

Boolesch Algebraische Generalisierung / Spezialisierung von Objekttypen (BAGS-Modellierung)

Gisbert Englmeyer
Dipl.-Betriebswirt (FH)
Löserweg 7
74722 Buchen
Telefon/Fax: 06281/2435

Sanz

- 0 Einführung
- 1 Beschreibung der Vorgehensweise
 - 1.1 Attribute und ihre Interpretation als Boolesche Variable
 - 1.2 Zweckorientierte Datensicht
 - 1.3 Konjunktive Darstellung einer zweckorientierten Datensicht
 - 1.4 Disjunktive Darstellung zweckorientierter Datensichten
 - 1.5 Graphische Interpretation der Boolesch algebraischen Klammersausdrücke
 - 1.6 Absorptionsfall (Verschmelzungsfall)
 - 1.7 Sich gegenseitig ausschließende Attribute (Disjunktion)
 - 1.8 Alternative graphische Darstellung einer Konjunktion
- 2 Hauptgeneralisierung / -spezialisierung
 - 2.1 Bestimmung der Wurzelobjekttypen bei Generalisierungen / Spezialisierungen
 - 2.2 Vernetzte Generalisierungs- / Spezialisierungsstrukturen (Mehrfachvererbung)
 - 2.3 Elementare Attribute (Variable)
 - 2.4 Mehrere gültige Lösungen bei Generalisierungen / Spezialisierungen
- 3 Instanzierungstypen (in Vorbereitung)
- 4 Systematisierte und integrierte Datenmodell- und Instanzenwarnung (in Vorbereitung)
- 5 Nebengeneralisierung / -spezialisierung (in Vorbereitung)

stand überführt. Darüber hinaus könnte es notwendig sein, den alten Zustand neben dem neuen gleichzeitig im Datenbanksystem operabel zu halten und durch versionierte Prozesskomponenten mit entsprechenden Dateninhalten unterschiedlich zu bearbeiten.

Die mit dieser Methode entwickelten referentiellen Integritätsbedingungen zwischen den generalisierten / spezialisierten Objekttypen können mit Hilfe eines aktiven Data Dictionary Systems dynamisch während der Ausführungszeit überwacht werden. Ebenso ist es möglich, den Konstruktionsprozess des Datenmodells zu automatisieren. Es sind erhebliche Fortschritte in der Produktivität bei der Entwicklung von Datenmodellen und in der Erhaltung der Integrität einer Datenbank während ihres Betriebes zu erwarten.

1 Beschreibung der Vorgehensweise
In dieser Vorgehensweise ist der organisatorisch existentielle Aspekt von Attributen [ScNe90] und deren Zuordnung zu zweckorientierten Datensichten von Interesse. In einer zweckorientierten Datensicht werden Attribute zusammengefaßt, die durch einen detaillierten Geschäftsvorgang im Zusammenhang benötigt werden. Attributeinhalte (Wertebereiche) werden nur am Rande behandelt. Nachdem die zweckorientierten Datensichten in die disjunktive Normalform [Bar77] [InDu89] überführt wurden, können sie durch Anwendung des Distributivgesetzes generalisiert / spezialisiert werden. Daraus werden die Generalisierungen / Spezialisierungen durch Anwendung des Distributivgesetzes gewonnen. Die sich ergebenden Klammersausdrücke können graphisch in einer gewünschten Notation interpretiert werden. Sie sichern die referentiellen Integritäten zwischen den einzelnen Objekttypen präzise ab.

Ziel ist es, die Ordnungsstruktur der Attribute unter Beibehaltung der Semantik der zweckorientierten Datensichten so zu gestalten, daß jedes Attribut in den generalisierten / spezialisierten Objekttypen [ScNe90] nur einmal vorkommt. Vom Umfang her werden die Attribute, je nach der Abgrenzung des zu untersuchenden Systems, global (z.B. unternehmensweit) oder nur lokal (z.B. bereichsweit) bearbeitet.

Diese Beschreibung behandelt in den praktischen Beispielen zweckorientierte Datensichten nur bereichsweit. Die Thematik wird von der pragmatischen Seite her aufgearbeitet. Insoweit mathematische Konzepte der Booleschen Algebra zweckdienlich sind, werden sie problemadäquat interpretiert und in das Konzept der Problemlösung eingebaut. Die Vorgehensweise wird als „Boolesch Algebraische Generalisierung / Spezialisierung von Objekttypen“ bezeichnet („BAGS“-Modellierung).

