



**TECHNISCHE  
UNIVERSITÄT  
DRESDEN**

**Seminar Algebra**  
Prof. Dr. B. Ganter

# **Rough Sets – Unexakte Mengen**

Francesco Kriegel

**TU Dresden  
Fakultät Mathematik  
Institut Algebra**

*WS 2008 / 2009*

*5. November 2008*

# Inhaltsverzeichnis

---

<b>Kapitel 1 Approximation</b>	<b>1</b>
1.1 Approximationsraum . . . . .	1
1.2 Allgemeiner Approximationsraum . . . . .	7

---

<b>Kapitel 2 Information</b>	<b>9</b>
2.1 Informationssystem . . . . .	9
2.2 Redukt & Kern . . . . .	10

---

<b>Kapitel 3 Entscheidung</b>	<b>17</b>
3.1 Entscheidungssystem . . . . .	17
3.2 Konsistenz . . . . .	18
3.3 Redukt & Kern . . . . .	20
3.4 Attributselektion . . . . .	22
3.5 Wertemengenreduktion . . . . .	23
3.6 Primentscheidungsregeln . . . . .	28

---

# 1 Approximation

## 1.1 Approximationsraum

### Definition 1.1 (Approximationsraum)

Sei  $U$  eine endliche nicht-leere Menge.  $U$  heißt *Universum* (engl. *universe*) und die Elemente heißen *Objekte* (engl. *object*). Für eine Äquivalenzrelation  $\approx$  auf  $U$ , die eine Ununterscheidbarkeit in  $U$  charakterisiert, heißt

$$\langle U, \approx \rangle$$

auch *Approximationsraum* (engl. *approximation space*). Für  $x \approx y$  sagen wir  $x$  und  $y$  sind *ununterscheidbar* (engl. *indiscernible*). Wir nennen ein Element von  $U/\approx$ , also eine Menge  $[x] := [x]_{\approx}$ , auch *elementare Menge* (engl. *elementary set*) und Vereinigungen von elementaren Mengen heißen *definierbare Mengen* (engl. *definable set*).

### Definition 1.2 (Approximation)

Für die Abbildung

$$\begin{aligned} \cdot_* : \varphi(U) &\rightarrow \varphi(U) \\ X &\mapsto X_* := \{x \in U \mid [x] \subseteq X\} \end{aligned}$$

heißt  $X_*$  *untere Approximation* (engl. *lower approximation*) von  $X$ , für

$$\begin{aligned} \cdot^* : \varphi(U) &\rightarrow \varphi(U) \\ X &\mapsto X^* := \{x \in U \mid [x] \cap X \neq \emptyset\} \end{aligned}$$

nennen wir  $X^*$  *obere Approximation* (engl. *upper approximation*) von  $X$  und das Paar  $(X_*, X^*)$  heißt nun *Approximation* (engl. *approximation*) von  $X$ . Die Menge  $X^\circ := X^* - X_*$  heißt *Rand* (engl. *boundary*) von  $X$ .

Die Abbildung 1.1 veranschaulicht diese Definition.

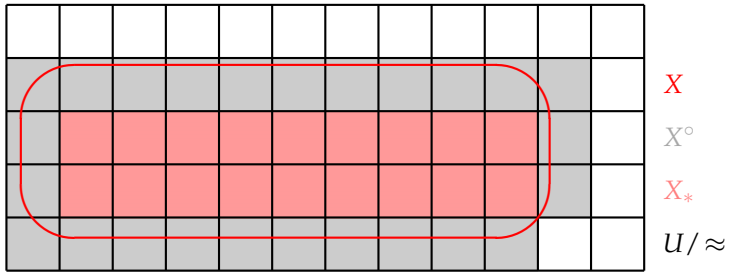


Abbildung 1.1: Approximation einer Menge  $X \subseteq U$

Es gilt

$$X_* = \bigcup_{[x] \subseteq X} [x] = \{x \in U \mid \forall y \approx x: y \in X\},$$

die untere Approximation von  $X$  enthält also diejenigen Objekte, die *sicher* in  $X$  liegen (sicheres Objekt, engl. *certain object*). Die obere Approximation

$$X^* = \bigcup_{[x] \cap X \neq \emptyset} [x] = \{x \in U \mid \exists y \approx x: y \in X\}$$

enthält die Objekte, die *möglicherweise* in  $X$  liegen (mögliches Objekt, engl. *possible object*). Der Rand besteht aus allen Objekten, die weder sicher in  $X$  noch sicher in  $-X$  sind (unentscheidbares Objekt, engl. *undecidable object*).

**Lemma 1.3** Für Mengen  $X, Y \subseteq U$  gelten:

- |   |   |
|---|---|
| (a) $X_* \subseteq X \subseteq X^*$         | (h) $X \subseteq Y \Rightarrow X_* \subseteq Y_*$ |
| (b) $\emptyset_* = \emptyset = \emptyset^*$ | (i) $X \subseteq Y \Rightarrow X^* \subseteq Y^*$ |
| (c) $U_* = U = U^*$                         | (j) $(-X)_* = -X^*$                               |
| (d) $(X \cap Y)_* = X_* \cap Y_*$           | (k) $(-X)^* = -X_*$                               |
| (e) $(X \cup Y)_* \supseteq X_* \cup Y_*$   | (l) $(-X)^\circ = X^\circ$                        |
| (f) $(X \cap Y)^* \subseteq X^* \cap Y^*$   | (m) $(X_*)_* = X_* = (X_*)^*$                     |
| (g) $(X \cup Y)^* = X^* \cup Y^*$           | (n) $(X^*)_* = X^* = (X^*)^*$                     |

**Beweis:**

- 
- (a) Die untere Approximation  $X_*$  ist eine Vereinigung von Teilmengen von  $X$ , muss also selbst eine Teilmenge von  $X$  sein. Die obere Approximation  $X^*$  ist die Vereinigung von allen (elementaren) Mengen, die mit  $X$  einen nicht-leeren Durchschnitt haben, und daher muss sie  $X$  enthalten.
- (b) Da elementare Mengen stets nicht-leer sind, ist die untere Approximation der leeren Menge stets die leere Vereinigung, also die leere Menge. Andererseits gibt es keine Menge, die mit der leeren Menge einen nicht-leeren Durchschnitt hat, also ist auch die obere Approximation der leeren Menge die leere Menge.
- (c) Die Menge der elementaren Menge ist eine Partition von  $U$ , daher ist deren Vereinigung  $U$  die untere Approximation von  $U$ . Die obere Approximation von  $U$  ist ebenso die Vereinigung von allen elementaren Mengen, denn keine von ihnen hat mit  $U$  einen leeren Durchschnitt.
- (d) Die untere Approximation des Schnitts  $X \cap Y$  ist die Vereinigung derjenigen elementaren Mengen, die in beiden Mengen enthalten sind. Dies ist nichts anderes als der Schnitt der beiden Vereinigungen der elementaren Mengen, die in  $X$  bzw. in  $Y$  enthalten sind.
- (e) Möglicherweise gibt es eine elementare Menge, die zwar in der Vereinigung  $X \cup Y$  enthalten ist, jedoch nicht in  $X$  oder  $Y$ . Daher gilt hier nicht die Gleichheit, sondern nur  $\supseteq$ .
- (f) Möglicherweise gibt es eine elementare Menge, die mit dem Schnitt  $X \cap Y$  einen leeren Durchschnitt hat, jedoch nicht mit  $X$  oder  $Y$ . Somit haben wir hier lediglich ein Enthaltensein  $\subseteq$ .
- (g) Die obere Approximation der Vereinigung  $X \cup Y$  ist die Vereinigung aller elementaren Mengen, die Elemente mit  $X$  oder mit  $Y$  gemeinsam haben, also die Vereinigung der Vereinigungen derjenigen elementaren Mengen, die mit  $X$  bzw. mit  $Y$  einen nicht-leeren Durchschnitt haben.
- (h) Ist  $X$  eine Teilmenge von  $Y$ , dann gibt es möglicherweise mehr elementare Mengen, die in  $Y$  enthalten sind, also ist die untere Approximation von  $X$  in der unteren Approximation von  $Y$  enthalten.
- (i) Ist  $X$  eine Teilmenge von  $Y$ , dann gibt es möglicherweise mehr elementare Mengen, die Elemente mit  $Y$  gemeinsam haben, also ist die obere Approximation von  $X$  in der oberen Approximation von  $Y$  enthalten.
- (j) Die untere Approximation von  $-X$  ist die Vereinigung von allen elementaren Mengen, die im Komplement von  $X$  liegen, d.h. die mit  $X$  einen leeren Durchschnitt haben. Das ist gleich dem Komplement der Vereinigung der elementaren Mengen, die mit  $X$  einen nicht-leeren Durchschnitt haben. Formal:  $[x] \subseteq -X \Leftrightarrow \neg([x] \cap X \neq \emptyset)$ .
-

- (k) Die obere Approximation von  $-X$  ist die Vereinigung von den elementaren Mengen, die mit dem Komplement von  $X$  einen nicht-leeren Durchschnitt haben, also das Komplement der Vereinigung derjenigen elementaren Mengen, die im Komplement von  $X$  liegen. Formal:  $[x] \cap -X \neq \emptyset \Leftrightarrow \neg([x] \subseteq X)$ .
- (l) Der Rand des Komplements von  $X$  ist die Vereinigung der elementaren Mengen, die mit  $-X$  einen nicht-leeren Durchschnitt haben, aber nicht in  $-X$  enthalten sind, das sind aber gerade die elementaren Mengen, die mit  $X$  einen nicht-leeren Durchschnitt haben und nicht in  $X$  enthalten sind. Formal:  $[x] \cap -X \neq \emptyset \wedge \neg([x] \subseteq X) \Leftrightarrow \neg([x] \subseteq X) \wedge [x] \cap X \neq \emptyset$ .
- (m) Die untere Approximation ist eine Vereinigung von elementaren Mengen, daher gilt hier Gleichheit, vgl. (a).
- (n) Die obere Approximation ist eine Vereinigung von elementaren Mengen, damit gilt hier Gleichheit, vgl. (a). ■

Nach obigen Lemma ist die Abbildung, die einer Menge ihre untere Approximation zuordnet, ein Kernoperator und dual ist die Abbildung, die einer Menge ihre obere Approximation zuordnet, ein Hüllenoperator. Die Menge der unteren bzw. oberen Approximationen bildet also einen vollständigen Verband. Genauer gilt nun: Die Menge aller Approximationen

$$\{(X_*, X^*) \mid X \subseteq U\} = \{(X, Y) \mid \exists Z \subseteq U: Z_* = X \wedge Z^* = Y\},$$

geordnet durch die komponentenweise Teilmengenrelation

$$(X_*, X^*) \subseteq (Y_*, Y^*) :\Leftrightarrow X_* \subseteq Y_* \wedge X^* \subseteq Y^*,$$

ist ein vollständiger Verband. Für eine Menge von Approximationen  $\{(X_t, Y_t) \mid t \in T\}$  sind das Infimum und das Supremum gegeben durch

$$\bigwedge_{t \in T} (X_t, Y_t) = ((\bigcap_{t \in T} X_t)_*, \bigcap_{t \in T} Y_t)$$

und

$$\bigvee_{t \in T} (X_t, Y_t) = (\bigcup_{t \in T} X_t, (\bigcup_{t \in T} Y_t)^*).$$

Diese Konstruktion führt uns zum Verband der Raumengenabstraktionen, für weiterführende Informationen siehe [4].

