

# Deduktives Lernen

Prof. Dr.-Ing. Rüdiger Dillmann  
Prof. Dr.-Ing. J. Marius Zöllner

Eine Definition  
Lernen allgemeiner Problemlösungen  
durch Beobachtung und Analyse  
Lösungen eines speziellen Problems



*“Hey! Look what Zog do!”*

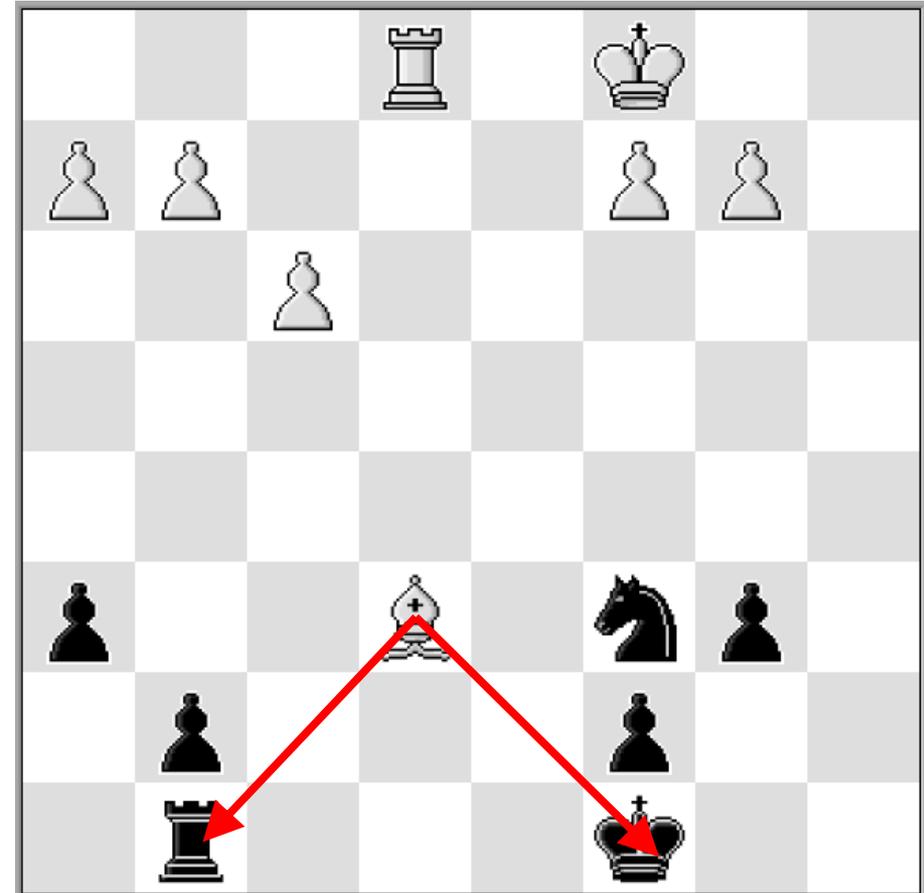
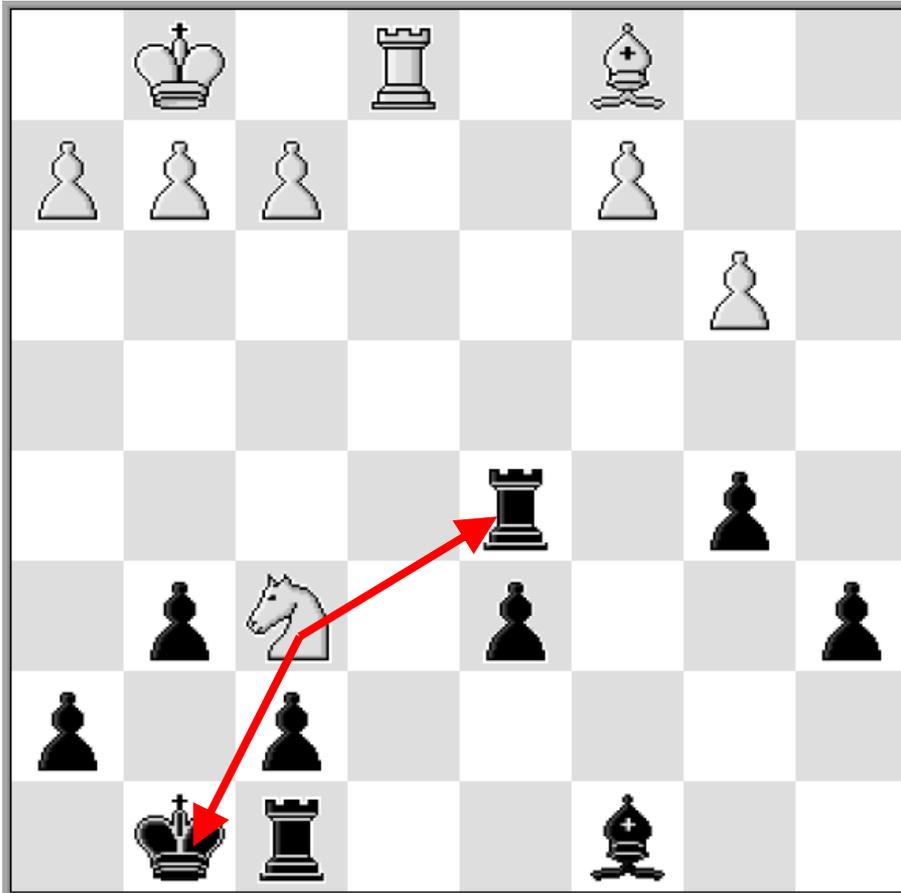
(drawn by Gary Larson)

- Deduktive Lernverfahren: Motivation und Einführung (3-7)
- Erklärungsbasiertes Lernen (EBL)
  - Prinzip (8–10)
  - Mitchell's EBG als Formalisierung (11–16)
  - Diskussion EBL (17–24)
  - Anwendung: STRIPS (25–31)
- Bewertung Deduktives Lernen (32–33)
- Hybride Lernverfahren (34-41)

- Deduktion = „Ableitung“
- Eigenschaften der Lernverfahren
  - Nutzung vorhandenen Wissens
  - Neuformulierung vorhandenen Wissens
  - Explizite Darstellung implizit vorhandenen Wissens
- Beispiele
  - Modus Ponens: 
$$\frac{A, A \rightarrow B}{B}$$
  - Reduktion abstrakter auf berechenbare Größen