Diese Art der Vorgehensweise ist nicht nur auf Objekttypen beschränkt. So können betriebliche Prozesse in gleicher Weise strukturiert werden, um z.B. funktionale Redundanz zu vermeiden. Ebenso können Ordnungssysteme in anderen Bereichen mit Hilfe dieser Vorgehensweise eindeutig strukturiert werden. Für alle Instanzen [ScNe90] (Ausprägungen von Objekttypen, Exemplare) der generalisierten / spezialisierten Objekttypen empfiehlt es sich, Eindeutigkeit durch Benutzung entsprechender Schlüsselssysteme herzustellen.

1.1 Attribute und ihre Interpretation als Boolesche Variable
In dieser Vorgehensweise wird ein Attribut als eine Boolesche Variable aufgefaßt, die den Wert „0“ oder „1“ annehmen kann. Der Wert „1“ bedeutet, daß ein Attribut in die Generalisierung / Spezialisierung einbezogen wird. Ein nicht in die Generalisierung / Spezialisierung einbezogenes Attribut ist mit dem Wert „0“ belegt. Es existiert damit nicht im zu untersuchenden System und wird deshalb weder erwähnt noch in die Bearbeitung einbezogen. Alle zu einem zu untersuchenden System gehörenden Attribute existieren somit explizit berechtigt. Sie sind mit dem Wert „1“ (existent) belegt.

Ein Attribut ist systemweit eindeutig festzulegen. Die Anzahl n der Attribute eines Systems ist endlich. Damit eine kurze aber dennoch aussagefähige Notation der Attribute vorgenommen werden kann, wird in dieser Abhandlung ein Attribut (Boolesche Variable) neben seinem jeweiligen Attributnamen auch als Variable a_i (mit i = 1, 2, ..., n) benannt.

Folgende Tabelle zeigt ein abstraktes Beispiel mit Attributen:

0 Einführung
Generalisierungen / Spezialisierungen gewinnen künftig immer mehr an Bedeutung. Die heute aktuelle objektorientierte Vorgehensweise in der Anwendungssystementwicklung verlangt korrekte Klassenhierarchien zur Vererbung von Attributen und Prozessen. Aber auch in herkömmlichen Methoden ist eine saubere Generalisierung / Spezialisierung Voraussetzung für qualitativ hochwertige Anwendungssysteme.

Die hier beschriebene Vorgehensweise ermöglicht es, Generalisierungen / Spezialisierungen formal und jederzeit reproduzierbar zu „berechnen“. Die Attribute von zu generalisierenden / spezialisierenden Objekttypen werden mit Operationen der Booleschen Algebra bearbeitet. Bestimmte Elemente der Booleschen Algebra werden problemadäquat interpretiert. Ausgangsbasis für die Vorgehensweise ist ein Datenmodell, das vorgegangsbegleitende zweckorientierte Datensichten enthält.

Je nach Problemstruktur erhält man mehrere Lösungen, die aber alle Boolesch algebraisch korrekt sind und damit in das Datenmodell als Generalisierungen / Spezialisierungen übernommen werden können. Falls man mehrere Lösungen erhält, können verschiedene Aspekte zur Bestimmung optimaler Strukturen herangezogen werden. Während der Hauptgeneralisierung / -spezialisierung entsteht eine streng, baumartige Hierarchie. Erfordert die Problemstellung eine Netzstruktur, wird diese in der Nebengeneralisierung / -spezialisierung ermittelbar.