**Definition 1.4 (unexakte Menge)**

Eine Menge  $X \subseteq U$  heißt *exakt* (engl. *crisp set*), falls ihr Rand  $X^\circ$  leer ist. Andernfalls heißt  $X$  *unexakt* (engl. *rough set*). Eine unexakte Menge  $X$  heißt

- (i) *unexakt definierbar* (engl. *roughly definable*), falls  $X_* \neq \emptyset$  und  $X^* \neq U$ ,
- (ii) *unten undefinierbar* (engl. *internally indefinable*), falls  $X_* = \emptyset$  und  $X^* \neq U$ ,
- (iii) *oben undefinierbar* (engl. *externally indefinable*), falls  $X_* \neq \emptyset$  und  $X^* = U$ ,
- (iv) *undefinierbar* (engl. *totally indefinable*), falls  $X_* = \emptyset$  und  $X^* = U$ .

Exakte Mengen sind genau die definierbaren Mengen, d.h. Vereinigungen von elementaren Mengen. Für eine unexakt definierbare Menge  $X$  gibt es mindestens ein Element das sicher zu  $X$  gehört und ein Element das sicher zu  $-X$  gehört. Für eine unten undefinierbare Menge  $X$  können wir für keinen Gegenstand entscheiden, ob dieser sicher zu  $X$  gehört, jedoch gibt es einen Gegenstand, der sicher zu  $-X$  gehört. Dual für oben undefinierbare Mengen. Entsprechend ist es für eine undefinierbare Menge  $X$  unmöglich zu entscheiden, ob ein Gegenstand sicher in  $X$  bzw. in  $-X$  liegt. Die reellwertige Abbildung

$$\begin{aligned} \wp(U) &\rightarrow \mathbb{R} \\ \alpha: \quad X &\mapsto \frac{|X_*|}{|X^*|} = 1 - \frac{|X^\circ|}{|X^*|} \end{aligned}$$

misst die *Approximationsgenauigkeit* (engl. *accuracy of approximation*) von  $X$ . Offensichtlich gilt stets  $0 \leq \alpha(X) \leq 1$  und  $X$  ist exakt genau dann, wenn  $\alpha(X) = 1$  ist, bzw. unexakt genau für  $\alpha(X) < 1$ .

Eine andere Möglichkeit ist  $\alpha(X) := 1 - \frac{|X^\circ|}{|U|}$ .

**Definition 1.5 (unexakte Enthaltenseinsfunktion)**

Die *unexakte Enthaltenseinsfunktion* (engl. *rough membership function*) ist definiert als Abbildung

$$\begin{aligned} U \times \wp(U) &\rightarrow \mathbb{R} \\ \mu: \quad (x, X) &\mapsto \frac{|[x] \cap X|}{|[x]|} \end{aligned}$$



Der Wert  $\mu(x, X)$  kann interpretiert werden als bedingte Wahrscheinlichkeit, dass  $x$  zu  $X$  gehört.

**Lemma 1.6** Es gelten für alle  $x \in U$  und  $X \subseteq U$

- (a)  $\mu(x, X) = 1 \Leftrightarrow x \in X_*$
- (b)  $\mu(x, X) = 0 \Leftrightarrow x \in -X^*$
- (c)  $0 < \mu(x, X) < 1 \Leftrightarrow x \in X^\circ$
- (d)  $\mu(x, -X) = 1 - \mu(x, X)$
- (e)  $\mu(x, X \cup Y) \geq \mu(x, X) \vee \mu(x, Y)$
- (f)  $\mu(x, X \cap Y) \leq \mu(x, X) \wedge \mu(x, Y)$

**Beweis:**

- (a)  $1 = \mu(x, X) = \frac{|[x] \cap X|}{|[x]|} \Leftrightarrow [x] \cap X = [x] \Leftrightarrow [x] \subseteq X \Leftrightarrow x \in X_*$
- (b)  $0 = \mu(x, X) = \frac{|[x] \cap X|}{|[x]|} \Leftrightarrow [x] \cap X = \emptyset \Leftrightarrow \neg(x \in X^*) \Leftrightarrow x \in -X^*$
- (c)  $0 < \mu(x, X) < 1 \Leftrightarrow x \notin X_* \wedge x \notin -X^* \Leftrightarrow x \in X^\circ$
- (d)  $\mu(x, -X) = \frac{|[x] \cap -X|}{|[x]|} = \frac{|[x] \setminus X|}{|[x]|} = \frac{|[x] - ([x] \cap X)|}{|[x]|} = 1 - \frac{|[x] \cap X|}{|[x]|} = 1 - \mu(x, X)$
- (e)  $[x] \cap (X \cup Y) = ([x] \cap X) \cup ([x] \cap Y)$   
 $\Rightarrow |[x] \cap (X \cup Y)| \geq |[x] \cap X| \vee |[x] \cap Y|$   
 $\Rightarrow \mu(x, X \cup Y) \geq \mu(x, X) \vee \mu(x, Y)$
- (f)  $[x] \cap (X \cap Y) = ([x] \cap X) \cap ([x] \cap Y)$   
 $\Rightarrow |[x] \cap (X \cap Y)| \leq |[x] \cap X| \wedge |[x] \cap Y|$   
 $\Rightarrow \mu(x, X \cap Y) \leq \mu(x, X) \wedge \mu(x, Y)$  ■

## 1.2 Allgemeiner Approximationsraum

### Definition 1.7 (allgemeiner Approximationsraum)

Ein *allgemeiner Approximationsraum* (engl. *generalized approximation space*) ist ein Tupel  $\langle U, N, \nu \rangle$ . Dabei ist  $N: U \rightarrow \wp(U)$  eine Abbildung,  $N(x)$  heißt *Nachbarschaft* (engl. *neighborhood*) von  $x$ , und  $\nu: \wp(U) \times \wp(U) \rightarrow \mathbb{R}$  ist eine Abbildung, sie heißt *Inklusionsfunktion* (engl. *inclusion function*).  $\nu(X, Y)$  misst den Grad der Inklusion von  $X$  in  $Y$ . Die untere Approximation einer Menge  $X \subseteq U$  definieren wir nun als

$$X_* := \{x \in U \mid \nu(N(x), X) = 1\}$$

und die obere Approximation als

$$X^* := \{x \in U \mid \nu(N(x), X) > 0\}.$$

Für einen Approximationsraum  $\langle U, \approx \rangle$  ist beispielsweise  $N: x \mapsto [x]_{\approx}$  eine Nachbarschaftsfunktion, und analog ist für ein Informationssystem  $\langle U, A \rangle$  stets  $N: x \mapsto [x]_B$  für  $B \subseteq A$  eine geeignete Nachbarschaftsfunktion.

Sei  $\mathcal{F}$  eine Menge von Formeln, und für  $\varphi \in \mathcal{F}$  bezeichne  $\|\varphi\|$  die Menge aller Objekte, für die  $\varphi$  gilt bzw. wahr ist. Wir setzen  $\mathcal{F}(x) := \{\varphi \in \mathcal{F} \mid x \in \|\varphi\|\}$ , dann ist  $N: x \mapsto \bigcup_{\varphi \in \mathcal{F}(x)} \|\varphi\|$  eine entsprechende Nachbarschaftsfunktion.

Eine geeignete Inklusionsfunktion ist stets

$$\nu: (X, Y) \mapsto \begin{cases} \frac{|X \cap Y|}{|X|} & (X \neq \emptyset) \\ 1 & (X = \emptyset) \end{cases},$$

denn dann gilt  $\nu(X, Y) = 1 \Leftrightarrow X \subseteq Y$  und  $\nu(X, Y) > 0 \Leftrightarrow X \cap Y \neq \emptyset$ .

Noch allgemeiner kann  $N: U \rightarrow \wp(\wp(U))$  eine Abbildung sein, die jedem Objekt eine Menge von Nachbarschaften zuordnet. Die untere Approximation ist dann

$$X_* := \{x \in U \mid \exists Y \in N(x): \nu(Y, X) = 1\}$$

und

$$X^* := \{x \in U \mid \forall Y \in N(x): \nu(Y, X) > 0\}$$

ist die obere Approximation von  $X \subseteq U$ .

Ein Beispiel für eine solche Abbildung  $N: U \rightarrow \wp(\wp(U))$  findet man für

eine Topologie  $\tau$  auf  $U$  als Abbildung, die jedem Objekt  $x$  die Menge aller Umgebungen von  $x$  bezüglich  $\tau$  zuordnet, also  $N = \mathcal{U}: x \rightarrow \{X \mid X \subseteq U \wedge \exists O \in \tau: x \in O \subseteq X\}$ . Im topologischen Fall ist die untere Approximation gerade das Innere von  $X$  und die obere Approximation ergibt den Abschluss von  $X$ . Insbesondere sind also genau diejenigen Teilmengen des Universums unexakt, die weder offen noch abgeschlossen bezüglich der Topologie sind.

