Situierte Systeme mit Lernfähigkeit

Katharina Morik
Fachbereich Informatik, Lehrstuhl für Künstliche Intelligenz

11. April 1996

Zusammenfassung

In diesem Aufsatz wird die Anwendung einer neuen Klasse von Lernverfahren, nämlich die der induktiven logischen Programmierung (ILP), auf Probleme der Navigation mobiler Roboter dargestellt. Aus klassifizierten Daten von Trainingsfahrten eines mobilen Roboters werden in einer Folge von Lernläufen ausführbare Begriffsdefinitionen gelernt. Diese operationalen Begriffe verbinden Objekterkennung Handlungen (Bewegungen) des Roboters.

In this paper, the application of inductive logic programming to navigation of a mobile robot is described. A series of learning passes uses classified data from training missions of the mobile robot in order to learn operational concepts. Operational concepts characterize objects and actions such that the robot is capable of recognizing the object and acting appropriately.

1 Lernende Roboter

In diesem Aufsatz wird die Anwendung einer neuen Klasse von Lernverfahren, nämlich die der induktiven logischen Programmierung (ILP), auf Probleme der Navigation mobiler Roboter dargestellt. Diese Kombination von Lernmethode und Anwendungsfeld wird zunächst überraschen, war es doch bisher üblich, zur Verbesserung der Leistung von Robotern numerische Lernverfahren (neuronale Netze, Verstärkungslernen) einzusetzen. Die so erzielten Erfolge sollen auch nicht geschmälert werden. Es gibt aber einige Gründe, die dafür sprechen, auch noch andere, neue Verfahren zu versuchen:

- Für Benutzer von Robotern sind die auf niedriger konzeptueller Ebene angesiedelten Repräsentationen schwer zu verstehen. Die Kommunikation gerade mit mobilen Service-Robotern erfordert eine dem Menschen angepaßte Benutzer– Schnittstelle.
- Wenn ein Weltmodell in Form einer Karte vorliegt, können graphische Benutzerschnittstellen mit direkter Manipulation gut eingesetzt werden. Die Erstellung solcher Karten ist aber aufwendig und führt

zu langen Einrichtungszeiten, wenn ein Roboter in neuer Umgebung eingesetzt werden soll.

- Numerische Repräsentationen sind übergenau. Abweichungen müssen mit Verfahren wie etwa der Fuzzy Logic behandelt werden. Für die Objekterkennung sollte aber nur die Struktur von Merkmalen eine Rolle spielen, nicht jedoch der exakte Zahlenwert.
- Numerische Lernverfahren können Hintergrundwissen nur auf sehr indirekte Weise einbeziehen. So muß bei neuronalen Netzen etwa durch geschickte Wahl der Topologie, beim Verstärkungslernen durch geschickte Wahl der Verstärkungsfunktion implizit Hintergrundwissen berücksichtigt werden.

Während einerseits (numerische) Verfahren für das Erlernen von Reflexen und andererseits (logikbasierte) Verfahren für das Erlernen von abstrakten Begriffen zur Verfügung stehen, ist die Verbindung der Ebenen noch zu wenig untersucht. Wir wollten den Zusammenhang zwischen der Ebene der Reflexe und der begrifflichen Ebene untersuchen, so daß Begriffe anhand konkreter Erfahrung (Wahrnehmung und Handlung in der natürlichen Umwelt) erworben und dort angewandt werden. Es geht um die Frage, wie Begriffe in der Umwelt verankert sind (Stichwort: *symbol grounding*). Um uns diesem Thema anzunähern, haben wir konkrete Fragen bearbeitet [1]:

- Ist es möglich, über die Ebene der Reflexe hinauszugehen und eine Hierarchie von immer abstrakteren Verhaltensweisen eines mobilen Roboters zu entwickeln, wobei jede Verhaltensweise sowohl die Wahrnehmungs- wie auch die Handlungsseite berücksichtigt?
- Können wissensbasierte Verfahren eingesetzt werden und dennoch die Realzeitanforderungen erfüllt werden?
- Sind Lernverfahren, die eine eingeschränkte Prädikatenlogik verwenden, in der Robotik überhaupt einsetzbar?

Stellen wir uns einen auf Büroumgebungen spezialisierten mobilen Roboter vor. Er soll etwa das folgende Kommando ausführen können:
fahre durch die Tür, dann links bis zum Schrank, stop.