Ein wesentliches Merkmal der Vorgehensweise ist ein geregeltes Warnungskonzept für generalisierte- / spezialisierte Datenmodelle. Das Datenmodell „altert“ damit nicht durch laufende Änderungen oder Anpassungen. Es sind keine „Notlösungen“ in Form von „Override“-Attributen notwendig. Vorliegende Instanzen [ScNe90] (Ausprägungen, Exemplare) werden, wie die Modellierungskomponenten selbst, vom alten in den neuen Zu-

LfdNr	Attribute (Boolesche Variable)	Attribute (abgekürzt)	Wertzuweisung
1	Attribut a	a ₁	=1
2	Attribut b	a ₂	=1
3	Attribut c	a ₃	=1
4	Attribut d	a ₄	=1
5	Attribut e	a ₅	=1

Tabelle 1: Systemweite Attributabelle (Beispiel mit abstrakten Attributen)

Alle Booleschen Variablen (Attribute), die ein zu untersuchendes System repräsentieren, bilden eine Menge M. Sie sind untereinander konjunktiv (mit „Logisch - Und“, Logische Multiplikation) zu verknüpfen. Diesem Booleschen algebraischen Produkt kann der Funktionswert „1“ zugewiesen werden, falls alle Booleschen Variablen den Wert „1“ besitzen, d.h. daß sie existieren. Dies bedeutet auch, daß für jedes Attribut ein gültiger Wertebereich existiert. Jeder Wert des Wertebereichs muß genau der Semantik des Attributes genügen. So dürfen z.B. beim Attribut „Steuerklasse eines Arbeitnehmers“ nur die tatsächlichen Steuerklassen 1 bis 6 vorkommen. Die Werte „leer“, „blank“, „keine Steuerklasse“ oder „0“ dürfen z.B. nicht im Wertebereich existieren.

Es gilt somit für die Attribute aus Tabelle 1:

$$\text{Attribut a} \wedge \text{Attribut b} \wedge \text{Attribut c} \wedge \text{Attribut d} \wedge \text{Attribut e} = 1$$

$$a_1 \wedge a_2 \wedge a_3 \wedge a_4 \wedge a_5 = 1$$

oder als „Logische Multiplikation“

$$a_1 * a_2 * a_3 * a_4 * a_5 = 1$$

oder kürzer als „Logische Multiplikation“

$$a_1 a_2 a_3 a_4 a_5 = 1$$

mit $a_1 = a_2 = a_3 = a_4 = a_5 = 1$.

Bemerkung: Für die folgenden „Logisch - Und“ Ausdrücke wird, von Ausnahmen abgesehen, die Form, wie sie in (1.4) gezeigt wird, verwendet.

Konkretes Beispiel mit Attributen:

Lfd Nr.	Attribute (Boolesche Variable)	Attribute (abgekürzt)	Wertzuweisung
1	KundenNr	a ₁	=1
2	Name	a ₂	=1
3	Anschrift	a ₃	=1
4	Bonität	a ₄	=1
5	Kreditgrenze	a ₅	=1
6	LieferantNr	a ₆	=1
7	PersonalNr	a ₇	=1
8	Steuerklasse	a ₈	=1
9	Finanzamt	a ₉	=1
10	Abteilung	a ₁₀	=1
11	Stundensatz	a ₁₁	=1
12	Lohngruppe	a ₁₂	=1

Tabelle 2: Systemweite Attributabelle (Beispiel mit konkreten Attributen)

Die Attribute aus Tabelle 2 sind untereinander „Logisch Und“ zu verknüpfen:

$$\text{KundenNr} \wedge \text{Name} \wedge \text{Anschrift} \wedge \text{Bonität} \wedge \text{Kreditgrenze} \wedge \text{LieferantNr} \wedge \text{PersonalNr} \wedge \text{Steuerklasse} \wedge \text{Finanzamt} \wedge \text{Abteilung} \wedge \text{Stundensatz} \wedge \text{Lohngruppe} = 1 \quad (2.1)$$

oder verkürzt

$$a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} a_{11} a_{12} = 1 \quad (2.2)$$

oder

$$\bigwedge_{i=1}^{12} a_i = \prod_{i=1}^{12} a_i = 1 \quad (2.3)$$

Für alle n Attribute a_i eines zu untersuchenden Systems gilt:

$$\bigwedge_{i=1}^n a_i = \prod_{i=1}^n a_i = 1 \quad (2.4)$$

Alle im System benutzten Attribute müssen ihrer Semantik entsprechend vorkommen und vorgesehen sein. Dies sagt noch nichts über tatsächlich vorkommende Instanzen mit den benutzten Attributen aus.

