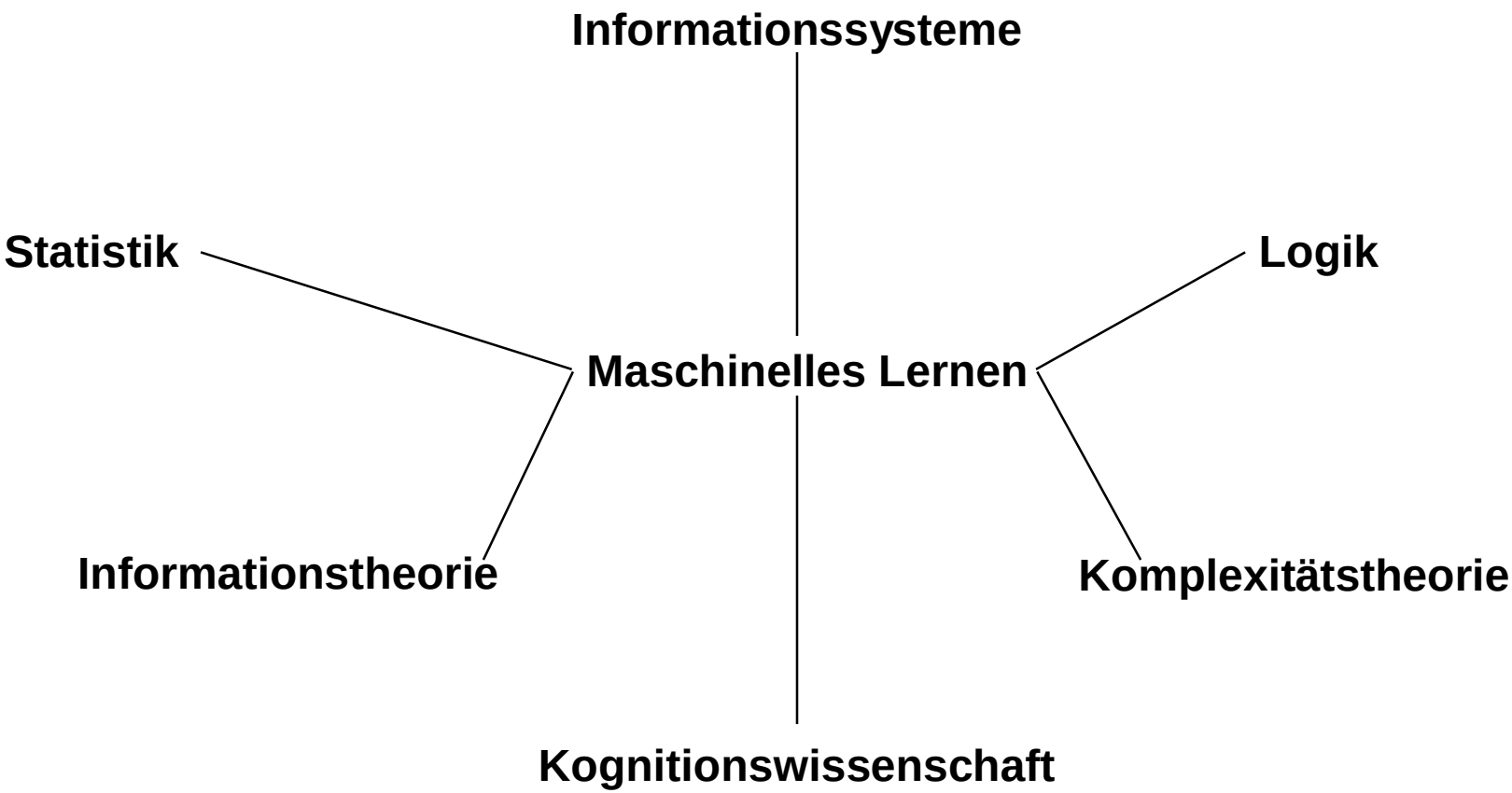
Maschinelles Lernen und Data Mining



Anwendungen

Entdecken von Mißbrauchmustern in Kreditkarten-Transaktionen

Entdecken von Merkmalen „guter" Kunden

Heraussuchen „interessanter" oder zu einem Thema gehörender Web-Seiten

Klassifikation von Objekten in Bilddaten

Vorhersage des Abverkaufs von Artikeln (Lagerhaltung)

Finden von Assoziationen zwischen Waren oder zwischen Kunden und Waren

Anpassen an einen Sprecher beim Verstehen gesprochener Sprache

Erkennen von handgeschriebenen Buchstaben, Zahlen

Spiele lernen, z.B. Backgammon durch Spiele eines Systems gegen sich selbst

Lernen einer Grammatik für eine Sprache

Lernen von Programm-Code aus Beispielen der gewünschten Ein-/Ausgabe

...

Lernen ist...

... jeder Vorgang, der ein System in die Lage versetzt, bei der zukünftigen Bearbeitung derselben oder einer ähnlichen Aufgabe diese besser zu erledigen. (Simon 1983)

Was heißt „besser"?
Was für den einen besser ist, ist für den anderen schlechter!
Lernen ohne Ziel!

... das Konstruieren oder Verändern von Repräsentationen von Erfahrungen. (Michalski 1986)

Lernen ist...

Wissenserwerb (Begriffe, Theorie, Sprache)

Definition eines Begriffs aus seinen Beispielen,
zusammenhängende Definitionen ergeben eine Theorie
Grammatik aus wohlgeformten Sätzen

Funktionslernen (Klassifikation, Regression)

$f(\vec{x}) = y, \quad y \in R \vee y \in [0,1]$

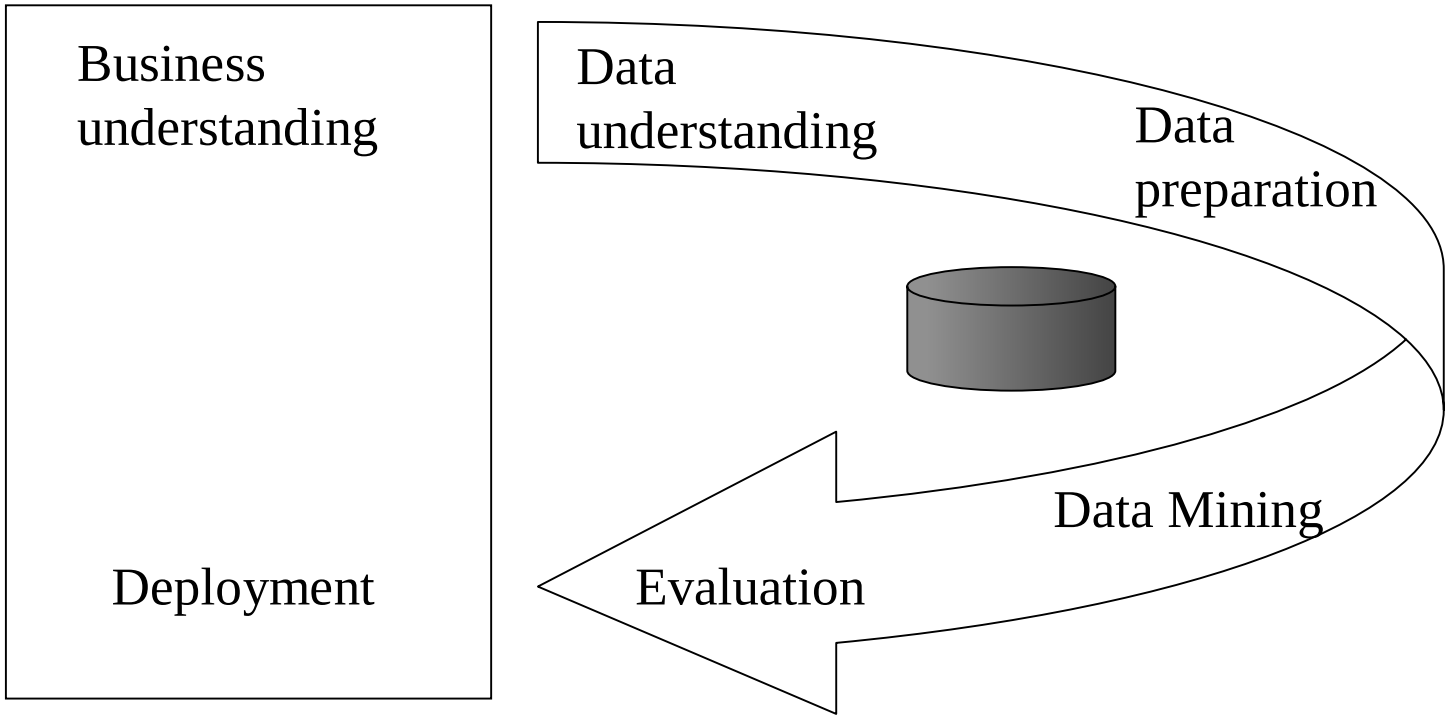
Suche im Hypothesenraum

Mögliche Lösungen werden geordnet aufgezählt,
bei der richtigen wird angehalten

der induktive Schluß

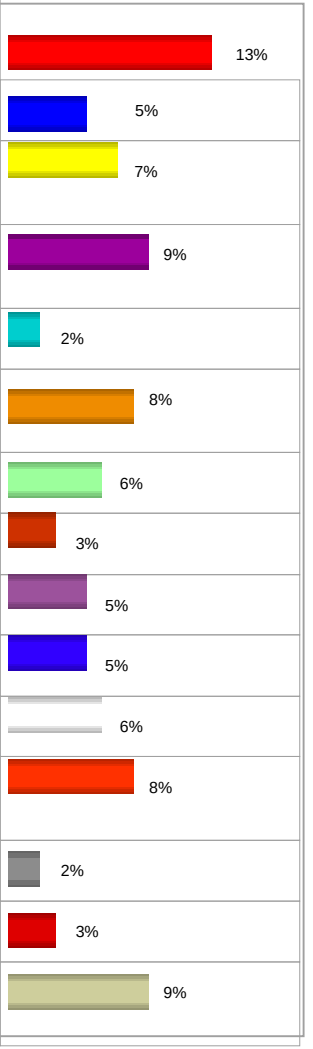
Uta ist ein Mensch, Uta ist sterblich, so auch Uli, Vroni, Sokrates...
also
alle Menschen sind sterblich

CRISP-DM Process Model



Kdnuggets 2002 Poll

Industries/fields where you currently apply data mining: [608 votes total]



Wissensentdeckung ist...

... der nichttriviale Prozess der Identifikation gültiger, neuer, potenziell nützlicher und schlussendlich verständlicher Muster in (sehr großen) Datenbeständen.

Maschinelles Lernen wird in der Wissensentdeckung als data mining step verwendet.

Sogar in der Datenvorverarbeitung werden neuerdings maschinelle Lernverfahren eingesetzt.

Einige Verfahren des maschinellen Lernens (Adaptivität und Optimierung) werden nicht in der Wissensentdeckung verwendet.

Wissensentdeckung benötigt Verfahren, die sehr große Datenmengen verarbeiten können.

Wissenschaftliche Fragen

Wieviele Beispiele muß ich mindestens haben, bis ich ein ausreichend korrektes und vollständiges Lernergebnis erzielen kann? Wie sicher bin ich bei meiner Beispielmenge?

Wie mächtig muß mein Repräsentationsformalismus sein, damit ich ein annähernd korrektes und vollständiges Lernergebnis ausdrücken kann? Wie schwach darf er sein?

Unter welchen Umständen wird der Lernalgorithmus zu einem Ergebnis kommen und anhalten? Wie schnell ist er?

Welche Zusicherungen kann der Algorithmus über sein Ergebnis garantieren? z.B.: dies ist die speziellste Generalisierung über allen Daten -- wenn sie falsch ist, sind auch die Daten falsch!
z.B.: Dies sind alle Regeln, die in den Daten verborgen sind -- wenn eine fehlt, fehlen auch die entsprechenden Daten!

Ausschnitt der Vorlesung

Algorithmen	Lernaufgaben	Theorie
Versionenraum	Begriffslernen	PAC
lgg	Begriffslernen	Logik
Entscheidungs-baumlernen	Begriffslernen	Statistik
SVM	Begriffslernen, Regression	Statistik
backprop	Begriffslernen	PAC
Data cube	Dateninspektion	Datenbanken
Apriori	Assoziationslernen	Datenbanken

Vorkenntnisse?

- Logik
  - Logisches Modell
  - Logische Folgerung
  - Resolution: Unifikation durch Substitution, Schnittregel
  - (Horn-)Klauseln
- Statistik
  - Zufallsvariable
  - (bedingte) Wahrscheinlichkeit
- Praktische Fertigkeiten
  - JAVA
  - SQL
  - Rechnerbedienung

Lernziele

- Unterschiedliche Definitionen als Präzisierung (Formalisierung) des Lernbegriffs verstehen und herleiten können.
  - Lernaufgabe
  - Formalismus
  - Zusämmenhänge zwischen den Paradigmen
- Algorithmen kennen und anwenden können.
  - Welches Verfahren werde ich als erstes für ein Problem probieren?
  - Wie mache ich das?
- Eigenschaften von Algorithmen kennen.
  - Komplexität (Effizienz)
  - Zusicherungen der Qualität (Effektivität)
- Originalliteratur verstehen können.

Übungsschein

- Zu jeder Vorlesungsstunde kommen und zuhören!
- Nachbereiten, indem Materialien gelesen,
  Fragen in der Gruppe diskutiert werden,
  Dies erfordert mindestens 2 Stunden pro Woche.
- Alle Übungsaufgaben bis auf 2 müssen richtig abgegeben werden.
  Dazu sind mindestens noch einmal 4 Stunden pro Woche nötig.
- Diesmal: erste Übungsstunde wird eine Vorlesung!

Arbeitsmaterial

Texte:

Tom Mitchell "Machine Learning" McGraw-Hill Companies, 1997

Ian Witten, Eibe Frank „Data Mining- Praktische Werkzeuge und Techniken für das maschinelle Lernen" Hanser, 2001

Stefan Wrobel, Katharina Morik, Thorsten Joachims „Maschinelles Lernen und Data Mining" in: „Handbuch der KI" Görz, Rollinger, Schneeberger (Hg.), Oldenbourg, 2000

Mein altes Skript

Folien:

www-ai.cs.uni-dortmund.de

Software:

Weka-Bibliothek: www.cs.waikato.ac.nz/ml/weka

Was haben wir gelernt?