Dazu sollen nicht – wie bisher üblich – die genauen Koordinaten für Wände, Türen, Tische bekannt sein müssen. Der Begriff der *Tür* soll nicht als ein Durchgang bestimmter Breite und Tiefe an einer bestimmten Stelle im Raum repräsentiert sein, sondern als ein *operationaler Begriff*, d.h. ein Konzept, das Handlung und Wahrnehmung integriert. Der operationale Begriff der Tür ist verankert in den Wahrnehmungs- und Handlungsmöglichkeiten des Roboters [Klingspor und Morik, 1995]. So ist jeder Durchgang für den Roboter eine Tür, wenn er sich hindurchbewegen kann. Ein Wahrnehmungsmuster signalisiert die Möglichkeit, daß an dem durch die Sensoren angemessenen Ort eine Tür ist. Der Erfolg der Handlung, sich durch die Tür zu bewegen, verifiziert, daß an der Stelle

[1] Die hier vorgestellten Verfahren wurden im Rahmen des ESPRIT Projektes BLearn II entwickelt.

eine Tür war. Der operationale Begriff ist ausführbar bei Türen beliebiger Abmessungen in verschiedensten Umgebungen.

2 Induktive logische Programmierung

Lernverfahren für relationales Wissen werden heute unter dem Begriff der *induktiven logischen Programmierung* zusammengefaßt. Diese Lernverfahren sind obendrein in der Lage, Hintergrundwissen zu berücksichtigen. Ein einfaches Beispiel für Sachverhalte, die Hintergrundwissen benötigen und sich nicht ohne Relationen ausdrücken lassen, sind Verwandschaftsbeziehungen. Gegeben für den Begriff *Tochter*:

die positiven Beispiele	die negativen Beispiele
$tochter(mary, ann)$	$\neg tochter(tom, ann)$
$tochter(eve, tom)$	$\neg tochter(eve, ann)$

Ein Lernverfahren, das nur Attribut-Werte bearbeiten kann, kann gar nicht aus diesen Beispielen lernen. Ein Lernverfahren, das eine eingeschränkte Prädikatenlogik als Formalismus für Hypothesen bearbeiten kann, kann aus diesen Beispielen allein auch noch keine vernünftige Regelhaftigkeit finden. Es fehlen Angaben. Wird nun als Hintergrundwissen noch angegeben

$elter(ann, mary)$	$weiblich(ann)$	$elter(ann, tom)$	$weiblich(mary)$
$elter(tom, eve)$	$weiblich(eve)$	$elter(tom, ian)$	

so kann die Regel gelernt werden:

$$\boxed{elter(Y, X), weibl(X) \quad \rightarrow tochter(X, Y)}$$

Die induktive logische Programmierung geht über die Möglichkeiten der klassischen Verfahren (Induktion von Entscheidungsbäumen, neuronale Netze, *conceptual clustering*) hinaus.

- Während die klassischen Verfahren nur Attribut-Werte verarbeiten können, sind Verfahren der induktiven logischen Programmierung in der Lage, Formeln einer eingeschränkten Prädikatenlogik als Hypothesen auszugeben, zum Beispiel Prolog-Klauseln.
- Während die klassischen Verfahren nur Beispiele als Eingabe nehmen können, sind die Verfahren der induktiven logischen Programmierung in der Lage, Hintergrundwissen zu berücksichtigen.

Die Schwierigkeit des Lernens hängt von dem Formalismus für die Hypothesensprache ab. Gordon Plotkin [Plotkin, 1971] hat gezeigt, daß ohne Hintergrundwissen in Prädikatenlogik gelernt werden kann. Allerdings ist die Generalisierung von n Formeln der Länge l im schlimmsten Fall l^n lang. Das Lernproblem mit Hypothesen in Form von Klauseln und Hintergrundwissen in Form von variablenfreien Fakten ist in exponentieller Zeit lösbar. Die Komplexität der Lernaufgabe muß also eingeschränkt werden, wenn wir wollen, daß das Lernverfahren auch bei vielen Beispielen in vertretbarer Zeit zum Ergebnis kommt. Die Forschung im Bereich der induktiven logischen Programmierung beschäftigt sich damit, wie die Prädikatenlogik so eingeschränkt werden kann, daß sie immer noch das ausdrücken kann, was wir lernen wollen, aber das Lernverfahren dennoch ein Ergebnis in polynomiell beschränkter Zeit findet (siehe [Kietz und Dzeroski, 1994] für einen Überblick).