1.2 Zweckorientierte Datensicht

Für verschiedene betriebliche Aspekte oder Vorgänge werden Attribute des zu untersuchenden Systems zu zweckorientierten Datensichten zusammengefaßt und in Zuordnungsmatrizen als Attributteilungen der Menge M ausgewiesen. Ein „+“-Zeichen in einem Matrixelement einer Zuordnungsmatrix von Attributen zu zweckorientierten Datensichten bedeutet, daß das entsprechende Attribut zur jeweiligen zweckorientierten Datensicht eines Bearbeitungsvorgangs gehört bzw. einem betrieblichen Aspekt entsprechend zugeordnet ist. Ist ein Matrixelement leer, gehört das in dieser Zeile referenzierte Attribut nicht zu der in dieser Spalte referenzierten zweckorientierten Datensicht.

Abstraktes Beispiel:

LfdNr	Attribute (Boolesche Variable)	Attribute (abgekürzt)	Zweckorientierte Datensicht S ₁	Zweckorientierte Datensicht S ₂
1	Attribut a	a ₁	+	+
2	Attribut b	a ₂	+	+
3	Attribut c	a ₃	+	+
4	Attribut d	a ₄	+	+
5	Attribut e	a ₅		+

Tabelle 3: Tabelle der zweckorientierten Datensichten A und B mit Attributzuzuweisung

Ein konkretes Beispiel mit den zweckorientierten Datensichten Kunde, Lieferant und Gewerblicher Arbeitnehmer und den ausgewählten Attributen Kundennummer, Name, Anschrift, Bonität, Kreditgrenze, Lieferantnummer, Personalnummer, Steuerklasse, Finanzamt, Abteilung, Stundensatz für gewerbliche Arbeitnehmer und Lohngruppe für gewerbliche Arbeitnehmer zeigt die folgende Tabelle.

Die Zuordnung der Attribute ist in den Spalten drei bis fünf durch den Eintrag eines „+“-Zeichens gekennzeichnet:

Verarbeitungsprozessen, um die tatsächliche Datensicht während des Prozessablaufs dynamisch zu erzeugen. Aus den dynamischen Verarbeitungsprozessen wird die intensive Datenlektionsverarbeitung herausgenommen und in das statische Datenmodell verlagert. Der bei solchen intensiven Prozessen notwendige Testaufwand wird damit erheblich reduziert.

Eine zweckorientierte Datensicht ist entweder vollständig (mit allen Attributen belegt) zu verwenden oder sie existiert nicht.

Für eine zweckorientierte Datensicht S_k gilt:

$$S_k = \bigwedge_{i \in I_k} a_i = 1 \quad (4.7)$$

- k : Laufindex der zweckorientierten Datensichten
- I_k : Menge der in einer zweckorientierten Datensicht S_k benötigten Indizes i der Attribute a
- i : Systemweit eindeutiger Index für jedes Attribut a
- a_i : Attribut a , eindeutig identifiziert durch Index i

Die maximale Anzahl p der zweckorientierten Datensichten S_k ergibt sich aus der Potenzmenge der Menge M aller Attribute ohne die leere Menge und ohne die Menge M selbst. Die Attribute a_i jeder zweckorientierten Datensicht bilden eine echte Teilmenge der Menge M aller Attribute.

Entspricht ein Attribut in einer Instanz einmal nicht der bedachten Semantik, dann ist die entsprechende Boolesche Variable 0. Damit wird der gesamte Ausdruck 0, d.h. für diese Instanz gibt es eigentlich keine vorgesehene Datensicht. Durch diese strenge Vorschrift werden die in verschiedenen Methoden / Vorgehensweisen vorkommenden sogenannten optionalen Attribute [Hohe93] nicht zugelassen. Falls solche optionalen Attribute in einer Datensicht vorkommen, handelt es sich im Sinne dieser Vorgehensweise um eine nicht korrekte (unvollständige) Abbildung der Realität.

Folgendes Beispiel soll diesen Sachverhalt verdeutlichen: In einer konkreten Instanz (Ausprägung) fehlen bei der Datensicht „Gewerblicher Arbeitnehmer“ die beiden Werte für Steuerklasse und Finanzamt, weil sie beim vorliegenden Mitarbeiter nicht relevant sind. Alle anderen Merkmale stimmen mit der Datensicht „Gewerblicher Arbeitnehmer“ überein. Damit wären für diesen Fall die beiden Booleschen Variablen (aus Tabelle 4) a_7 und $a_8 = 0$. Für diese Instanz gibt es somit keine definierte zweckorientierte Datensicht, da die obige Ausdruck (4.3) bzw. (4.6) zu Null wird.