# 2 Information

## 2.1 Informationssystem

### Definition 2.1 (Informationssystem)

Seien  $U$  und  $A$  endliche nicht-leere Mengen sowie  $V := \{V_a \mid a \in A\}$  eine Menge von endlichen nicht-leeren Mengen.  $U$  heißt *Universum* (engl. *universe*) und die Elemente heißen *Objekte* (engl. *object*), die Elemente von  $A$  heißen *Attribute* (engl. *attribute*) und sind Abbildungen  $A: U \rightarrow V_a$ , die Elemente von  $V_a$  heißen *Werte* (engl. *value*) zum Attribut  $a$ . Dann heißt

$$\langle U, A \rangle$$

*Informationssystem* (engl. *information system*).

**Bemerkung:** Eine Teilmenge  $B \subseteq A$  erzeugt eine binäre Relation  $\approx_B$  auf  $U$ , genannt *B-Ununterscheidbarkeitsrelation* (engl. *indiscernibility relation*), vermöge

$$x \approx_B y \quad :\Leftrightarrow \quad a(x) = a(y) \quad \forall a \in B.$$

Offensichtlich ist  $\approx_B$  eine Äquivalenzrelation und wir schreiben auch  $U/B := U / \approx_B$  und  $[x]_B := [x]_{\approx_B}$ . Für ein Informationssystem  $\langle U, A \rangle$  ist  $\langle U, \approx_B \rangle$  für alle Teilmengen  $B$  von  $A$  ein Approximationsraum. Umgekehrt ist für einen Approximationsraum  $\langle U, \approx \rangle$  nun  $\langle U, \{x \mapsto [x]_{\approx}\} \rangle$  ein Informationssystem. Daher sind die im vorigen Abschnitt definierte Begriffe auch in Informationssystemen verfügbar. Für die Approximationen von  $\langle U, \approx_B \rangle$  schreiben wir  $X_B$  und  $X^B$ .

Jedes Informationssystem  $\langle U, A \rangle$  ist ein mehrwertiger Kontext  $\langle U, A, \cup V, I \rangle$  mit  $(x, a, v) \in I \quad :\Leftrightarrow \quad a(x) = v$ . Umgekehrt ist jeder mehrwertige Kontext  $\langle G, M, W, I \rangle$  Informationssystem  $\langle G, M \rangle$  mit  $m(g) = w \quad :\Leftrightarrow \quad (g, m, w) \in I$ .

Für ein Objekt  $x \in U$  bezeichnen wir die Menge  $\text{Sig}_B(x) := \{(a, a(x)) \mid a \in B\}$  auch als *B-Signatur* (engl. *signature*) von  $x$ , dann sind zwei Objekt genau dann *B-ununterscheidbar*, wenn ihre *B-Signaturen* übereinstimmen, d.h. es gilt  $x \approx_B y \quad \Leftrightarrow \quad \text{Sig}_B(x) = \text{Sig}_B(y)$  für alle  $x, y \in U$ .

## 2.2 Redukt & Kern

In diesem Abschnitt schreiben wir  $\approx$  für die  $A$ -Ununterscheidbarkeitsrelation.

### Definition 2.2 (Redukt, Kern)

Für ein Informationssystem  $\langle U, A \rangle$  heißt eine Attributmenge  $B \subseteq A$  *Redukt* (engl. *reduct*) von  $A$ , wenn  $B$  die gleiche Ununterscheidbarkeit erzeugt wie  $A$ , also falls  $\approx_B = \approx$  gilt. Falls  $B$  minimal bezüglich der Teilmengeninklusion ist, so heißt  $B$  *Primredukt*. Für die Menge aller Redukte von  $A$  schreiben wir auch  $\text{Red}(A)$  und analog  $\text{PRed}(A)$  für die Menge der Primredukte. Den Durchschnitt aller Redukte

$$\text{Ker}(A) := \bigcap \text{Red}(A)$$

nennen wir *Kern* (engl. *core*).

Es gilt  $\text{Ker}(A) = \bigcap \text{Red}(A) = \bigcap \text{PRed}(A)$ . Im Folgenden zeigen wir eine allgemeine Möglichkeit zur Bestimmung der Redukte eines Informationssystems. Dafür definieren wir die *Unterscheidbarkeitsmatrix* (engl. *discernibility matrix*) als Matrix, die in Zeile  $x \in U$  und Spalte  $y \in U$  die Menge aller Attribute aus  $A$  hat, bezüglich derer sich  $x$  und  $y$  unterscheiden. Wir setzen also

$$A_{xy} := \{a \in A \mid a(x) \neq a(y)\}$$

und damit  $\text{Mat}(A) := (A_{xy})_{x,y \in U} \in \wp(A)^{U \times U}$ . Diese Matrix ist symmetrisch, denn es gilt stets  $A_{xy} = A_{yx}$ , und für alle ununterscheidbaren Objekte  $x \approx y$  gilt  $A_{xy} = \emptyset$ , also sind insbesondere alle Einträge  $A_{xx}$  der Hauptdiagonale  $\emptyset$ .

Die Potenzmengenalgebra  $\wp(A)$  ist isomorph zur booleschen Algebra  $\{0, 1\}^A$  vermöge der Abbildung  $B \mapsto B' := (a')_{a \in A}$  mit  $a' = 1$  für  $a \in B$  und  $a' = 0$  sonst bzw.  $(a')_{a \in A} \mapsto \{a \in A \mid a' = 1\}$ . Insbesondere ist  $A' = (1)_{a \in A}$  und  $\emptyset' = (0)_{a \in A}$ .

### Definition 2.3 (Unterscheidbarkeitsfunktion)

Die boolesche Funktion

$$f_A: (a')_{a \in A} \mapsto \bigwedge_{x \not\approx y} \bigvee_{a \in A_{xy}} a'$$

heißt *Unterscheidbarkeitsfunktion* (engl. *discernibility function*).

Es ist leicht zu sehen, dass  $f_A(A') = 1$  gilt, denn für jedes unterscheidbare Paar von Objekten muss es ein Attribut geben, dass sie unterscheidet. Für  $B_0 \subseteq B$  gilt  $f_A(B'_0) \leq f_A(B')$ . Der boolesche Term  $\bigwedge_{x \not\approx y} \bigvee_{a \in A_{xy}} a'$  ist in konjunktiver Normalform.

**Bemerkung:** Haben wir eine totale Striktordnung  $<$  der Objekte in  $U$  gegeben, dann setzen wir  $\prec := < \cap \not\approx$  und können damit den booleschen Term auf seine halbe Länge kürzen zu

$$f_A((a')_{a \in A}) = \bigwedge_{x \prec y} \bigvee_{a \in A_{xy}} a'$$

denn es gilt stets  $A_{xy} = A_{yx}$ .

**Lemma 2.4**  $B$  ist Redukt von  $A$  genau dann, wenn  $f_A(B') = 1$  gilt.

**Beweis:** Es gilt stets  $\approx \subseteq \approx_B$  wegen  $B \subseteq A$ . Weiter haben wir

$$\begin{aligned} B \in \text{Red}(A) &\iff \not\approx \subseteq \not\approx_B \\ &\iff \forall x \not\approx y \exists a \in A_{xy} : a \in B \\ &\iff \forall x \not\approx y \exists a \in A_{xy} : a' = 1 \\ &\iff \bigwedge_{x \not\approx y} \bigvee_{a \in A_{xy}} a' = 1 \\ &\iff f_A(B') = 1 \end{aligned}$$

für jede Attributmenge  $B \subseteq A$ . ■

Eine Attributmenge  $B$  ist ein Primredukt von  $A$  genau dann, wenn  $f_A(B') = 1$  und für alle echten Teilmengen  $B_0 \subsetneq B$  stets  $f_A(B'_0) = 0$  ist.

Das folgende Lemma zeigt, dass wir jeden booleschen Term in konjunktiver Normalform auch in disjunktiver Normalform darstellen können.

**Lemma 2.5** Es gilt

$$\bigwedge_{i \in I} \bigvee_{j_i \in J_i} x_{ij_i} = \bigvee_{j \in \prod_{i \in I} J_i} \bigwedge_{i \in I} x_{ij_i}$$

dabei ist  $\prod_{i \in I} J_i := \{j : I \rightarrow \bigcup_{i \in I} J_i \mid \forall i \in I : j_i \in J_i\}$  das kartesische Produkt. Ein Element  $j \in \prod_{i \in I} J_i$  ist eine Auswahlfunktion, die jedem  $i \in I$  ein Element der Menge  $J_i$  zuordnet.

**Beweis:** Es gelten folgende äquivalente Umformungen:

$$\begin{aligned} \bigwedge_{i \in I} \bigvee_{j_i \in J_i} x_{ij_i} = 1 &\iff \forall i \in I \exists j_i \in J_i: x_{ij_i} = 1 \\ &\iff \exists j \in \prod_{i \in I} J_i \forall i \in I: x_{ij_i} = 1 \\ &\iff \bigvee_{j \in \prod_{i \in I} J_i} \bigwedge_{i \in I} x_{ij_i} = 1 \end{aligned}$$

■

### Definition 2.6 (Implikant)

Ein Literal ist eine Variable  $x_i$  oder eine negierte Variable  $\neg x_i$ . Ein Konjunktionsterm  $k$  von Literalen bzw. eine boolesche Funktion

$$\begin{aligned} &\{0, 1\}^I \rightarrow \{0, 1\} \\ k: (x_i)_{i \in I} &\mapsto \bigwedge_{i \in J} y_i \end{aligned}$$

mit  $J \subseteq I$  und  $y_i \in \{x_i, \neg x_i\}$  heißt *Implikant* (engl. *implicant*) der Funktion  $f: \{0, 1\}^I \rightarrow \{0, 1\}$ , falls für alle  $x \in \{0, 1\}^I$  aus  $k(x) = 1$  stets  $f(x) = 1$  folgt. Wir sagen auch  $k$  *impliziert*  $f$  und schreiben  $k \Rightarrow f$ . Ein Implikant heißt *Primimplikant* von  $f$ , wenn es keinen Implikanten  $h \neq k$  von  $f$  gibt, der von  $k$  impliziert wird, d.h. falls  $\nexists h \neq k: k \Rightarrow h \Rightarrow f$ . Wir legen fest, dass  $\text{Imp}(f)$  die Menge aller Implikanten von  $f$  ist und entsprechend ist  $\text{PImp}(f)$  die Menge der Primimplikanten.