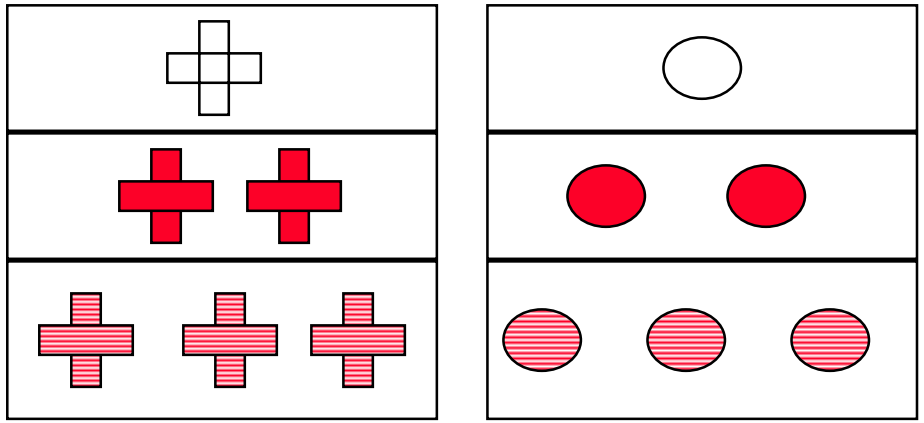
- Intuitive Einführung in die Definitionen von maschinellem Lernen und Wissensentdeckung.
- Umfang und Qualitätskriterien der Vorlesung

Was werden wir nächstes Mal lernen?

- Einführung in die Lernaufgabe des Begriffslernens
- Qualitätskriterien Lernergebnisse: Korrektheit, Vollständigkeit
- Lernen als Suche

Begriffsbildung Kategorisierung

- alles zusammenfassen zu einer Klasse, was gemeinsame Merkmale hat
- was sind Merkmale?



Bedarf für Kategorisierung:

Es gibt schon ein Wort dafür

Vorhersage ist nötig

Begriff erleichtert Definition anderer Begriffe

Begriffsbildung Charakterisierung

Kategorien abgrenzend beschreiben

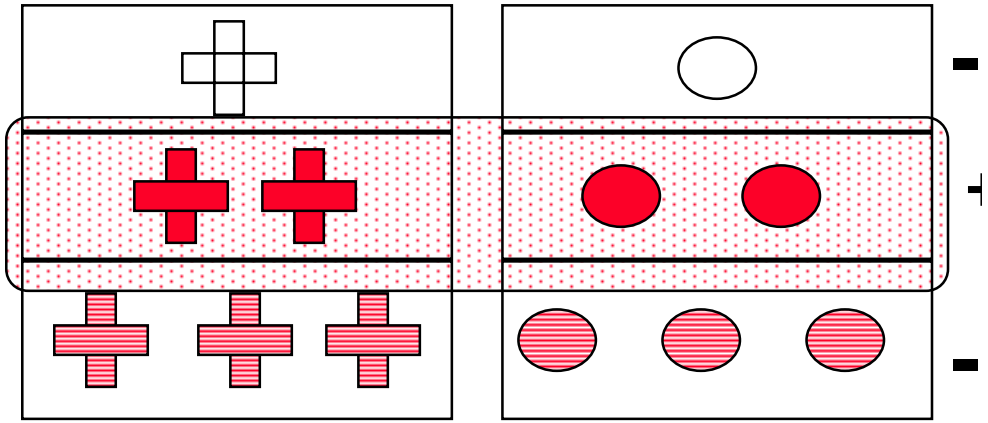
für Gegensätze dieselben Merkmale verwenden

so wenig Merkmale wie möglich

Vererbbarkeit der Merkmale

Operationalität der Merkmale

- sind Begriffe und Merkmale wirklich verschieden?



Alternative Lernergebnisse:

a) ist rot,

b) besteht aus 2 Teilobjekten,

c) ist rot und besteht aus 2 Teilobjekten

Probleme der Charakterisierung

Herkunft der Merkmale:

- verankert in der Wahrnehmung
- vermittelt über die Sprache/Kultur

Auswahl von Merkmalen:

- nur solche Merkmale, die auf alle Begriffe in Gegensatz-Relation anwendbar sind (Konsistenz)
- Vererbung definitorischer Merkmale
- beschreibende vs. erklärende Merkmale je nach Wissensstand

Begriffsrepräsentation:

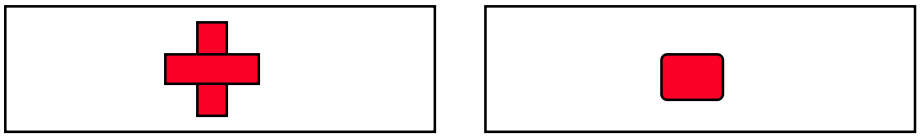
- Merkmale und Begriffe sind eigentlich nicht verschieden!

Anwendung des Begriffs Klassifikation

a) ist rot: +

b) besteht aus 2 Teilobjekten: -

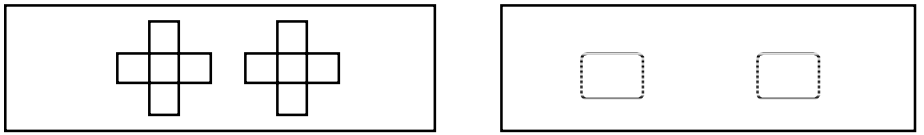
c) ist rot und besteht aus 2 Teilobjekten: -



a) ist rot: -

b) besteht aus 2 Teilobjekten: +

c) ist rot und besteht aus 2 Teilobjekten: -

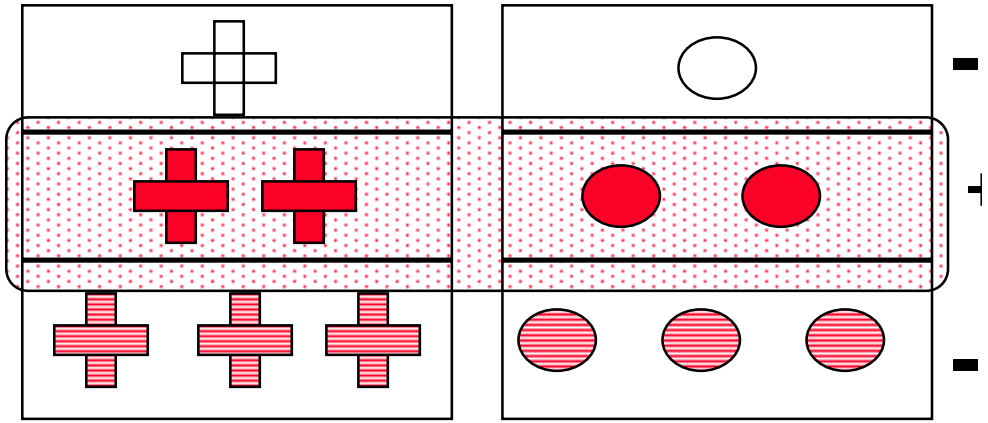


Lösung:
a) ist rot

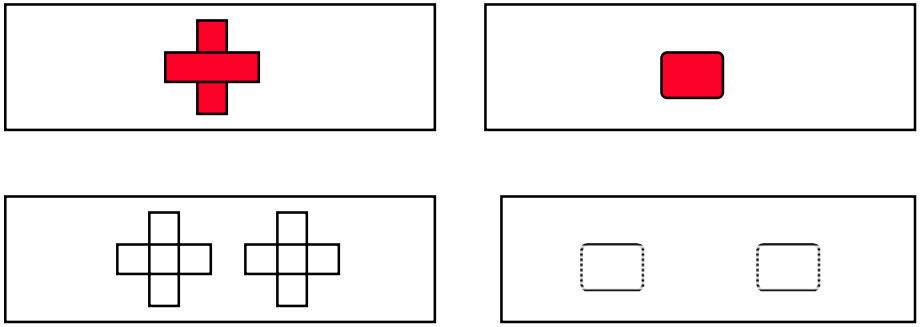
Lernmenge/Testmenge

Lernmenge:
Menge von Beispielen =
klassifizierte Beobachtungen

Lernen einer Definition



Testmenge:
Menge von Beispielen, bei denen
die tatsächliche Klassifikation mit
der von der Definition vorhergesagten
verglichen wird



Kreuzvalidierung

Man teile alle verfügbaren Beispiele in n Mengen auf, z.B. n= 10.

Für i=1 bis i=n:

Wähle die i-te Menge als Testmenge und

die restlichen n-1 Mengen als Lernmenge.

Messe Korrektheit und Vollständigkeit auf der Testmenge.

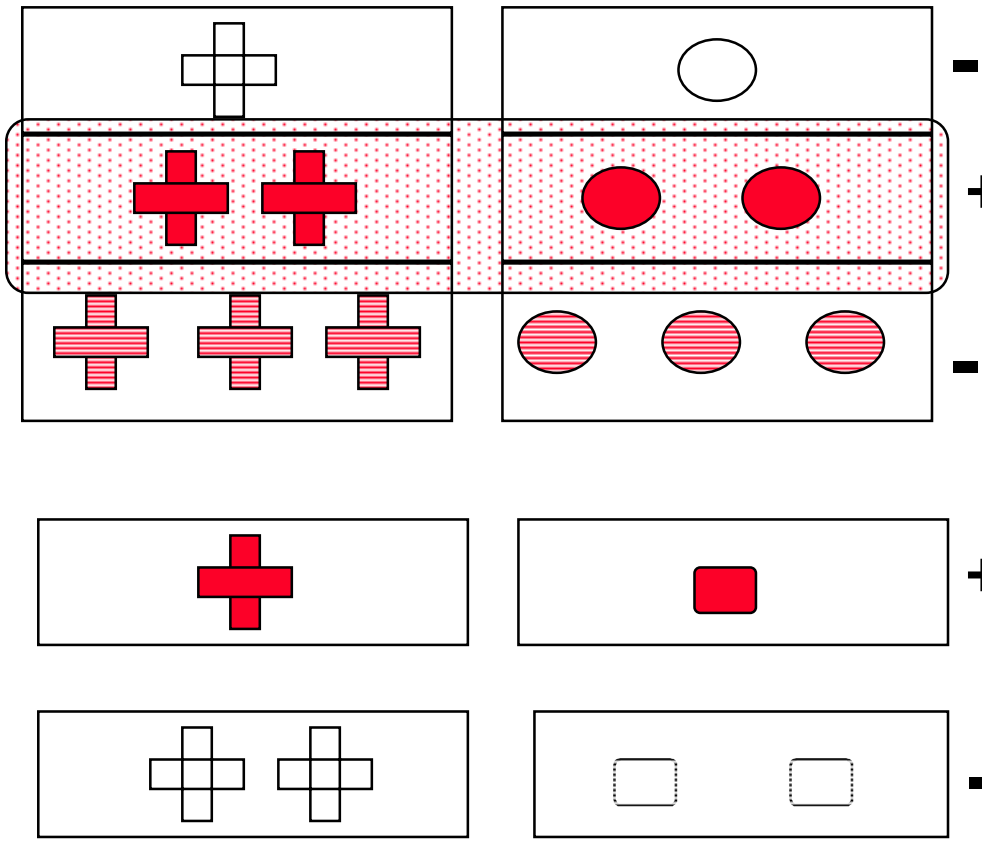
Bilde das Mittel der Korrektheit und Vollständigkeit
über allen n Lernläufen.

Das Ergebnis gibt die Qualität des Lernergebnisses an.

korrekt, vollständig

korrekt ist eine Definition, wenn sie kein negatives Beispiel abdeckt;

vollständig ist eine Definition, wenn sie alle positiven Beispiele abdeckt



a) ist rot:

korrekt und vollständig

b) besteht aus 2 Teilobjekten:

nicht korrekt, nicht vollständig

c) ist rot und besteht aus 2 Teilobjekten:

korrekt, unvollständig

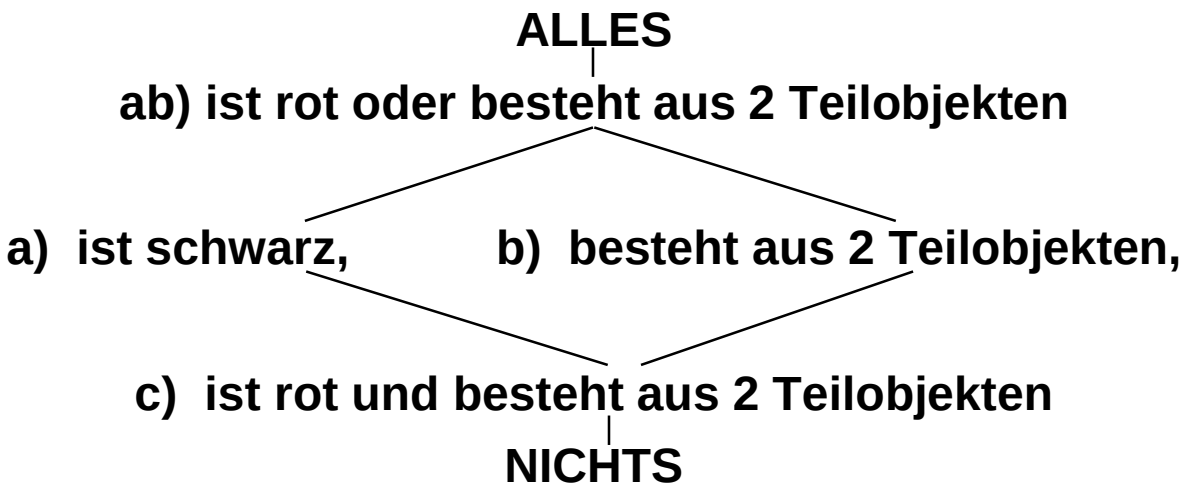
Sprachen

LE für die Beobachtungen oder Beispiele

zulässig!

LH für die Begriffsdefinitionen bzw. Hypothesen

Allgemeinheitsordnung der Hypothesen



Lernen als Suche

Top-down:

Beginne mit der allgemeinsten Hypothese;

solange noch negative Beispiele abgedeckt werden -- spezialisieren!

sonst -- anhalten.

Bottom-up:

Beginne mit den speziellsten Hypothesen;

solange noch positive Beispiele nicht abgedeckt werden -- generalisieren!

sonst -- anhalten.

