Data Mining und Maschinelles Lernen

Wintersemester 2015/2016

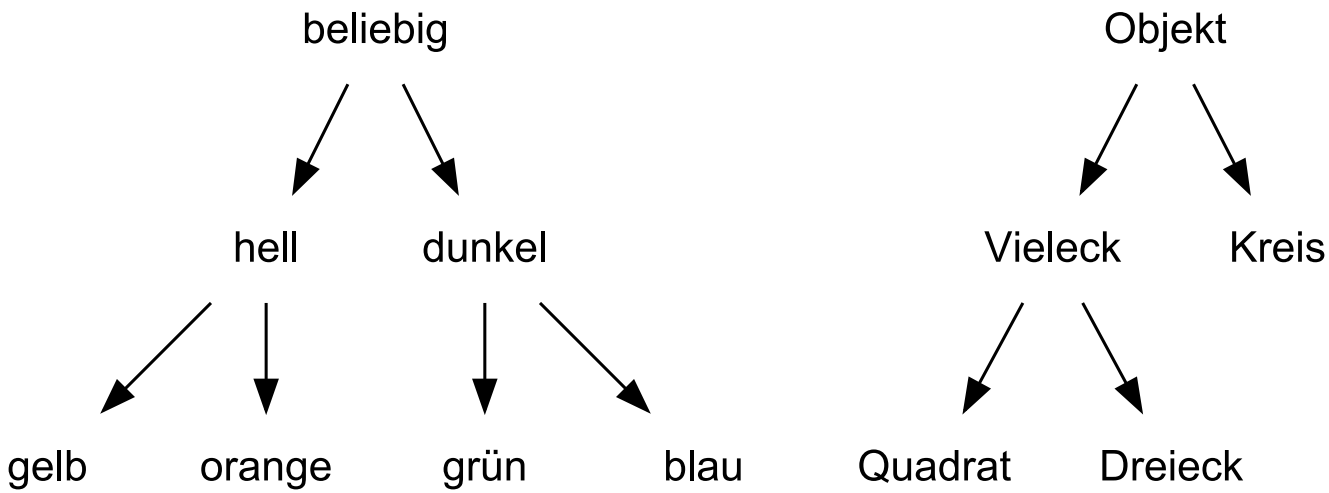
Lösungsvorschlag für das 3. Übungsblatt

TECHNISCHE UNIVERSITÄT DARMSTADT

Aufgabe 1: Version Space, Generalisierung und Spezialisierung (1)

TECHNISCHE UNIVERSITÄT DARMSTADT

Gegeben sei folgende Hierarchie von Begriffen:



Beobachtet werden Objekte, die durch Begriffspaare charakterisiert werden, die man an der untersten Ebene dieser Taxonomien finden kann (also z.B. „blaues Dreieck“). Konzepte können auch höherliegende Begriffe verwenden (also z.B. „dunkles Vieleck“). Überlegen Sie sich eine Generalisierungsvorschrift, die diese Taxonomien verwendet.

Aufgabe 1: Version Space, Generalisierung und Spezialisierung (2)

TECHNISCHE UNIVERSITÄT DARMSTADT

a) Wie sieht die minimale Generalisierung der Objekte „blauer Kreis“ und „grünes Dreieck“ aus?
Lösung: Gesucht wurde hier die gemeinsame Generalisierung zweier Objekte. Geht man von „grün“ und „blau“ den kürzesten Weg nach oben, landet man bei „dunkel“. Geht man von „Kreis“ und „Dreieck“ den minimalen Schritt nach oben, landet man bei „Objekt“.
$\Rightarrow$ *„dunkles Objekt“*
b) Wie sehen minimale Spezialisierungen des Konzepts „helles Objekt“ aus, sodaß das Beispiel „oranger Kreis“ nicht mehr abgedeckt wird?
Lösung: Entweder bei „hell“ eine Ebene weiter runter (ohne bei „orange“ zu landen) oder bei „Objekt“ (ohne bei „Kreis“ zu landen):
„gelbes Objekt“ und „helles Vieleck“. „Gelbes Vieleck“ ist keine zulässige Lösung, da diese Spezialisierung nicht minimal ist.