Ein Implikant ist Primimplikant, wenn der Konjunktionsterm eine minimale Anzahl von Literalen enthält, d.h. falls aus dem Konjunktionsterm kein Literal entfernt werden kann, ohne dass er seine Eigenschaft Implikant zu sein, verliert. In der disjunktiven Normalform sind alle Konjunktionsterme Implikanten und wir können jede boolesche Funktion als Disjunktion ihrer (Prim-)implikanten darstellen. Es gibt verschiedene Verfahren zur Bestimmung der (Prim-)implikanten einer booleschen Funktion. Hier sind einige Beispiele:

- *Algebraische Verfahren:* Anwendung der Rechengesetze von Booleschen Algebren, Nelson-Verfahren
- *Graphische Verfahren:* Karnaugh-Veitch-Diagramme
- *Tabellarische Verfahren:* Quine-McCluskey-Verfahren, Konsensus-Verfahren

**Lemma 2.7** Die Menge der Implikanten von  $f_A$  ist genau die Menge der booleschen Funktionen

$$\{0,1\}^A \rightarrow \{0,1\}$$

$$\beta': (a')_{a \in A} \mapsto \bigwedge_{x \not\approx y} \beta'_{xy}$$

für Abbildungen  $\beta \in \times_{x \not\approx y} A_{xy}$ .

**Beweis:** Nach Lemma 2.5 folgt

$$f_A(B') = \bigwedge_{x \not\approx y} \bigvee_{a \in A_{xy}} a' = \bigvee_{\beta \in \times_{x \not\approx y} A_{xy}} \bigwedge_{x \not\approx y} \beta'_{xy},$$

also

$$f_A = \bigvee_{\beta \in \times_{x \not\approx y} A_{xy}} \beta'.$$

Für  $\beta \in \times_{x \not\approx y} A_{xy}$  gilt stets  $\beta'(B') \leq f_A(B')$  für  $B \subseteq A$ , also  $\beta' \Rightarrow f_A$ . ■

Für  $B_0 \subseteq B$  gilt  $\beta'(B'_0) \leq \beta'(B')$ . Für einen Primimplikant  $\beta'$  hat die zugehörige Abbildung  $\beta$  (die nicht notwendig eindeutig ist) ein minimales Bild  $\text{im}\beta$ , d.h. es gibt keinen Implikanten  $\beta'_0$  von  $f_A$  mit  $\text{im}\beta_0 \subsetneq \text{im}\beta$ .

**Lemma 2.8** Es gilt  $\beta'(B') = 1$  genau dann, wenn  $\beta \in \times_{x \not\approx y} (A_{xy} \cap B)$ .

**Beweis:** Wir wissen, dass  $\beta'(B') = 1$  genau dann gilt, wenn für alle unterscheidbaren Objekte  $x \not\approx y$  stets  $\beta'_{xy} = 1$  ist. Das ist äquivalent dazu, dass jedes Attribut  $\beta_{xy}$  in  $A_{xy} \cap B$  liegt. ■

Insbesondere gilt also für  $\beta'(B') = 1$  stets  $\text{im}\beta \subseteq B$ .

**Satz 2.9** Für jeden Implikant  $\beta'$  von  $f_A$  ist das Bild  $\text{im}\beta$  ein Redukt von  $A$ . Falls  $\beta'$  ein Primimplikant ist, so ist  $\text{im}\beta$  ein Primredukt.

**Beweis:** Für jede Abbildung  $\beta \in \times_{x \not\approx y} A_{xy}$  gilt  $f_A((\text{im}\beta)') = 1$  und damit ist  $\text{im}\beta$  ein Redukt nach Lemma 2.4. Falls  $\beta'$  ein Primimplikant ist, dann gibt es keinen Implikant  $\beta'_0$  von  $f_A$  mit  $\text{im}\beta_0 \subsetneq \text{im}\beta$ , d.h. für alle  $B_0 \subsetneq \text{im}\beta$  gibt es keinen Implikant  $\beta'_0$  von  $f_A$  mit  $\beta'_0(B'_0) = 1$ . Also gilt für alle  $B_0 \subsetneq \text{im}\beta$  immer  $f_A(B'_0) = 0$  und somit existiert kein Redukt  $B_0$  mit  $B_0 \subsetneq \text{im}\beta$ . ■



Falls  $B$  ein Primredukt von  $A$  ist, dann gilt für einen Implikant  $\beta'$  von  $f_A$  mit  $\beta'(B') = 1$  immer  $\text{im}\beta = B$ . Da nach obigen Satz  $\text{im}\beta$  stets ein Redukt ist, würde  $\text{im}\beta \subsetneq B$  im Widerspruch zur Minimalität von  $B$  stehen.

**Satz 2.10** Ein  $B$  ist Redukt von  $A$  genau dann, wenn ein Implikant  $\beta'$  von  $f_A$  mit  $\text{im}\beta \subseteq B$  existiert. Weiter ist  $B$  ein Primredukt von  $A$  genau dann, wenn ein Primimplikant  $\beta'$  von  $f_A$  mit  $\text{im}\beta = B$  existiert.

**Beweis:** Eine Attributmenge  $B \subseteq A$  ist nach Lemma 2.4 genau dann ein Redukt, wenn  $f_A(B') = 1$  ist. Nach Lemma 2.7 ist das genau dann der Fall, wenn es eine Abbildung  $\beta \in \times_{x \not\approx y} A_{xy}$  gibt, die jedem unterscheidbaren Objektpaar  $x \not\approx y$  ein Attribut  $\beta_{xy}$  zuordnet, das sie unterscheidet, sodass  $\beta'(B') = 1$  ist. Das ist nach Lemma 2.8 äquivalent dazu, dass  $\beta'$  ein Implikant von  $f_A$  mit  $\beta \in \times_{x \not\approx y} (A_{xy} \cap B)$  ist.

$$\begin{aligned} B \in \text{Red}(A) &\stackrel{2.4}{\iff} f(B') = 1 \\ &\stackrel{2.7}{\iff} \exists \beta \in \times_{x \not\approx y} A_{xy}: \beta'(B') = 1 \\ &\stackrel{2.8}{\iff} \exists \beta' \in \text{Imp}(f_A): \text{im}\beta \subseteq B \end{aligned}$$

Weiter ist  $B$  genau dann ein Primredukt, wenn  $B$  ein Redukt ist und es kein Redukt  $B_0 \subsetneq B$  gibt. Das ist nach dem ersten Teil äquivalent dazu, dass ein Implikant  $\beta'$  von  $f_A$  mit  $\text{im}\beta \subseteq B$  existiert und es keinen Implikant  $\beta'_0$  von  $f_A$  mit  $\text{im}\beta_0 \subsetneq B$  gibt. Weil  $B$  ein Primredukt ist, gilt  $\text{im}\beta = B$  und damit ist  $\beta'$  ein Primimplikant von  $f_A$ .

$$\begin{aligned} B \in \text{PRed}(A) &\iff B \in \text{Red}(A) \wedge \forall B_0 \subsetneq B: B_0 \notin \text{Red}(A) \\ &\iff \exists \beta' \in \text{Imp}(f_A): \text{im}\beta \subseteq B \wedge \nexists \beta'_0 \in \text{Imp}(f_A): \text{im}\beta_0 \subsetneq B \\ &\iff \exists \beta' \in \text{PImp}(f_A): \text{im}\beta = B \end{aligned}$$

Umgekehrt sei  $\beta'$  Primimplikant von  $f_A$ , dann ist  $\beta'$  insbesondere ein Implikant von  $f_A$  und es gibt keinen Implikant  $\beta'_0$  von  $f_A$  mit  $\text{im}\beta_0 \subsetneq \text{im}\beta = B$ . ■

**Korollar 2.11** Jeder Implikant von  $f_A$  bestimmt vermöge der Abbildung

$$\rho: \begin{array}{l} \text{Imp}(f_A) \rightarrow \text{Red}(A) \\ \beta' \mapsto \text{im}\beta \end{array}$$

ein Redukt von  $A$  und jeder Primimplikant erzeugt ein Primredukt. Die Zuordnung  $\rho$  ist surjektiv, die Menge der Implikanten bestimmt also die Menge der Redukte vollständig. Weiter ist  $\rho(\text{PImp}(f_A)) = \text{PRed}(A)$ .

**Definition 2.12 (Reduktion)**

Für ein Redukt  $B$  von  $A$  ist  $\langle U/B, B \rangle$  mit  $a([x]_B) := a(x)$  für alle  $a \in B, x \in U$  ein Informationssystem und heißt  $B$ -Reduktion von  $\langle U, A \rangle$ .

In einer Reduktion werden ununterscheidbare Objekte in Mengen zusammengefasst und die Anzahl der Attribute reduziert.

**Bemerkung:** Wir können die Primredukte von  $A$  ermitteln, indem wir die Primimplikanten von  $f_A$  suchen. Dies kann formal dadurch geschehen, dass wir den Term

$$\bigwedge_{x \neq y} \bigvee_{a \in A_{xy}} a'$$

in eine minimale disjunktive Normalform

$$\bigvee \bigwedge a'$$

umformen, indem die Rechengesetze in Booleschen Algebren (z.B. Distributivgesetz, Absorptionsgesetz, etc.) so oft wie möglich angewandt werden. Die Konjunktionsterme  $\bigwedge a'$  definieren dann alle Primimplikanten.

**Beispiel:** Gegeben sei ein Informationssystem  $\langle U, A \rangle$  mit den Objekten  $U = \{\text{Auto1}, \text{Auto2}, \text{Auto3}\}$  und den Attributen  $A = \{\text{Motor}, \text{Farbe}, \text{Türen}\}$ , deren Werte in der folgenden Tabelle notiert sind.