3 Lernen operationaler Begriffe

Im Rahmen des Projektes BLearn II (ESPRIT 7274) [Kaiser et al., 1995] wurden nun aus den Sensormessungen und Bewegungsdaten während einiger Trainingsfahrten eines mobilen Roboters entlang und durch eine Tür operationale Begriffe gelernt. Dabei wird schrittweise von den numerischen Werten der Sensor- und Bewegungsdaten abstrahiert.

Im ersten Schritt werden die Abstandsmessungen der 24 Ultraschallsensoren inkrementell in Zeitintervalle aufgeteilt, während derer die Meßpunkte bei gleicher Orientierung eines Sensors dieselbe Tendenz aufweisen (steigend, fallend, gleichbleibend, plötzlich ansteigend, ...) Diese Umsetzung von (numerischen) Signalen in Prolog-Fakten der Form

`steigend(Fahrt, Orientierung, Sensor, Anfang, Ende)` [2]

abstrahiert von geringfügigen Abweichungen. Dennoch sind die Daten auf dieser Abstraktionsebene noch sehr verrauscht.

Im zweiten Schritt werden auf immer abstrakterer Ebene Regeln gelernt. Es wird also eine Folge von Lernläufen gestartet. Fakten, die die Messungen aller Sensoren während verschiedener Trainingsfahrten wiedergeben, werden für das Lernen klassifiziert. Es wird angegeben, in welchem Zeitintervall der Roboter an einem Objekt vorbei- oder durch eine Tür hindurchgefahren ist. Es werden nun anhand der klassifizierten Daten Regeln gelernt, die für jeweils einen Sensor und darauf aufbauend dann für Sensorgruppen charakteristische Wahrnehmungsmuster bei einer objektbezogenen Bewegung ausdrücken. Diese Regeln stellen nur noch die Beziehungen zwischen Zeitintervallen dar, nicht jedoch die konkrete Dauer der Messung eines Musters. Für jedes Sensormuster werden verschiedene Regeln gelernt, so daß unterschiedliche Situationen abgedeckt sind.

`gleich(Fahrt, Orientierung, Sensor,T1, T2, Gradient1) &`

[2] Das Prädikat gibt die Abstandsmessungen in einem Zeitintervall an. Die Argumente geben die Fahrtnummer, die Orientierung des Sensors, den Sensor und das Zeitintervall mit Anfangs- und Endzeitpunkt an.

ploetz_hoch(Fahrt, Orientierung, Sensor, T2, T3, Gradient2) &
gleich(Fahrt, Orientierung, Sensor,T3, T4, Gradient3)
→ s_sprung(Fahrt, Sensor, T1, T4, parallel)

Auf der Ebene der Sensorgruppen wird nur das ausgedrückt, was die Mehrheit der Sensoren einer Klasse gemessen haben [3]. Dadurch ergibt sich ein Filter, der die Daten bereinigt.

```
s_sprung(Fahrt, S1, T1, T2, parallel) &
sklasse(Fahrt, S1, T0, T7, rechts_seite) &
s_sprung(Fahrt, S2, T3, T4, parallel) &
sklasse(Fahrt, S2, T0, T7, rechts_seite) &
s_sprung(Fahrt, S3, T5, T6, parallel) &
sklasse(Fahrt, S3, T0, T7, rechts_seite) &
succ(T1, T3) & succ(T3, T5) & T0 ≤ T1 & T6 ≤ T7
→ sg_sprung(Fahrt, T1, T6,rechts_seite, parallel)
```

Das Prädikat succ aus dem Hintergrundwissen gibt Nachfolger an, etwa T1 + 1 = T3 oder T1 + 2 = T3. Auf der nächsten Abstraktionsebene werden komplexe Wahrnehmungs- und Handlungsmuster gelernt: [4]

```
sg_sprung(Fahrt,T1,T2,rechts_seite,parallel) &
sg_sprung(Fahrt,T1,T2,links_seite,parallel)
→ durch_tuer(Fahrt, T1, T2, parallel)
```

So wird ausgedrückt, daß Sensoren auf der rechten und der linken Seite gleichzeitig den Türrahmen anmessen, wenn der Roboter weitgehend gerade durch die Tür fährt (parallel). Eine entsprechende Regel für zeitverschobene Wahrnehmungen auf den Seiten, also die schräge Durchfahrt, wurde ebenfalls gelernt.