Aufgabe 1: Version Space, Generalisierung und Spezialisierung (3)

TECHNISCHE UNIVERSITÄT DARMSTADT

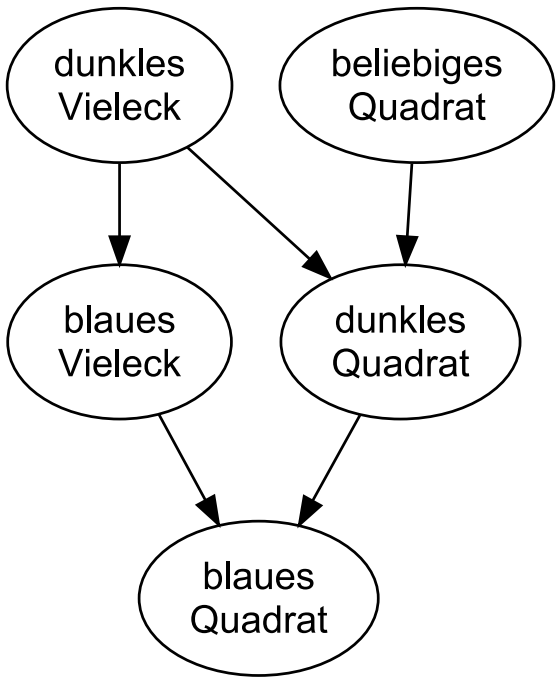
c) Gegeben seien folgende S und G-Sets:

G: { dunkles Vieleck, beliebiges Quadrat }

S: { blaues Quadrat }

Skizzieren Sie den Version Space, der durch diese Mengen definiert wird.

Lösung:



Aufgabe 1: Version Space, Generalisierung und Spezialisierung (4)

TECHNISCHE UNIVERSITÄT DARMSTADT

d) Wie würden Sie mit Hilfe des oben gegebenen Version Spaces die folgenden Beispiele klassifizieren (mit Begründung):

Objekt	Klasse
blaues Quadrat	
blauer Kreis	
blaues Dreieck	

Lösung: Das erste Beispiel ist im S-set enthalten und wird daher als positiv klassifiziert. Das zweite Beispiel ist nicht im G-set enthalten und wird daher als negativ klassifiziert. Beispiel 3 ist ein „dunkles Vieleck“, aber kein „beliebiges Quadrat“. Im S-set ist es nicht enthalten. Daher würde es nicht klassifizierbar sein und ein „?“ erhalten. Wenn Beispiele teilweise im S-Set und im G-Set sind (d.h., von diesen Hypothesen abgedeckt werden), können diese nie eindeutig klassifiziert werden. (Vergleiche *Lernen einzelner Regeln*, Folien *How to Classify these Examples?*)

Zusatzfrage: Wie würde man das Beispiel *blaues Vieleck* klassifizieren?

Aufgabe 1: Version Space, Generalisierung und Spezialisierung (5)

TECHNISCHE UNIVERSITÄT DARMSTADT

e) Gegeben seien wiederum die S- und G-sets aus c). Wie verändern sich die Sets nach Eintreffen des Beispiels (es wird der Candidate Elimination Algorithmus betrachtet):

gelbes Dreieck +

Lösung: Betrachtet wird der Candidate Elimination Algorithmus für das Auftreten eines positiven Beispiels.

G-Set: Hier wird jede Hypothese im G-Set entfernt, die das Beispiel nicht abdeckt. Da „gelbes Dreieck“ $\notin G$ ist, werden beide Hypothesen entfernt. Dann gilt $G = \{\}$.

S-Set: Da zum S-set nur die Hypothesen hinzugefügt werden, für die es eine Hypothese im G-set gibt, die genereller ist, werden keine Hypothesen hinzugefügt (da das G-set leer ist). Außerdem wurde das aktuelle Beispiel nicht abgedeckt. Daher gilt $S = \{\}$.

Wie interpretieren Sie dieses Ergebnis?

Lösung: Mit der Hinzunahme des Beispiels „gelbes Dreieck“ + ist das Konzept mit dem vorliegenden Version Space nicht mehr lernbar. Wenn man sich die Beispiele betrachten würde, die zur Entstehung des Version Spaces geführt haben, und das aktuelle Beispiel hinzunehmen würde, so wäre diese Beispielmenge mit dem Candidate Elimination Algorithmus nicht mehr lernbar.

