RDT/DB
Rule Discovery Tool

Wissensentdeckung
in SQL-Datenbanken

Übersicht

1. KDD-Motivation
2. ILP-Crashkurs
3. ILP-Werkzeug RDT
4. RDT/DB

Wissensentdeckung in Datenbanken

Knowledge Discovery in Databases (KDD)

- Untersuchen von unübersichtlichen Datensammlungen nach Regularitäten bzw. finden von allen gültigen und interessanten Regeln
- Ausgabe von verständlichen Regeln (nicht nur einfache Statistiken)
- Hypothesen für gültige Regeln vom System selbst ausgestellt [Morik 98]

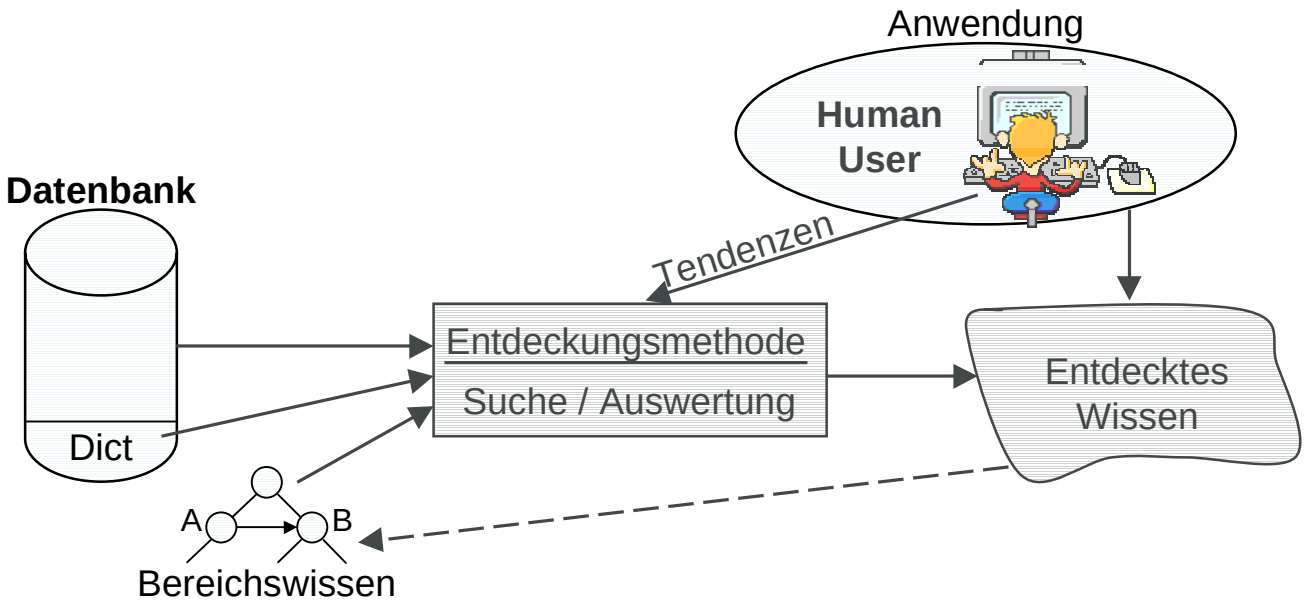


Abb.:Architektur eines prototypischen Entdeckungslernsystems für Datenbanken [Frawley et al 92]

KDD Anwendungsbeispiele

- Medizin: Nebenwirkungen von Medikamenten
- Finanzwirtschaft: Vorhersagen für den Aktienmarkt
- Sozialwissenschaft: Trendanalyse bei Wahlen
- Marketing: Identifikation von Personengruppen mit ungewöhnlichem Kaufverhalten
- Versicherungen: Entdeckung von exzessiven und betrügerischen Ansprüchen
- Physik: Erforschung von Supraleitern
- Militär: (geheim) ☺
- Verbrecherbekämpfung: Abgleich von Fingerabdrücken
- Raumfahrt: Suche nach außerirdischen Intelligenz

Logikorientiertes induktives Lernen (ILP)

Crashkurs, Teil 1

ILP (inductive logik programming) ist gemeinsames Forschungsgebiet des maschinellen Lernens und logischen Programmierens [Muggleton 92].

Begriffslernen bzw. Regellernen (Wissensentdeckung) erfolgt durch Induktion von prädikatenlogischen Formeln aus Beispielen unter Einbeziehung von Hintergrundwissen.