Bidirektionale Suche Versionenraum

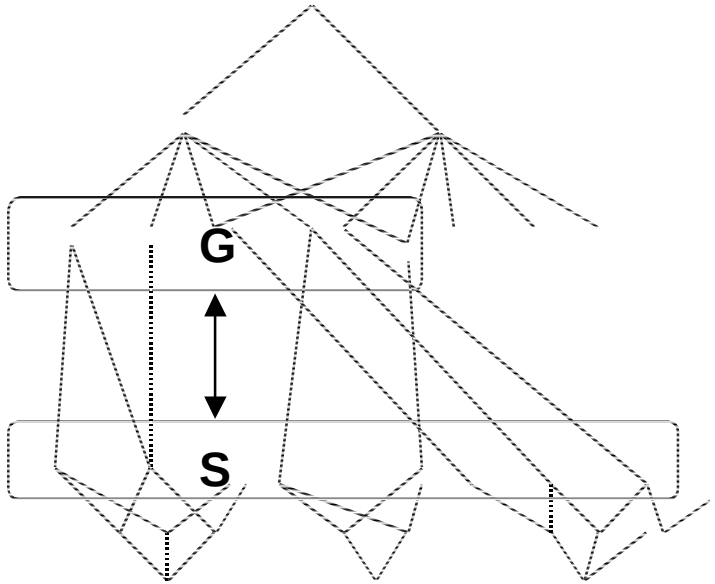
G: { g |∀ p ∈P: covers (g, p),
∀ n ∈ N: ¬ covers (g, n),

es gibt kein g', das genereller ist als g und konsistent mit P und N ,

es gibt ein s ∈ S, das spezieller ist als g }

S: {s | ∀ p ∈ P: covers (s, p),
∀ n ∈ N: ¬ covers (s, n),

es gibt kein s', das spezieller ist als s und konsistent mit P und N
es gibt ein g ∈ G, das genereller ist als s }



Verfahren

Initialisiere G mit den generellsten, S mit den speziellsten Begriffen.

Solange G und S disjunkt sind, lies ein neues Beispiel e ein.

Falls $e \in N$:

entferne alle s aus S, die e abdecken;

für alle $g \in G$, die e abdecken:

entferne g aus G,

füge alle schrittweisen Spezialisierungen h von g hinzu, so daß gilt:

h deckt nicht e ab,

es gibt ein $s \in S$, das spezieller ist als h;

entferne alle $g \in G$, die echt spezieller sind als ein anderes $g' \in G$

Verfahren (cont'd)

Falls e $\in$ P:

entferne alle g aus G, die e nicht abdecken;

für alle s $\in$ S, die e nicht abdecken:

entferne s aus S,

füge alle schrittweisen Generalisierungen h von s hinzu, so daß gilt:

h deckt e ab,

es gibt ein g $\in$ G, das genereller ist als h;

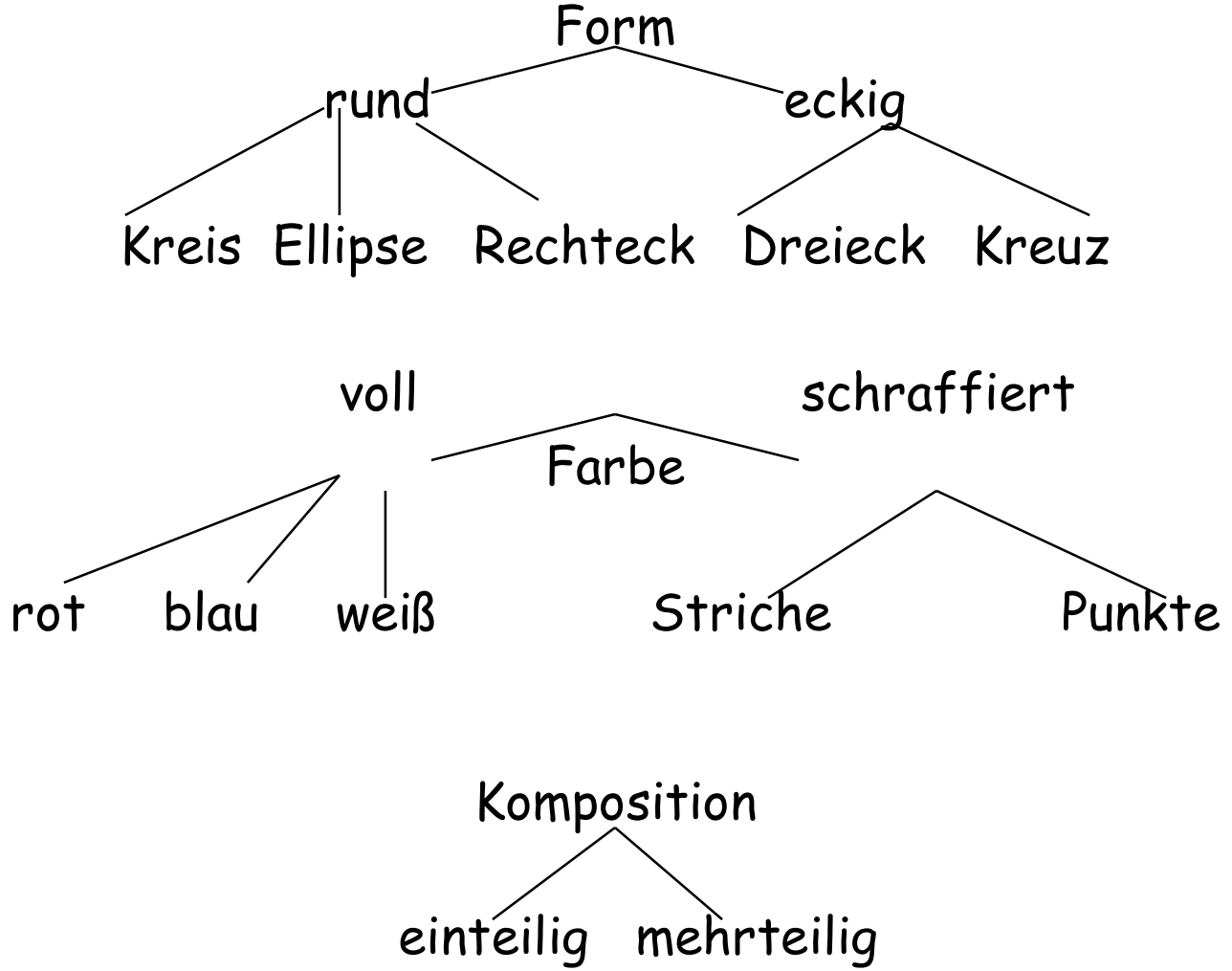
entferne alle s $\in$ S, die echt genereller sind als ein anderes s' $\in$ S

Sobald S = G , so ist das Ergebnis gefunden:

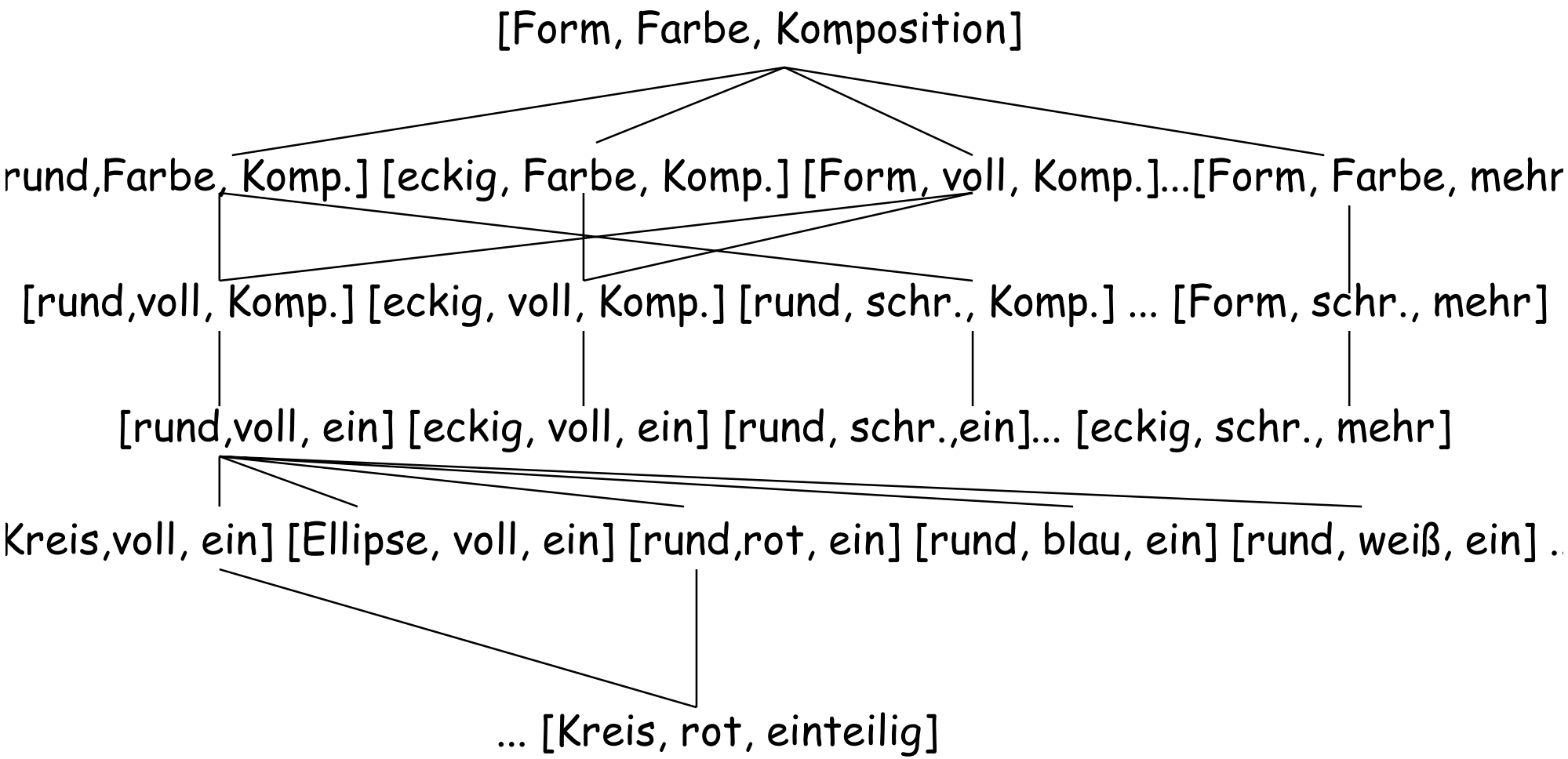
gib die Hypothese aus und halte an!

Sonst: Mißerfolg (zu wenig Beispiele)

Beispiel: Beispielsprache LE



Beispiel: Hypothesenraum LH



Beispiel

- **[Kreuz,rot, mehrteilig] p**

  G= {[Form,Farbe, Komposition]}
  S= {[Kreuz,rot,mehrteilig]}

- **[Kreuz,schraffiert,mehrteilig] n**

  G= { [Form, voll, Komposition] } ¬[rund, Farbe, Komposition] > s
  S= {[Kreuz,rot,mehrteilig]}

- **[Kreuz,rot, einteilig] p**

  G= { [Form,voll,Komposition] }
  S= { [Kreuz, rot, Komposition] } ¬covers([eckig, rot, mehrteilig],e)

- **[Ellipse, weiß, einteilig] n**

  G= { [Form, rot, Komposition]} >> [eckig, voll, Komposition]
  S={ [Kreuz,rot,Komposition]}

- **[Ellipse, rot, mehrteilig] p**

  G= { [Form, rot, Komposition]}
  S= { [Form, rot, Komposition]}

Eigenschaften des Verfahrens

Der Versionenraum enthält alle Hypothesen S und G und dazwischen.

Die Mengen S und G enthalten alternative Hypothesen.

Gelernt wird eine (die richtige) Hypothese, wenn

- die Beispiele keine Fehler enthalten,
- die richtige Hypothese in LH ausdrückbar ist,
- der Hypothesenraum sich (halb-)ordnen läßt
- und die Notwendigkeit zu generalisieren und zu spezialisieren gegeben ist.

Was haben wir gelernt?

- Suche beinhaltet
  - einen Raum,
  - Operatoren zur Erzeugung der Nachfolger und
  - eine Suchstrategie
- Lernen als Suche
  - Halbgeordneter Hypothesenraum als Suchraum
  - Schrittweise Generalisierung, schrittweise Spezialisierung als Operatoren
  - Versionenraumalgorithmus als bidirektionale Suchstrategie
- Die Halbordnung ist in LH begründet.
- Sie erlaubt, ohne Risiko viele Hypothesen auszulassen.

Was kommt heute?

- Lernaufgabe Begriffslernen (Klassifikationslernen) als Spezialfall des Funktionslernens
- Was ist ein Fehler?
- Erste Theorie: PAC-Einführung
- Wieviele Beispiele braucht der Versionenraum?

Funktionslernen

Gegeben:

Beispiele X in LE

die anhand einer Wahrscheinlichkeitsverteilung D auf X erzeugt wurden und

mit einem Funktionswert Y = t(X) versehen sind. t(X) kann mit einer bestimmten Wahrscheinlichkeit falsch sein (verrauschte Daten).

H die Menge von Funktionen in LH

Ziel:

Eine Hypothese $h(X) \in H$, die das Fehlerrisiko minimiert

Risiko:

$$R(H) = \sum_{i=1}^{n} Q(X_i, H)\ P(X)$$

Fehler Q(X, H)

P(X) Wahrscheinlichkeit, daß das Beispiel X aus der Menge aller Beobachtungen gezogen wird

Q(X, H) Fehlerfunktion, die angibt, wie genau H die Funktion approximiert

Klassifikation:

t(X) ist jeweils eine Klassenbezeichnung

Q(X, H) ist 1, falls $t(X) \neq h(X)$,
0, falls $t(X) = h(X)$

Regression:

t(X) ist eine reelle Zahl

quadratischer Fehler $\mathbf{Q}(\mathbf{X}, \mathbf{H}) = (\mathbf{t}(\mathbf{X}) - \mathbf{h}(\mathbf{X}))^2$

Minimierung des beobachteten Fehlers

Da wir die tatsächliche Funktion t(X) nicht kennen, können wir nur
eine hinreichend große Lernmenge nehmen
und für diese den Fehler minimieren.

empirical risk minimization

Lernbarkeitstheorie

LH

Größe des Hypothesenraums, Schwierigkeit der Suche

Zusicherungen an das Lernergebnis

Korrektheit, speziellste Generalisierung, generellste Diskriminierung

Konvergenz

Wann, unter welchen Umständen, mit welcher Wahrscheinlichkeit wird ein angemessenes Lernergebnis ausgegeben?