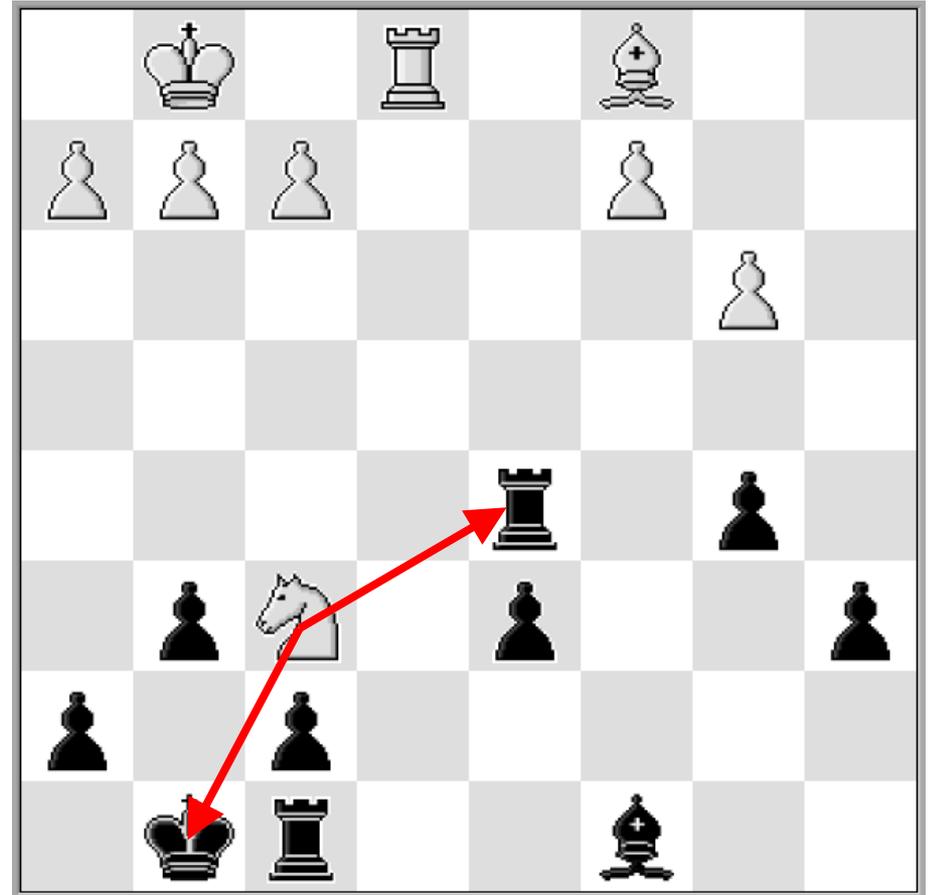
- **Deduktive / Analytische Lernverfahren**  
benutzen hauptsächlich vorhandenes Hintergrundwissen  
und benötigen wenige oder gar keine Lernbeispiele
- **Gegensatz: Induktive / Empirische Lernverfahren**

# Motivation: Schach-Gabel



# Motivation: Schach–Gabel II

- Lernziel:
  - Erkennen von Situationen, bei denen der König sowie eine wertvolle Figur gleichzeitig bedroht werden.
- Für Schachspiel reichen normalerweise:
  - Zugregeln für Figuren
  - Spiel ist verloren, wenn König geschlagen wird
- Suchraum ist zu groß
- Lernen relevanter Situationen auf Basis von Hintergrundwissen



# Motivation: Schach–Gabel III

- Induktiv: extrem viele Beispiele notwendig.
- Mensch ist wesentlich effektiver! Ein paar Beispiele reichen.
- Extrahiertes Wissen:
  - Wenn König und Turm gleichzeitig bedroht sind
  - König muss bewegt werden
  - Turm kann geschlagen werden
- Bereichstheorie:
  - Mögliche Züge im Schach
  - Spiel wird verloren, wenn König geschlagen wird
- Bereichstheorie erlaubt, die wesentlichen Merkmale zu extrahieren

- „The key insight behind explanation-based generalization is that it is possible to form a justified generalization of a single positive training example provided the learning system is endowed with some **explanatory capabilities**. In particular, the system must be able to explain to itself **why the training example is an example of the concept** under study. Thus, the generalizer is presumed to possess a definition of the concept under study as well as domain knowledge for constructing the required explanation.“

Mitchell et al.: [2], S. 49

## ■ Gegeben:

- *Zielkonzept*: Eine Beschreibung des zu lernenden Konzepts (die nicht das *Operationalitätskriterium* erfüllt).
- *Trainingsbeispiel*: Ein Beispiel für das Zielkonzept
- *Bereichstheorie* (engl. Domain Theory): Regeln und Fakten, die erklären, warum das Trainingsbeispiel ein Beispiel für das Zielkonzept ist.
- *Operationalitäts- (Anwendbarkeits-) Kriterium*: Ein Prädikat über Konzeptbeschreibungen, das die Form spezifiziert, in der erlernte Beschreibungen vorliegen müssen.

## ■ Gesucht:

- Eine Generalisierung des Trainingsbeispiels, die eine hinreichende Definition des Zielkonzeptes darstellt und das Operationalitätskriterium erfüllt.

# Erklärungsbasierte Generalisierung I (EBG)

- Ziel: Trainingsbeispiel  $\leftrightarrow$  Generalisierung
- Zweischnitt-Verfahren für jedes positive Beispiel:
  - Explain: Finden einer Erklärung
  - Generalize: Generalisierung der Erklärung

Formalisierung des Erklärungs-basierten Lernens durch Mitchell et.al. [2]  
Andere Formalisierungen u.a. durch DeJong/Mooney [3]

- Zielkonzept: robust (x)
- Trainingsbeispiele:
  - robot(Num5), r2d2 (Num5), age(Num5, 5), manufacturer(Num5, GR) ...
- Bereichstheorie:
  - fixes(u, u)  $\rightarrow$  robust(u)
  - sees (x, y)  $\wedge$  habile(x)  $\rightarrow$  fixes(x, y)
  - robot(w)  $\rightarrow$  sees(w, w)
  - r2d2 (x)  $\rightarrow$  habile (x), c3po (x)  $\rightarrow$  habile(x)
- Operationalitätskriterium:
  - Das Zielkonzept ist beschrieben in Termen der Beispielbeschreibung oder einfach auswertbarer Prädikate aus der Bereichstheorie (z.B. less (x, y))