Lernen aus Beispielen kann als Suche im Hypothesenraum, geordnetem nach Generalisierungsrelation (Allgemeinheit), betrachtet werden [Mitchell 82].

Aussagenlogischer (attribut-orientierter) Repräsentationsformalismus:

- endlicher Hypothesenraum
- wenige Generalisierungen jeder Hypothese

} + effiziente Lernprogramme
} - keine Objektrelationen

! Es können keine relationalen Begriffe gelernt werden.

Prädikatenlogischer Repräsentationsformalismus erster Ordnung:

- *un*endlicher Hypothesenraum
- schlechte Generalisierungseigenschaften

} + Objektrelationen möglich
} - ineffizient

! Einschränkungen notwendig, um rel. Begriffe effizient lernen zu können.

Logikorientiertes induktives Lernen (ILP)

Crashkurs, Teil 2

Generalisierung (bzw. Spezialisierung) ist eine partielle Ordnungsrelation auf Literalen bzw. Klauseln.

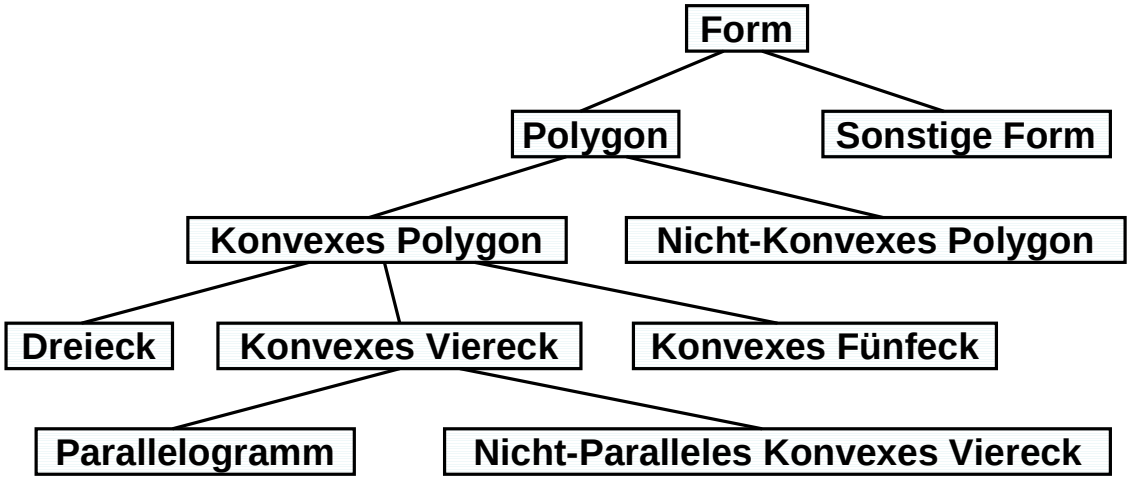


Abb.: Beispiel eines Generalisierungsbaumes [Herrmann 94]

Def.: Eine Klausel $C1$ ist genereller als eine andere Klausel $C2$ bzgl. einer Theorie B, gdw. gilt: $B, C1 \models C2$.

Logikorientiertes induktives Lernen (ILP)

Crashkurs, Teil 3

Logische Folgerung zwischen Klauseln ist im allgemeinen nicht entscheidbar.

! Ein schwächeres Generalisierungsmodell ist erforderlich.

Logische Folgerung gilt immer, wenn θ-**Subsumtion** gilt, umgekehrt aber nicht.

Def.: Eine Klausel $C1$ ist *genereller* als eine andere Klausel $C2$, gdw. gilt:
$C1$ subsumiert $C2$ ($C1 =_\theta C2$).

Def.: *θ-Subsumtion:* $C1 =_\theta C2$ gdw. $C1\theta \subseteq C2$,
θ ist eine geeignete Substitution.

Bsp.: **weiblich (X)** ← **mutter (X,Z)** $=_\theta$ **weiblich (X)** ← **mutter (X,Y), tochter (X,Z)**
mit θ = {Z/Y}.

Def.: *Äquivalenz* unter θ-Subsumtion:
$C1 =_\theta C2$ gdw. $C1 =_\theta C2$ und $C2 =_\theta C1$.