Wieviele Beispiele werden für das Erreichen eines angemessenen Lernergebnisses gebraucht?

Wahrscheinlich annähernd korrektes Lerner (PAC)

X in LE: alle möglichen Beobachtungen, repräsentiert in LE

Grafische Objekte, beschrieben durch Form, Farbe, Kompositiohn

x in X: ein Beispiel

[Kreuz, rot, einteilig]

C in LH: mögliche Begriffe, die zu lernen sind

c in C: Zielbegriff

rote Objekte [Form, rot, Komposition]

als Funktion: X --> {0,1}

D: Distribution der Beispiele -- hier: nicht bekannt, aber fest

Beispiele werden gemäß D zufällig gezogen

Wenn es viele rote Kreuze gibt, werden auch viele als Beispiel vorkommen.

PAC-learning

Eine Begriffsklasse C mit Bezug auf Beispiele X ist PAC-lernbar durch einen Lernalgorithmus L und einen Hypothesenraum H, wenn für alle c in C, Verteilungen D über X, $0 < \varepsilon < 1/2$ und $0 < \delta < 1/2$

- L mit einer Wahrscheinlichkeit von mindestens $1 - \delta$
- eine Hypothese h aus H ausgibt, deren Fehler über der Verteilung D nicht größer ist als ε
- in einer Zeit, die durch ein Polynom über $1/\varepsilon$, $1/\delta$, $|c|$ und $|x|$ begrenzt ist.

L liefert also

wahrscheinlich

ein annähernd korrektes Ergebnis

in poynomieller Zeit ab.

L muß aus nur polynomiell vielen Beispielen lernen,

wobei die Verarbeitung jeden Beispiels nur polynomielle Zeit benötigt.

Wieviele Beispiele braucht man im Versionenraum?

- Der Hypothesenraum ist endlich. |H|
- Alle Hypothesen in H sind konsistent mit den Beispielen.

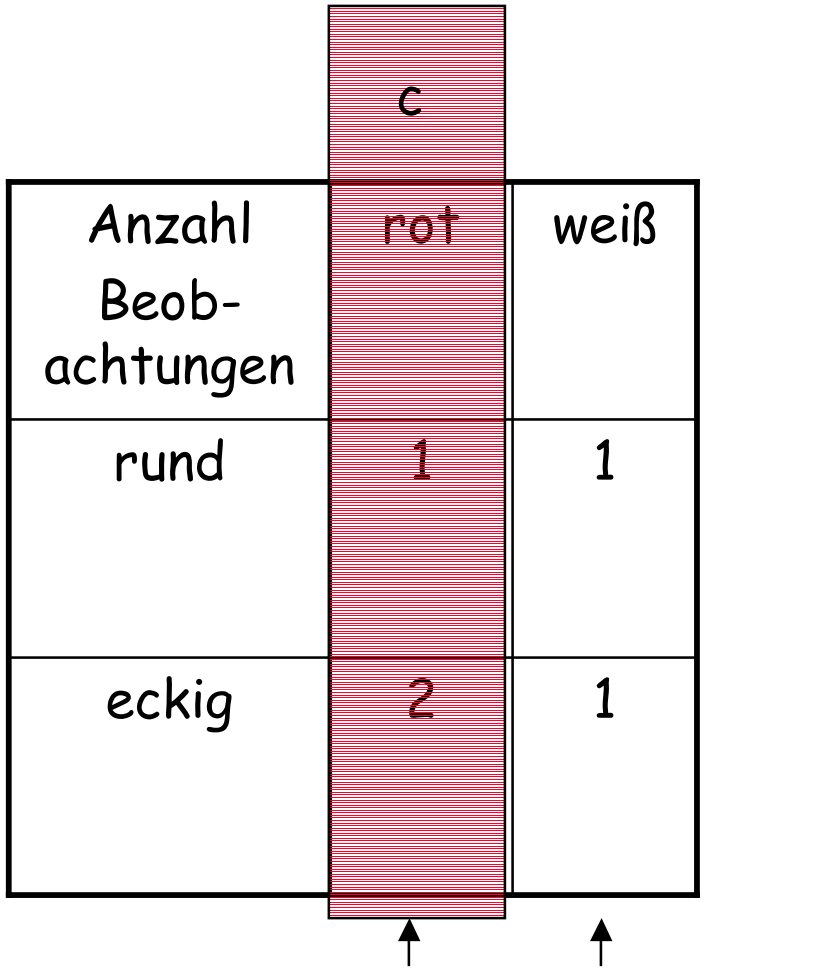
Fehler einer Hypothese: Wahrscheinlichkeit, daß ein Beispiel gezogen wird, das eine andere Klassifikation hat als die Hypothese angibt.

Nach wievielen Beispielen gibt das Verfahren mit der Wahrscheinlichkeit 1 - δ eine Hypothese aus, deren Fehler höchstens ε beträgt?

Wann brauchen wir noch Beispiele, um falsche Hypothesen auszuschließen?

Falsche Hypothesen sind korrekt bzgl. der Beispiele, aber in Wirklichkeit nicht korrekt, werden also möglicherweise durch weitere Beispiele ausgeschlossen.

Beispiel



h1: [Form, Farbe] Fehler: 2/5

h2: [rund, Farbe] Fehler: 3/5

h3: [eckig, Farbe] Fehler: 2/5

h4: [Form, rot] Fehler: 0

h5: [Form,weiß] Fehler: 5/5

h6: [rund, rot] Fehler: 2/5

h7: [rund,weiß] Fehler: 4/5

h8: [eckig, rot] Fehler: 1/5

h9: [eckig, weiß] Fehler: 4/5

Obere Schranke

H: { h_1, ..., h_k}

Die Wahrscheinlichkeit, daß eine konsistente Hypothese mit Fehler > ε

konsistent mit einem neuen Beispiel ist, ist höchstens (1 - ε).

Die Wahrscheinlichkeit, daß die Hypothese mit m Beispielen konsistent ist, beträgt also höchstens

$$(1 - \varepsilon)^m$$

Die Wahrscheinlichkeit, daß mindestens eine der Hypothesen mit m Beispielen konsistent ist (obwohl sie tatsächlich einen Fehler > ε hat), beträgt höchstens

$$|H| (1 - \varepsilon)^m$$

Da $0 \leq \varepsilon \leq 1$ begrenzen wir die Wahrscheinlichkeit, daß m Beispiele nicht die schlechten Hypothesen entfernen, durch

$$|H| \varepsilon^{- \varepsilon m}$$

Für das PAC-Lernen muß dies kleiner sein als δ.

Anzahl der Beispiele

Wahrscheinlichkeit, daß m Beispiele nicht die schlechten Hypothesen entfernen, kann nicht größer sein als

$$|H| \; \varepsilon^{-em}$$

Für das PAC- Lernen:

$$|H| \; e^{-em} \leq \delta$$

Also brauchen wir mindestens m Beispiele:

$$m \geq 1/\varepsilon \; (\ln |H| + \ln (1/\delta))$$

Beispiel – cont'd

- H enthält bei 2 Attributen mit je 2 Werten + „egal"
  $3^2 = 9$ Hypothesen
- Nehmen wir $\varepsilon = 0{,}2$ und $\delta = 0{,}1$
- $m \geq 1/0{,}2\ (\ln 9 + \ln (1/0{,}1))$
  $m \geq 5\ (2{,}2 + 2{,}3)$
- Wir brauchen also mindestens 22,5 Beispiele.
- Mehrmals die Beispiele ziehen, um verteilungsunabhängig zu sein!

Was haben wir gelernt?

- Definition Funktionslernen
- Definition PAC-learning
- Beispielkomplexität des Lernens
- Abschätzung derBeispielkomplexität bei endlichen Hypothesenräumen
- Anwendung des PAC-learning auf den Versionenraum: Bestimmung der Mindestanzahl von Beispielen

- Mehr PAC gibt es erst wieder bei neuronalen Netzen.

Was kommt jetzt?

Logik!

- Begriffslernen aus Beispielen
- Logische Bedingungen an das Lernergebnis
- Ordnung des Hypothesenraums nach Allgemeinheit durch
  - Implikation
  - Subsumtion

Begriffslernen aus Beispielen

- gegeben: Hypothesensprache LH für den Begriff
- Hintergrundwissen B in einer Sprache LB
- Menge E = P ˙ N in einer Sprache LE
- wobei B, E |≠ □ , B |≠ E
- Ziel:
- H ∈ LH mit
- B, H, E |-/- □ (Konsistenz)
- B, H |-- P (Vollständigkeit)
- ∀e ∈ N: B, H |-/- e (Korrektheit)
- Präferenzkriterium:
- z.B.: speziellste Generalisierung, generellste Diskriminierung
- u.a.: triviale Lösung (Aufzählung von P) ausschließen!

Zuverlässige Lernoperatoren

LH wird durch eine Generalisierungsrelation geordnet.

Diese soll eindeutig sein:

für zwei Klauseln existiert genau eine Generalisierung.

Dann ist Lernen die Suche entlang der Generalisierungsbeziehung.

Zuverlässige Lernoperatoren liefern die speziellste Generalisierung oder generellste Spezialisierung.

Heuristische Verfahren liefern irgendeine konsistente Hypothese.

Suchoperatoren

Die Suche wird operationalisiert durch

- Operator für die Generalisierung,
- Perator für die Spezialisierung.

Dabei wollen wir alle speziellsten Generalisierungen oder alle generellsten Spezialisierungen erzeugen.

Beschneiden des Hypothesenraums

Der Hypothesenraum kann sicher beschnitten werden:

- Wenn beim Generalisieren bereits die Hypothese C1 ein negatives Beispiel abdeckt, so wird dies auch jede Generalisierung von C1 tun. Von C1 ausgehend braucht nicht generalisiert zu werden.
- Wenn beim Spezialisieren bereits die Hypothese C1 ein positives Beispiel nicht abdeckt, so wird dies auch jede Spezialisierung nicht tun. Von C1 ausgehend braucht nicht spezialisiert zu werden.

Generalisierung

Kümmern wir uns zunächst um die Generalisierungsrelation!

Danach betrachten wir Generalisierungsoperatoren.

Generalisierungsrelation: Implikation

Eine Hornklausel C1 ist genereller als eine andere, C2, gdw.

C1 --> C2

C1 ist genereller als C2 bezüglich einer Theorie T , gdw.

T, C1 --> C2

Äquivalenz:

Sei T eine Konjunktion von Hornklauseln, dann sind die Klauseln C1 und C2 logisch äquivalent bzgl. T gdw.

T, C1 --> C2 und T, C2 --> C1

Redundanz :

Ein Literal L ist redundant in einer Klausel C bzgl. T gdw.

C und C \ {L} sind äquivalent bzgl. T.

Beweisen1: Widerspruchsbeweis

- Wir wollen zeigen, dass T, C1 → C2 allgemeingültig ist.
- Dazu zeigen wir, dass T, C1, ¬ C2 widersprüchlich ist.
- Unsere Formeln sind in konjunktiver Normalform:
  - Jede Klausel enthält geODERte Literale.
  - Jede Formel besteht aus geUNDeten Klauseln.
- Wir haben also $T \wedge C1 \wedge \neg C2$
- $\neg (A \vee B)$ $(\neg A \wedge \neg B)$ (De Morgan)

Beweisen2: Skolem

- Unsere Formeln sind quantorenfrei.
  Der äußere Allquantor kann weggelassen werden.
  Innere Existenzquantoren werden durch eine Skolemfunktion ersetzt.
  - Eine Skolemfunktion ersetzt eine existenzquantifizierte Variable im Geltungsbereich von allquantifiierten Variablen V_i durch eine Funktion über V_i .

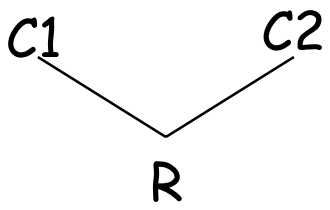
$$\forall x, y \exists z \left| p(x, y), q(x, z) \right.$$

$$p(x, y), q(x, f(x, y))$$

Beweisen3: Resolution

- Resolution zweier Klauseln C1 und C2 ergibt eine Resolvente R, falls es ein Literal L in C1 gibt und ¬ L in C2:

  $R = (C1 - \{L\}) \cup (C2 - \{\neg L\})$



- Unifikation zweier Literale durch eine Substitution, die die generellste Spezialisierung der Argumente darstellt.
  - $L1\ \sigma = L2\ \sigma$
  - Für alle σ' gilt: es gibt ein σ'' , so dass $L\ \sigma' = L\sigma\ \sigma''$

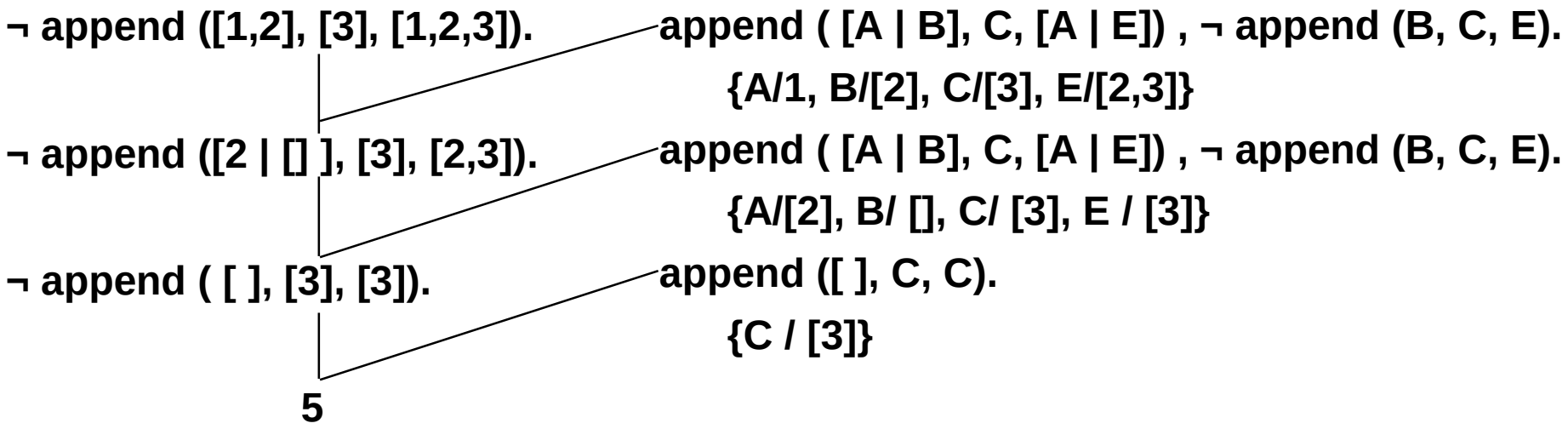
Beispiel Generalisierung

T: append ([ ], C, C).