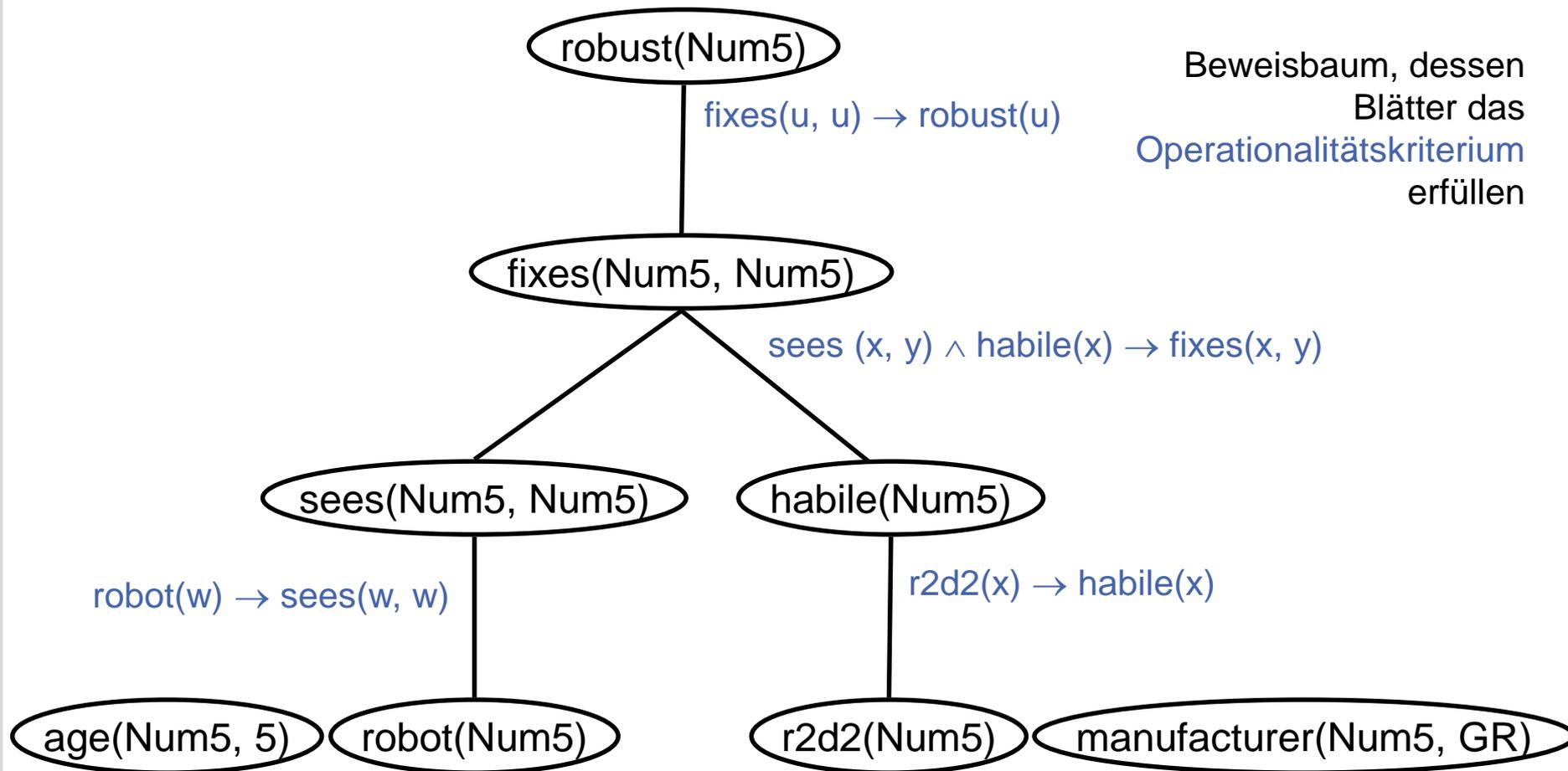
## ■ Explain:

- Konstruiere in Termen der Bereichstheorie eine Erklärung, die zeigt, **wie das Trainingsbeispiel die Definition des Zielkonzeptes erfüllt.**
- Konstruiere diese Erklärung so, dass jeder Ast der Erklärungsstruktur in einem Ausdruck endet, der das **Operationalitätskriterium** erfüllt.

## ■ Generalize:

- Bestimme **hinreichende Bedingungen**, unter denen die oben gefundene Erklärungsstruktur gültig ist und formuliere diese Kriterien in Termen, die das **Operationalitätskriterium** erfüllen.

# EBG-Beispiel: 1. Schritt: Explain



## ■ Explain:

- Konstruiere in Termen der Bereichstheorie eine Erklärung, die zeigt, **wie das Trainingsbeispiel die Definition des Zielkonzeptes erfüllt.**
- Konstruiere diese Erklärung so, dass jeder Ast der Erklärungsstruktur in einem Ausdruck endet, der das **Operationalitätskriterium** erfüllt.

## ■ Generalize:

- Bestimme **hinreichende Bedingungen**, unter denen die oben gefundene Erklärungsstruktur gültig ist und formuliere diese Kriterien in Termen, die das **Operationalitätskriterium** erfüllen.