Aufgabe 2: Version Space und Candidate Elimination Algorithmus (1)

TECHNISCHE UNIVERSITÄT DARMSTADT

Überlegen Sie sich eine geeignete Sprache, um den Candidate Elimination Algorithmus um die Behandlung von numerischen Daten zu erweitern.
a) Wie sieht eine passende Generalisierungs/Spezialisierungsvorschrift aus?
Lösung: Wir verwenden als Hypothesensprache Intervalle. Bei Generalisierungen sind diese abgeschlossen und bei Spezialisierungen entsprechend offen. Demnach sind die Hypothesen des G-Sets offen beziehungsweise die des S-Sets abgeschlossen.
Spezialisierungen erfolgen durch Einschränkung beziehungsweise Aufteilung von Intervallen.
Generalisierungen entstehen durch Erweiterung von Intervallen.

Aufgabe 2: Version Space und Candidate Elimination Algorithmus (2)

TECHNISCHE UNIVERSITÄT DARMSTADT

b) Berechnen Sie den Version Space für folgende Beispiele:

Nr.	**A1**	**A2**	**Klasse**
1	0.5	1.5	−
2	1.1	1.2	+
3	1.8	1.0	+
4	1.5	2.1	−
5	2.1	1.2	−

Lösung: Wir verwenden den wie oben beschrieben modifizierten Candidate Elimination Algorithmus.

1. *Wir beginnen mit:*
   - $G_0 = \{<?, ?>\}$
   - $S_0 = \{<\emptyset, \emptyset>\}$
2. *Als nächsten erhalten wir folgendes Beispiel* (0.5, 1.5, −):
   - $S_1 = S_0$, *da keine Hypothese in* S_0 *das Beispiel abdeckt*
   - $\{<?, ?>\}$ *deckt das Beispiel ab, deshalb muß diese Hypothese entfernt und minimal spezialisiert werden. Alle möglichen minimalen Spezialisierung werden hinzugefügt, da* G_0 *die generellste Hypothese beinhaltet.*
     $\Rightarrow G_1 = \{<(-\infty, 0.5), ?>, <(0.5, \infty), ?>, <?, (-\infty, 1.5)>, <?, (1.5, \infty)>\}$

Aufgabe 2: Version Space und Candidate Elimination Algorithmus (3)

TECHNISCHE UNIVERSITÄT DARMSTADT

3. *Nächstes Beispiel* (1.1, 1.2, +):
   - $S_1 = \{< \emptyset, \emptyset >\}, G_1 = \{< (-\infty, 0.5), ? >, < (0.5, \infty), ? >, <?, (-\infty, 1.5) >, <?, (1.5, \infty) >\}$
   - *Wir entfernen alle Hypothesen aus G_1, die das Beispiel nicht abdecken:* $< (-\infty, 0.5), ? >, <?, (1.5, \infty) > \} \Rightarrow G_2 = \{< (0.5, \infty), ? >, <?, (-\infty, 1.5) >\}$
   - $< \emptyset, \emptyset >$ *aus S_1 deckt das Beispiel nicht ab und muß generalisiert werden:* $\Rightarrow S_2 = \{< [1.1, 1.1], [1.2, 1.2] >\}$ *(alle Hypothesen in G_2 sind genereller)*
4. *Nächstes Beispiel* (1.8, 1.0, +):
   - *Alle Hypothesen aus G_2 decken das Beispiel ab:* $\Rightarrow G_3 = G_2$
   - $< [1.1, 1.1], [1.2, 1.2] >$ *aus S_2 deckt das Beispiel nicht ab und muß generalisiert werden:* $\Rightarrow S_3 = \{< [1.1, 1.8], [1.0, 1.2] >\}$

Aufgabe 2: Version Space und Candidate Elimination Algorithmus (4)

TECHNISCHE UNIVERSITÄT DARMSTADT

5. *Nächstes Beispiel* $(1.5, 2.1, -)$:
   - $S_3 = \{< [1.1, 1.8], [1.0, 1.2] >\}$, $G_3 = \{< (0.5, \infty), ? >, <?, (-\infty, 1.5) >\}$
   - *Keine Hypothese in S_3 deckt das Beispiel ab:*
     $\Rightarrow S_4 = S_3$
   - *Wir spezialisieren alle Hypothesen in G_3, die das Beispiel abdecken:*
     $< (0.5, \infty), ? >$
     Mögliche Spezialisierungen werden hinzugefügt, falls eine Hypothese in S_3 spezifischer ist:
     - $< (0.5, 1.5), ? >$, *keine Hypothese (in S) ist spezifischer*
     - $< (1.5, \infty), ? >$, *keine Hypothese (in S) ist spezifischer*
     - $< (0.5, \infty), (-\infty, 2.1) >, < [1.1, 1.8], [1.0, 1.2] >$ *ist spezifischer*
     - $< (0.5, \infty), (2.1, \infty) >$, *keine Hypothese (in S) ist spezifischer*

