkmal-Sequenzen.

Um eine Vorstellung der Sensor-Merkmale geben zu können, sind zu einigen ausgewählten die gelernten Regeln im Anhang abgedruckt, unter dem Abschnitt B.2. Das Sensor-Merkmal „durch_tür-parallel" wird durch einige wenige Regeln beschrieben. Zu allen drei Sensorklassen sind die Sequenzen nicht länger als vier Basis-Merkmale. Anders dagegen die Sequenzen bei diagonalen Fahrten. Das Sensor-Merkmal „s_entlang_tür-diagonal" wird nur durch Sequenzen der Längen fünf und sechs beschrieben. Die Regelmenge zu dem Sensor-Merkmal „s_durch_dür-diagonal" ist um einiges größer. Allein das Merkmal der seitlichen Sensoren wird durch 45 Sequenzen beschrieben, deren Längste aus acht Basis-Merkmalen besteht.

Die Regelmengen zeigen, daß teilweise Wahrnehmungen der Umgebung, von der sich ein Objekt abhebt, zur Charakterisierung in das Merkmal mit einbezogen werden. Die Merkmale sind demnach insofern nicht Ortsinvariant, als sie für verschiedene Umgebungen („Schrank direkt hinter der Tür", „Blumentopf in geringem Abstand vor der Tür") neu gelernt werden müssen. Zur Unterscheidung dieser Fälle bietet es sich an, jeweils ein eigenes Sensor-Merkmal zu bilden.

Bei der Untersuchung der Regelmengen wurde beobachtet, daß die Regelmengen der fünf Sensor-Merkmale vollkommen disjunkt sind![5] Damit gestaltet sich das Durchlaufen der Hierarchie der Wahrnehmungs-Merkmale sehr einfach, da die Sensor-Merkmale bereits zwischen den verschiedenen handlungsorientierten Wahrnehmungs-Merkmalen diskriminieren.

Nicht disjunkt sind, aufgrund des Designs überlappender Sensorklassen, die Regelmengen verschiedener Sensorklassen zu demselben Merkmal. Dort gibt es einige Überschneidungen.

5.2 Vergleich mit überwachtem Lernen

In diesem Teil der Arbeit soll ein Vergleich der gelernten Sensor-Merkmale mit den beim überwachten Lernen verwendeten Kantenmerkmalen stattfinden. Aufgrund der Unterschiedlichkeit der Merkmale können sie nicht direkt verglichen werden. Stattdessen wird für die Sensor-Merkmale „s_durch_tür-parallel" und „s_durch_tür-diagonal" gezeigt, wie in sehr einfacher Weise handlungsorientierte Wahrnehmungs-Merkmale aus ihnen abgeleitet werden

[5] Diese Untersuchung wurde ganz zu Beginn von Kapitel 4 motiviert.

können. Der Vergleich soll auch deswegen auf oberster Ebene der handlungsorientierten Wahrnehmungen geführt werden, da die Ableitung der richtigen Merkmale auf dieser Ebene ein in sich abgeschlossenes Zielkriterium bildet.

Die Ergebnisse werden verglichen mit den handlungsorientierten Wahrnehmungs-Merkmalen, die nach dem Lernen der Regeln in den drei Hierarchieebenen mit den Kantenmerkmalen abgeleitet wurden. Die Hypothese ist: Mit den unüberwacht gelernten Sensor-Merkmalen können auf der Ebene der handlungsorientierten Wahrnehmungs-Merkmale bessere Ergebnisse erzielt werden, als es bisher mit überwacht gelernten Kantenmerkmalen gelungen ist.

Sensorgruppen-Merkmale: Ein Aspekt des Lernens auf dieser Hierarchieebene, welche aus einer Vielzahl von Sensorgruppen überhaupt vorkommen, entfällt, weil hier nur die wenigen mit Hilfe von KBG [6] empirisch ermittelten Sensorgruppen vorgegeben wurden. Es muß nur noch gelernt werden, wie viele Sensor-Merkmale mit jeweils derselben Sensorgruppe zusammen ein Sensorgruppen-Merkmal ableiten.

Dazu wird dasselbe Bewertungskriterium wie auch schon bei den Sensor-Merkmalen angewandt: Die Abdeckung des Oberbegriffes, des entsprechenden handlungsorientierten Wahrnehmungs-Merkmales. Die Zahl der Sensor-Merkmale, die zusammen ein gleichnamiges Sensorgruppen-Merkmal ableiten, ergibt sich aus zwei gegensätzlichen Kriterien. Sie sollte:

a) So niedrig liegen, daß die **Vollständigkeit** des Sensorgruppen-Merkmales im Vergleich zu dem Sensor-Merkmal nicht sinkt.

b) So hoch liegen, daß die **Korrektheit** möglichst verbessert wird.

Handlungsorientierte Wahrnehmungs-Merkmale: Zur Ableitung handlungsorientierter Wahrnehmungs-Merkmale wird das in [Klingspor et al., 1996] in Abschnitt 5.3 verwendete Regelmodell in abgewandelter Form benutzt. Zu dem dort angegebenen Regelmodell können Regeln gelernt werden, die aus zwei Sensorgruppen-Merkmalen, die gleichzeitig wahrgenommen werden, ein handlungsorientiertes Wahrnehmungs-Merkmal ableiten. Das Regelmodell wird hier so abgewandelt, daß die Sensorgruppen-Merkmale nicht gleichzeitig, sondern nur innerhalb eines Intervalls wahrgenommen werden müssen.

[6] siehe Anhang A.2

Im Prinzip soll gelernt werden, in **welchen** beiden Sensorgruppen ungefähr gleichzeitig dasselbe Sensorgruppen-Merkmal wahrgenommen werden muß, damit das entsprechende handlungsorientierte Wahrnehmungs-Merkmal abgeleitet werden kann. Bei der überschaubaren Zahl an Sensorgruppen kann hier dieser Schritt auch von Hand vorgenommen werden. Insbesondere zu dem Merkmal *durch_tür* läßt sich die folgende Regel leicht angeben:

```
sg_durch_tuer(Tr, seite_rechts ,T1,T2,diagonal) &
sg_durch_tuer(Tr, seite_links  ,T3,T4,diagonal) &
-10 < (T1-T3) =< 0
--> durch_tuer(Tr,T1,T4,diagonal)
```

Die Regel wird auch mit „parallel" anstelle von „diagonal" verwandt.