C2: append ([1,2], [3], [1,2,3]).

C1: append([A | B], C, [A | E]) :- append (B, C, E).

C1, T --> C2



Beweis von C2 mithilfe von T und C1

Beispiel Redundanz

T: member(X, [X| Y]).

C: member (X, [Y|Z]) :- member (X, Z), member(Y, (Y |Z]).

C': C \ { member(Y, (Y | Z])} ist äquivalent zu C.

T, C --> C' und T, C' --> C

T beschreibt den Fall, daß das Element am Anfang der Liste steht.

C' beschreibt den Fall, daß das Element im Rest der Liste steht.

C beschreibt beide Fälle.

Beispiel Übungsaufgabe

T: $\forall X \mid \neg o(X) \vee w(X)$

C2: $\forall X \mid \neg s(X,10) \vee o(X)$

C1: $\forall X,Y \mid \neg s(X,Y) \vee o(X)$

C1': $\forall X \mid \neg s(X,10) \vee w(X)$

C1'': $\forall X,Y \mid \neg s(X,Y) \vee w(X)$

- Wir ordnen die Klauseln nach der Generalisierungsrelation (hier: Implikation) an, indem wir die für alle Paare von Klauseln Widerspruchsbeweise führen.
  - C1, C2
  - C1, C1'' C2, C1''
  - C1, C1' C2, C1' C1', C1''

T, C1 → C2 ?

- T ∧ C1 ist allgemeingültig.
- T ∧ C1 ∧ ¬ C2 ist widersprüchlich.

C1: ∀X,Y | ¬s(X,Y) ∨ o(X)
≡ {¬s(X,Y) ∨ o(X) }

¬ C2: ∃X | s(X,10) ∧ ¬ o(X)
≡ {s(a,10)} ∧ {¬ o(a)}

{X/a, Y/10}

{o(a) }

□

C1 ist genereller als C2 bei der Implikation als Generalisierungsrelation.

T, C2 → C1 ?

$C2$: $\forall X \mid \neg s(X,10) \vee o(X)$

$\equiv \{\neg s(X,10) \vee o(X)\}$

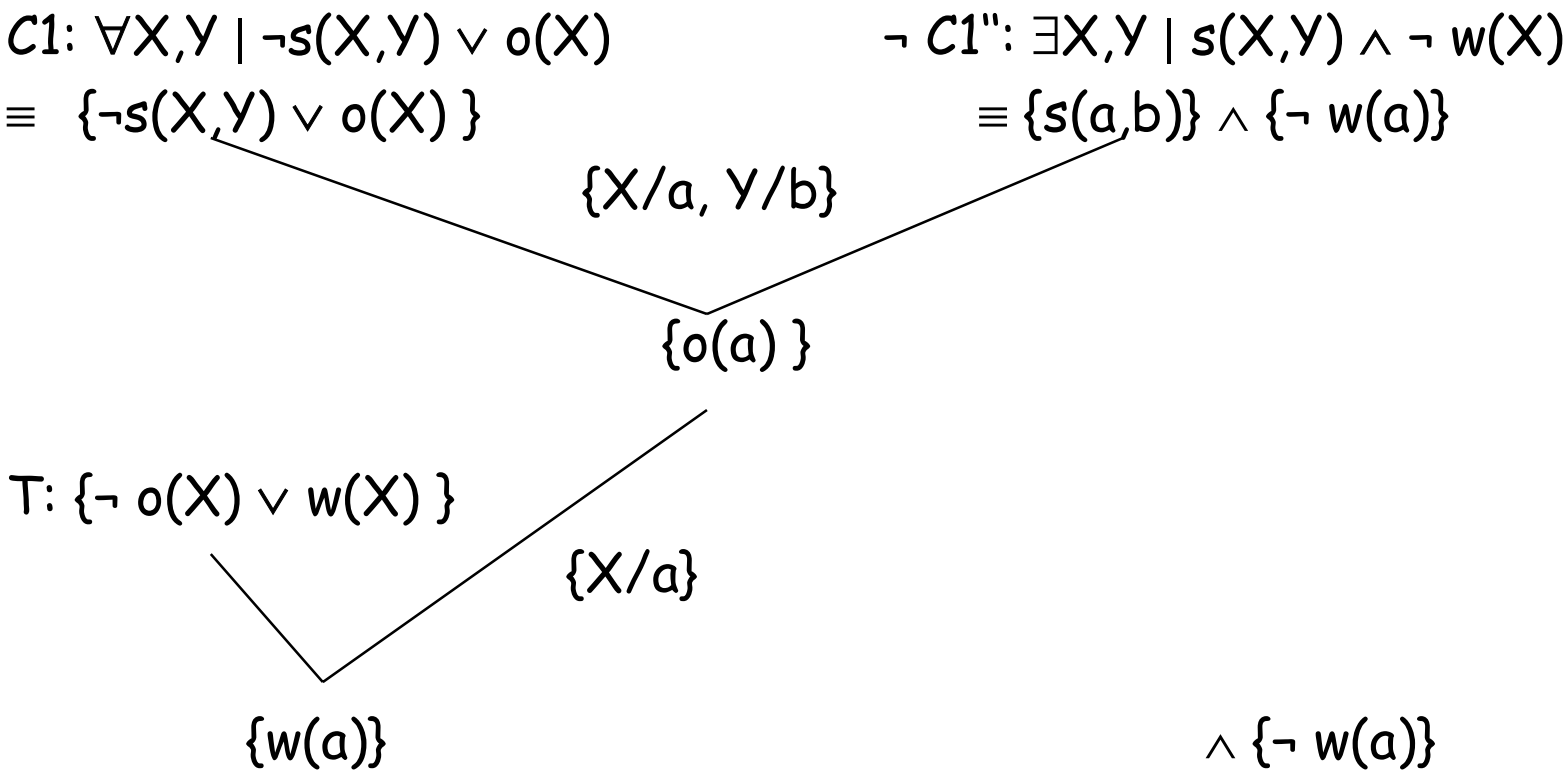
$\neg\, C1$: $\exists X,Y \mid s(X,Y) \wedge \neg\, o(X)$

$\equiv \{s(a,b)\} \wedge \{\neg\, o(a)\}$

unifiziert nicht!

C2 ist nicht genereller als C1 bei der Implikation als Generalisierungsrelation.

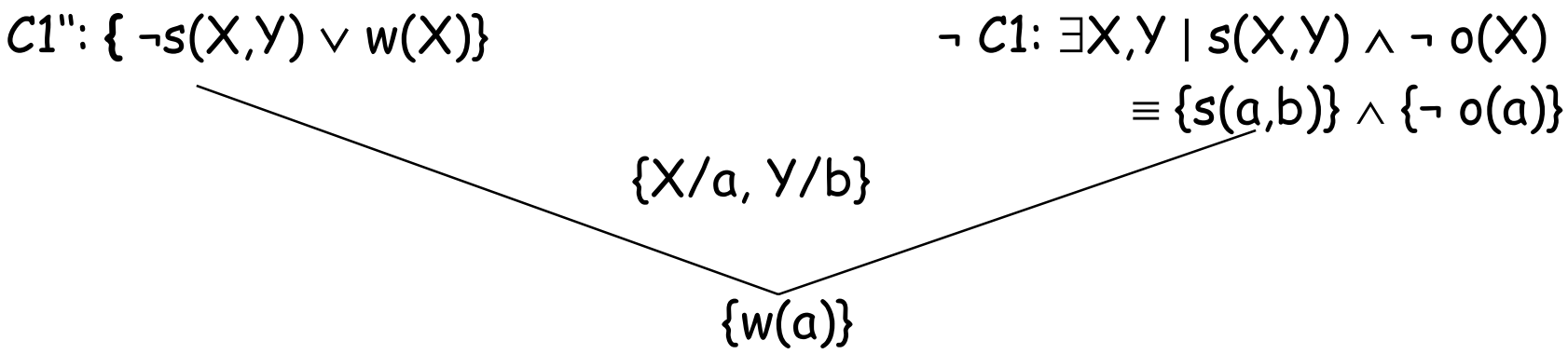
T, C1 → C1“?



ist widersprüchlich!

C1 ist allgemeiner als C1" unter der Theorie T.

T, C1“ → C1?

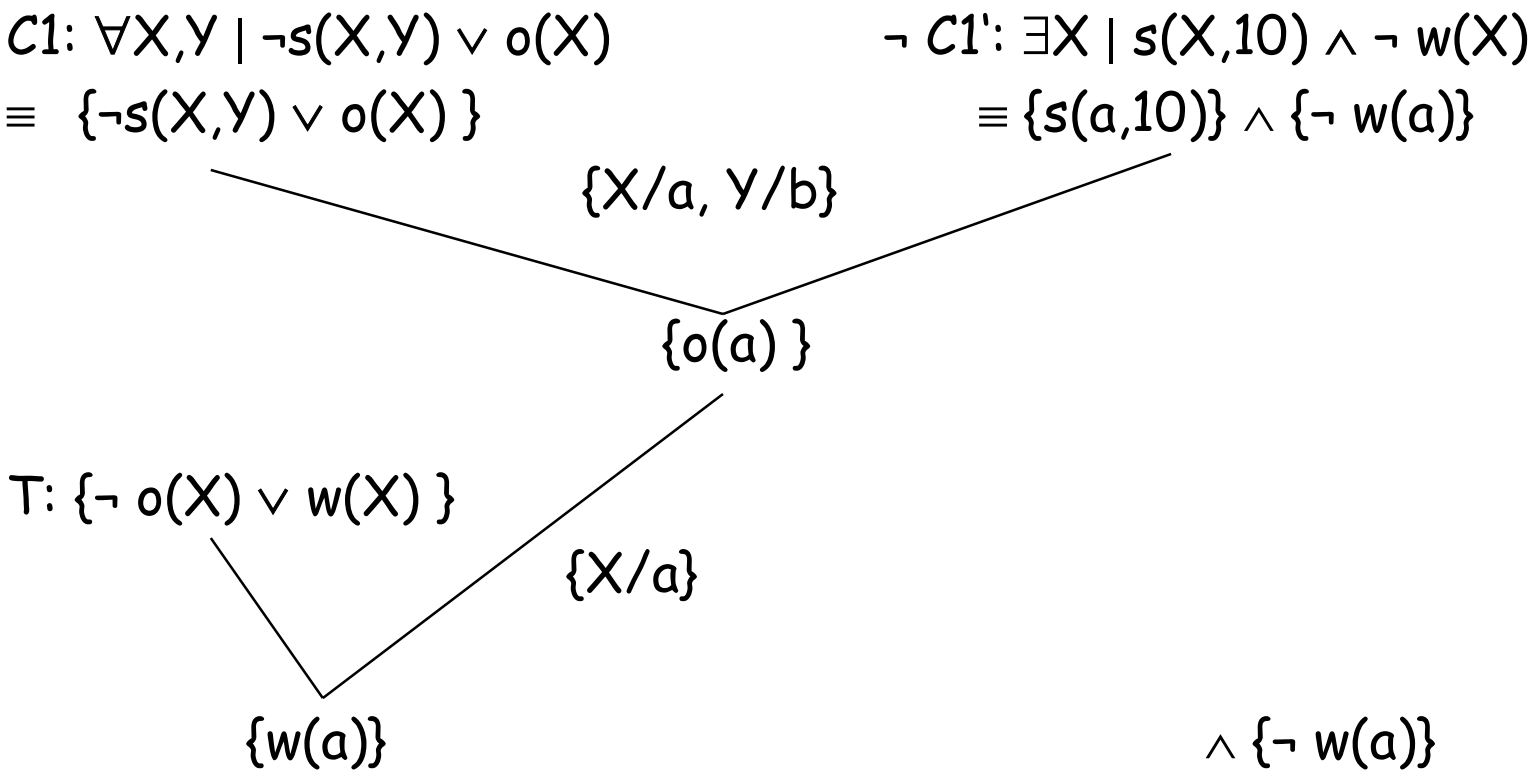


T: {¬o(X) ∨ w(X)}

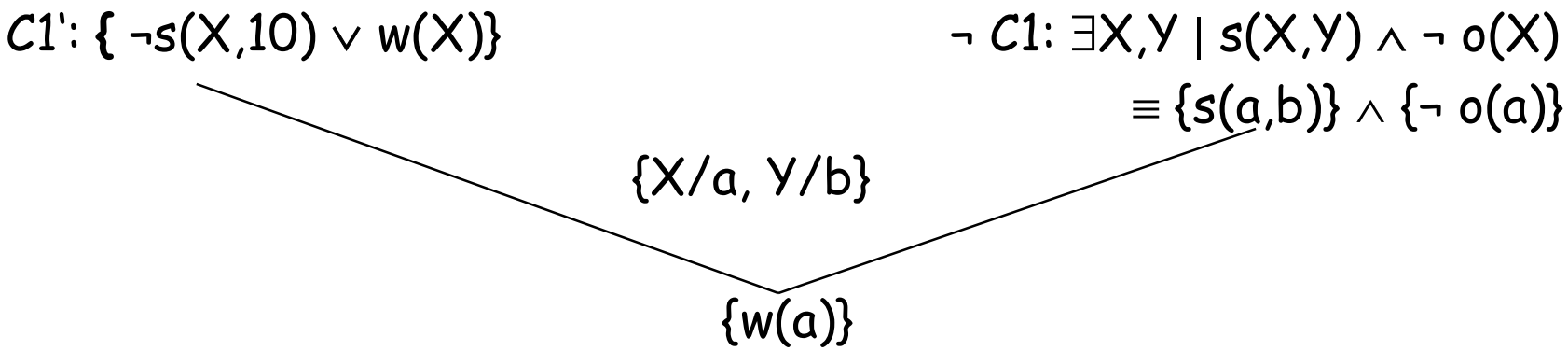
Kein Widerspruch!

C1" ist nicht genereller als C1 .

T, C1 → C1‘?