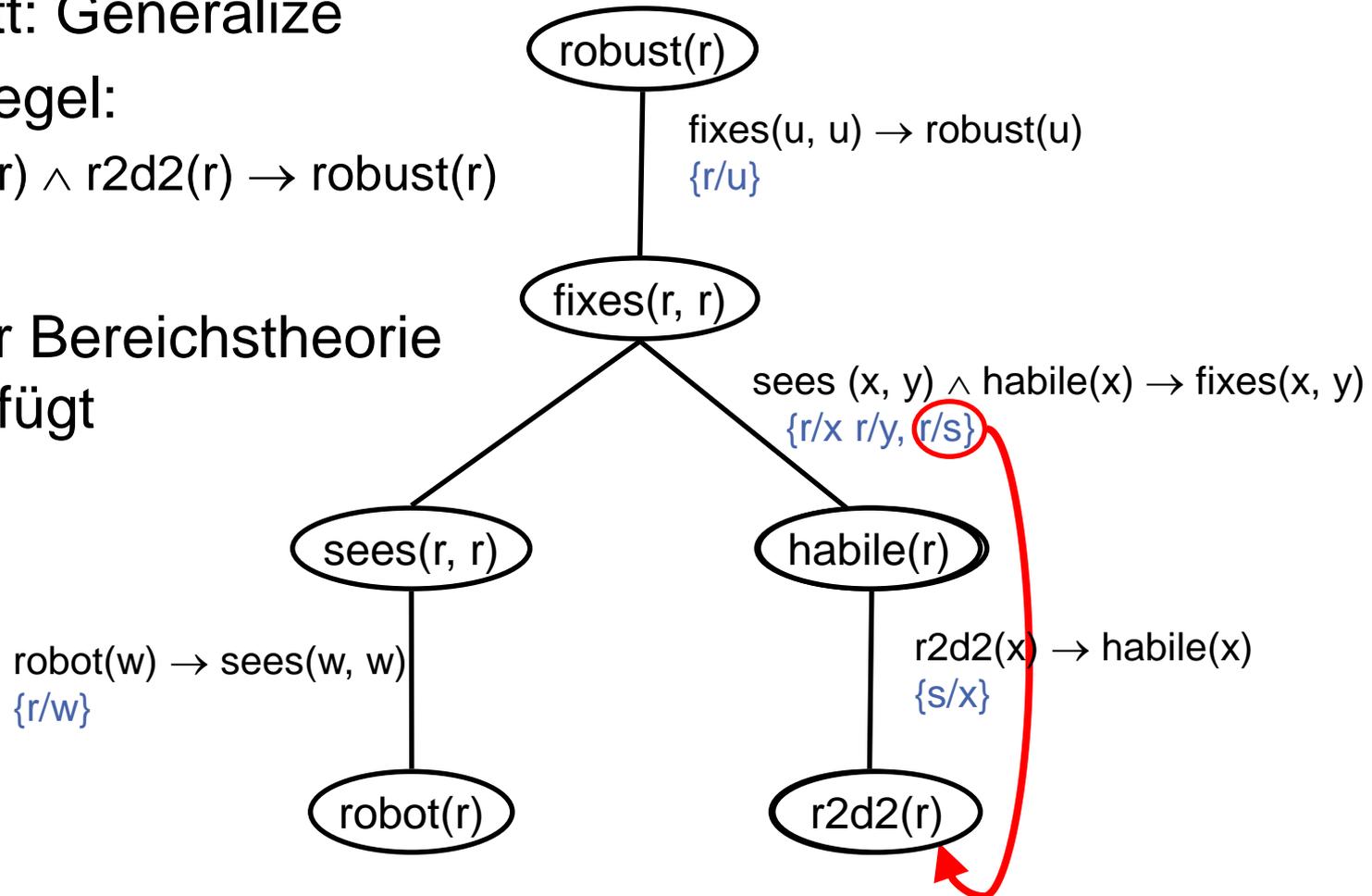
# EBG-Beispiel: 2. Schritt: Generalize

## 2. Schritt: Generalize

### Neue Regel:

- $\text{robot}(r) \wedge \text{r2d2}(r) \rightarrow \text{robust}(r)$

### Wird der Bereichstheorie hinzugefügt

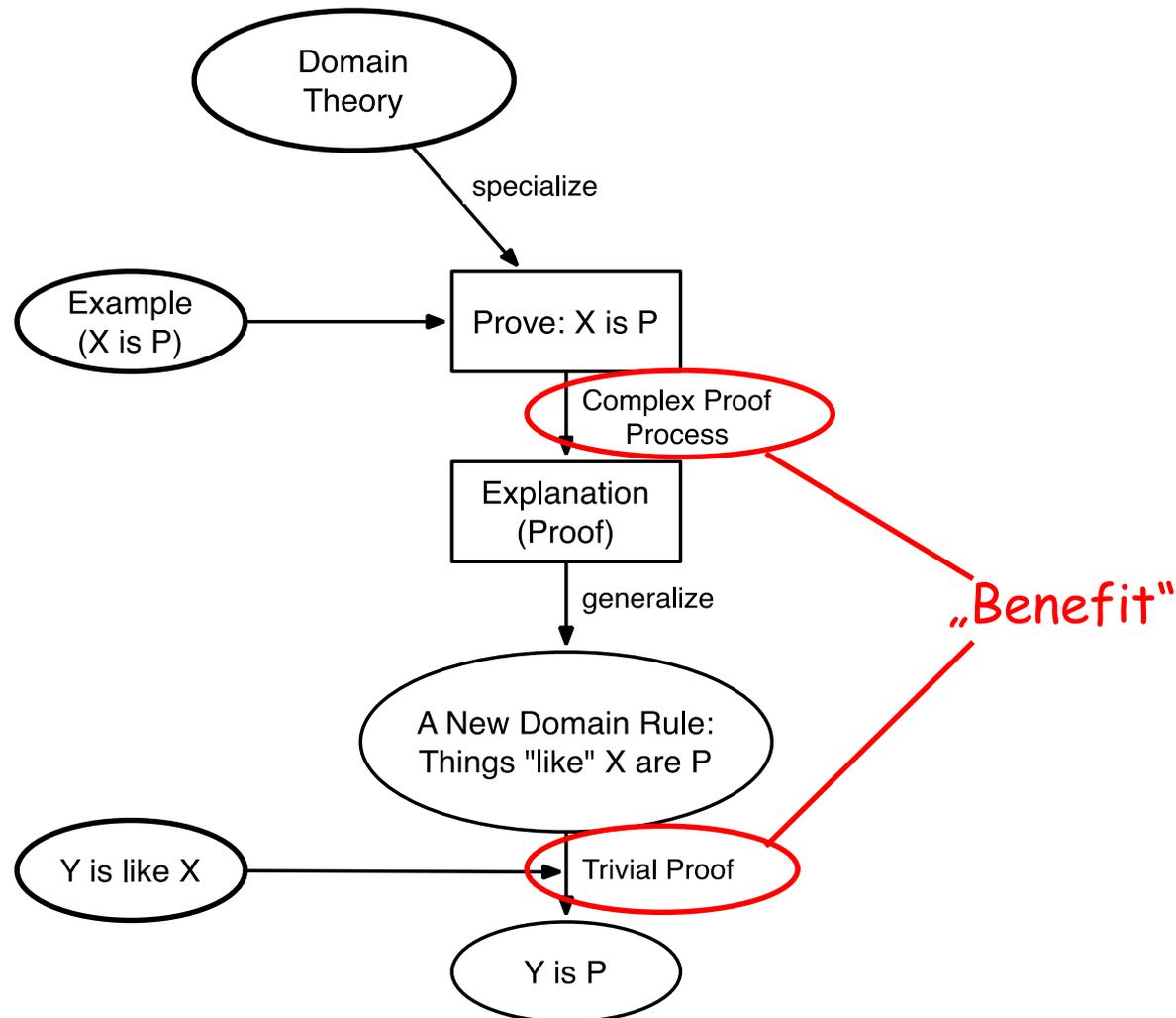


# Was ist Erklärungsbasiertes Lernen? III

Karlsruhe Institute of Technology

- Prozess, der implizites Wissen in explizites umwandelt
- Für jedes positive Trainingsbeispiel wird eine Generalisierung erstellt.
- Warum EBL?
  - Analogien zur menschlichen Wissensverarbeitung
  - Biasproblematik bei induktivem Lernen
- Wesentlicher Aspekt: **Speedup-Learning**

# Erklärungsbasierte Generalisierung III



- **Voraussetzungen:**
  - Vollständigkeit
  - Korrektheit / Konsistenz
  - Anwendbarkeit
- **Problemfälle**
  - Erklärung nicht möglich oder nicht berechenbar
  - Inkonsistente (Beispiel und Gegenbeispiel) oder multiple Erklärungen
- **Wie soll mit Hintergrundwissen / Bereichstheorien umgegangen werden, die die Anforderungen nicht erfüllen?**
  - Approximierungen (Annahme „etwas sei wahr / falsch“)
  - Explorierung

- EBL-System kann Beispiel erklären. Kann es damit auch Beispiele selbst erzeugen ?
  - In manchen Bereichen gilt nicht
    - Erklärungsfähigkeit => Generierungsfähigkeit
      - (z.B. NP-vollständige Probleme)