Experiment: Für den Vergleich mit überwachtem Lernen werden in beiden Fällen die gleichen Datensätze zum Lernen der drei Hierarchieebenen benutzt. Es sind dies die bereits in den Kapiteln 3 und 4 verwendeten Daten (siehe Abbildung 3.1 in Kapitel 3).

Aus den in 4.2 gelernten Sensor-Merkmalen werden direkt Sensorgruppen-Merkmale abgeleitet. Die Zahl der Sensor-Merkmale, die zusammen ein gleichnamiges Sensorgruppen-Merkmal ableiten, liegt damit gemäß des Kriteriums a) bei 1. Dieser Wert wurde hier empirisch ermittelt. Bereits bei einem Wert von 2 sinkt die **Vollständigkeit** um einiges. Um die Ausmaße zeigen, ist der Fall in der Ergebnistabelle unten auch gezeigt (Spalte sf_sgf_2). Der Wert kann hier so niedrig angesetzt werden, da die Ergebnisse bereits bei den Sensor-Merkmalen so gute Korrektheitswerte aufweisen, daß Fehler durch die Regel auf der Ebene der handlungsorientierten Wahrnehmungsmerkmale kompensiert werden. Den Wert höher anzusetzen, macht erst bei unregelmäßigeren Roboterdaten Sinn, bei denen weniger **korrekte** Ergebnisse zu erwarten sind.[7]

Mit der oben angegebenen Regel werden für „parallel" und „diagonal", aus den Sensorgruppen-Merkmalen *durch_tür/4*-Merkmale abgeleitet. Die Evaluierung der abgeleiteten handlungsorientierten Wahrnehmungs-Merkmale kann dieses Mal direkt anhand der gegebenen Beispiele vorgenommen werden. In Tabelle 5.7 ist das Ergebnis gezeigt. Das Ergebnis wird

[7] Dort kann der Wert (gegebenenfalls unter Verzicht auf 100% **Vollständigkeit**), zugunsten der **Korrektheit** nach Kriterium b) höher gewählt/gelernt werden.

durch_tür-parallel				
	sf_sgf_1		sf_sgf_2	
geg.	abgedeckt	neg.	abgedeckt	neg.
10	100%	0%	90%	0%

durch_tür-diagonal				
	sf_sgf_1		sf_sgf_2	
geg.	abgedeckt	neg.	abgedeckt	neg.
11	91%	0%	82%	0%

Tabelle 5.7: *durch_tür/4* mit Sensor-Merkmalen aus Kapitel 4

plausibel, wenn man sich die Ergebnisse zu den Sensor-Merkmalen der rechten und linken Seite (Kapitel 4, Tabelle 4.1) ansieht: Die Vollständigkeit ist aufgrund der zuletzt angewendeten Regel höchstens gleich der geringeren Vollständigkeit einer der beiden Sensorgruppen (Spalte „sf_sgf_1“). Werden Sensorgruppen-Merkmale erst bei mindestens 2 gleichen Sensor-Merkmalen abgeleitet, sinkt die Vollständigkeit (Spalte „sf_sgf_2“). Die Zahl der unkorrekt abgeleiteten handlungsorientierten Wahrnehmungs-Merkmale liegt bei 0, da keines der unkorrekt abgeleiteten Sensor-Merkmale in beiden Sensorgruppen „seite_links“ **und** „seite_rechts“ wahrgenommen wurde.

Zum Vergleich sind in Tabelle 5.8 die Ergebnisse zitiert, die mit den Kantenmerkmalen erzielt werden konnten [Klingspor et al., 1996]. Das „Training Set 1“ der in Abschnitt 5.3 des Artikels verwendeten „training sets“ entspricht dem hier verwendeten Lernset (Tabelle 3.1). Die Ergebnisse sind hier getrennt nach paralleler und diagonaler Fahrt angegeben. Die Werte für die Vollständigkeit sind im Vergleich zu den in dem genannten Artikel angegebenen Werten schlechter, da dort einige Fahrten nachträglich aus der Bewertung hinausgenommen wurden. Die Messungen aus dem Simulator waren für Kantenmerkmale teilweise unbrauchbar. In Klammern sind die Ergebnisse angegeben, die sich ergeben, wenn die Fahrten nicht bewertet werden.

In dem genannten Artikel sind beim Testen der Korrektheit einige Fahrten mehr benutzt worden, dies wurde hier berücksichtigt; ebenso die Tatsache, daß hier die Korrektheit als Anzahl der Fehler dividiert durch die Anzahl möglicher Fehler definiert worden ist.

Fazit: Die Tabellen zeigen, daß die Lern- und Testergebnise handlungsorientierter Wahrnehmungs-Merkmale mit Sensor-Merkmalen, die beim dem

	geg.	abgedeckt	neg.
durch_tür-parallel	10 (7)	40% (57%)	6%
durch_tür-diagonal	11 (10)	73% (80%)	0%

Tabelle 5.8: *durch_tür/4* mit Kantenmerkmalen

Zählen von Teilsequenzen gelernt werden, verbessert werden können.

In den Vergleich spielen viele Aspekte der Vorgehensweise auf den mittleren Hierarchieebenen hinein. Eine ähnliche Vorgehensweise wie hier bei der Ableitung der Sensorgruppen-Merkmale könnte die Ergebnisse der Kantenmerkale auf der Sensorgruppen-Ebene wahrscheinlich verbessern. Allerdings muß gesagt werden, daß das, was hier aufgrund der Übersichtlichkeit weniger Sensorgruppen empirisch ermittelt werden konnte, in dem dortigen Lernschritt dann nicht herausgekommen ist.