Dann werden Regeln der folgenden Art gelernt:

```
fahr(Fahrt, T1,T2, Tempo, BewegRichtung) &
wahrnehm (Fahrt,T1,T2,Wahrnehm, WahrnehmRichtung, Seite, parallel)
→ parallel_fahr(Fahrt, T1, T2, Tempo, BewegRichtung,
Wahrnehm, WahrnehmRichtung)
```

Die Regel notiert in kompakter Form, daß während eines Zeitintervalls (T1 - T2) eine kontinuierliche Bewegung und Wahrnehmung (z.B. durch_tuer) erfolgt ist, so daß beides zusammengefaßt werden kann. Die Regeln auf dieser Abstraktionsebene sind bereits zu 78% korrekt.

Schließlich werden operationale Begriffe gelernt:

```
steh(Fahrt, T1, T2, vor_tuer, WarnehmRichtung, schmal_seite, VorW)
&
parallel_fahr(Fahrt, T2, T3, Tempo, WarnehmRichtung, durch_tuer,
rechts_links) &
steh(Fahrt, T3, T4, vor_tuer, hinten, schmal_seite, durch_tuer)
→ fahr_durch_tuer(Fahrt, T1, T4)
```

Durch die Unifikation von Wahrnehmungsrichtung vor der Handlung (steh) und der Bewegungsrichtung bei der Handlung (parallel_fahr) ist

[3] Eine Klasse wird durch den Ort, dan dem sich der Sensor am Roboter findet, gebildet.

[4] Wir folgen der Prolog-Konvention und drücken Variable durch Großbuchstaben, konstante Werte durch Kleinbuchstaben aus.

das Geradeaus- Fahren ausgedrückt. Die aktuelle Wahrnehmungsrichtung wird als `vorn` interpretiert. Die Wahrnehmung während der Handlung ist das Muster `durch_tuer`. [5]

Die Wahrnehmung nach der Handlung ist das Muster `vor_tuer`, diesmal jedoch von den hinteren Sensoren gemessen. Als `hinten` werden die den Sensoren der ersten `stand`-Handlung gegenüberliegenden Sensoren betrachtet. Die kompakten Regeln für operationale Begriffe sind 100% korrekt.

Die Folge von Lernläufen produziert eine Menge von Regeln. Zum Testen werden dann neue, ebenfalls vom Benutzer klassifizierte Fahrtdaten (Handlungs- und Wahrnehmungsdaten) eingegeben und in Fakten umgewandelt, aus denen dann per Vorwärtsinferenz abgeleitet wird, welches Objekt in welcher Bewegung zu erkennen ist. In dieser Weise werden die Korrektheitsangaben der vorigen Absätze hergeleitet. Wichtiger als die Korrektheit ist jedoch, daß die gelernten Regeln in ganz unterschiedlichen Umgebungen anwendbar sind. Was anhand Fahrten entlang oder durch *eine* Tür gelernt wurde, ist dafür geeignet, völlig unterschiedliche Türen zu erkennen. Diese Flexibilität lohnt den Lernaufwand und auch eine Nachbesserung per Hand.

Entgegen der gängigen Meinung, für Robotik müsse man numerische Verfahren anwenden, ist die induktive logische Programmierung sehr wohl dafür geeignet, für Robotikanwendungen höhere Konzepte zu lernen. Anders herum formuliert: mithilfe der induktiven logischen Programmierung können Begriffe in der realen Welt verankert werden.

4 Anwenden operationaler Begriffe

Da das Lernen off-line erfolgt, kann es ruhig ohne enge Zeitvorgaben arbeiten. Die Verwendung des gelernten Wissens erfordert allerdings, daß die Realzeitanforderungen eingehalten werden. Deshalb werden die gelernten Regeln in dreierlei Hinsicht optimiert:

- Die Regeln für die Wahrnehmungsmuster eines Sensors werden in Präfixbäume überführt. Aktuelle Sensormessungen setzen oder verschieben Marker in diesen Bäumen. Erreicht ein Marker ein Blatt, so ist das betreffende Wahrnehmungsmuster erkannt [Rieger, 1996].
- Die Daten aller Sensoren werden parallel verarbeitet. Parallel werden die numerischen Werte in Fakten umgewandelt und diese dann mithilfe des Marker- Passings zu Wahrnehmungsmustern zusammengefaßt.
- Die Inferenz höherer Begriffe wird ebenenweise durchgeführt (von Sensormustern zu Sensorgruppenmustern, von diesen zu komplexen Wahrnehmungs- Handlungsmustern und schließlich zu operationalen Begriffen), so daß die jeweilige Inferenztiefe stets nur 1 ist.