     Alle anderen Hypothesen in G_3 (also $<?, (-\infty, 1.5) >$) bleiben unverändert.
     $\Rightarrow G_4 = \{<?, (-\infty, 1.5) >, < (0.5, \infty), (-\infty, 2.1) >\}$

Aufgabe 2: Version Space und Candidate Elimination Algorithmus (5)

TECHNISCHE UNIVERSITÄT DARMSTADT

6. *Nächstes Beispiel* (2.1, 1.2, −):
   - $S_4 = \{< [1.1, 1.8], [1.0, 1.2] >\}$, $G_4 = \{<?, (-\infty, 1.5) >, < (0.5, \infty), (-\infty, 2.1) >\}$
   - *Keine Hypothese in S_4 deckt das Beispiel ab:*
     $\Rightarrow S_5 = S_4$
   - *Wir spezialisieren alle Hypothesen in G_4, die das Beispiel abdecken:*
     $< (0.5, \infty), (-\infty, 2.1) >, <?, (-\infty, 1.5) >$
     Betrachten wir die möglichen Spezialisierungen der ersten Hypothese:
     - $< (0.5, 2.1), (-\infty, 2.1) >, < [1.1, 1.8], [1.0, 1.2] >$ *ist spezifischer*
     - $< (2.1, \infty), (-\infty, 2.1) >$, *keine Hypothese ist spezifischer*
     - $< (0.5, \infty), (-\infty, 1.2) >$, *keine Hypothese ist spezifischer*
     - $< (0.5, \infty), (1.2, 2.1) >$, *keine Hypothese ist spezifischer*

     Die möglichen Spezialisierungen der zweiten Hypothese lauten wie folgt:
     - $< (-\infty, 2.1), (-\infty, 1.5) >, < [1.1, 1.8], [1.0, 1.2] >$ *ist spezifischer*
     - $< (2.1, \infty), (-\infty, 1.5) >$, *keine Hypothese ist spezifischer*
     - $<?, (-\infty, 1.2) >$, *keine Hypothese ist spezifischer*
     - $<?, (1.2, 1.5) >$, *keine Hypothese ist spezifischer*

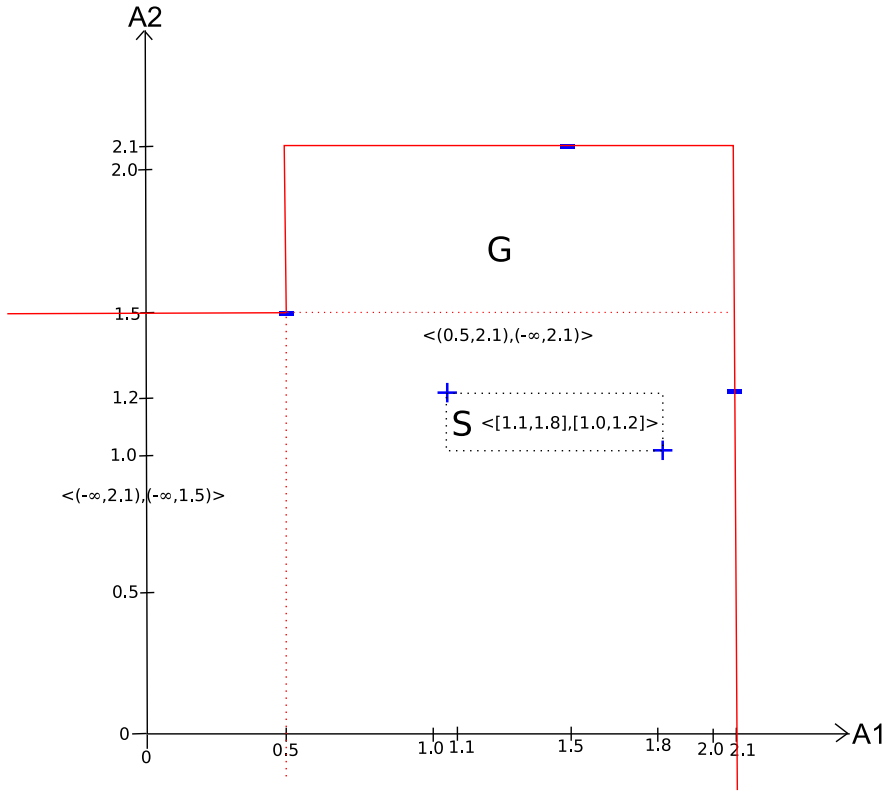
     $\Rightarrow G_5 = \{< (0.5, 2.1), (-\infty, 2.1) >, < (-\infty, 2.1), (-\infty, 1.5) >\}$