	Motor	Farbe	Türen
Auto1	Otto	grün	vier
Auto2	Otto	gelb	zwei
Auto3	Diesel	blau	vier

Es ergibt sich die Unterscheidbarkeitsmatrix

	Auto1	Auto2	Auto3
Auto1	$\emptyset$	{Farbe, Türen}	{Motor, Farbe}
Auto2	{Farbe, Türen}	$\emptyset$	{Motor, Farbe, Türen}
Auto3	{Motor, Farbe}	{Motor, Farbe, Türen}	$\emptyset$

und damit haben wir für  $\text{Auto1} \prec \text{Auto2} \prec \text{Auto3}$  nun die Unterscheidbarkeitsfunktion

$$\begin{aligned} f_A &= (\text{Farbe}' \vee \text{Türen}') \wedge (\text{Motor}' \vee \text{Farbe}') \wedge (\text{Motor}' \vee \text{Farbe}' \vee \text{Türen}') \\ &= (\text{Farbe}' \vee \text{Türen}') \wedge (\text{Farbe}' \vee \text{Motor}') \\ &= \text{Farbe}' \vee (\text{Motor}' \wedge \text{Türen}') \end{aligned}$$

mit den Primimplikanten  $\text{Farbe}'$  und  $\text{Motor}' \wedge \text{Türen}'$ . Damit haben wir also die beiden Primredukte  $\{\text{Farbe}'\}$  und  $\{\text{Motor}', \text{Türen}'\}$ . Die  $\{\text{Farbe}'\}$ -Reduktion ergibt sich zu

	Farbe
{Auto1}	grün
{Auto2}	gelb
{Auto3}	blau

□

**Bemerkung:** Sei  $|\approx| > |A|$  und  $A^\approx := \times_{x \approx y} A = \{\beta: \approx \rightarrow A\}$ . Wir definieren auf  $A^\approx$  eine Quasiordnung vermöge  $\beta_1 \sqsubseteq \beta_2 :\Leftrightarrow \beta_1(\approx) \subseteq \beta_2(\approx)$  und dann ist  $\equiv := \sqsubseteq \cap \supseteq$  eine Äquivalenzrelation, nach der wir  $A^\approx$  faktorisieren zu der geordneten Menge  $\langle A^\approx / \equiv, \sqsubseteq \rangle$ . Dann ist  $\langle A^\approx / \equiv \cup \{\emptyset\}, \sqsubseteq \rangle$  mit  $\forall X \in A^\approx / \equiv: \emptyset \sqsubseteq X$  ein vollständiger Verband. Das Supremum ist gegeben durch  $\emptyset \vee [\beta]_\equiv = [\beta]_\equiv$  und  $\bigvee_{i \in I} [\beta_i]_\equiv = [\beta]_\equiv$  mit  $\beta(\approx) = \bigcup_{i \in I} \beta_i(\approx)$  und das Infimum ist  $\bigwedge_{i \in I} [\beta_i]_\equiv = [\beta]_\equiv$  für  $\beta(\approx) = \bigcap_{i \in I} \beta_i(\approx) \neq \emptyset$  und  $\bigwedge_{i \in I} [\beta_i]_\equiv = \emptyset$  für  $\bigcap_{i \in I} \beta_i(\approx) = \emptyset$ . Die Menge der  $\beta$  aller Implikanten  $\beta'$  von  $f_A$  faktorisiert nach  $\equiv$  bildet einen zu einem vollständigen Supremumunterhalbverband des eben definierten vollständigen Verbands isomorphen Supremumhalbverband. Die Menge der  $\beta$  aller Primimplikanten  $\beta'$  faktorisiert nach  $\equiv$  bildet eine maximale Antikette.

Analog ist die Potenzmenge von  $A$  mit der Teilmengeninklusion ein vollständiger Verband, in dem die Menge der Redukte von  $A$  einen vollständigen Supremumunterhalbverband bildet. Die Menge der Primredukte von  $A$  ist eine maximale Antikette. Diese Strukturen sind isomorph zu den obigen.

# 3 Entscheidung

## 3.1 Entscheidungssystem

### Definition 3.1 (Entscheidungssystem)

Sei  $\langle U, A \rangle$  ein Informationssystem mit einer Partition  $\{C, D\}$  der Attributmenge  $A$ , die Attribute aus  $C$  heißen *Bedingungen* (engl. *condition*) und die aus  $D$  bezeichnen wir als *Entscheidungen* (engl. *decision*).

$$\langle U, C, D \rangle$$

heißt *Entscheidungssystem* (engl. *decision system*).

### Definition 3.2 (Formel)

Sei  $B \subseteq C \uplus D$ . Ein Ausdruck der Form  $a = v$  mit  $a \in B$ ,  $v \in V_a$  wird als *Deskriptor* (engl. *descriptor*) oder *Selektor* (engl. *selector*) über  $B$  und  $V$  bezeichnet. Die Menge  $\mathcal{F}(B, V)$  von Formeln über  $B$  und  $V$  wird nun induktiv aufgebaut: Jeder Deskriptor über  $B$  und  $V$  ist eine Formel über  $B$  und  $V$ . Für zwei Formeln über  $B$  und  $V$  sind auch deren Konjunktion, Disjunktion und Negation stets Formeln über  $B$  und  $V$ . Formal legen wir also fest:

- (i)  $a \in B, v \in V_a \Rightarrow (a = v) \in \mathcal{F}(B, V)$
- (ii)  $\varphi, \psi \in \mathcal{F}(B, V) \Rightarrow (\varphi \wedge \psi) \in \mathcal{F}(B, V)$
- (iii)  $\varphi, \psi \in \mathcal{F}(B, V) \Rightarrow (\varphi \vee \psi) \in \mathcal{F}(B, V)$
- (iv)  $\varphi \in \mathcal{F}(B, V) \Rightarrow (\neg\varphi) \in \mathcal{F}(B, V)$

Mit  $\|\varphi\|$  beschreiben wir die Bedeutung der Formel  $\varphi$  als Menge derjenigen Objekte, die die Eigenschaft  $\varphi$  haben, d.h. für die  $\varphi$  wahr ist. Diese Mengen werden entsprechend dem Aufbau der Menge der Formeln auch induktiv aufgebaut:

- (i)  $\|a = v\| := \{x \in U \mid a(x) = v\} = a^{-1}(v)$
- (ii)  $\|\varphi \wedge \psi\| := \|\varphi\| \cap \|\psi\|$
- (iii)  $\|\varphi \vee \psi\| := \|\varphi\| \cup \|\psi\|$
- (iv)  $\|\neg\varphi\| := -\|\varphi\|$

Die Elemente von  $\mathcal{F}(C, V)$  heißen *Bedingungsformeln*. Jedes Objekt  $x \in U$  gehört zu der *Bedingungsklasse*  $\|\bigwedge_{c \in C} c = c(x)\|$ . Die Menge aller Bedingungsklassen bildet eine Partition von  $U$ , nämlich  $U/C$ . Ein Objekt  $y \in U$  gehört zu der Bedingungsklasse  $\|\bigwedge_{c \in C} c = c(x)\|$  genau dann, wenn  $x$  und  $y$   $C$ -ununterscheidbar sind, d.h.  $\|\bigwedge_{c \in C} c = c(x)\| = [x]_C$ . Analog heißen Formeln aus  $\mathcal{F}(D, V)$  *Entscheidungsformeln*. Die Menge aller Entscheidungsformeln  $\|\bigwedge_{d \in D} d = d(x)\|$  ist die Partition  $U/D$  von  $U$ . Es gilt  $y \in \|\bigwedge_{d \in D} d = d(x)\| \Leftrightarrow x \approx_D y$ , also  $\|\bigwedge_{d \in D} d = d(x)\| = [x]_D$ .

## 3.2 Konsistenz

### Definition 3.3 (Entscheidungsregel)

Für Formeln  $\varphi \in \mathcal{F}(C, V)$  und  $\psi \in \mathcal{F}(D, V)$  heißt

$$\varphi \Rightarrow \psi$$

*Entscheidungsregel* (engl. *decision rule*). Wir nennen  $\varphi$  *Vorgänger* (engl. *predecessor*) und  $\psi$  *Nachfolger* (engl. *successor*) der Entscheidungsregel  $\varphi \Rightarrow \psi$ . Eine Entscheidungsregel  $\varphi \Rightarrow \psi$  heißt *wahr*, wenn  $\psi$  wahr ist für alle Objekte, für die bereits  $\varphi$  wahr ist, d.h. falls  $\|\varphi\| \subseteq \|\psi\|$  gilt.

Oft werden Entscheidungsregeln auch als *wenn-dann-Regeln* (engl. *if then rules*) bezeichnet. Jedes Objekt  $x \in U$  bestimmt eine Entscheidungsregel

$$\left( \bigwedge_{c \in C} c = c(x) \right) \Rightarrow \left( \bigwedge_{d \in D} d = d(x) \right).$$

Diese ist wahr genau dann, wenn  $[x]_C \subseteq [x]_D$  gilt, d.h. wenn für jedes Objekt  $y \in U$ , für das  $x, y$  ununterscheidbar bezüglich  $C$  sind, stets  $x, y$  auch  $D$ -ununterscheidbar sind.

### Definition 3.4 (Konsistenz)

Wir definieren die sogenannte *Entscheidungsfunktion*

$$\delta_C^D: \begin{aligned} U &\rightarrow \wp\left(\prod_{d \in D} V_d\right) \\ x &\mapsto \left\{v \in \prod_{d \in D} V_d \mid \exists y \in U: y \approx_C x \wedge \forall d \in D: d(y) = v_d\right\}. \end{aligned}$$

Das Entscheidungssystem  $\langle U, C, D \rangle$  heißt *konsistent*, wenn  $\delta_C^D(x)$  für alle  $x \in U$  einelementig ist; andernfalls heißt  $\langle U, C, D \rangle$  *inkonsistent*.

Ein konsistentes Entscheidungssystem heißt auch *deterministisch*.