C1: ∀X,Y | ¬s(X,Y) ∨ o(X)
≡ {¬s(X,Y) ∨ o(X) }
¬ C1': ∃X | s(X,10) ∧ ¬ w(X)
≡ {s(a,10)} ∧ {¬ w(a)}
{X/a, Y/b}
{o(a) }
T: {¬ o(X) ∨ w(X) }
{X/a}
{w(a)}
∧ {¬ w(a)}
ist widersprüchlich!
C1 ist allgemeiner als C1' unter der Theorie T.

T, C1‘ → C1?



T: {¬o(X) ∨ w(X)}

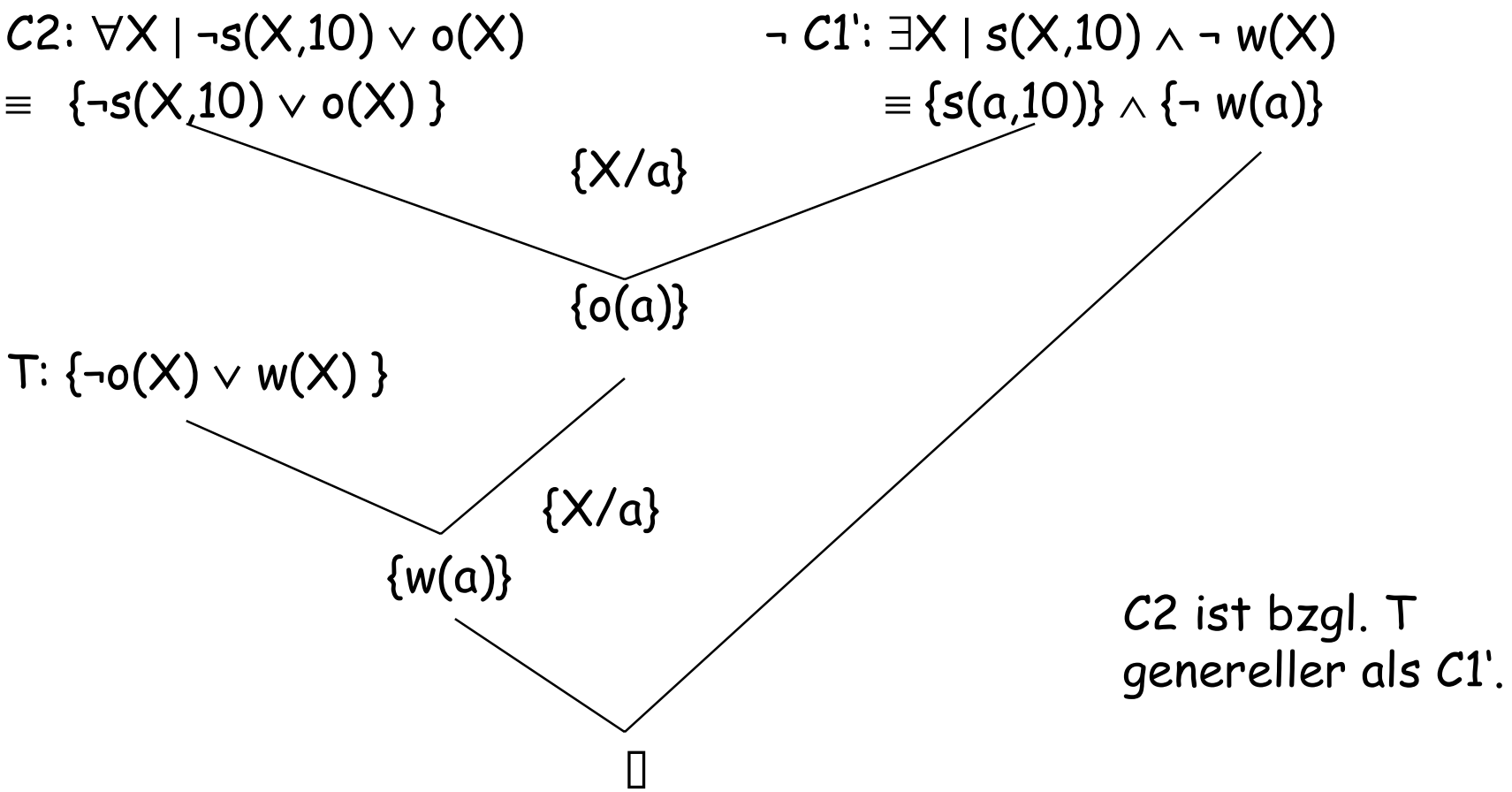
Kein Widerspruch!

C1' ist nicht genereller als C1 .

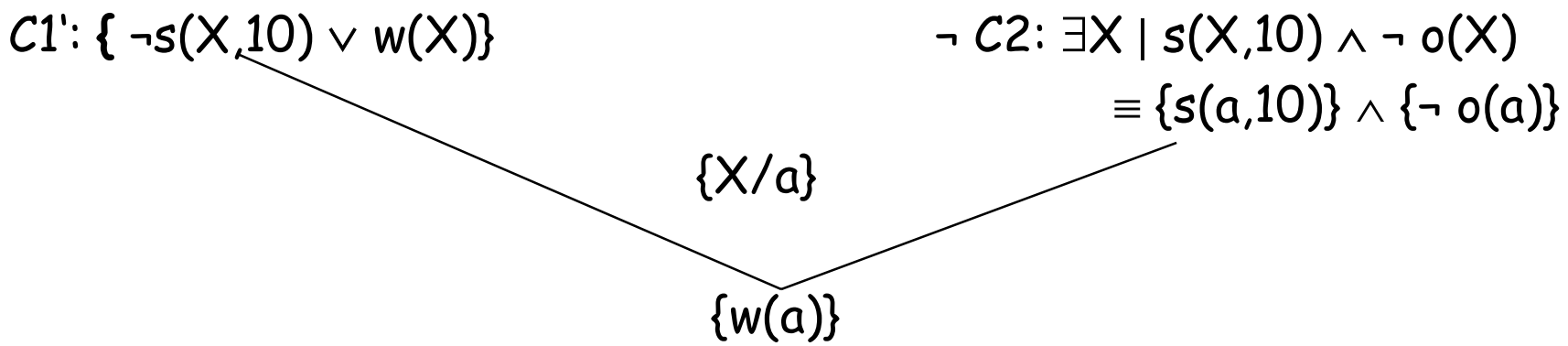
T, C2 → C1“?
T, C1“ → C2?

- Weder ist C2 genereller als C1" noch C1" genereller als C2.
- Die beiden Klauseln sind nicht vergleichbar, werden im Halbverband auf derselben Allgemeinheitsstufe angeordnet.

T, C2 → C1‘?


C2: ∀X | ¬s(X,10) ∨ o(X)
≡ {¬s(X,10) ∨ o(X) }
¬ C1': ∃X | s(X,10) ∧ ¬ w(X)
≡ {s(a,10)} ∧ {¬ w(a)}
{X/a}
{o(a)}
T: {¬o(X) ∨ w(X) }
{X/a}
{w(a)}
□
C2 ist bzgl. T
genereller als C1'.

T, C1‘ → C2?

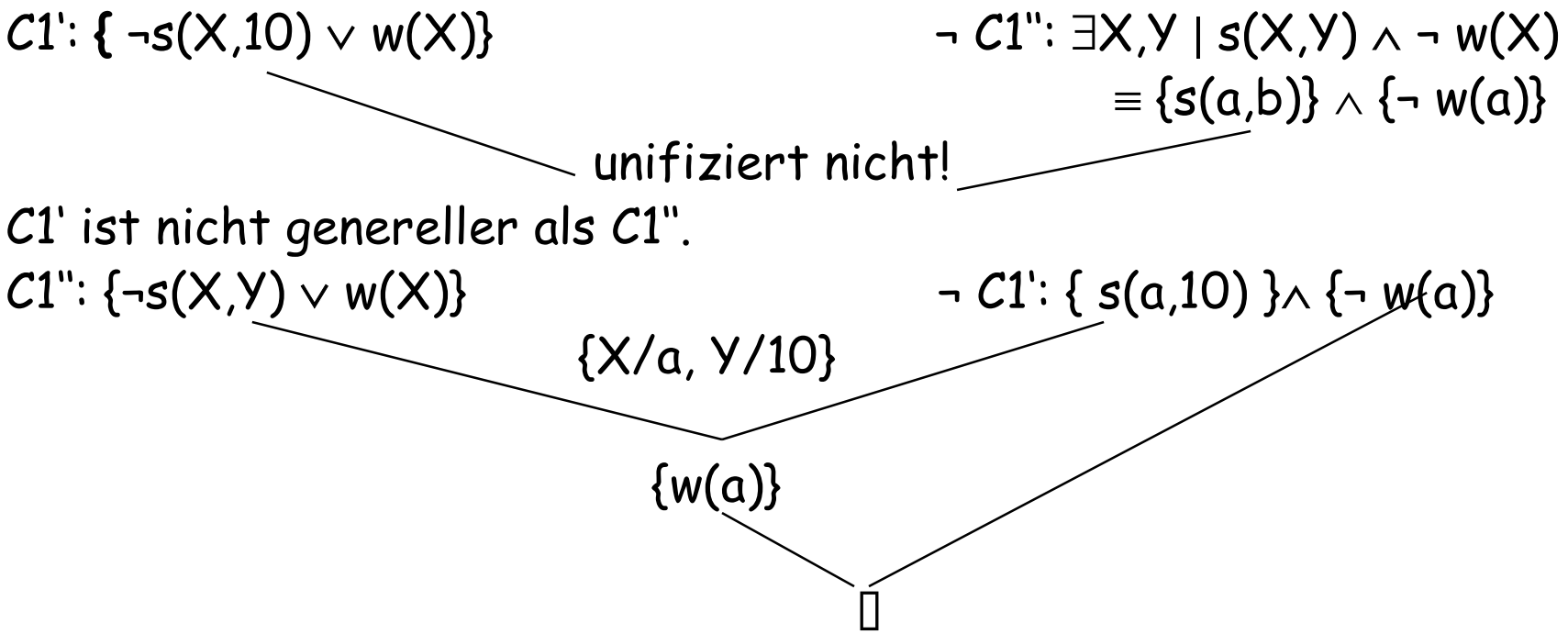


T: {¬o(X) ∨ w(X)}

Kein Widerspruch!

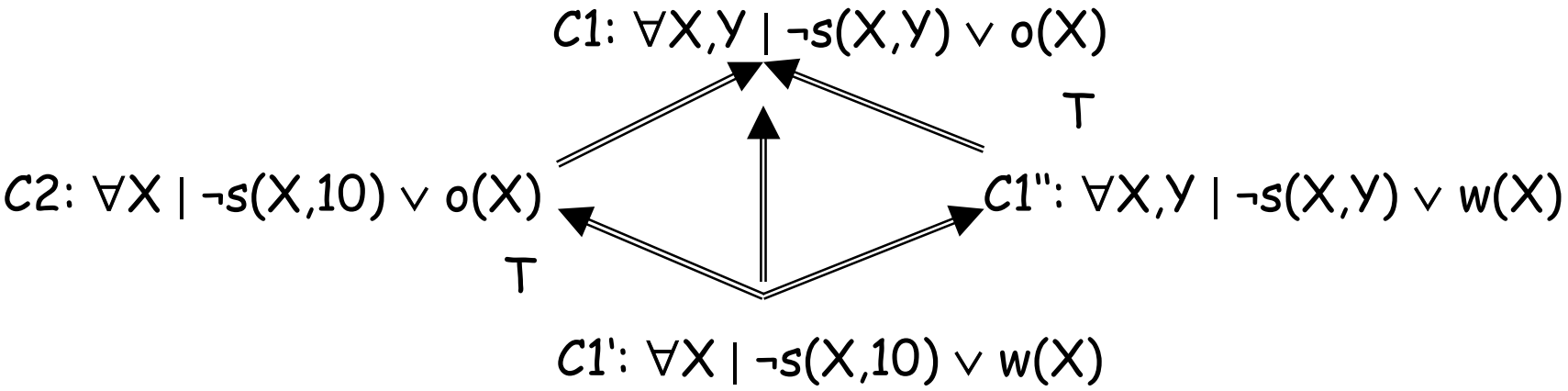
C1' ist auch mit T nicht genereller als C2.

T, C1‘ → C1“?
T, C1“ → C1‘?



C1" ist genereller als C1'.

Halbordnung der gegebenen Klauseln per Implikation



Vorteile

- Wir können Hintergrundwissen (T) einbeziehen.
- Die Klauseln können unterschiedliche Prädikate im Klauselkopf haben.

Nachteile

Der Hypothesenraum ist nicht so strukturiert, daß

bei jedem Generalisierungsschritt

der Ausschnitt der erreichbaren Hypothesen kleiner wird.

Wir können keinen effizienten Generalisierungsoperator konstruieren.

Also ist die Implikation als Generalisierungsbeziehung zum Lernen als Suche nicht geeignet.

Generalisierungsrelation: Subsumtion

Eine Hornklausel C1 ist genereller als eine andere, C2, gdw.

C1 subsumiert C2.

Ein Literal L1 subsumiert ein Literal L2 gdw.

$L1\ \sigma = L2$.

Eine Klausel C1 subsumiert eine andere, C2, gdw.

$C1\ \sigma \subseteq C2$

Eine Hornklausel C1 subsumiert eine andere, C2, gdw.

$C1kopf\ \sigma = C2kopf$ und

$C1körper\ \sigma\ \theta \subseteq C2körper\theta$

Die Subsumtion ist eine korrekte, aber nicht vollständige Ableitung, d.h.

C1 subsumiert C2 ==> C1 impliziert C2, aber nicht umgekehrt.

Beispiel Generalisierung

C1: member (X, [Y| Z]):- member(X,Z).

C2: member (3, [1,2,3]):- member(3, [2,3]), member(3, [3]).

σ : { X/3, Y/1, Z/ [2,3] }

C1kopf σ = member (3, [1 | [2,3]) = C2kopf

C1körper $\sigma\theta$ = member (3, [2, 3]) $\subseteq$ C2körper θ

Nicht alles, was unter der Implikation eine Generalisierung ist,

ist es auch unter der Subsumtion!

T: append ([ ], C, C).

C2: append ([1,2], [3], [1,2,3]).

C1: append([A | B], C, [A | E]) :- append (B, C, E).

kein Hintergrundwissen!

Generalisierung kann nicht länger sein als Generalisiertes!

Generellere Klausel kann länger sein!