Lfd. Nr.	Attribute (Boolesche Variable)	Attribute (abgekürzt)	Zweckorientierte Datensichten			
			Kunde	Lieferant	Gewerblicher Arbeitnehmer	Geringfügig Beschäftigte
	1	2	S_1	S_2	S_3	S_4
1	KundenNr	a_1	3	4	5	6
2	Name	a_2	+	+	+	+
3	Anschrift	a_3	+	+	+	+
4	Bonität	a_4	+	+	+	+
5	Kreditgrenze	a_5	+	+	+	+
6	LieferantNr	a_6				
7	PersonalNr	a_7			+	+
8	Steuerklasse	a_8			+	+
9	Finanzamt	a_9			+	+
10	Abteilung	a_{10}			+	+
11	Stundensatz	a_{11}			+	+
12	Lohngruppe	a_{12}			+	+

Tabelle 5: Attributabelle mit den zweckorientierten Datensichten „Kunde“, „Lieferant“, „Gewerblicher Arbeitnehmer“ und „Geringfügig Beschäftigter“

Lfd. Nr.	Attribute (Boolesche Variable)	Attribute (abgekürzt)	Zweckorientierte Datensichten			
			Kunde	Lieferant	Gewerblicher Arbeitnehmer	Geringfügig Beschäftigte
	1	2	S_1	S_2	S_3	S_4
1	KundenNr	a_1	3	4	5	6
2	Name	a_2	+	+	+	+
3	Anschrift	a_3	+	+	+	+
4	Bonität	a_4	+	+	+	+
5	Kreditgrenze	a_5	+	+	+	+
6	LieferantNr	a_6				
7	PersonalNr	a_7			+	+
8	Steuerklasse	a_8			+	+
9	Finanzamt	a_9			+	+
10	Abteilung	a_{10}			+	+
11	Stundensatz	a_{11}			+	+
12	Lohngruppe	a_{12}			+	+

Tabelle 4: Attributabelle mit den zweckorientierten Datensichten „Kunde“, „Lieferant“ und „Gewerblicher Arbeitnehmer“

1.3 Konjunktive Darstellung einer zweckorientierten Datensicht

Alle Attribute (Booleschen Variablen), die eine zweckorientierte Datensicht repräsentieren, sind untereinander konjunktiv (mit „Logisch - Und“) zu verknüpfen. Jedem Boolesch algebraischen Produkt kann der Funktionswert „1“ zugewiesen werden, falls alle zu einer zweckorientierten Datensicht gehörenden Booleschen Variablen existieren (den Wert „1“ besitzen).

Abstraktes Beispiel Tabelle 3:

$$S_1 = a_1 a_2 a_3 a_4 = 1 \quad (3.1)$$

$$S_2 = a_1 a_2 a_5 = 1 \quad (3.2)$$

mit $a_1 = a_2 = a_3 = a_4 = a_5 = 1$.

Konkretes Beispiel aus Tabelle 4:

$$S_1 = \text{Kunde} = \text{KundenNr} \wedge \text{Name} \wedge \text{Anschrift} \wedge \text{Bonität} \wedge \text{Kreditgrenze} = 1 \quad (4.1)$$

$$S_2 = \text{Lieferant} = \text{Name} \wedge \text{Anschrift} \wedge \text{LieferantNr} = 1 \quad (4.2)$$

$$S_3 = \text{Gewerblicher Arbeitnehmer} = \text{Name} \wedge \text{Anschrift} \wedge \text{PersonalNr} \wedge \text{Steuerklasse} \wedge \text{Finanzamt} \wedge \text{Abteilung} \wedge \text{Stundensatz} \wedge \text{Lohngruppe} = 1 \quad (4.3)$$

oder verkürzt:

$$S_1 = \text{Kunde} = a_1 a_2 a_3 a_4 a_5 = 1 \quad (4.4)$$

$$S_2 = \text{Lieferant} = a_2 a_3 a_6 = 1 \quad (4.5)$$

$$S_3 = \text{Gewerblicher Arbeitnehmer} = a_2 a_3 a_7 a_8 a_9 a_{10} a_{11} a_{12} = 1 \quad (4.6)$$

$$\text{mit } a_1 = a_2 = a_3 = a_4 = a_5 = a_6 = a_7 = a_8 = a_9 = a_{10} = a_{11} = a_{12} = 1$$