**Lemma 3.5**  $\langle U, C, D \rangle$  ist konsistent genau dann, wenn für jedes Objekt  $x \in U$  die Entscheidungsregel  $\bigwedge_{c \in C} c = c(x) \Rightarrow \bigwedge_{d \in D} d = d(x)$  wahr ist.

**Beweis:**  $\langle U, C, D \rangle$  ist konsistent genau dann, wenn

$$\delta_C^D(x) = \{v \in \bigtimes_{d \in D} V_d \mid \exists y \in U: y \approx_C x \wedge \forall d \in D: d(y) = v_d\}$$

für alle Objekte  $x \in U$  einelementig ist. Da  $x$  und  $x$  stets  $C$ -ununterscheidbar sind, gilt  $(d(x))_{d \in D} \in \delta_C^D(x)$ . Damit ist also  $\delta_C^D(x)$  für alle  $x \in U$  einelementig genau dann, wenn für jedes Objekt  $y \in U$ , für das  $x, y$   $C$ -ununterscheidbar sind, stets  $x, y$  auch  $D$ -ununterscheidbar sind. ■

Ein Entscheidungssystem  $\langle U, C, D \rangle$  ist nach obigem Lemma konsistent genau dann, wenn  $\approx_C \subseteq \approx_D$  gilt, also wenn die Partition  $U/C$  feiner als  $U/D$  ist.

Wenn die Entscheidungen in  $D$  vollständig von den Bedingungen in  $C$  abhängen, d.h. wenn die Werte der Bedingungen in  $C$  die Werte der Entscheidungen in  $D$  eindeutig bestimmen, dann schreiben wir auch

$$C \Rightarrow D.$$

Also hängt  $D$  vollständig von  $C$  ab, wenn es einen funktionalen Zusammenhang zwischen den Werten der Elemente von  $C$  und  $D$  gibt. Formal definieren wir nun die (partielle) Abhängigkeit.

### Definition 3.6 (partielle Abhängigkeit)

Wir setzen

$$\text{Pos}(C, D) := \bigcup_{X \in U/D} X_C$$

als den *positiven Bereich* der Partition  $U/D$  bezüglich  $C$ . Nun heißt  $D$  *abhängig* von  $C$  zum Grad  $k$ , falls

$$k = \gamma(C, D) := \frac{|\text{Pos}(C, D)|}{|U|}$$

gilt, und wir schreiben dafür auch

$$C \Rightarrow_k D.$$

Für  $k = 1$  ist  $D$  *vollständig abhängig* von  $C$ , symbolisch auch schlichtweg  $C \Rightarrow D$  geschrieben. Für  $k < 1$  nennen wir  $D$  *partiell abhängig* von  $C$  zum Grad  $k$ .

Der positive Bereich ist die Vereinigung aller unteren  $C$ -Approximationen von  $D$ -elementaren Mengen, daher enthält  $\text{Pos}(C, D)$  alle Objekte, die bezüglich  $C$  sichere Objekte von Klassen der Partition  $U/D$  sind. Es ist  $\text{Pos}(C, D) = \bigcup_{x \in U} ([x]_D)_C$ .

**Lemma 3.7** Ein Entscheidungssystem  $\langle U, C, D \rangle$  ist konsistent genau dann, wenn  $C \Rightarrow D$  gilt.

**Beweis:** Sei  $\langle U, C, D \rangle$  konsistent. Für  $x \in U$  gilt  $x \in [x]_C$  und  $[x]_C \subseteq [x]_D$ . Damit liegt  $x$  auch in der Vereinigung derjenigen  $C$ -elementaren Mengen, die in  $[x]_D$  als Teilmenge enthalten sind. Diese Vereinigung ist nichts anderes als  $([x]_D)_C$  und  $[x]_D \in U/D$ , also liegt  $x$  im positiven Bereich  $\text{Pos}(C, D)$ . Umgekehrt sei nun  $\text{Pos}(C, D) = U$ . Angenommen,  $\langle U, C, D \rangle$  wäre nicht konsistent, es gäbe also ein Objekt  $x \in U$  mit  $[x]_C \not\subseteq [x]_D$ , dann läge  $x$  jedoch nicht in  $\text{Pos}(C, D)$ .  $\nexists$  Widerspruch! ■

### 3.3 Redukt & Kern

#### Definition 3.8 (Redukt, Kern)

Eine Attributmengende  $B \subseteq C$  heißt  $D$ -(*Prim*-)Redukt oder *entscheidungsabhängiges* (*Prim*-)Redukt (engl. *decision-relative reduct*) von  $C$ , falls  $B$  eine (minimale) Teilmenge von  $C$  mit  $\gamma(B, D) = \gamma(C, D)$  oder  $\delta_B^D = \delta_C^D$  ist. Für die Menge aller  $D$ -Redukte von  $C$  schreiben wir auch  $\text{Red}(C, D)$  und analog  $\text{PRed}(C, D)$  für die Menge der Primredukte. Den Durchschnitt aller  $D$ -Redukte von  $C$

$$\text{Ker}(C, D) := \bigcap \text{Red}(C, D)$$

bezeichnen wir auch als den  $D$ -Kern (engl. *core*) von  $C$ .

Der  $D$ -Kern von  $C$  ist in allen  $D$ -Redukten von  $C$  enthalten, also enthält er die „wichtigen“ Bedingungen, deren Entfernen stets eine Verkleinerung des positiven Bereichs, also eine Verminderung des Klassifikationsvermögens, bewirken.

Zur Bestimmung von entscheidungsabhängigen Redukten gehen wir ähnlich wie bei der Bestimmung der Redukte von Informationssystemen vor. Dazu sei  $\langle U, C, D \rangle$  ein konsistentes Entscheidungssystem und  $\text{Mat}(C)$  sei die Unterscheidbarkeitsmatrix von  $\langle U, C \rangle$ . Wir konstruieren eine *entscheidungsabhängige Unterscheidbarkeitsmatrix* (engl. *decision-relative discerni-*

bility matrix)  $\text{Mat}(C, D) := (C_{xy}^D)_{x,y \in U}$  mit

$$C_{xy}^D := \begin{cases} \emptyset & (x \approx_D y) \\ C_{xy} & (x \not\approx_D y). \end{cases}$$

Analog definieren wir wieder eine *entscheidungsabhängige Unterscheidbarkeitsfunktion* (engl. *decision-relative discernibility function*) vermöge

$$f_C^D((c')_{c \in C}) := \bigwedge_{\substack{x \not\approx_C y \\ x \not\approx_D y}} \bigvee_{c \in C_{xy}^D} c'$$

und erhalten die Menge der entscheidungsabhängigen (Prim-)redukte aus den (Prim-)implikanten der Funktion  $f_C^D$ .

**Beispiel:** Wir erweitern das vorige Beispiel eines Informationssystems zu einem Entscheidungssystem  $\langle U, C, D \rangle$  mit den Objekten  $U = \{\text{Auto1}, \text{Auto2}, \text{Auto3}\}$ , den Bedingungen  $C = \{\text{Motor}, \text{Farbe}, \text{Türen}\}$  und einer Entscheidung  $D = \{\text{Fahren}\}$ .

	Motor	Farbe	Türen	Fahren
Auto1	Otto	grün	vier	nein
Auto2	Otto	gelb	zwei	nein
Auto3	Diesel	blau	vier	ja

Es ergibt sich die entscheidungsabhängige Unterscheidbarkeitsmatrix

	Auto1	Auto2	Auto3
Auto1	$\emptyset$	$\emptyset$	$\{\text{Motor}, \text{Farbe}\}$
Auto2	$\emptyset$	$\emptyset$	$\{\text{Motor}, \text{Farbe}, \text{Türen}\}$
Auto3	$\{\text{Motor}, \text{Farbe}\}$	$\{\text{Motor}, \text{Farbe}, \text{Türen}\}$	$\emptyset$

und damit haben wir für  $\text{Auto1} \prec \text{Auto2} \prec \text{Auto3}$  nun die entscheidungsabhängige Unterscheidbarkeitsfunktion

$$\begin{aligned} f_C^D &= (\text{Motor}' \vee \text{Farbe}') \wedge (\text{Motor}' \vee \text{Farbe}' \vee \text{Türen}') \\ &= \text{Motor}' \vee \text{Farbe}' \end{aligned}$$

mit den Primimplikanten  $\text{Motor}'$  und  $\text{Farbe}'$ . Damit haben wir also die beiden entscheidungsabhängigen Primredukte  $\{\text{Motor}\}$  und  $\{\text{Farbe}\}$ . Die  $\{\text{Motor}\}$ -Reduktion ergibt sich zu

	Motor	Fahren
$\{\text{Auto1}, \text{Auto2}\}$	Otto	nein
$\{\text{Auto3}\}$	Diesel	ja



### 3.4 Attributselektion

In vielen Fällen sind die Attribute in Entscheidungssystemen redundant, das heißt wir können Attribute entfernen, ohne dabei das Klassifikationsvermögen zu verringern. Das *Attributauswahlproblem* (engl. *attribute selection problem*) ist das Problem der Auswahl einer ausreichenden Attributmenge, während die unwichtigen Attribute entfernt werden. Dazu eignet sich zuallererst natürlich der Kern eines Entscheidungssystems, denn die Attribute im Kern liegen in allen Redukten, und sind somit zur Aufrechterhaltung der maximalen Klassifikationsfähigkeit unbedingt nötig. Es kann allerdings vorkommen, dass der Kern leer ist.

Um das Attributauswahlproblem zu lösen, benötigen wir den Begriff eines approximativen Redukts. Für ein konsistentes Entscheidungssystem  $\langle U, C, \{d\} \rangle$  heißt jede Bedingungs Menge  $B \subseteq C$  *approximatives Redukt* von  $C$  (engl. *approximate reduct*). Die Abbildung

$$\varepsilon_C^{\{d\}}: \wp(C) \rightarrow \mathbb{R} \\ B \mapsto \frac{\gamma(C, \{d\}) - \gamma(B, \{d\})}{\gamma(C, \{d\})} = 1 - \frac{\gamma(B, \{d\})}{\gamma(C, \{d\})}$$

misst den Fehler der Reduktapproximation. Der Reduktapproximationsfehler gibt an, wie gut die Bedingungen aus  $B$  die Bedingungen aus  $C$  unter Berücksichtigung der Entscheidung  $d$  annähern. Der Fehler  $\varepsilon_C^{\{d\}}(B)$  liegt stets im reellen Intervall  $[0, 1]$ , wobei  $\varepsilon_C^{\{d\}}(B) = 0$  genau dann gilt, wenn  $B$  ein Redukt ist.