S: p(X,X) → h(X,X)
G1: p(X1, X2) → h(X1, X2)
G2: p(X3, X2), p(X2, X1) → h(X1, X2)
G3: p(X4, X3), p(X3, X2), p(X2, X1) → h(X1, X2)

G1σ: p(X,X) → h(X,X) mit σ:{X1/X, X2/X}
Also Gi genereller als S.

Gi genereller als Gi+1.
$G2 \subset G3$

echt genereller

Zwei Klauseln C1 und C2 sind äquivalent, wenn gilt:
C1 subsumiert C2 und C2 subsumiert C1.

Eine Klausel C1 ist echt genereller als eine andere, C2, gdw.
C1 subsumiert C2 und C1 ist nicht äquivalent C2.

Redundanz

Ein Literal L in der Klausel C ist redundant, wenn gilt:

C subsumiert $C \setminus \{L\}$.

Eine Klausel heißt reduziert, wenn sie keine redundanten Literale enthält.

Algorithmus, der eine Klausel C reduziert (Plotkin):

1. Initialisiere D mit C.
2. Finde ein Literal in D und eine Substitution σ, so daß

   $D\sigma \subseteq D \setminus \{L\}$.

   Gelingt dies nicht, STOP.
3. D:= $D\sigma$, gehe zu 2.

Die reduzierte Form einer Klausel ist eindeutig.

Beispiel Redundanz

C1: member(X, [Y | Z]):- member(X, Z), member (X, U).

σ ={U/Z}

C1 $\sigma \subseteq$ C1 \ { member (X, U) }

C1 σ : member(X, [Y | Z]):- member(X, Z) = C1 \ { member (X, U)}

Was wissen Sie jetzt?

- Allgemeinheitsrelation im Versionenraum:
  - Attributwerte sind nach Allgemeinheit angeordnet (vorgegeben) Form > rund, Form > eckig, rund > Kreis, rund > Ellipse...
  - Werte eines Attributs werden mit Konstanten verglichen (covers)
- Allgemeinheitsrelation in der Prädikatenlogik:
  - Formeln werden miteinander verglichen
  - Eine Formel deckt eine andere ab (covers), wenn sie sie impliziert oder subsumiert.
  - Die Implikation (Deduktion) ist in der Prädikatenlogik nur semi-entscheidbar. Die Subsumtion ist korrekt, aber nicht vollständig und daher entscheidbar.
  - Die Substitution einer Variablen durch eine Konstante spezialisiert.

Was kommt jetzt?

- Bottom-up Suche im Hypothesenraum von nach Allgemeinheit geordneten Klauseln.
- Schrittweise Generalisierung von zwei Klauseln:
  least general generalization (lgg)
- Als Vorverarbeitung das Hintergrundwissen einberechnen

Least General Generalization Plotkin

LGG (C1, C2):

Für alle Paare von Literalen L1i $\subseteq$ C1, L2i $\subseteq$ C2, suche die mit gleichem Prädikatensymbol und gleichem Vorzeichen heraus --

bilde LGG (L1i , L2i)

Die Generalisierung von C1 und C2 ist die Vereinigung aller generalisierten Literale.

Aus dieser Generalisierung werden alle redundanten Literale entfernt.

Generalisierung von Literalen Anti-Unifikation

zwei Literale mit demselben Prädikatsymbol und Vorzeichen

p(s1, ..., sn) p (t1, ..., tn) von links nach rechts durchgehen

LGG (si, ti) = X, falls si, ti konstante Terme oder Variablen ungleich X sind;

LGG (f(s1, ..., sn), f(t1, ..., tn)) = f (LGG(s1, t1), ..., LGG(sn, tn))

LGG (f(s1, ..., sn), g(t1, ..., tm)) = X

Beispiel

L1: unterhalt (ulf, maria, alimente(ulf, 1000))

L2: unterhalt(udo, marion, alimente(udo,300))

LGG(L1, L2): unterhalt (X, Y, alimente(X, V))

wobei es für jede andere Generalisierung G(L1, L2) eine Substitution gibt, so daß $G(L1, L2)\ \sigma = LGG(L1, L2)$

LGG (C1, C2)

C1: member(2, [1,2]) :- member (2, [2]).

C2: member (c, [a, b, c]):- member(c, [b,c]), member (c, [c]).

alle Paare:

[¬ m(2, [2]), ¬ m(2, [2]), ¬m(2, [2]), m(2, [1,2]), m(2, [1,2]), m(2, [1,2])]

[¬ m(c, [b,c]), ¬ m((c, [c]), m(c, [a, b, c]), –m(c, [b,c]), –m((c, [c]), m(c, [a, b, c])]

LGG (¬ m(2, [2]), –m(c, [b,c])) = ¬m(A, [C|D])

LGG (¬ m(2,[2]), ¬ m(c, [c])) = ¬ m(A, [A])

LGG (m(2, [1,2]), m(c, [a, b, c])) = m (A, [B, C|D])

LGG (C1, C2) = m (A, [B, C|D]) :- m (A, [C|D]), m (A, [A]).

Bei jedem Literal probieren, ob es weggelassen werden kann, ohne zu generalisieren. Dieser Schritt ist leider NP-schwierig, weil der Subsumtionstest NP-schwierig ist.

Eigenschaften des LGG

kommutativ

$$lgg(e1, e2) = lgg(e2, e1)$$

assoziativ

$$lgg(lgg(e1, e2), e3) = lgg(e1, lgg(e2, e3))$$

idempotent

$$lgg(e, e) = e$$

Das bedeutet auch: reihenfolgeunabhängig

$$lgg(e1, e2, ..., en) = lgg(... lgg(lgg(e1, e2), e3), ... en)$$

und eindeutig.

In einem Verband von Äquivalenzklassen von Klauseln

ist das Supremum zweier Klauseln ihr LGG.

Aufwand

Die gute Nachricht:

Die Länge des LGG ist linear in der Anzahl der Selektionen.

Der Aufwand der Generalisierung ist linear in der Anzahl der Selektionen.

Die schlechte Nachricht:

Hat die längste Klausel in den Beispielen m Literale und

gibt es n Klauseln als positive Beispiele,

dann gibt es höchstens m^n Selektionen.

Es werden also exponentiell viele Selektionen jeweils in linearer Zeit bearbeitet.

Und dann kommt die Reduktion, die für jedes Literal noch einmal den aufwendigen Subsumtionstest braucht...

Hintergrundwissen

LE: Grundfakten LB: Grundfakten

eneu = e :- K wobei K die Konjunktion aller Fakten aus dem Hintergrundwissen ist.

ggf. werden die neuen Beispiele auf verbundene Klauseln beschränkt.

Eine Klausel heißt verbunden, wenn alle ihre Literale verbunden sind. Ein Literal heißt verbunden, wenn mindestens einer seiner Terme verbunden ist. Ein Term heißt verbunden mit der Tiefe 0, wenn er im Kopf der Klausel vorkommt. Ein Term heißt verbunden mit der Tiefe d+1, wenn ein anderer Term desselben Literals mit der Länge d verbunden ist.

oma (X, Z) :- mutter (X,Y), vater (Y, Z). X, Z haben die Tiefe 0, Y die Tiefe 1.

LH: funktionsfreie, nicht rekursive Klauseln

Dann ist die θ-Subsumtion eine korrekte und vollständige Ableitung!

Beispiel

Beispiele:

oma(anna, christof).

oma(anita, cecilie).

Hintergrundwissen:

mutter(anna, bernd). vater (bernd, christof).

mutter (anita, bruno). vater (bruno, cecilie).

Beispiele neu:

oma(anna, christof):- mutter (anna, bernd), vater(bernd,christof).

oma(anita, cecilie):- mutter (anita, bruno), vater (bruno, cecilie).

LGG:

oma (A, C) :- mutter (A, B), vater (B, C).

Und wenn LB Klauseln sind?

funktionsfreie Klauseln:

Jedes Argument eines Prädikats ist entweder einer Variable oder eine Konstante -- Funktionen sind ausgeschlossen.

generative Klauseln (bereichsbeschränkt):

Jede Variable im Klauselkopf kommt auch im Körper vor.

oma (X,Z) :- mutter (X,Y), vater (Y, Z).

Wenn man alle Variablen im Hintergrundwissen durch die in den Beispielen vorkommenden Konstanten ersetzt, so wird das Hintergrundwissen variablenfrei.

Wenn LB auf funktionsfreie, generative Klauseln beschränkt ist, so kann man durch einen (tiefenbeschränkten) Ableitungsprozeß ebenfalls variablenfreies Hintergrundwissen herstellen.

Und dann kann man das Hintergrundwissen in die Beispiele hineinrechnen und den LGG bilden.

Beispiel

Beispiele:

oma(anna, christof).

oma(anita, cecilie).

Hintergrundwissen:

vater(Y,Z) :- kind (Z, Y), mann (Y).

kind(christof, bernd). mann(bernd).

kind(cecilie, bruno). mann (bruno).

Beispiele neu:

oma(anna, christof):- mutter (anna, bernd), vater(bernd,christof).

oma(anita, cecilie) :- mutter (anita, bruno), vater (bruno, cecilie).

Saturierung

Rouveirol

Sei C1 die Klausel H1 :- B1 und

C2 die Klausel H2:- B2, wobei B2 θ-subsumiert B1.

Dann ist die elementare Saturierung von C1 durch C2

D: H1:- B1, H2h

C1: oma(anna, christof):- mutter(anna, bernd), kind (christof, bernd), mann (bernd).

C2: vater(Y,Z) :- kind (Z, Y), mann (Y).

D: oma(anna, christof):- mutter (anna, bernd), kind (christof,bernd), mann(bernd), vater(bernd,christof).

Und jetzt ein bißchen Theorie

- Welche Hypothesenräume erlauben polynomielles Lernen?
- Wichtigster Punkt: im Lernen (Induktion) ist die Deduktion versteckt!
- Wie müssen wir die Prädikatenlogik beschränken, so dass die Deduktion schnell ist?
- Können wir die schwierigsten Sprachen LH feststellen, die gerade noch das Lernen in polynomieller Zeit erlauben?

Sprachbeschränkungen

deterministische Klauseln (bzgl. des Hintergrundwissens):

jede Variable in jedem Literal hat eine eindeutige Substitution durch das Hintergrundwissen.

Klausel: **oma(anna, Z):- mutter(anna, X), vater(X, Z).**

Hintergrundwissen:

mutter(anna, bernd). vater (bernd, christof).

hier: nur {X/ bernd, Z/christof}

Aber wenn Hintergundwissen:

mutter(anna, bernd). vater(bernd, christof). vater (bernd, christiane).

dann ist die Klausel indeterministisch, weil es zwei Substitutionen für Z gibt.

ij-deterministisch

Ein Term aus dem Klauselkopf K ist mit einer Kette 0 deterministisch verbunden.

Für den Klauselkörper nehmen wir an, daß die Literale nach Verbundenheit mit dem Klauselkopf geordnet sind:

{ ¬ L1, ..., ¬ Lm, ¬ Lm+1, ..., ¬ Ln }

Ein Term t aus Lm+1 ist genau dann durch eine deterministische Kette der Länge d+1 verbunden, wenn

alle Terme im Klauselkopf und in {¬ L1, ..., ¬ Lm } verbunden sind durch deterministische Ketten, die höchstens d lang sind,

es für jede Substitution θ, die K mit einem Beispiel und die ersten Literale mit dem Hintergrundwissen unifiziert (d.h. $K\theta \in E+$ und $\{ \{L1\}, ..., \{\neg Lm\}\}\theta \subseteq B$) genau eine eindeutige Substitution σ gibt, so daß $Lm+1\ \theta\sigma \in B$.

Die minimale Länge der deterministisch verbindenden Ketten ist die deterministische Tiefe eines Terms.

Eine Klausel mit maximaler deterministischer Tiefe i und maximaler Stelligkeit j heißt ij-deterministisch.

Beispiel

oma(X, Z) :- mutter (Y, Z) , mutter (X, Y)

oma(X, Z) :- mutter (X, Y), elternteil (Y, Z)

12-deterministisch

tante(X1, Z) :- geschwister (X1, Liste),
member (X2, Liste),
elternteil (X2, Z).

indeterministisch, insofern eine Mutter mehrere Kinder haben kann und ein Kind 2 Elternteile hat.

indeterministisch, insofern als die Liste mehr als ein Element enthalten kann und X2 nicht im Kopf gebunden ist.

tante(X, Z) :- mutter (Y, Z), schwester (X, Y)

tante(X, Z) :- vater (Y, Z), schwester (X, Y)

12-deterministisch

Lernbarkeit

Sei LE Grundfakten mit höchstens t Termen,

LB Grundfakten mit m verschiedenen Prädikaten, die höchstens f Terme enthalten,

LH ij-deterministische Klauseln, wobei i und j festgelegt sind,

so werden Hypothesen mit höchstens

$O((t\ f\ m)^{i^j})$ Literalen gelernt.

Wegen der Tiefenbeschränkung ist die Länge der Klauseln also nicht mehr exponentiell. Ij-deterministische Klauseln sind polynomiell lernbar.

Indeterministische Klauseln sind auch bei Tiefenbeschränkung nicht polynomiell lernbar.

k- lokale Klauseln

Bestehe eine Klausel aus einem deterministischen Teil DDET und einem indeterministischen Teil DNONDET.

D_0 :- D_{DET} , D_{NONDET}

Sei vars eine Funktion, die alle Variablen einer Klausel findet.

Als lokalen Teil LOC einer Klausel bezeichnen wir die Literale aus D_{NONDET}, für die gilt:

(vars (LOC) \ vars ({D_0, D_{DET} })) ∩ vars (D_{NONDET} \LOC) = {}

Minimaler lokaler Teil für eine Konstante k

k-vlokal gdw. k ≥ | vars(LOC) \ vars({ D0, D_{DET} }) | nicht lernbar

k-llokal gdw. k ≥ | LOC | lernbar

Beispiel

Geschwister, deren Mutter Oma ist:

geschwister(X, Z) :- mutter (Y1, X), mutter (Y1, Z),
mutter (Y1, Y2),
elter (Y2, Y3).

vars ({D0, DDET }): X, Y1, Z

vars (DNONDET): Y1, Y2, Y3

LOC 1: mutter (Y1, Y2)

LOC 2: elter (Y2, Y3)

LOC 3: mutter (Y1, Y2),
elter (Y2, Y3)

(vars (LOC) \ vars ({D0, DDET })) ∩ vars (DNONDET \LOC) = {}

LOC1:

({Y1, Y2 } \ {X, Y1, Z}) ∩ vars({mutter(Y1,Y2),elter(Y2,Y3)}\{mutter(Y1,Y2}) = {Y2} ∩ {Y2,Y3} ≠ {}

LOC2:

({Y2, Y3} \ {X, Y1, Z}) ∩ ({Y1, Y2}) = {Y2, Y3} ∩ {Y1,Y2} ≠ {}

LOC3:

({Y1, Y2, Y3} \ {X, Y1, Z}) ∩ ({ }) = {Y2, Y3} ∩ {} = { }

2-vlokal, 2-llokal

Was wissen wir jetzt?