- Beispiele dienen als Fingerzeige
  - ohne Beispiele ist Suche im Bereichswissen und Aufbau der Erklärungsstruktur komplexer.
- Vorgegebene Beispiele können als "typisch" ausgewählt werden. Diese Auswahl erfordert Wissen über das Zielkonzept, das i.a. nicht im System enthalten ist.

# EBL: Wird tatsächlich gelernt?

Wdh. aus VL 1: Definition Maschinelles Lernen

## Definition:

Ein System lernt aus Erfahrung  $E$  in Hinblick auf eine Klasse von Aufgaben  $T$  und einem Performanzmaß  $P$ , wenn seine Leistungen bei Aufgaben aus  $T$  gemessen mit  $P$  durch Erfahrung aus  $E$  steigt.

## Beispiel: Lernen Schach zu spielen

$T$  = Schachspielen

$P$  = Prozent der gewonnenen Spiele

$E$  = Spiele gegen sich selbst

# EBL: Wird tatsächlich gelernt?

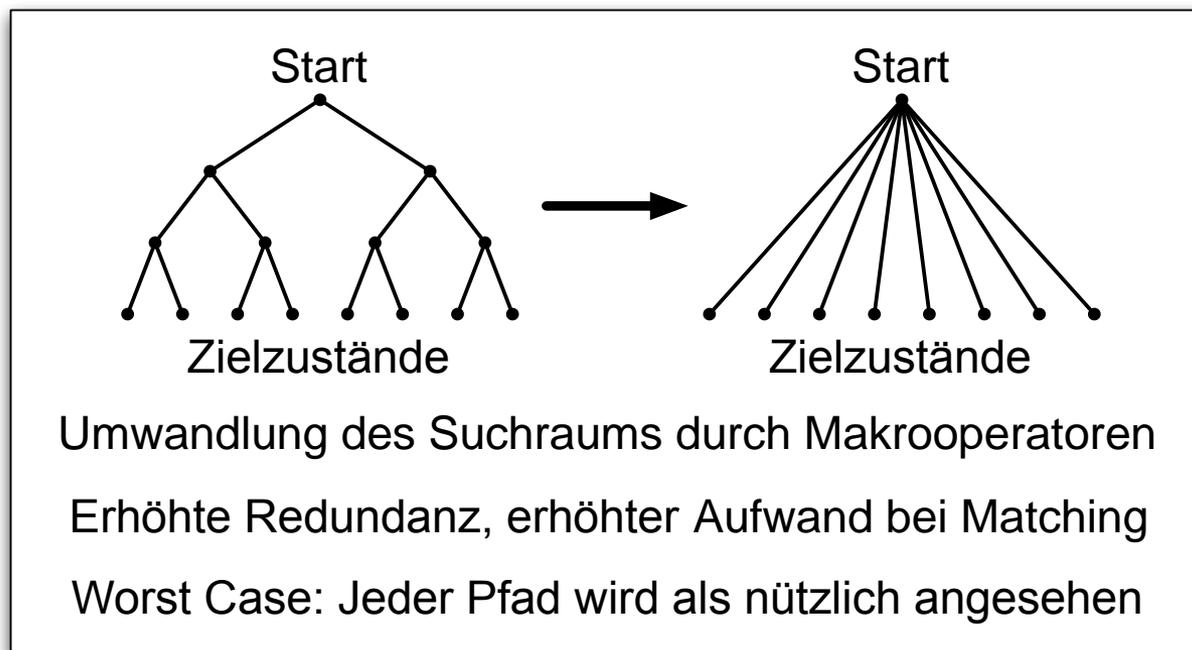
- Wird Wissen erlernt, das nicht bereits im Hintergrundwissen enthalten ist („knowledge-level learning“)?
- EBG/EBL verändert die "deduktive Hülle" der Wissensbasis NICHT.  
⇒ Unbegrenzte Ressourcen vorausgesetzt wird nichts neues erlernt.
- Aber: In der Praxis sind Zeit/Rechenleistung begrenzt.
- Erst durch Makros/Generalisierung können manche Ziele erreicht/Aufgaben durchgeführt werden.
  - Unter diesen Voraussetzungen wird tatsächlich "gelernt".
  - sog. **Speedup-Learning**