Auch ist es nicht gelungen Regeln zu lernen, die verschiedene Kantenmerkmale zu handlungsorientierten Wahrnehmungs-Merkmalen zusammensetzen (zum Beispiel *s_jump-s_line-s_jump* in Folge ergeben ein *durch_tür*). Die Regeln, die gelernt werden konnten, leiten aus *sg_jump* oder *sg_concave* das Merkmal *durch_tür* ab. Es ist gar nicht klar, wie im Gegensatz dazu Regeln für zum Beispiel das Merkmal *entlang_schrank* gelernt werden soll. Dieses Problem ergibt sich bei den hier vorgeschlagenen Sensor-Merkmalen nicht.

Kapitel 6

Zusammenfassung und Ausblick

6.1 Zusammenfassung

Oberstes Ziel dieser Arbeit war das Lernen intervallbezogener Sensor-Merkmale. Insbesondere sollte ein unüberwachter Ansatz gewählt werden, unter Anwendung von Verfahren des Conceptual-Clustering.

Zur Lösung wurde ein Vorgehen gewählt, bei dem die einzelnen Sensor-Merkmale nach Oberbegriffen getrennt gelernt werden. Dazu wurde nach Clustern in den Basis-Merkmals-Wahrnehmungen gesucht, die im Rahmen jeweils eines der Oberbegriffe stattgefunden haben. Die Cluster wurden auf der Basis verschiedener Ähnlichkeiten in den Basis-Merkmals-Wahrnehmungen gebildet. Die untersuchten Formen von Ähnlichkeiten sind:

1. Das Ähnlichkeitsmaß des Verfahrens KBG, daß sich auf 2 verschiedene Repräsentationen der Basis-Merkmals-Wahrnehmungen in KBG unterschiedlich auswirkt.

2. Die Gleichheit von Teilsequenzen innerhalb der (Sequenzen von) Basis-Merkmals-Wahrnehmungen (modifizierter Präfixbaum).

Die Bildung der Cluster erforderte Aggregations- und Charakterisierungsschritte. Die bei der Charakterisierung erhaltenen Beschreibungen wurden in einem weiteren Schritt zu Regeln, die Sensor-Merkmale aus Basis-Merkmalen ableiten, umformuliert. Aggregations- und Charakterisierungsmethoden wurden im Zusammenhang mit den eingesetzten Ähnlichkeitsmaßen gewählt:

zu 1. Die Cluster- (Aggregation) und Generalisierungsmethode (Charakterisierung) des Verfahrens KBG.

zu 2. Mehrfach auftretende, gleiche Teile von Basis-Merkmal-Sequenzen einzelner Sensoren werden zu Clustern aggregiert. Die Basis-Merkmal-Sequenzen der Cluster bilden gleichzeitig ihre Charakterisierug. Ein Generalisierungsschritt ist nicht notwendig.

Die Cluster wurden anhand ihrer Charakterisierungen bewertet, um eine Auswahl vorzunehmen. Kriterien hierzu wurden bei den beiden Verfahren angegeben. In beiden Fällen wird zur Bewertung der Charakterisierungen der Parameter Mindestlänge hinzugezogen, der sich empirisch aus der Anwendungsdomäne ergibt.

Zu den beiden Verfahren wurden in Kapitel 3 und 4 Experimente mit einem Sensor-Merkmal, das die Wahrnehmung einer Fahrt durch eine Tür beschreibt, mit jeweils denselben Lern- und Testsets durchgeführt und untereinander verglichen.

Mit dem Verfahren, das auf dem modifizierten Präfixbaum aufbaut, wurden in Kapitel 5.1 Experimente zu weiteren Sensor-Merkmalen mit verschiedenen Parametersätzen und Lernvarianten durchgeführt, die gezeigt haben, daß auf diese Weise flexibel intervallbezogene Wahrnehmungs-Merkmale gelernt werden können.

Schließlich sind in Kapitel 5.2 die mit dem Verfahren des modifizierten Präfixbaum gelernten Sensor-Merkmale verglichen worden, mit Sensor-Merkmalen aus überwachtem Lernen, zu denen sie eine Alternative darstellen sollten. Die Ergebnisse haben gezeigt, daß die unüberwacht gelernten Sensor-Merkmale eine Verbesserung auf der Ebene der handlungsorientierten Wahrnehmungs-Merkmale ermöglichen.

6.2 Ausblick

Aufbauend auf den aus den Ergebnissen gewonnenen Erkenntnissen sind verschiedene Verbesserungen denkbar. Ideen zur Verbesserung beim Einsatz der Verfahren KBG und modifizierter Präfixbaum setzen bei dem Bewertungskriterium zur Auswahl relevanter Cluster an. Für den Parameter Mindestlänge wäre eine flexible Handhabung wie ansatzweise bereits mit dem

„extra Parameter“ der Lernvariante Nr. 6 in Kapitel 5 versucht erwünscht. Beispielsweise könnten kürzere Sequenzen einzeln daraufhin untersucht werden, ob sie nicht aufgrund selten auftretender Basis-Merkmale eine diskriminierende Beschreibung eines Sensor-Merkmales ermöglichen.

Ein weiterer Ausblick richtet sich auf die Anwendung des Ansatzes dieser Arbeit auf reale Daten von Fahrten des mobilen Roboters. Erste nicht repräsentative Versuche mit realen Daten haben gezeigt, daß die dort wahrgenommenen Basis-Merkmal-Sequenzen länger, und untereinander wesentlich verschiedener sind, so daß gleiche Teilsequenzen von Basis-Merkmalen zur Beschreibung von Sensor-Merkmalen nicht lernbar sind. Demnach könnten zum Beispiel die in Kapitel 3.4 mit der 2. Repräsentation lernbaren Sensor-Merkmale der Form „nach einem Sequenzteil X folgt irgendwann ein Sequenzteil Y“ interessant werden. Da allerdings KBG mit dem Ähnlichkeitsmaß bei der entsprechenden Repräsentation zu keinen guten Ergebnissen führt[1], bietet es sich an, andere Verfahren zu entwickeln, die auf die Entdeckung von Mustern in sequenziellen Daten ausgerichtet sind [Laird, 1993; Mannila et. al., 1995].