Aufgabe 2: Version Space und Candidate Elimination Algorithmus (6)

TECHNISCHE UNIVERSITÄT DARMSTADT

c) Skizzieren Sie das S-Set, das G-Set, und den Version Space im $\mathbb{R}^2$.

Lösung:



Aufgabe 2: Version Space und Candidate Elimination Algorithmus (7)

TECHNISCHE UNIVERSITÄT DARMSTADT

d) Vertauschen Sie die Rolle der positiven und negativen Beispiele. Was passiert dann?
Lösung:

1. *Wir beginnen mit:*
   - $G_0 = \{<?, ?>\}$
   - $S_0 = \{<\emptyset, \emptyset>\}$
2. *Als nächsten erhalten wir folgendes Beispiel* (0.5, 1.5, +):
   - *Alle Hypothesen aus* G_0 *decken das Beispiel ab:*
     $\Rightarrow G_1 = G_0$
   - $<\emptyset, \emptyset>$ *aus* S_0 *deckt das Beispiel nicht ab und muß generalisiert werden:*
     $\Rightarrow S_1 = \{<[0.5, 0.5], [1.5, 1.5]>\}$

Aufgabe 2: Version Space und Candidate Elimination Algorithmus (8)

TECHNISCHE UNIVERSITÄT DARMSTADT

3. *Nächstes Beispiel* $(1.1, 1.2, -)$:
   - *Keine Hypothese in* S_1 *deckt das Beispiel ab:*
     $\Rightarrow S_2 = S_1$
   - $\{<?, ?>\}$ *aus* G_1 *deckt das Beispiel ab und muß spezialisiert werden:*
     - $<(-\infty, 1.1), ?>, <[0.5, 0.5], [1.5, 1.5]>$ *ist spezifischer*
     - $<(1.1, \infty), ?>$, *keine Hypothese ist spezifischer*
     - $<?, (-\infty, 1.2)>$, *keine Hypothese ist spezifischer*
     - $<?, (1.2, \infty)>, <[0.5, 0.5], [1.5, 1.5]>$ *ist spezifischer*

   $\Rightarrow G_2 = \{<(-\infty, 1.1), ?>, <?, (1.2, \infty)>\}$
4. *Nächstes Beispiel* $(1.8, 1.0, -)$:
   - *Keine Hypothese in* S_2 *deckt das Beispiel ab:*
     $\Rightarrow S_3 = S_2$
   - *Keine Hypothese in* G_2 *deckt das Beispiel ab:*
     $\Rightarrow G_3 = G_2$

Aufgabe 2: Version Space und Candidate Elimination Algorithmus (9)

TECHNISCHE UNIVERSITÄT DARMSTADT

5. *Nächstes Beispiel* (1.5, 2.1, +):
   - *Wir entfernen alle Hypothesen aus G_3, die das Beispiel nicht abdecken:*
     $< (-\infty, 1.1), ? >$
     $\Rightarrow G_4 = \{<?, (1.2, \infty) >\}$
   - $< [0.5, 0.5], [1.5, 1.5] >$ *aus S_3 deckt das Beispiel nicht ab und muß generalisiert werden:*
     $\Rightarrow S_3 = \{< [0.5, 1.5], [1.5, 2.1] >\}$
6. *Nächstes Beispiel* (2.1, 1.2, +):
   - *Wir entfernen alle Hypothesen aus G_4, die das Beispiel nicht abdecken:*
     $<?, (1.2, \infty) > \Rightarrow G_5 = \emptyset$
   - $< [0.5, 1.5], [1.5, 2.1] >$ *aus S_4 deckt das Beispiel nicht ab und muß generalisiert werden. Es können aber keine gültigen Generalisierungen gefunden werden, da $G_5 = \emptyset$ (Vergleiche "Lernen einzelner Regeln", Folie "Candidate Elimination Algorithm", Textstelle "some hypothesis...")*
     $\Rightarrow S_5 = \emptyset$

Da sowohl das G-set und das S-set leer sind, folgt, dass diese Beispielmenge nicht mit dem Algorithmus und der gegebenen Hypothesensprache lernbar ist.