Ausgehend vom Kern können wir nun eine endliche Folge  $(B_n)_{n=0}^k$  minimaler Länge von Attributmengen  $B_n$  mit  $B_0 := \text{Ker}(C, \{d\})$  und  $B_{n+1} := B_n \cup \{c\}$  für  $c \notin B_n$  finden, in der der Reduktapproximationsfehler immer kleiner wird, und schließlich 0 erreicht.

$$1 \geq \varepsilon_C^{\{d\}}(B_0) > \varepsilon_C^{\{d\}}(B_1) > \dots > \varepsilon_C^{\{d\}}(B_k) = 0$$

Dann ist  $B_k$  ein geeignetes Primredukt.

Betrachten wir das Attributauswahlproblem von der anderen Seite. Dazu nehmen wir die Menge  $C$  aller Entscheidungen und reduzieren diese schrittweise. Dazu benötigen wir den Begriff der Attributsignifikanz, der die "Wichtigkeit," von Attributen angibt, indem der Effekt des Attributentfernens gemessen wird. Die *Attributsignifikanz* einer Bedingung  $c \in$

$C$  eines Entscheidungssystems  $\langle U, C, \{d\} \rangle$  ist definiert als Wert  $\sigma_C^{\{d\}}(\{c\})$  der Abbildung

$$\sigma_C^{\{d\}}: \begin{array}{l} \wp(C) \rightarrow \mathbb{R} \\ B \mapsto \frac{\gamma(C, \{d\}) - \gamma(C - B, \{d\})}{\gamma(C, \{d\})} = 1 - \frac{\gamma(C - B, \{d\})}{\gamma(C, \{d\})}. \end{array}$$

Der Koeffizient  $\sigma_C^{\{d\}}(B)$  gibt den Klassifikationsfehler an, der nach dem Entfernen der Bedingungen aus  $C$  auftritt. Der Wert  $\sigma_C^{\{d\}}(B)$  liegt im Intervall  $[0, 1]$  und ist 0, wenn kein Fehler auftritt.

Nun können wir eine endliche Folge  $(B_n)_{n=0}^k$  maximaler Länge von Attributmengen  $B_n$  mit  $B_0 := C$  und  $B_{n+1} := B_n - \{c\}$  für  $c \in B_n$  finden, in der die Attributsignifikanz der entfernten Bedingungen stets 0 ist.

$$\sigma_{B_0}^{\{d\}}(B_0 - B_1) = \sigma_{B_1}^{\{d\}}(B_1 - B_2) = \dots = \sigma_{B_{k-1}}^{\{d\}}(B_{k-1} - B_k) = 0$$

Dann ist  $B_k$  ein geeignetes Primredukt.

### 3.5 Wertemengenreduktion

In einem Entscheidungssystem mit einer großen Anzahl an Werten für die einzelnen Attribute wird es selten der Fall sein, dass die Signatur eines neuen Objekts mit der Signatur eines vorhandenen Objekts genau übereinstimmt, d.h. die beiden Objekte als ununterscheidbar erkannt werden. Um also ein gutes Klassifikationsvermögen zu erreichen, ist es nötig die Anzahl der möglichen Werten für die einzelnen Attribute zu verringern. Dieses Problem nennen wir auch das *Wertemengenreduktionsproblem* (engl. *value set reduction problem*). Es gibt mindestens zwei Methoden zur Lösung dieses Problems. Einerseits kann für reellwertige Attribute eine Diskretisierung der Wertemenge vorgenommen werden, d.h. einem Intervalle wird eine endliche Menge mit Zahlen aus dem Intervall zugeordnet. Das Verfahren wird hier nicht vorgestellt; der interessierte Leser findet es in [3]. Andererseits ist es auch für Attribute mit Symbolen als Werte möglich, die Mächtigkeit der Wertemengen zu verringern. Die *Gruppierung von symbolischen Attributwerten* (engl. *symbolic attribute value grouping*) ist ein geeignetes Verfahren und soll nun vorgestellt werden.

Sei  $\langle U, C, \{d\} \rangle$  ein Entscheidungssystem. Für ein Attribut  $c \in C$  heißt eine Funktion  $g_c: V_c \rightarrow \{1, \dots, m\}$  mit  $m \leq |V_c|$  *Gruppierungsfunktion* (engl.

clustering function) für  $V_c$ . Der Rang (engl. *rank*) von  $g_c$  ist definiert als  $\text{rang}(g_c) := |g_c(V_c)|$ . Der Rang von  $g_c$  ist die Anzahl der verschiedenen Werte von  $g_c$ , also insbesondere höchstens  $m$ . Für eine Menge  $B \subseteq C$  heißt eine Menge  $\{g_c \mid c \in B\}$  von Gruppierungsfunktionen  $B$ -konsistent, wenn

$$(\forall c \in B: g_c(c(x)) = g_c(c(y))) \implies (x \approx_B y \vee x \approx_{\{d\}} y)$$

gilt. Eine Menge von Gruppierungsfunktionen ist also  $B$ -konsistent, wenn je zwei Objekte, die bezüglich der Werte der Gruppierungsfunktionen für alle Attribute aus  $B$  ununterscheidbar sind, auch bezüglich den Attributen aus  $B$  oder der Entscheidung  $d$  ununterscheidbar sind.

Nun betrachten wir das (engl. *symbolic value partition grouping problem*): Für ein gegebenes Entscheidungssystem  $\langle U, C, \{d\} \rangle$  und eine Menge  $B \subseteq C$  suchen wir eine  $B$ -konsistente Menge  $\{g_c \mid c \in B\}$  von Gruppierungsfunktionen  $g_c$ , sodass die Summe der Ränge  $\sum_{c \in B} \text{rang}(g_c)$  minimal ist. Um dieses Problem zu lösen, sind die folgenden Schritte möglich.

- (1) Definiere eine Menge von booleschen Variablen

$$S := \{c_v^w \mid c \in B \wedge v, w \in V_c \wedge v <_c w\}$$

Dabei ist  $<_c$  eine beliebige lineare Ordnung auf der entsprechenden Wertemenge  $V_c$ .

- (2) Konstruiere eine Matrix  $M := (M_{xy})_{x,y \in U}$  mit

$$M_{xy} := \{c_v^w \in S \mid v = c(x) \wedge w = c(y) \wedge d(x) \neq d(y)\}$$

- (3) Konstruiere aus der Matrix  $M$  die boolesche Funktion

$$\bigwedge_{\substack{x,y \in U \\ M_{xy} \neq \emptyset}} \bigvee_{c_v^w \in M_{xy}} c_v^w.$$

Falls die Entscheidung  $d$  nur zwei Werte hat, die angenommen werden, dann lässt sich die Matrix auch reduziert darstellen, indem die Objekte, die auf den einen Wert abgebildet werden, als Zeilen gewählt werden, und die Objekte, die den anderen Wert haben, wählt man entsprechend als Spalten.

- (4) Bestimme den kürzesten Primimplikanten  $\beta'$ .

- (5) Konstruiere für jedes Attribut  $c \in B$  den ungerichteten Graphen  $\Gamma_c := \langle V_c^\Gamma, E_c^\Gamma \rangle$  mit der Knotenmenge  $V_c^\Gamma := V_c$  und der Kantenmenge

$$E_c^\Gamma := \{(v, w) \mid c_w^v \in \text{im}\beta\}.$$

- (6) Suche eine minimale Knotenfärbung von  $\Gamma_c$ . Die Färbung definiert eine Partition von  $V_c^\Gamma$ , indem die Knoten entsprechend ihrer Farbe in Klassen eingeteilt werden. Den Klassen der Partition werden aufeinanderfolgende natürliche Zahlen  $i$  zugeordnet. Die Gruppierungsfunktion für das Attribut  $c \in B$  ergibt sich nun vermöge  $g_c(v) = i$ , wenn  $v$  in der  $i$ -ten Klasse der Partition von  $V_c$  liegt.

### Definition 3.9 (Reduktion)

Jede Menge  $g = \{g_c \mid c \in B\}$  von Gruppierungsfunktionen erzeugt ein reduziertes Entscheidungssystem  $\langle U, C^g, \{d\} \rangle$  mit  $C^g := \{c^g \mid c \in B\}$  und  $c^g(x) := g_c(c(x))$  für  $x \in U$ . Wir nennen  $\langle U, C^g, \{d\} \rangle$  die *g-Reduktion* von  $\langle U, C, \{d\} \rangle$ .