# EBL: Ressourcenaufwand für das Lernen neuer Regeln

- Hardware (Parallelrechner, paralleles "Matchen")
  - Keine echte "Lösung", Lösung wird "vertagt"
- **Indizierung** von Regeln/Makrooperatoren
  - Keine Reduzierung der Regelmenge
- Vermutlich bester Ansatz:
  - Beschränkung der Aufnahme neuer Regeln bzw. Makrooperatoren
  - **Messen der "Nützlichkeit"** neuer Regeln/ Operatoren (bzgl. neu zu lösender Probleme)
  - Ggfs. Lösung von Beispielproblemen zur Unterstützung dieser "Messungen"
  - Erhalten oder Verwerfen der Regel, je nach Ergebnis

# EBL: Nützlichkeit von Regeln

- Nützlichkeit einer einzelnen gelernten Regel bzw. eines Makrooperators
- Nützlichkeit der Kombination aller Regeln (MOs) (gelernt & vorgegeben)



- Gelerntes Wissen ist korrekt
  - Korrektheit des ursprünglichen Wissens vorausgesetzt
- Kein (impliziter) induktiver Bias
- Stattdessen explizite Formulierung der Domänentheorie
  - Ausnutzung vorhandenen domänenspezifischen Wissens
  - Bessere Überprüfbarkeit
  - Domänenunabhängiger, da Bereichswissen separat repräsentiert

- Erzeugung von Makrooperatoren (automatisches Planen)
  - STRIPS (STanford Research Institute Problem Solver)
  - SOAR (Symbolic Cognitive Architecture, CMU)
  
- Lernen, Suche zu kontrollieren
  - Effektivitätssteigerung beim Planen, besonders bei großen Zustandsräumen
  - PRODIGY (Lernen von Planungsstrategien, CMU)
    - Bewertung von Regeln
    - Einbeziehung von negativen Beispielen

# Erzeugung von Makrooperatoren

## ■ Gegeben:

- Wissen über (Elementar-) Operatoren in einer Problemlösungsdomäne
- Erfahrung mit gültigen Problemlösungen in der Domäne

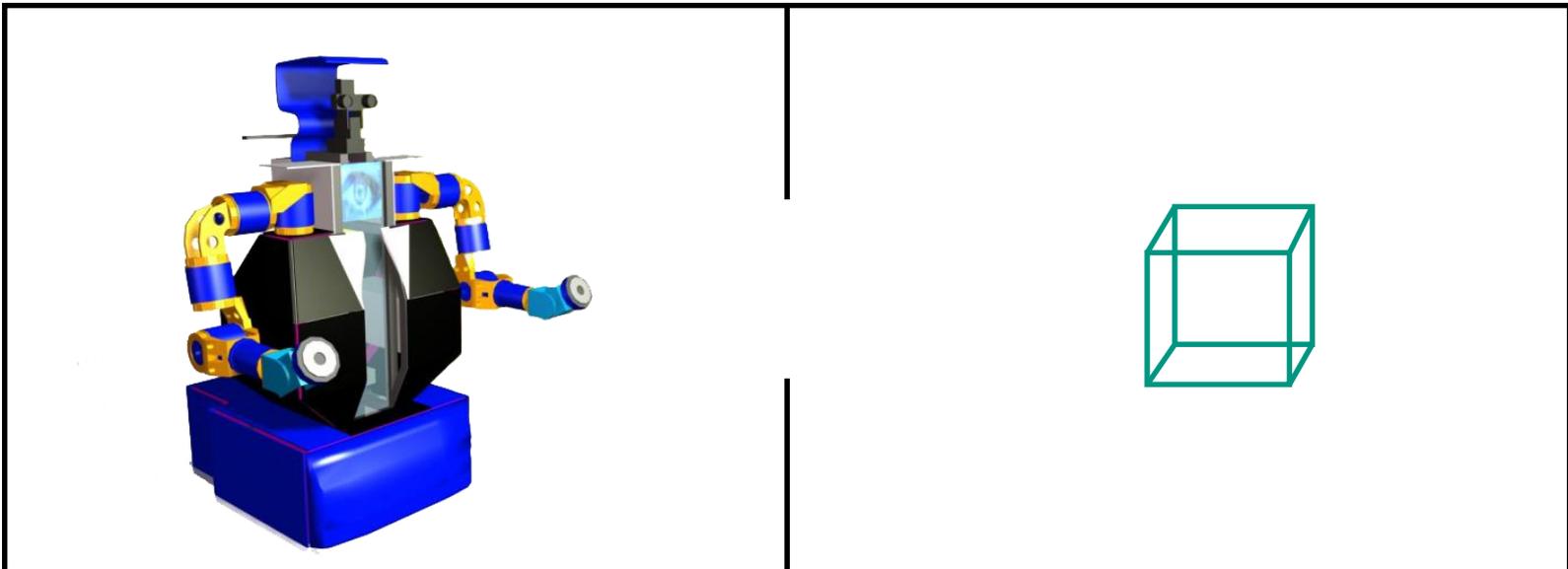
## ■ Gesucht:

- Zusammengefasste Operatorsequenzen, welche die Kosten für das Finden von Problemlösungswissen reduzieren

# STRIPS: Makrooperatoren I

- **Planungsaufgabe:**  
Finde eine Operatorsequenz, die einen Startzustand S in einen Zielzustand T überführt.