Ein anderer Schritt in die Richtung, nicht nur exakt gleiche Teilsequenzen zur Beschreibung eines Sensor-Merkmales zuzulassen, wäre der Versuch, das in [Rieger, 1996a] vorgestellte *Post-Processing* auf die unüberwacht gelernten Sensor-Merkmale anzuwenden. Dort wird Hintergrundwissen über irrelevante Basis-Merkmale dazu benutzt, rekursive Regeln zu bilden. An den Stellen, an denen in den gelernten Basis-Merkmal-Sequenzen irrelevante Basis-Merkmale stehen, dürfen diese bei der Ableitung der Sensor-Merkmale dann fehlen oder beliebig oft vorkommen.

Beispielgenerierung Im Zusammenhang mit realen Daten sei noch auf den Vorteil hingewiesen, den die Vorgabe der handlungsorientierten Wahrnehmungs-Merkmale im Vergleich zur Beispielgenerierung bei den Kantenmerkmalen mit sich bringt: Zur Erzeugung der Beispiele für die Kantenmerkmale wird die Information benötigt, welche Kanten jeweils das Echo für die Messungen geliefert haben. Dieses Wissen ist aber nur bei von einem Simulator erzeugten Sensordaten erhältlich. Bei realen Daten müßte eine neue Vorgehensweise erdacht werden. Bei dem in dieser Arbeit verwendeten Ansatz ist die in Kapitel 1.2.1 dargestelle Vorgehensweise auch auf reale Daten anwendbar.

[1] siehe Kapitel 3.4, beziehungsweise die Erklärung in 4.2

Anhang A

KBG-Ergebnisse

A.1 Ergebnisgraph

```
GR-1  = [keine_messung,abfallend,stabil,aufsteigend,keine_messung].
GR-2  = [keine_messung,stabil,etwas_passiert,keine_messung].
GR-3  = [keine_messung,stabil,aufsteigend,keine_messung].
GR-4  = [keine_messung,etwas_passiert,aufsteigend,stabil].
GR-5  = [keine_messung,stabil,aufsteigend_sprung,stabil].
GR-6  = [keine_messung,stabil,keine_messung].

GR-7  = [etwas_passiert,aufsteigend_sprung,stabil], keine_messung.
GR-8  = [keine_messung,abfallend,stabil], keine_messung.
GR-10 = [keine_messung,stabil].
GR-12 = [keine_messung,stabil], keine_messung.
```

Abbildung A.1: Ergebniscluster

Abbildung A.2! zeigt das KBG-Ergebnis zu einer Eingabe von 42 Sequenzen, von jeweils 6 Seitensensoren aus 7 simulierten Roboterfahrten. Pro Zeile ist jeweils ein Knoten des Graphen (GR-X), beziehungsweise ein Cluster angegeben. Die Cluster sind entsprechend der Reihenfolge ihrer Entstehung im Clusterprozeß numeriert. Hinter den Doppelpunkten sind die Beobachtungen der Cluster angegeben. Das sind entweder Cluster, die eher entstanden sind oder an den Blattknoten die *Case*-Sequenzen.

```
* GR-18 comes from : (GR-17 T80S18)
* GR-17 comes from : (GR-15 GR-16)
* GR-16 comes from : (GR-14 GR-8)
* GR-15 comes from : (GR-10 GR-13)
* GR-14 comes from : (T81S7 GR-11)
* GR-13 comes from : (GR-6 GR-12)
* GR-12 comes from : (GR-2 GR-3)
* GR-11 comes from : (GR-9 GR-5)
* GR-10 comes from : (T60S5  T60S7  T71S5  T71S6 T71S7)
* GR-9  comes from : (GR-7 T51S19)
* GR-8  comes from : (GR-1 T71S18)
* GR-7  comes from : (GR-4 T80S7)
* GR-6  comes from : (T80S17 T80S19 T81S18 T81S19)
* GR-5  comes from : (T50S17 T50S18 T50S19 T50S5  T50S6
                      T50S7  T51S17 T51S18 T51S6  T60S17 ...)
* GR-4  comes from : (T51S5  T51S7  T60S18 T60S6  T70S5 T70S6
                      T70S7  T80S6  T81S5  T81S6)
* GR-3  comes from : (T70S17 T71S17)
* GR-2  comes from : (T70S19 T81S17)
* GR-1  comes from : (T70S18 T71S19)
```

Abbildung A.2: s_durch_tür(parallel,kleine_seite)

Für einige Cluster ist in Abbildung A.1! ihre Beschreibung angegeben. Basis-Merkmale in eckigen Klammern bilden jeweils eine zusammenhängende Sequenz, ansonsten ist **keine** zeitliche Reihenfolge gegeben. Um die Struktur des Graphen zeigen zu können, ist in Abbildung A.3 eine andere Darstellung gewählt.

A.2 Sensorgruppen

Hinweise für die Bildung der Sensorgruppen wurden mit dem im Verlauf der Arbeit eingesetzten Conceptual-Clustering Verfahren KBG erarbeitet.

Mit Hilfe des Verfahrens können die Wahrnehmungen verschiedener Sensoren in Cluster mit ähnlichen Wahrnehmungen gruppiert werden. Die Be-

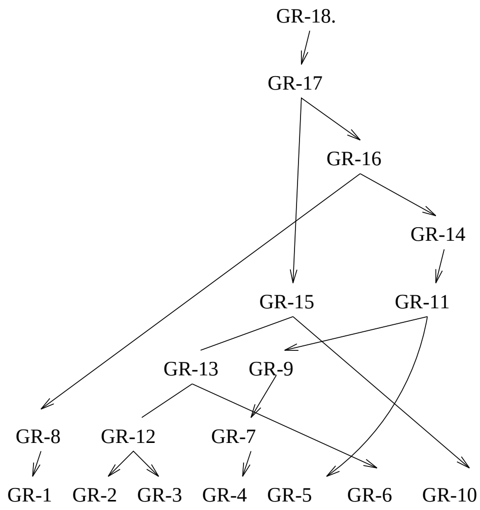


Abbildung A.3: s_durch_tür(parallel,kleine_seite)

obachtung, welche Sensoren (über die wahrgenommenen Basis-Merkmal-Sequenzen) Cluster bilden, soll dazu dienen Hinweise zu liefern, welche Sensorgruppen beim Lernen der Sensor-Merkmale verwandt werden können.