**Beispiel:** Betrachten wir nocheinmal das Entscheidungssystem aus dem vorigen Beispiel. Es ergeben sich folgende Schritte:

- (1) Wir haben die booleschen Variablen

$$S = \{\text{Motor}_{\text{Otto}}^{\text{Diesel}}, \text{Farbe}_{\text{grün}}^{\text{gelb}}, \text{Farbe}_{\text{grün}}^{\text{blau}}, \text{Farbe}_{\text{gelb}}^{\text{blau}}, \text{Türen}_{\text{zwei}}^{\text{vier}}\}.$$

- (2) Damit ergibt sich die Matrix

	Auto1	Auto2	Auto3
Auto1	$\emptyset$	$\emptyset$	$\emptyset$
Auto2	$\emptyset$	$\emptyset$	$\emptyset$
Auto3	$\{\text{Motor}_{\text{Otto}}^{\text{Diesel}}, \text{Farbe}_{\text{grün}}^{\text{blau}}\}$	$\{\text{Motor}_{\text{Otto}}^{\text{Diesel}}, \text{Farbe}_{\text{gelb}}^{\text{blau}}, \text{Türen}_{\text{zwei}}^{\text{vier}}\}$	$\emptyset$

und diese können wir reduzieren zu

	Auto1	Auto2
Auto3	$\{\text{Motor}_{\text{Otto}}^{\text{Diesel}}, \text{Farbe}_{\text{grün}}^{\text{blau}}\}$	$\{\text{Motor}_{\text{Otto}}^{\text{Diesel}}, \text{Farbe}_{\text{gelb}}^{\text{blau}}, \text{Türen}_{\text{zwei}}^{\text{vier}}\}$

(3) Aus der Matrix lesen wir die boolesche Funktion ab:

$$(\text{Motor}_{\text{Otto}}^{\text{Diesel}} \vee \text{Farbe}_{\text{grün}}^{\text{blau}}) \wedge (\text{Motor}_{\text{Otto}}^{\text{Diesel}} \vee \text{Farbe}_{\text{gelb}}^{\text{blau}} \vee \text{Türen}_{\text{zwei}}^{\text{vier}})$$

(4) Diese hat den kürzesten Primimplikanten:

$$\text{Motor}_{\text{Otto}}^{\text{Diesel}}$$

(5) Damit haben wir für das Attribut Motor den Graphen



und es ergeben sich  $g_{\text{Motor}}(\text{Otto}) = 1$  und  $g_{\text{Motor}}(\text{Diesel}) = 2$ . Die übrigen Graphen sind kantenfrei, also ist  $g_{\text{Farbe}} = 1$  und  $g_{\text{Türen}} = 1$ .

(6) Die g-Reduktion ergibt sich nun zu:

	Motor <sup>g</sup>	Farbe <sup>g</sup>	Türen <sup>g</sup>	Fahren
Auto1	1	1	1	nein
Auto2	1	1	1	nein
Auto3	2	1	1	ja

□

**Beispiel:** Gegeben sei folgendes Entscheidungssystem

$\langle U, C, \{d\} \rangle$	a	b	d
$x_1$	♥	△	0
$x_2$	♠	△	0
$x_3$	♣	▽	1
$x_4$	◇	△	1
$x_5$	◇	▽	1

mit den linearen Ordnungen  $\diamond < \heartsuit < \spadesuit < \clubsuit$  und  $\triangle < \nabla$ .

(1) Wir führen die booleschen Variablen ein:

$$S = \{a_{\diamond}^{\heartsuit}, a_{\diamond}^{\spadesuit}, a_{\diamond}^{\clubsuit}, a_{\heartsuit}^{\spadesuit}, a_{\heartsuit}^{\clubsuit}, a_{\spadesuit}^{\clubsuit}, a_{\clubsuit}^{\nabla}, b_{\triangle}^{\nabla}\}$$

(2) Die Matrix  $M$  ergibt sich zu

$M$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$x_1$	$\emptyset$	$\emptyset$	$\{a_{\heartsuit}, b_{\Delta}^{\nabla}\}$	$\emptyset$	$\{b_{\Delta}^{\nabla}\}$
$x_2$	$\emptyset$	$\emptyset$	$\{a_{\clubsuit}, b_{\Delta}^{\nabla}\}$	$\emptyset$	$\{b_{\Delta}^{\nabla}\}$
$x_3$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$x_4$	$\{a_{\heartsuit}^{\diamond}\}$	$\{a_{\spadesuit}^{\diamond}\}$	$\emptyset$	$\emptyset$	$\emptyset$
$x_5$	$\{a_{\heartsuit}^{\diamond}\}$	$\{a_{\spadesuit}^{\diamond}\}$	$\emptyset$	$\emptyset$	$\emptyset$

Ändern wir nun die lineare Ordnung der Werte von  $a$  zu

$$\heartsuit < \spadesuit < \clubsuit < \diamond$$

und entsprechend die booleschen Variablen zu

$$S = \{a_{\heartsuit}^{\spadesuit}, a_{\heartsuit}^{\clubsuit}, a_{\heartsuit}^{\diamond}, a_{\clubsuit}^{\spadesuit}, a_{\clubsuit}^{\diamond}, a_{\spadesuit}^{\diamond}, b_{\Delta}^{\nabla}\},$$

dann können wir die Matrix reduzieren zu

$M$	$x_3$	$x_4$	$x_5$
$x_1$	$\{a_{\heartsuit}^{\spadesuit}, b_{\Delta}^{\nabla}\}$	$\{a_{\heartsuit}^{\diamond}\}$	$\{a_{\heartsuit}^{\diamond}, b_{\Delta}^{\nabla}\}$
$x_2$	$\{a_{\clubsuit}^{\spadesuit}, b_{\Delta}^{\nabla}\}$	$\{a_{\spadesuit}^{\diamond}\}$	$\{a_{\spadesuit}^{\diamond}, b_{\Delta}^{\nabla}\}$

(3) Damit ergibt sich die boolesche Funktion

$$(a_{\heartsuit}^{\spadesuit} \vee b_{\Delta}^{\nabla}) \wedge a_{\heartsuit}^{\diamond} \wedge (a_{\heartsuit}^{\diamond} \vee b_{\Delta}^{\nabla}) \wedge (a_{\clubsuit}^{\spadesuit} \vee b_{\Delta}^{\nabla}) \wedge a_{\spadesuit}^{\diamond} \wedge (a_{\spadesuit}^{\diamond} \vee b_{\Delta}^{\nabla}).$$

(4) Diese hat den kürzesten Primimplikanten

$$a_{\heartsuit}^{\diamond} \wedge a_{\spadesuit}^{\diamond} \wedge b_{\Delta}^{\nabla}.$$

(5) Es ergeben sich folgende Graphen



und damit ergeben sich die Gruppierungsfunktionen  $g_a(\diamond) = 1$  und  $g_a(\heartsuit) = g_a(\spadesuit) = g_a(\clubsuit) = 2$  sowie  $g_b(\Delta) = 1$  und  $g_b(\nabla) = 2$ .

(6) Die  $g$ -Reduktion ist also

$\langle U, C, \{d\} \rangle$	$a^g$	$b^g$	$d$
$x_1$	2	1	0
$x_2$	2	1	0
$x_3$	2	2	1
$x_4$	1	1	1
$x_5$	1	2	1

□

### 3.6 Primentscheidungsregeln

#### Definition 3.10 (Primentscheidungsregel)

Eine Entscheidungsregel  $\varphi \Rightarrow \psi$  heißt *Primentscheidungsregel* (engl. *minimal decision rule*), wenn sie wahr ist und jedes Entfernen eines Deskriptors in  $\varphi$  bewirkt, dass sie falsch wird.

Eine minimale Entscheidungsregel hat als Vorgänger eine minimale Konjunktion von Deskriptoren und diese sind unbedingt nötig, um den Wert der Entscheidung  $d$  eindeutig und korrekt zu bestimmen.

(1) Definiere für jedes Objekt  $x \in U$  aus der entsprechenden Zeile in der entscheidungsabhängigen Matrix  $\text{Mat}(C, D)$  die boolesche Funktion

$$f_x((c')_{c \in C}) := \bigwedge_{\substack{x \neq_C y \\ x \neq_D y}} \bigvee_{c \in C_{xy}^D} c'.$$

(2) Bestimme für jede Funktion  $f_x$  die Primimplikanten.

(3) Konstruiere für jeden Primimplikanten  $\beta'$  die Entscheidungsregel

$$\left( \bigwedge_{a \in \text{im} \beta'} a = a(x) \right) \Rightarrow (d = d(x)).$$

# Literaturverzeichnis

- [1] Zdzislaw Pawlak and Andrzej Skowron. Rudiments of rough sets. *Institute of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland, Information Sciences 177 (2007):3–27, 2006.*
- [2] Zdzislaw Pawlak and Andrzej Skowron. Rough sets: Some extensions. *Institute of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland, Information Sciences 177 (2007):28–40, 2006.*
- [3] Zdzislaw Pawlak and Andrzej Skowron. Rough sets and boolean reasoning. *Institute of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland, Information Sciences 177 (2007):41–73, 2006.*
- [4] Bernhard Ganter. Lattices of rough set abstractions as p-products. *Institut für Algebra, Dresden University of Technology, D-01062 Dresden, 2007.*





# Index

- Abhängigkeit, 19
- allgemeiner Approximationsraum,  
7
- Approximation, 1
  - obere, 1
  - untere, 1
- Approximationsgenauigkeit, 5
- Approximationsraum, 1
- Attribut, 9
- Attributauswahlproblem, 22
  
- Bedingung, 17
- Bedingungsformel, 18
- Bedingungsklasse, 18
  
- definierbare Menge, 1
- Deskriptor, 17
- Determiniertheit, 18
  
- elementare Menge, 1
- Entscheidung, 17
- Entscheidungsformel, 18
- Entscheidungsklasse, 18
- Entscheidungsregel, 18
- Entscheidungssystem, 17
- exakte Menge, 5
  
- Formel, 17
  
- Gruppierungsfunktion, 23
  
- Implikant, 12
- Informationssystem, 9
- Inklusionsfunktion, 7
  
- Kern, 10, 20
- Konsistenz, 18, 24
  
- mögliches Objekt, 2
  
- Nachbarschaft, 7
- Nachfolger, 18
  
- oben undefinierbare Menge, 5
- obere Approximation, 1
- Objekt, 1, 9
  
- partielle Abhängigkeit, 19
- Primentscheidungsregel, 28
- Primimplikant, 12
- Primredukt, 10
  
- Rand, 1
- Rang, 24
- Redukt, 10, 20
- Reduktion, 15, 25
  
- Selektor, 17
- sicheres Objekt, 2
- Signatur, 9
  
- undefinierbare Menge, 5
- unexakt definierbare Menge, 5
- unexakte Enthaltenseinsfunktion,  
5
- unexakte Menge, 5
- Universum, 1, 9
- unten undefinierbare Menge, 5
- untere Approximation, 1

Unterscheidbarkeitsfunktion, 10

Unterscheidbarkeitsmatrix, 10

ununterscheidbar, 1

Ununterscheidbarkeitsrelation, 9

Vorgänger, 18

wahr, 18

Wert, 9

Wertemengenreduktionsproblem,

23