Logikorientiertes induktives Lernen (ILP)

Crashkurs, Teil 4

Def.: Eine Klausel $C1$ ist *echt genereller* als eine andere Klausel $C2$, gdw. gilt:

$$C1 =_\theta C2 \text{ und } \neg(C1 =_\theta C2).$$

Def.: Ein Literal L einer Klausel C ist *redundant* unter θ-Subsumtion gdw. gilt:

$$C =_\theta C \setminus \{L\}.$$

Def.: Eine Klausel C ist *reduziert* gdw. sie keine redundanten Literale enthält.

Bsp.: **{weiblich (X), ¬mutter (X,Z), ¬mutter (X,Y)}**

$=_\theta$

{weiblich (X), ¬mutter (X,Y), ¬tochter (X,Z)}

mit θ = {Z/Y}.

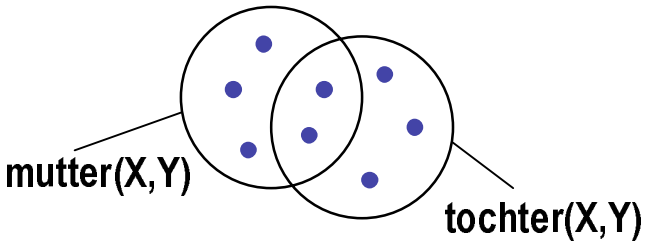


Abb.: Objektmengen

Logikorientiertes induktives Lernen (ILP)

Crashkurs, Teil 5

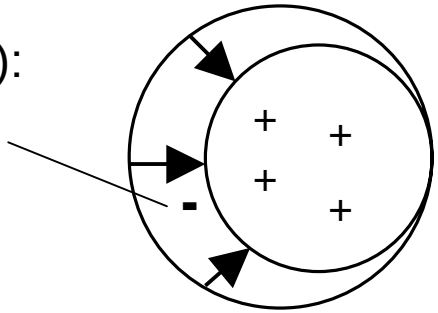
Die **generalisierte θ-Subsumtion** generalisiert zwei funktionsfreie Hornklausel bzgl. gegebenem Hintergrundwissen (Theorie) B:

$C1 =_B C2$ gdw. $\exists\sigma$, so dass $C1_{Kopf}\,\sigma = C2_{Kopf}$

und B, $\neg C2_{Körper}\,\theta \;?\; \exists\,(\neg C1_{Körper}\,\sigma\theta)$,

θ eine Skolemsubstitution, σ eine (Term-)Substitution.

Schrittweises Spezialisieren (Definition durch ein Bild):



Top-down Lernverfahren:

Beginne mit einer generellsten erzeugbaren Hypothese;
solange noch negative Beispiele abgedeckt werden,
wende auf die Hypothese das schrittweise Spezialisieren so an,
dass die positiven Beispiele weiterhin abgedeckt werden;
wenn die Hypothese kein negatives Beispiel abdeckt, gib die Hypothese aus und halte an.

Logikorientiertes induktives Lernen (ILP)

Crashkurs, Teil 6

ILP Semantik (*Begriffslernen*):

Geg.: pos. und neg. Beispiele $E = E^+ \cup E^- \subseteq$ L E,
Hintergrundwissen $B \subseteq$ L B , wobei $B \cup E$? ? und B ? E.

Ziel: Finden einer Hypothese $H \in$ L H , für die gilt:
$B \cup H \cup E$? ? (***Konsistenz***)
$B \cup H$? E^+ (***Vollständigkeit***)
$\forall e \in E^-$: $B \cup H$? e (***Korrektheit***)

ILP Semantik (*Wissensentdeckung*):

Geg.: Hintergrundwissen $B \subseteq$ L B , Beobachtungen $E \subseteq$ L E.

Ziel: Finden einer Menge von Hypothesen $H \in$ L H , für die gilt:
$M^+(B \cup E) \subseteq M(H)$ (***Gültigkeit***)
$\forall h \in H$ $\exists e \in E$: B, E - {e} ? e und B, E - {e}, {h} ? e (***Notwendigkeit***)
$\forall h \in$ L H , die gültig und notwendig sind, gilt: H ? h (***Vollständigkeit***)
H ist minimal (***Minimalität***)

Begriffslernen findet nur den gesuchen Begriff, Wissensendeckung findet dagegen alle wahren und nicht redundanten Regeln.

RDT/DB

RDT/DB ist das erste ILP- Wissensentdeckungswerkzeug, dass direkt mit einem Datenbank-Managementsystem interagiert [Brockhausen und Morik 96]. RDT/DB ist eine Weiterentwicklung des RDT-Werkzeugs aus dem Modellierungssystem MOBAL.