Alle Basis-Merkmal-Sequenzen einer einzelnen Roboterfahrt bilden die Eingabe des Verfahrens,[1] so daß die an den verschiedenen Stellen des Roboters angeordneten Sensoren anhand von unter denselben Bedingungen entstandenen Wahrnehmungen verglichen werden. Welche Sensoren sich im allgemeinen ähnlich verhalten, läßt sich nur aus den Ergebnissen mehrerer Fahrten zusammen ablesen.

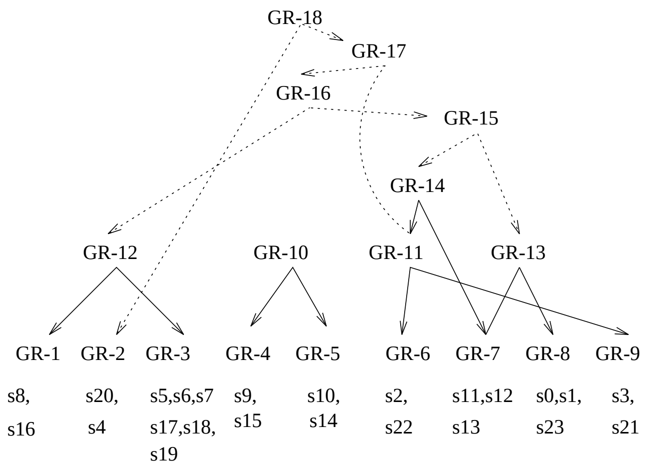
Aufgrund der Anordnung der 24 Sensoren des Roboters gibt es grundsätzlich Seitensensoren, Ecksensoren sowie vordere und hintere Sensoren. Es wurde versucht, in den Clustern von 10 verschiedenen Fahrten des Roboters durch eine Tür Gemeinsamkeiten zu entdecken. Einer der Graphen ist in Abbildung A.4 abgebildet.

Folgende Beobachtungen wurden gemacht:

- Es lassen sich Gruppen gemäß der Anordnung der Sensoren wiederfinden, speziell die Seitensensoren bilden häufig Cluster.

- Eine feinere Einteilung als die Grobeinteilung gemäß der Anordnung

[1] Unter Verwendung der einfacheren der beiden in Kapitel 3 ausprobierten Repräsentationen.



GR-10 : s9,s10,s14,s15 sind hintere Ecksensoren

GR-11: s21,s22,s2,s3 sind vordere Ecksensoren

GR-12: s5,s6,s7,s8,s16,s17,s18,s19 sind Seitensensoren und benachbarte Ecksensoren

GR-13: s23,s0,s1,s11,s12,s13 sind vordere und hintere Sensoren

GR-14: s9 bis s15 sind hintere- und Ecksensoren

Abbildung A.4: Gruppierungen von Sensoren

der Sensoren läßt sich nicht finden. Die Sensoren bilden mal kleinere, mal größere Cluster in wechselnder Konstellation.

- Die Ecksensoren bilden keine eigenständige Sensorklasse. Es kommen zwar Cluster vor, die nur Ecksensoren enthalten, aber auch Cluster, die Ecksensoren mit Seitensensoren, sowie Ecksensoren mit vorderen und hinteren Sensoren mischen.

Zur Bildung der Sensorgruppen werden daraus folgende Schlüsse gezogen: Die vorderen und hinteren Ecksensoren bilden einerseits jeweils eine Gruppe mit den vorderen und hinteren Sensoren, andererseits sind sie auch Bestandteil der Seitensensoren, indem zu der Gruppe der Seitensensoren die direkt benachbarten Ecksensoren hinzugenommen werden. Das Ergebnis sind große, überlappende Klassen. Die Sensorgruppen „seite“, „vorne_groß“ und „hinten_groß“, die beim Lernen der Sensor-Merkmale verwendet werden sollen, sind in Abbildung A.5 am Roboter gezeigt. Die beiden Rechtecke bilden zusammen die Gruppe „seite“, in der zwischen „links“ und „rechts“ nicht

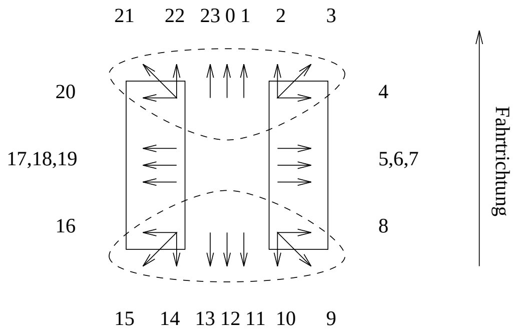


Abbildung A.5: Die verwendeten Sensorgruppen

unterschieden wird.

Die Abbildung der Sensorgruppen auf die darin enthaltenen Sensoren ist abhängig von der Fahrtrichtung. Entsprechende *sclass/5* Fakten werden aufgrund der vorgenommenen Einteilung der Sensoren und den Basis-Handlungs-Merkmalen, die angeben, welche Seite des Roboters sich in Fahrtrichtung befindet, gebildet.

A.3 2. Repräsentation: Regelergebnisse

Im Folgenden sind die aus der Repräsentation mit dem folgt/2 Prädikat von KBG gelernten Regeln für die Seitensensoren des Sensor-Merkmales *s_durch_tür,parallel* gezeigt.