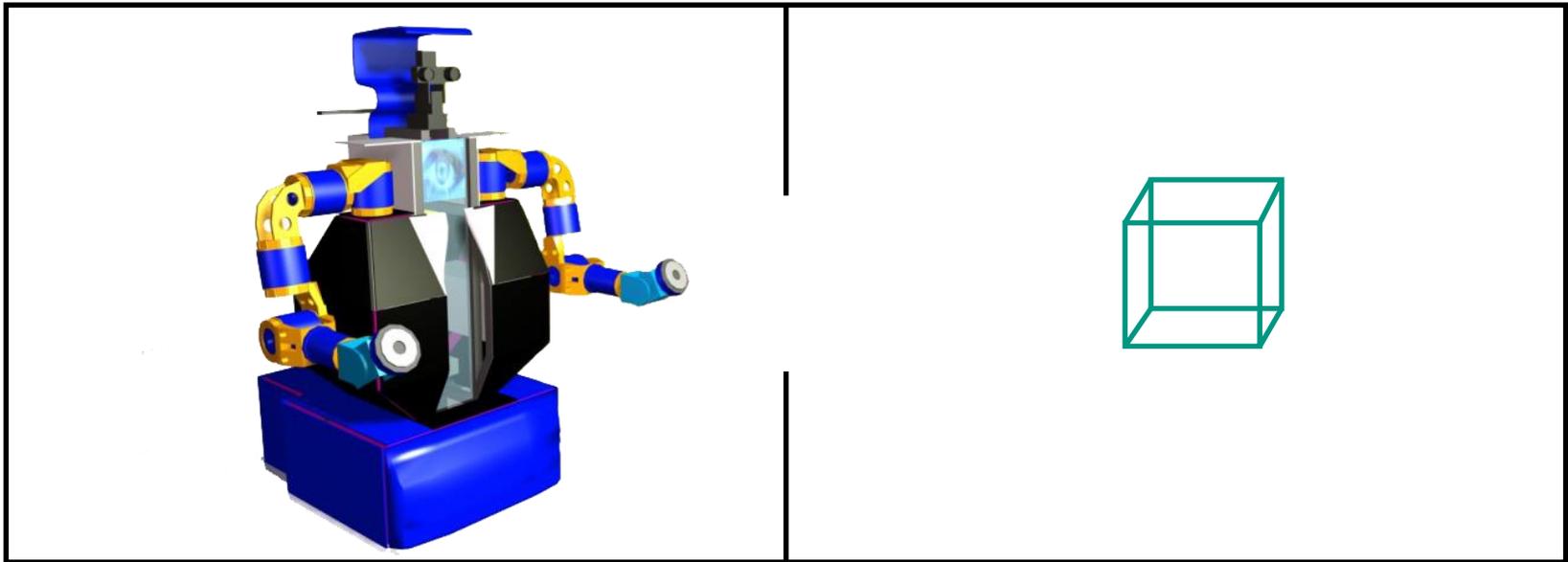
- **Beispiel:**  
S = INROOM(ROBOT,ROOM1) &  
INROOM(BOX1,ROOM2)  
  
T = INROOM(ROBOT, ROOM1) &  
INROOM(BOX1,ROOM1)



# STRIPS: Makrooperatoren II

- STRIPS Operatoren sind wie folgt beschrieben:
  - Vorbedingung (Precondition)
  - Liste zu löschender Aussagen (Delete list)
  - Liste hinzuzufügender Aussagen (Add list)

# STRIPS: Makrooperatoren III



*Operator:*  
**GOTROUGH(d,r1,r2)**  
**PC : INROOM(ROBOT,r1) &  
CONNECTS(d,r1,r2)**  
**DL : INROOM(ROBOT,r1)**  
**AL : INROOM(ROBOT,r2)**

*Operator:* **PUSHTHROUGH(b,d,r1,r2)**  
**PC : INROOM(ROBOT,r1) & CONNECT(d,r1,r2)  
& INROOM(b,r1)**  
**DL : INROOM(ROBOT,r1), INROOM(b,r1)**  
**AL : INROOM(ROBOT,r2), INROOM(b,r2)**

- Neuer Makrooperator TRANSFER
- Holen einer Box aus einem benachbarten Raum

***TRANSFER(b,r1,r2):***

**PC: INROOM(ROBOT,r1) & CONNECTS(d,r1,r2) &  
INROOM(b,r2) & CONNECTS(d,r2,r1)**

**AL: INROOM(b,r1)**

**DL: INROOM(b,r2)**

# STRIPS: Zusammenfassung

- **Vorgehensweise:**
- **Explain:**
  - Aufbau der Makrooperatoren aus Elementaroperatoren
- **Generalisierung:**
  - Ersetze Konstante durch Variable
  - Stelle Verbindungen zwischen Variablen her (auf Basis von Operator-Beziehungen und Vorbedingungen)
- **Operationalisierung:**
  - Explizite Angabe der Vorbedingungen für Anwendung
  - Indizierung über Vorbedingungen
- **Erweiterung:**
  - Probabilistische STRIPS – mehrere Zielzustände mit einer gewissen Wahrscheinlichkeit erreichbar [6]

# Einordnung EBL

Typ der Inferenz	<i>induktiv</i>	↔	<i>deduktiv</i>
Ebenen des Lernens	<i>symbolisch</i>	↔	<i>subsymbolisch</i>
Lernvorgang	<i>überwacht</i>	↔	<i>unüberwacht</i>
Beispielgebung	<i>inkrementell</i>	↔	<i>nicht inkrementell</i>
Umfang der Beispiele	<i>umfangreich</i>	↔	<i>gering</i>
Hintergrundwissen	<i>empirisch</i>	↔	<i>axiomatisch</i>

# Deduktives vs. Induktives Lernen I

	Induktives Lernen	Deduktives Lernen
Ziel	<ul style="list-style-type: none"><li>• Hypothese passt zu den Daten</li></ul>	<ul style="list-style-type: none"><li>• Hypothese passt zur Bereichstheorie</li></ul>
Vorteile	<ul style="list-style-type: none"><li>• wenig a-priori Wissen</li></ul>	<ul style="list-style-type: none"><li>• wenig Beispiele notwendig</li></ul>
Nachteile	<ul style="list-style-type: none"><li>• schlecht bei geringen Datenmengen</li><li>• schlecht bei inkorrektem Bias</li></ul>	<ul style="list-style-type: none"><li>• schlecht, falls imperfekte Bereichstheorie</li></ul>

# Deduktives vs. Induktives Lernen II

- Deduktive Methoden:
  - Durch Logik gerechtfertigte Hypothesen
- Induktive Methoden:
  - Statistisch gerechtfertigte Hypothesen



# Kombination induktiver und deduktiver Lernmethoden

- Reale Lernprobleme: zwischen induktiv und deduktiv
  - Hintergrundwissen vorhanden, aber nicht beliebig viel
  - Hintergrundwissen korrekt?
  - Trainingsdaten in begrenzter Menge
  - Trainingsdaten korrekt?