[5] Eine Regel überführt den Begriff `durch_tuer` in ein Argument des Prädikats `wahrnehm`:
`durch_tuer(Fahrt, T1, T2,WarnehmRichtung, Seite, Orientierung)`
→ `wahrnehm(Fahrt,T1,T2,durch_tuer, WarnehmRichtung, Seite, Orientierung)`

Eine einfache deduktive Planungskomponente erstellt einen groben Handlungsplan, für den die Basishandlungen des Roboters durch Rückwärtsinferenz ermittelt werden. Der Befehl *fahre durch die Tür* wird überführt in die Sequenz *finde eine Tür* und *fahre durch die Tür*. Die Regeln für das Finden einer Tür sind nicht gelernt. Sie verwenden aber gelernte Begriffe wie `entlang_tuer, entlang_wand`. Im wesentlichen wird eine Wand gesucht, an ihr entlanggefahren bis eine Ecke oder eine Tür erkannt wird. In einer Ecke wird so gedreht, daß der anstoßenden Wand gefolgt werden kann. Wird eine Tür erkannt (`entlang_tuer`), so fährt der Roboter zurück und dreht sich, bis er `vor_tuer` wahrnimmt und steht dann. Damit ist die erste Handlung des hier als Beispiel dargestellten operationale Begriffs der Fahrt durch die Tür wahr. Die nächste Handlung wird gemäß der gelernten Regeln für `parallel_fahr` ausgeführt, bis die nun hinteren Sensoren das Muster `vor_tuer` wahrnehmen. Dann hält der Roboter an. Diese Planungskomponente ist noch simpel. Für unsere Machbarkeitsstudie war sie jedoch ausreichend. Es ist jetzt gelungen, die gelernten Begriffe *Fahrt entlang einer Tür* und *Fahrt durch eine Tür* von dem mobilen Roboter ausführen zu lassen [Klingspor et al., 1996].

Die eingangs gestellten Fragen können also positiv beantwortet werden.

- Eine Verbindung von numerischen Sensor- und Bewegungsdaten bis hin zu operationalen Begriffen konnte hergestellt werden. Dazu wurde eine Signatur für eingeschränkte Prädikatenlogik entwickelt, die aufeinander aufbauende Abstraktionsebenen beschreibt. In diesem Artikel wurde ein Durchgang durch die Ebenen anhand des Begriffs *Fahrt durch die Tür* vorgestellt. Daran ist nachzuvollziehen, wie Bewegungs- und Wahrnehmungsmerkmale integriert werden.
- Die Wissensbasis wird auf der unteren Abstraktionsebene durch Transformation in Präfixbäume und Parallelisierung optimiert. Es wird ausgenutzt, daß die Zeit linear verläuft. Eine weitere, noch nicht implementierte Optimierung betrifft die Planungskomponente, deren Regeln durch den aus der logischen Programmierung bekannten Operator *unfolding* effizienter gemacht werden können.
- Große Teile der Wissensbasis wurden durch das Lernen im Paradigma der induktiven logischen Programmierung erworben. Durch die Einteilung in Ebenen, wobei die Verbindung zwischen den Ebenen gelernt wurde, wird das Lernen auch aus großen Datenmengen möglich.

Nachdem nun die Machbarkeit unseres Ansatzes gezeigt ist, lohnt sich dessen Verbesserung und der systematische Vergleich mit anderen Ansätzen.

5 Ausblick

Eine Motivation für das Lernen von Begriffen im Rahmen der Robotik stammt aus der kognitionswissenschaftlichen Diskussion um den Erwerb