- Die Induktion logischer Programme verfügt über eine ausgereifte Theorie, die festlegt, welche Beschränkungen der Prädikatenlogik hinreichend und notwendig sind, damit gelernt werden kann.
- Ij-deterministische Klauseln sind polynomiell lernbar.
  - Trick: der Beweisbaum wird in der Tiefe und Breite beschränkt.
- Kl-lokale Klauseln sind polynomiell lernbar.
  - Trick: die nichtdeterministischen Klauseln dürfen nur einen endlichen Teil des Problems „infizieren".

ILP für das data mining

- ILP kann Relationen zwischen Attributen (nicht nur zwischen Attributwerten) ausdrücken.
- ILP kann mehrere Objekte mit ihren Eigenschaften ausdrücken, nicht nur Objekte einer Klasse.
  - Wissen über Kunden, Wissen über Produkte, Relationen zwischen Eigenschaften von Kunden, Relationen zwischen Eigenschaften von Produkten, Relationen zwischen Kunden und Produkten...
- Datenbanken haben viele Relationen.
  - Eine Tabelle zu Kunden, eine Tabelle zu Produkten.
- ILP kann direkt auf Datenbanken angewandt werden.

ILP Regellernen

Gegeben:

Hypothesensprache für Mengen von Regeln LH

Beobachtungen E in einer Sprache LE

Hintergrundwissen Tin einer Sprache LT

T, E |-/- □ (Konsistenz)

Ziel:

$C \in LH$, so daß $M+(T, E) \subseteq M(C)$ (Gültigkeit)

$\forall c \in C, \exists e \in E$: T, E-{e} |-/- e,

aber T, E-{e}, c |-- e (Notwendigkeit)

Wenn c gültig und notwendig ist, dann C |-- c (Vollständigkeit)

Es gibt keine echte Teilmenge von C,

die gültig und vollständig ist (Minimalität)

(minimales) Modell

Eine Interpretation bildet eine Menge von Formeln
auf die Wahrheitswerte {0,1} ab.

Gegeben eine Interpretation I für eine Menge von Formeln F.

I ist ein Modell von F, geschrieben: M (F),
wenn alle Formeln von F in I wahr sind.

Wenn es keine Interpretation I' gibt, mit $I' \subset I$ und I' ist ein Modell von F,

ist I ein minimales Modell für F, geschrieben: M+(F).

RDT/db

Rule Discovery Tool (Kietz, Wrobel 1992), RDT/db (Morik, Brockhausen 1996)

Regelschemata in Allgemeinheitsordnung

Benutzer geben an, welche Form von Regeln sie interessant finden

$$m1(C,P,Q): P(X,C) \rightarrow Q(X)$$

$$m2(P1,P2,Q): P1(X,Y) \,\&\, P2(X) \rightarrow Q(Y)$$

$$m3(C,P1,P2,Q): P1(X,Y) \,\&\, P2(X,C) \rightarrow Q(Y)$$

Regelschemata werden mit allen Prädikaten instantiiert, C mit allen vorhandenen Werten.

Jede Instanz ist eine Regel. Ihre Gültigkeit wird mithilfe von SQL-Anfragen über der Datenbank getestet.

Akzeptanzkriterien

Benutzer geben Kriterium für die Akzeptanz einer Hypothese an.

Beispiele:

$$\frac{pos(H)}{concl(H)} - \frac{neg(H)}{concl(H)} > 0.8$$

a posteriori **a priori** **bei 2 Klassen**

$$\frac{pos(H)}{pos(H)+ neg(H)} > \frac{concl(H)}{concl(H)+ negconcl(H)}$$

pos(H): Prämisse und Konklusion kommen gemeinsam vor neg(H): Prämisse und Negation der Konklusion kommen gemeinsam vor	concl(H): Konklusion kommt vor negconcl(H): Konklusions-prädikat ist anwendbar, kommt aber nicht vor

Abbildungen der Datenbank auf Prädikate

Jede Datenbanktabelle ist ein Prädikat, ihre Attribute sind die Argumente.

customer (Person, Income, Customer) married (Husband, Wife)

Jedes Datenbank-Attribut wird ein Prädikat, der Schlüssel und der Attributwert die Argumente.

income (Person, Income) .., wife (Husband, Wife)

Jeder Wert eines Datenbankattributs wird Prädikat, Schlüssel das Argument

customer (Person), inc_10_20 (Person)

customer			**married**	
Person	Income	Customer	Husband	Wife

Größe des Hypothesenraums

$$r \left(p \; i^c\right)^k$$

r Anzahl der Regelschemata

p Anzahl der Prädikate

k max. Anzahl von Literalen in einem Regelschema

Bei Konstantenlernen:

c Anzahl der zu lernenden Konstanten

i max. Anzahl von Werten für eine Konstante

Je nach gewählter Abbildung ist die Größe des Hypothesenraums sehr unterschiedlich.

Hypothesentest

- Für eine Hypothese (Regel) werden SQL-Anfragen an die Datenbank gestellt:
  - Eine Anfrage holt die Unterstützung (support) für die Regel.
  - Eine Anfrage holt die Evidenz gegen die Regel.
- Die Abbildung von Prädikaten auf Datenbankrelationen ist in einer Datei gespeichert (Meta-Daten).
- Die Ergebnisse der SQL-Anfragen werden gemäß des Akzeptanzkriteriums ausgewertet.

Beispiel

vehicles			regions	
ID	produced	licensed	place	region

regions(X1, europe), licensed(Y, X1), produced(Y, X2)
→ regions(X2, europe)

```
SELECT COUNT (*)
        FROM vehicles veh1, vehicles veh2,
                regions reg1, regions reg2
        WHERE reg1.place = veh1.produced and
                veh1.ID = veh2.ID   and
               reg2.place = veh2.produced and
               reg1.region = ‘europe‘ and
               reg2.region = ‘europe‘ ;
```

pos(h)

```
SELECT COUNT (*)
        FROM vehicles veh1, vehicles veh2,
                regions reg1, regions reg2
        WHERE reg1.place = veh1.produced and
                veh1.ID = veh2.ID   and
               reg2.place = veh2.produced and
               not reg1.region = ‘europe‘ and
               reg2.region = ‘europe‘ ;
```

neg(h)

SQL-Generator

- Induktive Datenbankanfragen
- Metadaten-getriebene Code-erzeugung:
  FROM
  - Welche Tabellen sind betroffen?
  - Wieviele unterschiedliche Objekte kommen vor?

  WHERE
  - Relationen zwischen Attributen: Gleichheitsbedingungen

Beispiel – cont‘d

Metadaten

DB:

key(vehicle, [ID])

key(regions, [place])

RDT—DB:

map1(regions, regions)

%arg**n**_Prädikat, Tabelle.Attr**n**

map2(produced, vehicles.produced)

map2(licensed, vehicles.licensed)

%arg1_Prädikat, Tabelle.key

%arg2_Prädikat, Tabelle.Attr

arg1_regions, regions.place

arg2_regions, regions.region

arg1_produced, vehicles.key

arg1_licensed, vehicles.key

arg2_produced, vehicles.produced

arg2_licensed, vehicles.licensed

FROM erzeugen

Hypothese analysieren

Welche Tabellen sind betroffen?

Wieviele Literale je Tablle?
2 Literale regions
2 Literale vehicles

SQL-Anfrage aufbauen

```
SELECT COUNT (*)
FROM vehicles ...., regions....
```

```
SELECT COUNT (*)
FROM vehicles veh1, vehicles veh2,
     regions reg1, regions reg2
```

WHERE erzeugen

Welche Variablen kommen vor?

X1: Literal1: arg1_regions,
Literal2: arg2_licensed

X2: Literal3: arg1_regions,
Literal4: arg2_produced

Y: Literal2: arg1_licensed,
Literal3: arg1_produced

Welche Konstanten?

' europe': Literal1: arg2_regions
Literal4: arg2_regions

WHERE
reg1.place = veh1.licensed and

reg2.place = veh2.produced and

veh1.ID = veh2.ID and

reg1.region = ‘europe‘ and
reg2.region = ‘europe‘ ;

Anwendungen

Daimler Chrysler AG

2,6 Gigabyte Datenbank: alle Fahrzeuge und ihre Garantiefälle

40 Tabellen mit je bis zu 40 Attributen

1. Unterschiedlich großer Hypothesenraum:
   4913 ≤ Größe ≥ 2,8 E 41
2. Unterschiedlich großer Datenbankauszug:
   max. 23 Tabellen
   max. 750 000 Tupel
3. Verwendung von Hintergrundwissen
4. Vergleich mit anderen ILP-Verfahren

Verwendung von Hintergrundwissen

Gegeben:

elektronisch verfügbares Werkstattbuch für PKW mit allen Fahrzeugteilen

Finde:

Gruppen von Teilen, die räumlich, funktional oder bzgl. ihrer Schadensart zusammenhängen

Umformen der Datei in einstellige Fakten, wobei das Fahrzeugteil als Argument, der Zusammenhang als Prädikat ausgedrückt ist

Verwendung von STT (Kietz 1989) zum Finden einer Subsumtionshierachie

Klassen von Fahrzeugteilen der Datenbank hinzufügen

Lernergebnisse

rel_niveauregulierung (FIN) → beanstandet (FIN)

motor_e-typ (FIN, Typ) & mobr_cyl (Typ, 6) → beanstandet (FIN)

rel_garantie (X1, X2, RB, X4, X5, X6, X7, Konfig, Teil) &
rel_garantie_benzman (X1, X2, RB, X4, X5, X6, X7, FIN) &
rel_motor_typ (FIN, 206) & italien(RB) → class_419 (Konfig, Teil)

rel_garantie (X1, X2, X3, X4, X5, X6, X7, Konfig, Teil) & class_35 (Konfig, Teil)
→ kostbean_0_500 (X1, X2, X3, X4, X5, X6, X7)

Lernende Roboter

Bisherige Ansätze:

Neuronale Netze zum Lernen eines kollisionsfreien Pfades oder

zur automatischen Synthese von Fuzzy Controllern.

Dies sind sinnvolle Lernläufe auf der untersten Ebene eines Roboters .

Erklärungsbasiertes Lernen zur Optimierung der Planung.

Dies sind sinnvolle Lernläufe auf der höchsten Ebene eines Roboters.

Wie aber kann die Reflex-Ebene mit höheren Ebenen kombiniert werden?

Können Wahrnehmung und Handlung in Merkmalen (Begriffen) aller Ebenen integriert werden?

Zeitbehandlung durch Relationen zwischen Zeitpunkten.

Abstrakte Wahrnehmungsmerkmale, die Sensordaten komprimieren.

Anwendbarkeit, — **standing (Trc, T1, T2, *in_front_of_door* , PDir, small_side, PrevP) &**

Handlung mit Wahrnehmung, — **parallel_moving(Trc, T2, T3, Speed, PDir, *through_door*, right_and left) &**

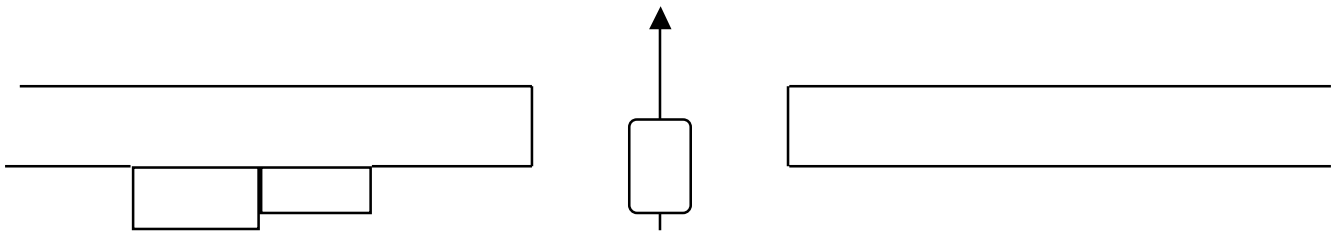
Verifikation — **standing (Trc, T3, T4, *in_front_of_door* , back, small_side, *through_door*)**

--> move_through_door (Trc, T1, T4)

through_door (...) --> parallel_moving (...)

Basismuster --> through_door (...)

Sensor, Bewegung --> incr_peak (...)



Sie wissen jetzt schon viel

Lernen als Suche bedeutet:

- alle Hypothesen werden entlang ihrer Allgemeinheit angeordnet;
- von den Beispielen ausgehend wird generalisiert und/oder
  von der allgemeinsten Hypothese ausgehend wird spezialisiert,
  bis die Bedingungen der Lernaufgabe erfüllt sind.

Man muß also

- den Hypothesenraum ordnen
- aufgrund der Beispiele von einer Hypothese zur nächsten kommen

Also:

Wahl der richtigen Sprache LH:

- der Hypothesenraum muß die richtige Hypothese enthalten,
  soll dabei aber so klein wie möglich sein!

Erreichbarkeit der Hypothesen:

- Generalisierungs-/Spezialisierungsschritte müssen minimal sein, um nicht an der richtigen Hypothese "vorbeizulaufen";
- vom Ausgangspunkt aus muß die richtige Hypothese durch eine Kette von minimalen Schritten erreichbar sein.

θ-Subsumtion als Ordnung

θ-Subsumtion induziert eine Halbordnung, d.h.

reflexiv: $C \geq_\theta C$

transitiv: $C \geq_\theta D, D \geq_\theta E \models C \geq_\theta E$ aber

nicht antisymmetrisch: $C \geq_\theta D, D \geq_\theta C \not\models C = D$, sondern nur

C äquivalent D.

ILP für KDD

- Relationen der Datenbank können als Prädikate aufgefasst werden.
- Regeln in (eingeschränkter) Prädikatenlogik sind Relationen zwischen Relationen.
- ILP kann also Relationen zwischen verschiedenen Tabellen lernen.
- Metadaten-gesteuerte SQL-Generierung kann den Hypothesentest direkt über der Datenbank laufen lassen.
- Vision: induktive Datenbanken