Die Regeln sind in Form von Metafakten in der Syntax von MOBAL/RDT gegeben. Die entsprechenden Metaprädikate verlangen, daß die Basis-Merkmale über ihre Zeitpunkte und durch Fahrt- und Sensornummer miteinander verbunden sind und leiten daraus das Sensor-Merkmal mit der relativen Bewegung und der Sensorgruppe als Konstanten ab.[2] Die Kleinbuchstaben im Prädikatssymbol der Metafakten bedeuten, daß a: nach dem ersten Basis-Merkmal (b,c: nach dem zweiten, dritten ...) der Anfangszeitpunkt des folgenden Basis-Merkmales gleich, **oder größer** sein darf, als der jeweilige Endzeitpunkt des vorhergehenden Basis-Merkmales. Das Druckformat der Metafakten ist so gewählt, daß die Basis-Merkmal-Sequenzen jeweils in die Lücken eingesetzt zu lesen sind:

```
bf_sf_4(                          parallel,s_through_door,side).
```

[2]Die englischen Begriffe entsprechen jeweils dem deutschen Äquivalent.

```
decr_peak,decreasing,something_h,incr_peak,
decr_peak,something_h,incr_peak,stable,
no_measurement,something_h,incr_peak,stable,
no_measurement,stable,incr_peak,stable,
no_measurement,stable,something_h,no_measurement,
no_measurement,stable,increasing,no_measurement,
something_h,something_h,incr_peak,stable,

bf_sf_4a(                      parallel,s_through_door,side).
decr_peak,something_h,incr_peak,stable,
no_measurement,something_h,incr_peak,stable,

bf_sf_4b(                      parallel,s_through_door,side).
no_measurement,something_h,incr_peak,something_h,

bf_sf_5(                       parallel,s_through_door,side).
no_measurement,something_h,incr_peak,something_h,stable,
decr_peak,something_h,something_h,incr_peak,stable,
no_measurement,stable,increasing,incr_peak,stable,

bf_sf_5b(                      parallel,s_through_door,side).
no_measurement,something_h,incr_peak,something_h,stable,

bf_sf_5c(                      parallel,s_through_door,side).
decr_peak,decreasing,something_h,something_h,no_measurement,
```

Anhang B

Zählen von Teilsequenzen

B.1 Abgeleitete Sensor-Merkmale

```
LERNSET 2.
***** durch_tuer,parallel *****
$ Fahrten mit korrekt abgeleiteten Merkmalen
t50 seite_links 3 seite_rechts 4 vorne 3 hinten 4
t51 seite_links 4 seite_rechts 4 vorne 3 hinten 2
t60 seite_links 4 seite_rechts 2 vorne 4 hinten 1
t61 seite_links 5 seite_rechts 2 vorne 3 hinten 2
t70 seite_links 0 seite_rechts 5 vorne 3 hinten 2
t71 seite_links 0 seite_rechts 1 vorne 3 hinten 1
t80 seite_links 1 seite_rechts 3 vorne 2 hinten 1
t81 seite_links 2 seite_rechts 4 vorne 3 hinten 4
t90 seite_links 4 seite_rechts 5 vorne 1 hinten 1
t91 seite_links 4 seite_rechts 3 vorne 4 hinten 0
$ Unkorrekt abgeleitete Merkmale
t30 seite_links 0 seite_rechts 0 vorne 0 hinten 1
t62 seite_links 0 seite_rechts 0 vorne 1 hinten 1
t83 seite_links 0 seite_rechts 0 vorne 0 hinten 2
t92 seite_links 0 seite_rechts 0 vorne 0 hinten 1
t93 seite_links 0 seite_rechts 0 vorne 2 hinten 1
$ Fehlende Fahrten

***** durch_tuer,diagonal *****
$ Fahrten mit korrekt abgeleiteten Merkmalen
```

```
t30 seite_links 4 seite_rechts 5 vorne 4 hinten 3
t52 seite_links 4 seite_rechts 5 vorne 2 hinten 4
t53 seite_links 4 seite_rechts 5 vorne 5 hinten 3
t62 seite_links 5 seite_rechts 5 vorne 2 hinten 3
t63 seite_links 1 seite_rechts 5 vorne 5 hinten 3
t72 seite_links 1 seite_rechts 1 vorne 4 hinten 0
t73 seite_links 3 seite_rechts 5 vorne 3 hinten 1
t82 seite_links 3 seite_rechts 2 vorne 0 hinten 4
t83 seite_links 1 seite_rechts 4 vorne 0 hinten 2
t92 seite_links 5 seite_rechts 2 vorne 5 hinten 3
t93 seite_links 4 seite_rechts 5 vorne 5 hinten 4
$ Unkorrekt abgeleitete Merkmale
t51 seite_links 1 seite_rechts 0 vorne 0 hinten 1
t90 seite_links 0 seite_rechts 1 vorne 0 hinten 0
t91 seite_links 1 seite_rechts 0 vorne 0 hinten 0
t96 seite_links 0 seite_rechts 3 vorne 0 hinten 0
$ Fehlende Fahrten

***** entlang_tuer,parallel *****
$ Fahrten mit korrekt abgeleiteten Merkmalen
t64 seite_links 0 seite_rechts 0 vorne 1 hinten 0
t74 seite_links 0 seite_rechts 0 vorne 1 hinten 0
t84 seite_links 0 seite_rechts 0 vorne 1 hinten 0
t94 seite_links 0 seite_rechts 0 vorne 1 hinten 0
$ Unkorrekt abgeleitete Merkmale
t80 seite_links 0 seite_rechts 0 vorne 1 hinten 0
t90 seite_links 0 seite_rechts 0 vorne 1 hinten 0
$ Fehlende Fahrten
t54 t55 t65 t75 t85 t95

***** entlang_tuer,diagonal *****
$ Fahrten mit korrekt abgeleiteten Merkmalen
t31 seite_links 0 seite_rechts 1 vorne 1 hinten 1
t56 seite_links 0 seite_rechts 4 vorne 2 hinten 2
t66 seite_links 0 seite_rechts 3 vorne 3 hinten 3
t76 seite_links 0 seite_rechts 1 vorne 3 hinten 0
t96 seite_links 0 seite_rechts 1 vorne 1 hinten 0
$ Unkorrekt abgeleitete Merkmale
t53 seite_links 0 seite_rechts 0 vorne 1 hinten 0
```

```
$ Fehlende Fahrten
t86

***** entlang_schrank,parallel *****
$ Fahrten mit korrekt abgeleiteten Merkmalen
t105 seite_links 3 seite_rechts 0 vorne 3 hinten 3
t123 seite_links 3 seite_rechts 0 vorne 0 hinten 3
t131 seite_links 4 seite_rechts 0 vorne 3 hinten 5
t132 seite_links 4 seite_rechts 0 vorne 3 hinten 3
t142 seite_links 4 seite_rechts 0 vorne 0 hinten 2
t152 seite_links 3 seite_rechts 0 vorne 0 hinten 0
$ Unkorrekt abgeleitete Merkmale
t95 seite_links 0 seite_rechts 3 vorne 0 hinten 0
$ Fehlende Fahrten
```