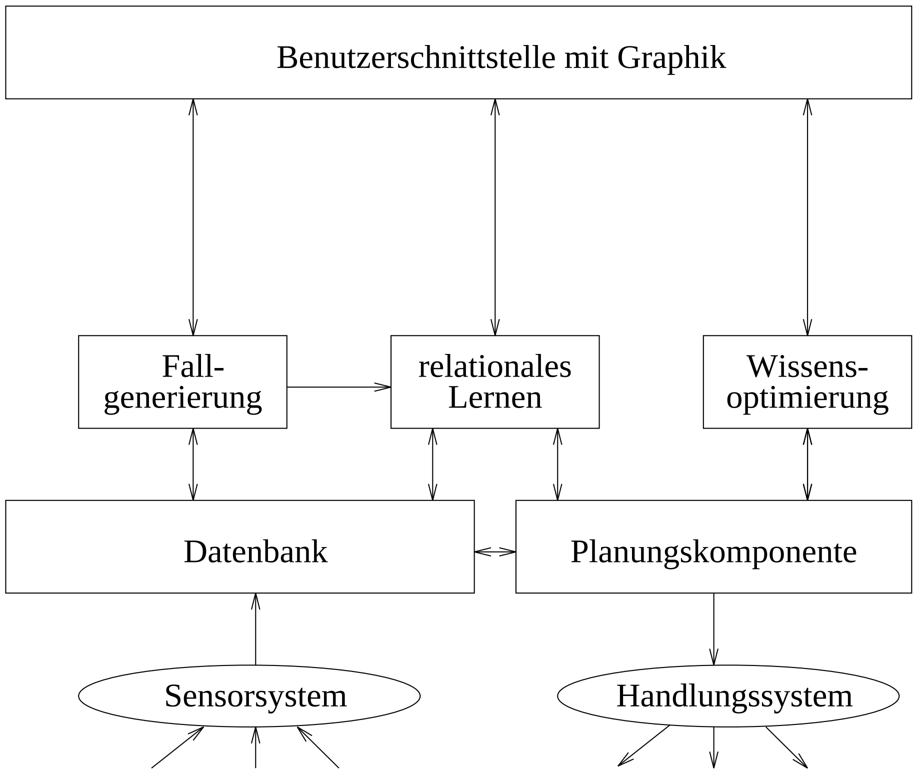

Abbildung 1: Ein situiertes System

und die Verankerung von Begriffen. Die Begriffe sollen nicht nur aus eingegebenen und damit schon vorformulierten Beispielen gebildet werden, sondern auf sinnlichen Erfahrungen beruhen. Eine mögliche Erweiterung wäre die Kopplung mit einer natürlichsprachlichen Schnittstelle, die ja höhere Begriffe benötigt. Insofern kann unser Ansatz als notwendige Vorarbeit zu einer Kopplung von Roboter und natürlichsprachlichem System betrachtet werden, bei der nicht alles begriffliche Wissen von den Systementwicklern eingegeben werden muß.

Das Ziel solcher Arbeiten kann durch ein Schaubild illustriert werden. Ein situiertes System verankert Lern- und Kommunikationsprozesse mit der realen Welt. Dabei müssen Sensordaten und Handlungssystem nicht unbedingt aus der Robotik stammen. Dieselbe Architektur ist gültig für *software agents* oder *softbots*. Hier ergibt sich ein reiches Arbeitsfeld beispielsweise für die Informationssuche im Internet.

6 Danksagung

Der Artikel faßt Arbeiten zusammen, die im Rahmen des ESPRIT Projektes BLearn II (P 7274) von Volker Klingspor, Anke Rieger, Stefan Sklorz, Stephanie Wessel und der Autorin entwickelt wurden. Der mobile Roboter ist PRIAMOS, der an der Univ. Karlsruhe bei Prof. Dr. Dillmann entwickelt wurde.

Literatur

[Kaiser et al., 1995] Kaiser, M., Klingspor, V., del R. Millán, J., Accame, M., Wallner, F., und Dillmann, R. (1995). Using Machine Learning Techniques in Real-World Mobile Robots. *IEEE-Expert (special track on Intelligent Robotic Systems)*, 10(2):37–45.

[Kietz und Dzeroski, 1994] Kietz, J.-U. und Dzeroski, S. (1994). Inductive Logic Programming and Learnability. *SIGART–Bulletin*, 5(1):22–32.

[Klingspor und Morik, 1995] Klingspor, V. und Morik, K. (1995). Towards Concept Formation Grounded on Perception and Action of a Mobile Robot. In Rembold, U., Dillmann, R., Hertzberger, L., und Kanade, T., Hrsg., *IAS–4, Proc. of the 4th Intern. Conference on Intelligent Autonomous Systems*, Seiten 271–278, Amsterdam. IOS Press.

[Klingspor et al., 1996] Klingspor, V., Morik, K., und Rieger, A. (1996). Learning Concepts from Sensor Data of a Mobile Robot. *Machine Learning Journal*.

[Plotkin, 1971] Plotkin, G. D. (1971). A further note on inductive generalization. In Meltzer, B. und Michie, D., Hrsg., *Machine Intelligence*, Kapitel 8, Seiten 101–124. American Elsevier.

[Rieger, 1996] Rieger, A. (1996). Optimizing Chain Datalog Programs and Their Inference Procedures. Technical Report 20, Univ. Dortmund, Computer Science Dept., LS 8.