RDT aus der Sicht des maschinellen Lernens:

- funktionsfreie Hornklauseln als Repräsentationformalismus
- Hintergrundwissen in Form von Fakten (ground unit clause)
- Top-down-Breitensuche-Lernverfahren
- Hypothesenraum durch *Regelschemata* syntaktisch eingeschränkt
- weitere Einschränkung des Hypothesenraumes durch *Prädikatentopologie*
- Sortentaxonomie (sortenbehaftete Prädikatsattribute)

Bsp.:	Fakten	Beispiele
	alter (mary,27)	**ehemann (john,vivian)**
	¬verheiratet (peter,janice)	**verheiratet (X,Y) ← ehemann (X,Y)**

RDT/DB, Teil 2

Geg.: Hintergrundwissen und eine Menge positiver und negativer Beispiele für einen zu lernenden Begriff C in funktionsfreier Klausellogik.

Ziel: Finde eine Hypothese H in funktionsfreier Klausellogik, die einem vom Benutzer definierten *Akzeptanzkriterium* genügt.

Mögliche Faktoren des Akzeptanzkriteriums:

- pos(H)
- neg(H)
- pred(H)
- total(H) := pos(H) ∪ neg(H) ∪ pred(H)
- concl(H)
- uncover(H) := concl(H)\pos(H)

Bsp.: pos(H) >5, neg(H) < 2,
pos(H)/total(H) > 0.7,
pred(H)/pos(H) > 0.3,
uncover(H)/concl(H) < 0.5

Pruningkriterium erlaubt eine weitere Einschränkung des Hypothesenraums. Es werden keine Spezialisierungen eines Regelschematas mehr getestet, falls das Regelschema:

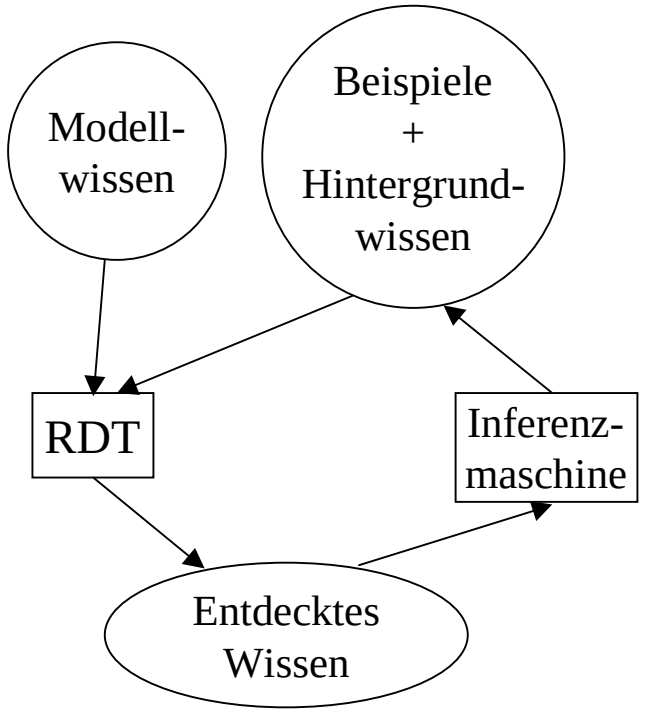
- akzeptiert wurde oder
- nicht dem Akzeptanzkriterium genügt.

RDT/DB, Teil 3

Bereits gelernte oder vorgegebene Regeln werden bei der Regelgenerierung berücksichtigt. Sie werden von der Inferenzmaschine angewandt und saturieren damit die Wissensbasis.

Gelernte Regeln können also zukünftige Lernschritte unterstützen, indem sie die Beispielbeschreibungen um weitere grundinstanziierte Fakten erweitern (*closed-loop learning*).

[Herrmann 94]



RDT/DB, Teil 4

Da bei RDT die Hypothesensprache eingeschränkte Prädikatenlogik ist (funktionsfreie Hornklauseln ergänzt um negative Literale), wird somit ein sehr großer Hypothesenraum beschrieben.