B.2 Einige Sequenzen

Die hier gezeigten Metafakten enthalten die gelernten Sequenzen. Sie sind der Lernvariante Nr. 2 entnommen. Die entsprechenden Metaprädikate verlangen, daß die Basis-Merkmale über ihre Zeitpunkte und durch Fahrt- und Sensornummer miteinander verbunden sind und leiten daraus das Sensor-Merkmal mit der relativen Bewegung und der Sensorgruppe als Konstanten ab. Das Druckformat der Metafakten ist so gewählt, daß die Basis-Merkmal-Sequenzen jeweils in die Lücken eingesetzt zu lesen sind:

```
*** s_durch_tuer-parallel ***
bf_sf_4(                      ,parallel,s_through_door,side).
 decr_peak,decreasing,something_h,something_h
 decr_peak,something_h,incr_peak,stable
 incr_peak,stable,incr_peak,stable
 no_measurement,something_h,incr_peak,stable
 no_measurement,something_h,increasing,incr_peak
 no_measurement,stable,incr_peak,stable
 something_h,something_h,incr_peak,stable

bf_sf_4(                 ,parallel,s_through_door,frontlarge).
 decr_peak,decreasing,something_h,something_h
 decr_peak,something_h,incr_peak,stable
 something_h,something_h,incr_peak,stable
```

```
 straight_to,something_h,incr_peak,stable
 straight_to,something_h,no_measurement,straight_to

bf_sf_4(               ,parallel,s_through_door,backlarge).
 no_measurement,something_h,increasing,incr_peak
 no_measurement,something_h,increasing,straight_away
 no_measurement,something_h,something_h,straight_away
 no_measurement,something_h,straight_away,stable

*** s_entlang_schrank-parallel *** (nur Seitenklasse)
bf_sf_4(                            ,parallel,s_desk,side).
 decr_peak,stable,incr_peak,stable
 incr_peak,stable,decr_peak,stable
 stable,decr_peak,stable,incr_peak
 stable,incr_peak,stable,decr_peak

*** s_entlang_tuer-diagonal ***
bf_sf_5(                        ,diagonal,s_along_door,side).
decreasing,stable,increasing,something_h,something_h
no_measurement,decreasing,stable,increasing,something_h
bf_sf_6(                        ,diagonal,s_along_door,side).
no_measurement,decreasing,stable,increasing,
     something_h,something_h

bf_sf_5(                  ,diagonal,s_along_door,frontlarge).
decreasing,decreasing,increasing,something_h,straight_to
decreasing,something_h,increasing,something_h,straight_to
increasing,something_h,straight_to,something_h,something_h
something_h,increasing,something_h,straight_to,something_h
bf_sf_6(                   ,diagonal,s_along_door,frontlarge).
something_h,increasing,something_h,
     straight_to,something_h,something_h

bf_sf_5(                   ,diagonal,s_along_door,backlarge).
increasing,decr_peak,increasing,something_h,something_h
increasing,increasing,decr_peak,increasing,something_h
bf_sf_6(                   ,diagonal,s_along_door,backlarge).
increasing,increasing,decr_peak,increasing,
     something_h,something_h
```

```
*** s_durch_tuer-diagonal *** (nur Seitenklasse)
bf_sf_4(                       ,diagonal,s_through_door,side).
decreasing,something_h,something_h,incr_peak
decreasing,something_h,something_h,something_h
decreasing,something_h,something_h,straight_away
incr_peak,increasing,no_measurement,straight_to
increasing,incr_peak,increasing,no_measurement
no_measurement,decreasing,incr_peak,decreasing
...

bf_sf_5(                      ,diagonal,s_through_door,side).
decreasing,something_h,something_h,straight_away,decreasing
increasing,incr_peak,increasing,no_measurement,straight_to
no_measurement,decreasing,something_h,something_h,str_away
something_h,decreasing,something_h,something_h,incr_peak
...

bf_sf_6(                      ,diagonal,s_through_door,side).
something_h,increasing,incr_peak,
     increasing,no_measurement,straight_to
something_h,something_h,increasing,
     incr_peak,increasing,no_measurement
something_h,something_h,something_h,
     incr_peak,increasing,no_measurement
something_h,something_h,something_h,
     something_h,incr_peak,increasing
straight_to,something_h,something_h,
     increasing,incr_peak,increasing

bf_sf_7(                       ,diagonal,s_through_door,side).
something_h,something_h,increasing,incr_peak,
     increasing,no_measurement,straight_to
straight_to,something_h,something_h,increasing,
     incr_peak,increasing,no_measurement

bf_sf_8(                       ,diagonal,s_through_door,side).
straight_to,something_h,something_h,increasing,
     incr_peak,increasing,no_measurement,straight_to
```

Literaturverzeichnis

[Bisson, 1992a] G. Bisson
Conceptual Clustering in a First Order Logic Representation
In: Proceedings of 10th ECAI 1992 Wien
Bernd Neumann (Hrsg.), John Wiley & Sons, England

[Bisson, 1992b] G. Bisson
Learning in FOL with a Similarity Measure
In: Proceedings of 10th AAAI Conference, San-Jose, 82-87 AAAI-Press, California 1992

[LRI, 1992] Laboratoire de Recherche en Informatique
Deliverable Guides for KBG
Université Paris-sud, Bâtiment 490, 91450 Orsay Cedex France