Aufgabe 3: Candidate Elimination Algorithmus (1)

TECHNISCHE UNIVERSITÄT DARMSTADT

Gegeben sei ein Datensatz mit drei Attributen:

Haarfarbe:	*blond*, *braun*, *schwarz*
Größe:	*klein*, *groß*
Augenfarbe:	*grün*, *blau*

Der Hypothesenraum besteht aus Disjunktionen (Oder-Verknüpfungen) von maximal einem Wert pro Attribut, einer speziellsten Theorie *false*, die keine Beispiele abdeckt, und einer allgemeinsten Theorie *true*, die alle Beispiele abdeckt.
Zum Beispiel deckt die Hypothese *blond* $\vee$ *blau* alle Personen ab, die entweder blond oder blauäugig sind (in der Datenmenge aus Aufgabe 3b) sind das z.B. die Beispiele 1, 3, 4).
Beachte: Hypothesen wie *blond* $\vee$ *braun*, die mehrere Werte desselben Attributs verwenden, sind nicht im Hypothesenraum.

Aufgabe 3: Candidate Elimination Algorithmus (2)

TECHNISCHE UNIVERSITÄT DARMSTADT

a) Geben Sie in dieser Hypothesensprache alle minimalen Generalisierungen und Spezialisierungen der Hypothese *blond* $\vee$ *blau* an.
Lösung: Durch Hinzufügen einer weiteren disjunktiven Terms kann man nur mehr Beispiele abdecken, daher generalisiert diese Operation. Wegstreichen eines Teilterms spezialisiert dagegen.
Minimale Generalisierungen = { *blond* $\vee$ *blau* $\vee$ *klein*, *blond* $\vee$ *blau* $\vee$ *groß*}
Minimale Spezialisierungen = { *blond*, *blau*}

Aufgabe 3: Candidate Elimination Algorithmus (3)

TECHNISCHE UNIVERSITÄT DARMSTADT

b) Folgende Beispiele treffen in dieser Reihenfolge ein:

1	*braun*	*groß*	*blau*	+
2	*braun*	*klein*	*grün*	−
3	*schwarz*	*klein*	*blau*	−
4	*blond*	*klein*	*grün*	+

Das erste Beispiel kodiert also eine Person, die braune Haare und blaue Augen hat und groß ist.
Führen Sie auf diesen Beispielen den Candidate-Elimination Algorithmus zur Berechnung des Version Spaces durch und geben Sie nach jedem Schritt das *S*-Set und das *G*-Set an.

Aufgabe 3: Candidate Elimination Algorithmus (4)

TECHNISCHE UNIVERSITÄT DARMSTADT

Lösung:

G_0 = { true }
S_0 = { false }
G_1 = { true }
S_1 = { *braun*, *groß*, *blau*} (das sind *drei* Hypothesen!)
G_2 = { *blond* $\vee$ *groß* $\vee$ *blau*, *schwarz* $\vee$ *groß* $\vee$ *blau*}
S_2 = { *groß*, *blau*} (die Hypothese *braun* würde nun inkonsistent sein).
G_3 = { *blond* $\vee$ *groß*} (*schwarz* $\vee$ *groß* $\vee$ *blau* wird inkonsistent und zu *groß* spezialisiert, was spezieller ist als *blond* $\vee$ *groß*)
S_3 = { *groß*} (*blau* wird nun inkonsistent)
G_4 = { *blond* $\vee$ *groß*}
S_4 = { *blond* $\vee$ *groß*} (auf *groß* $\vee$ *grün* können wir nicht generalisieren, da diese Theorie keine Spezialisierung eines Elements in G ist).
Der Version Space konvergiert also zu einer einzigen Lösung.