	Induktives Lernen	Deduktives Lernen
<b>Ziel</b>	Hypothese passt zu Daten	Hypothese passt zu Hintergrundwissen
<b>Rechtfertigung</b>	Statistische Inferenz	Deduktive Inferenz
<b>Vorteile</b>	Benötigt wenig Vorwissen	Lernt aus spärlichen Daten
<b>Probleme</b>	Spärliche Daten, inkorrekt gewählter Bias	Unvollständiges Hintergrundwissen

[4]

- Hybride Verfahren sollten ...
  - Ohne Hintergrundwissen so effektiv lernen wie rein induktive Verfahren
  - Mit perfektem Hintergrundwissen so effektiv lernen wie rein deduktive Verfahren
  - Mit unvollständigem Hintergrundwissen und Trainingsdaten besser als rein induktive und rein deduktive Verfahren sein
  - Mit unbekannter Menge von Fehlern in Trainingsdaten zurechtkommen
  - Mit unbekannter Menge von Fehlern in Hintergrundwissen zurechtkommen
- Realität: Wunschliste nur teilweise realisierbar

# Definition Lernproblem

## ■ Gegeben:

- Menge von Trainingsbeispielen  $D$ , möglicherweise mit Fehlern
- Bereichstheorie  $B$ , möglicherweise fehlerhaft
- Hypothesenraum  $H$

## ■ Ziel: Finde Hypothese $h$ , die am besten sowohl zu Trainingsbeispielen als auch zu Bereichstheorie passt.

## ■ Entscheidung über beste Hypothese z.B. mittels

$$\arg \min_{h \in H} k_D E_D(h) + k_B E_B(h)$$

- $E_D, E_B$ : Fehlerrate bezüglich Trainingsdaten/Bereichstheorie
- $k_D, k_B$ : relatives Gewicht für Trainingsdaten/Bereichstheorie

## ■ Frage: Wie $k_D, k_B$ wählen?

# Ansätze zur Konstruktion hybrider Verfahren

- Lernen als Suche im Hypothesenraum
  - Hypothesenraum  $H$
  - Initiale Hypothese  $h_0$
  - Suchoperatoren  $O$
  - Zielkriterium  $G$
- Verschiedene Möglichkeiten zur Nutzung von Vorwissen um induktive Suche zu verfeinern
- Verwende Vorwissen
  - um initiale Hypothese  $h_0$  abzuleiten → Bsp. KBANN
  - um Zielkriterium  $G$  der Suche zu ändern
  - um mögliche Suchschritte zu ändern

- Initialisiere Neuronales Netz mittels Bereichstheorie
- Verfeinere initiales Netz durch Backpropagation und Trainingsbeispiele
- Verwendung:
  - Bereichstheorie korrekt → alle Beispiele korrekt klassifiziert
  - Beispiele inkorrekt klassifiziert → Fehler in Bereichstheorie, induktive Verfeinerung durch Backpropagation
- **Intuition:**
  - Sogar wenn Bereichstheorie nur annähernd korrekt, Starthypothese besser als zufällige Initialisierung

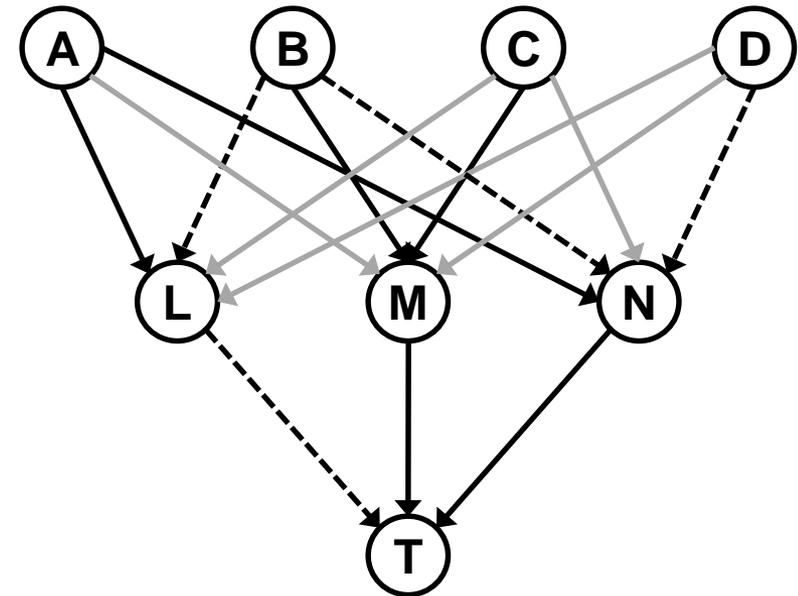
- Gegeben:
  - Menge von Trainingsbeispielen
  - Bereichstheorie aus nicht-rekursiven Prolog-ähnlichen logischen Ausdrücken
- Gesucht:
  - NN das zu Trainingsdaten passt, angelehnt an Bereichstheorie
- Verfahren in 2 Schritten
  - 1) Aufbau eines initialen NNs äquivalent zur Bereichstheorie
  - 2) Verfeinerung des initialen NNs

[4] [5]

## ■ Algorithmus

1. Pro Instanzattribut ein Netz-Input
2. Für jede Klausel der Bereichstheorie ein Neuron wie folgt einfügen:
  - Eingang mit geprüften Attributen verknüpfen, verwende Gewicht  $W$  für nicht-negierte Attribute und  $-W$  für negierte
  - Setze Schwellwert auf  $-(n - 0.5)W$  (mit  $n$  = Anzahl der nicht-negierten Bedingungssteile)
3. Zusätzliche Verbindungen, um jedes Element auf Schicht  $i$  mit jedem auf Schicht  $i+1$  zu verbinden, zufällige kleine Gewichte zuweisen
4. Backpropagation-Algorithmus mit den Trainingsbeispielen auf initiales Netz anwenden

■ Beispiel:  $T \leftarrow \neg L, M, N$   
 $L \leftarrow A, \neg B$   
 $M \leftarrow B, C$   
 $N \leftarrow A, \neg B, \neg D$



- Lernen von physikalischen Objektklassen [4]
  - Instanzen beschrieben durch Angaben über Material
  - Beispielaufgabe: Zielkonzept *Tasse* lernen
  - signifikante Verfeinerung des initialen Netzes durch Backpropagation-Algorithmus
  
- Erkennung biologischer Konzepte in DNS-Sequenzen [5]
  - Gen-Sequenzen repräsentiert als Strings
  - gesucht: bestimmte genetische Regionen (Promotoren)
  - Bereichswissen aus biol. Forschung extrahiert
  - KBANNs zeigen gute Resultate

# Was wir heute gemacht haben...

- Einführung / Motivation des Deduktiven Lernens
- Explanation Based Learning
- Explanation Based Generalization mit Beispiel
  - Anwendung: STRIPS
- Hybride Verfahren
  - Beispiel KBANN