[Dillmann et al., 1993] R. Dillmann, J. Kreuziger, F. Wallner
PRIAMOS – An Experimental Platform for Reflexive Navigation
In: IAS-3: Intelligent Autonomous Systems, Kap. 18
Groen, Hirose und Thorpe (Hrsg.), IOS Press 1993

[Emde, 1994a] W. Emde
Inductive Learning of Characteristic Concept Descriptions from Small Sets of Classified Examples
In: Proceeding of ECML-94, Vol. 784 of lecture Notes in AI, 103-124
Springer, Berlin 1994

[Emde, 1994b] W. Emde
Inductive Learning of Characteristic Concept Descriptions
In: Proc. 4th Int. Workshop in Inductive Logic Programming (ILP-94), 51-69

S. Wrobel (Hrsg.) 1994

[Everitt, 1980] B. Everitt
Cluster Ananlysis
Heinemann Educational, London 1980

[Gennari et al., 1989] J. H. Gennari, P. Langley, D. Fisher
Models of Incremental Concept Formation
In: Artificial Intelligence, Nr. 40 (1989), 11-61
Elsevier, Amsterdam

[Hui, 1992] L.C.K. Hui
Color set size problem with applications to string matching
In: Combinatorial Pattern Matching, Lecture Notes in Computer Science, 230-243
A. Apostolico, M. Crochemore, Z. Galil und U. Manber (Hrsg.), Springer Verlag 1992

[Kietz/Wrobel, 1991] J.U. Kietz, S. Wrobel
Controlling the Complexity of Learning in Logic through Syntactic and Task-Oriented Models
In: Inductive Logic Programming, 353-359
S. Muggleton (Hrsg.), Academic Press 1991

[Klingspor, 1994] V. Klingspor
Enhancing Model-Based Learning for Its Application in Robot Navigation
LS-8 Report 5, Univ. Dortmund 1994

[Klingspor et al., 1996] V. Klingspor, K.J. Morik, A.D. Rieger
Learning concepts from Sensor Data of a Mobile Robot
Machine Learning Journal, erscheint 1996
Kluwer Academic Publishers, Boston

[Klingspor et. al., 1994] V.Klingspor, M. Kaiser, J. del R. Millan, M. Accame, F. Wallner, R. Dillmann
Achieving Intelligence in Mobility -
Incorporating Learning Capabilities in Real-World Mobile Robots
LS-8 Report 13, Univ. Dortmund 1994

[Laird, 1993] P. Laird
Identifying and Using Patterns in Sequential Data
In: Algorithmic Learning Theory, 4th International Workshop, 1-18
Springer-Verlag 1993

[Lebowitz, 1987] M. Lebowitz
Experiments with incremental concept formation: UNIMEM
Machine Learning Journal 2 (1987), 103-138
Kluwer Academic Publishers, Boston

[Mannila et. al., 1995] H. Mannila, H. Toivonen, A.I. Verkamo
Discovering frequent episodes in sequences
Department of Computer Science, P.O.Box 26, FIN-00014 University of Helsinki, Finnland

[McCreight, 1976] E.M. McCreight
A Space-Economical Suffix Tree Construction Algorithm
In: Journal of the ACM, Nr. 23(2) (1976), 262-272

[Michalski/Stepp, 1980] R.S. Michalski, R.E. Stepp
Learning from Observation: Conceptual Clustering
In: Machine Learning: An AI Approach
Michalski, Carbonell, Mitchell (Hrsg.), 1983

[Mitchell, 1982] T. Mitchell
Generalization as search
In: Artificial Intelligence, Nr. 18(2) 1982
Elsevier, Amsterdam

[Morik et al., 1993] K. Morik, S. Wrobel, J.-U. Kietz, W. Emde
Kap. 8: Concept Formation
In: Knowledge Acquisition and Machine Learning: Theory, Methods and Applications
Academic Press, London 1993

[Morik/Rieger, 1993] K.J. Morik, A.D. Rieger
Learning Action-oriented Perceptual Features for Robot Navigation
LS-8 Report 3, Univ. Dortmund 1993

[Rieger, 1995a] A.D. Rieger
Inferring Probabilistic Automata from Sensor Data for Robot Navigaton
LS-8 Report 18, Univ. Dortmund 1995

[Rieger, 1995b] A.D. Rieger
Data Preparation for Inductive Learning in Robotics
LS-8 Report 19, Univ. Dortmund 1995

[Rieger, 1996a] A.D. Rieger
Optimizing Chain Datalog Programs and their Inferece Procedures
LS-8 Report 20, Univ. Dortmund 1995

[Rieger, 1996b] A.D. Rieger
A Data Preparation Tool for Relational Inductive Learning in Robotics
In: Proc. of the 13th European Meeting on Cybernetics and Systems Research, 1153-1159
R. Trappl (Hrsg.), 1996

[Rieger, 1996d] A.D. Rieger
New Directions in AI Planning
In: Learning to Guide a Robot via Perceptions, 383-394
M. Ghallab and A. Milani (Hrsg.), IOS Press, 1996

[Sklorz, 1995] S. Sklorz
Repräsentation operationaler Begriffe zum Lernen aus Roboter-Sensordaten
LS-8 Diplomarbeit, Univ. Dortmund 1995

[Smith/Medin, 1981] E.E. Smith, D.L. Medin
Categories and Concepts
Harvard University Press, Cambridge MA 1981

[Wang et al., 1994] J.T-L. Wang, G-W. Chirn, T.G. Marr, B. Shapiro, D. Shasha, K. Zhang
Combinatorial Pattern Discovery for Scientific Data: Some Preliminary Results
In: International Conference on Management of Data, SIGMOD-1994, 115-125

[Wessel, 1995] S. Wessel
Lernen qualitativer Merkmale aus numerischen Robotersensordaten
LS-8 Diplomarbeit, Univ. Dortmund 1995

[Zitzler, 1996] E. Zitzler
Entwurf eines interaktiven Werkzeuges zur Datenverwaltung in einem situierten Lernsystem
LS-8 Diplomarbeit, Univ. Dortmund 1996

Abbildungsverzeichnis

Tabellenverzeichnis