Dieser wird durch Vorgabe der syntaktischen Form der möglichen Regeln,

durch so genannte *Regelschemata* (*Regelmodelle*) eingeschränkt:

- anstatt der Sachbereichsprädikate *Prädikatsvariablen*
- nach Allgemeinheit partiell geordnet

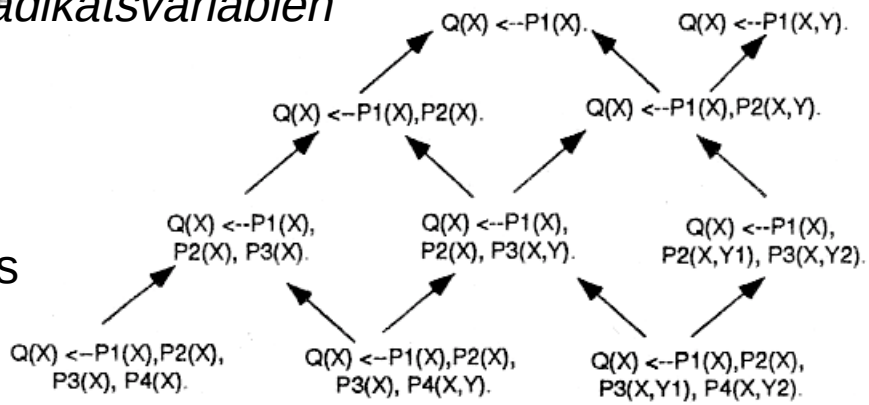


Abb.: Generalisierungsrelation zwischen Regelmodellen [Kietz und Wrobel 92].

Bsp.: Durch Vorgabe des Regelschemas

großmutter(X,Y) ← P1(U,Y), mutter (X,U)

sind aus

großmutter(X,Y) ← elternteil(U,Y), mutter(X,U)
großmutter(X,Y) ← vater(U,Y), mutter(X,U)
großmutter(X,Y) ← vater(Z,Y), mutter(X,Z), vater(Z,V)

nur die ersten zwei Klauseln Instanzen dieses Regelschemas.

RDT/DB, Teil 5

Eine Substitution Σ substituiert Prädikatsvariablen durch Bereichsprädikate der gleichen Stelligkeit, ohne verschiedene Prädikatssymbole zu unifizieren.

Def.: Ein Regelschema RS ist genereller als ein anderes RS , wenn es ein Σ gibt, so dass $RS\ \Sigma\sigma \subseteq RS$.

Regelschemata können auch *teilweise* instanziiert werden.

Ein vollständig instanziiertes Regelschema ist eine *Regel*.

Basiert auf der Menge der im RDT eingegebenen Regelschemata R und der Menge der Bereichsprädikate P im Hintergrundwissen, wird der Hypothesenraum für das Regellernen als die Menge:

$$H = \{R\Sigma \mid R \in R \wedge \mathit{range}(\Sigma) \subseteq P \wedge R\Sigma \text{ ist Regel}\}.$$

definiert, d.h. als die Menge aller möglichen Instanzen für alle Regelschemata.

RDT/DB, Teil 6

Die *Prädikatentopologie* beschreibt semantische Beziehungen zwischen den Prädikaten der Sachbereichstheorie (Hintergrundwissen).

- Gruppierung $T = \{T_1,...,T_m\}$, mit T_i Topologieknoten (Mengen von Prädikaten, evtl. zusammenhängend),
- die Topologieknoten können eine Hierarchie zusammenbilden, die weitere Einschränkung des Hypothesenraumes H^T darstellt:

$$H^T = \{H \in H \mid H = p_{concl} \leftarrow P_{prems}: \exists T_i \in T : p_{concl} \in T_i \wedge P_{prems} \subseteq T_i \cup children(T_i)\},$$

$children(T_i)$ bezeichnet die Vereinigung der direkten Nachkommen von T_i [Kietz und Wrobel 92].

Den Argumenten eines Prädikats kann eine bestimmte *Sorte* zugewiesen werden:

p/n: <sorte1>,...,<sorten>

Bsp.:

mutter/2: <frau>,<person>

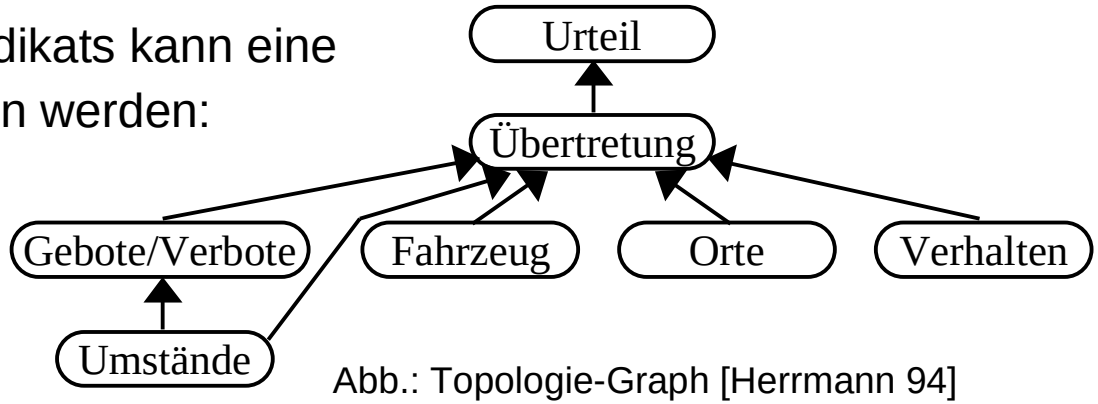


Abb.: Topologie-Graph [Herrmann 94]

RDT/DB, Teil 7

Erweiterung des RDT zum RDT/DB:

- Ersetzung der Sortenkompatibilität durch *Datentypkompatibilität.*
- Abbildung der tabellarischen Darstellung der zu untersuchenden Datenbank in Prädikatenform unter Verwendung der Metainformation der DB).
- *Redundanztest* (Erkennung der redundanten Prädikate) durch Ausnutzung der Primärschlüsseleigenschaft.
- Für Hypothesentest werden SQL-Anfragen generiert und an die Datenbank geschickt, z.B. für pos(H) in der Form:

  select count(*primkey*(q)) from *tabelle*(q), *tabelle*(P)
  where ν (P(c$_1$,...,c$_m$));

- Negative Beispiele können im MOBAL eingegeben werden.

(detaillierte Beschreibung ist in [Lindner 94] zu finden).

RDT/DB, Teil 8

Vorgehensweise von RDT/DB:

- **Beginne mit den generellsten Regelschematas.**
- **Gehe TOP-DOWN durch die Ordnung der Regelschematas**
  - **Instanziiere ein weiteres Prädikat P in P, das folgendem genügt:**
    1. **Stelligkeitskompatibel**
    2. **Prädikatentopologiekompatibel**
    3. **Datentypkompatibel**
    4. **Test auf redundant instanziierte Prädikate**
  - **Redundanztest θ-Subsumtion mit bisher akzeptierten oder zu speziellen Hypothesen.**
  - **Berechnung der Faktoren für das Akzeptanzkriterium.**
  - **Auswertung des Akzeptanzkriteriums.**

[Lindner 94].

Literatur:

[Brockhausen und Morik 96] Peter Brockhausen und Katharina Morik. Direct Access of an ILP Algorithm to a Database Management System. LS VIII, FB Informatik, Univ. Dortmund, 1996.

[Frawley et al 92] W.Frawley, G.Piatetsky-Shapiro, C.J.Matheus. Knowledge Discovery in Databases: An Overview. *AI Magazine,* Vol. 13, No 3, Fall 1992.

[Herrmann 94] Jürgen Herrmann. Maschinelles Lernen. *Skript zur Spezialvorlesung*. LS VIII, FB Informatik, Univ. Dortmund, WS 93/94.

[Kietz und Wrobel 92] Jörg-Uwe Kietz and Stefan Wrobel. Controlling the complexity of learning in logic through syntactic and task-oriented models. In Stephen Muggleton, editor, *Inductive Logic Programming*, chapter 16, pages 335-360. Academic Press, London, 1992.

[Lindner 94] Guido Lindner. Logikbasiertes Lernen in relationalen Datenbanken. *Report 12*, LS VIII, FB Informatik, Univ. Dortmund, 1994.

[Lübbe 95] Marcus Lübbe. Datengesteuertes Lernen von syntaktischen Einschränkungen des Hypothesenraumes für modellbasiertes Lernen. *Report 15*, LS VIII, FB Informatik, Univ. Dortmund, 1995.

[Mitchell 82] T.M.Mitchell. Generalization as search. *Artificial Intelligence*, 18(2):203-226, 1982.

[Morik 98] Katharina Morik. Maschinelles Lernen. *Skript zur Spezialvorlesung*. LS VIII, FB Informatik, Univ. Dortmund, WS 97/98.

[Muggleton 92] Stephen Muggleton. Inductive Logic Programming. In Stephen Muggleton, editor, *Inductive Logic Programming*, Kap. 1, S. 3-28. Academic Press, London, 1992.