Durch die Erfüllung dieser Gleichungen kommt inhaltlich zum Ausdruck, daß die jeweilige zweckorientierte Datensicht nur dann berechtigt existiert, wenn in ihren vorkommenden Instanzen (Ausprägungen) alle Attribute in jedem Fall immer mit Werten aus ihren jeweiligen Wertebereichen belegt sind. Dabei ist zu beachten, daß die Werte der bedachten Semantik des Attributes entsprechen müssen. Es entfallen somit Fallbehandlungen in

Das Fehlen der beiden Werte kann bei der Anwendung von Operationen (objektorientiert: Methoden, Services) auf diese Attribute zu unvorhergesehenen Fehlern führen. Ursache für diesen Zustand ist ein Konstruktionsfehler der zweckorientierten Datensicht „Gewerblicher Arbeitnehmer“: Sie kann keine Instanzen von Mitarbeitern repräsentieren, die fachlich gesehen kein persönlich zu versteuerndes Einkommen beziehen. Es handelt sich dabei um Mitarbeiter, die mit ihrem Einkommen unter der Mindesteinkommengrenze liegen. Für sie ist vom beschäftigenden Unternehmen eine pauschale Lohnsteuer an das Finanzamt zu entrichten ist. Deshalb ist eine weitere zweckorientierte Datensicht „Geringfügig Beschäftigter“ einzuführen, die genau die Attribute enthält, die bei dieser Datensicht relevant sind. Die Tabelle 5 ist um diese Datensicht erweitert. Damit ist es möglich, „Geringfügig Beschäftigte“ semantisch korrekt im Modell abzubilden.

1.4 Disjunktive Darstellung zweckorientierter Datensichten

Die als Konjunktionen vorliegenden zweckorientierten Datensichten des zu bearbeitenden Systems sind untereinander disjunktiv („Inklusive Logisch Oder“, Logische Addition) zu verknüpfen. Sie können jetzt einer Generalisierung / Spezialisierung unterzogen werden, indem die gemeinsamen als Boolesche Variable vorliegenden Attribute ausgeklammert (thetrasfaktoriert [Mart88]) werden. Durch Zuweisung des Funktionswertes „1“ zur entstandenen Disjunktion erhält man die disjunktive Normalform des Gesamtsystems. Der entstandene Ausdruck ist anschließend mit Hilfe des Distributivgesetzes der Booleschen Algebra zu bearbeiten. Durch das Ausklammern erhält man originär die Generalisierungs- / Spezialisierungsstrukturen:

Abstraktes Beispiel aus (3.1) und (3.2):

$$S_1 \vee S_2 = \quad (5.1)$$

$$a_1 a_2 a_3 a_4 \vee a_1 a_2 a_3 = 1 \quad (5.2)$$

$$a_1 a_2 (a_3 a_4 \vee a_3) = 1 \quad (5.3)$$

Durch Anwendung des Distributivgesetzes auf den obigen Ausdruck (Ausklammern der gemeinsamen Faktoren $a_1 a_2$) erhält man:

$$a_1 a_2 (a_3 a_4 \vee a_3) = 1$$

Damit ist die Generalisierung / Spezialisierung vollzogen. Die beiden Booleschen Variablen außerhalb der Klammer gelten bezüglich der drei Booleschen Variablen innerhalb der Klammer als generalisiert. Die drei Booleschen Variablen innerhalb der Klammer gelten bezüglich der beiden außerhalb der Klammer befindlichen Booleschen Variablen als spezialisiert.

Es findet eine neue Zuweisung der Attribute bzw. der sie repräsentierenden Booleschen Variablen zu den jetzt generalisierten / spezialisierten Objekttypen statt. Aus den zwei zweckorientierten, nicht generalisierten / spezialisierten Datensichten S_1 und S_2 sind die drei generalisierten / spezialisierten Objekttypen A , B und C mit den folgenden Attributzuweisungen entstanden:

$$A = a_1 a_2, B = a_3 a_4 \text{ und } C = a_3 \quad (5.4)$$

Damit gilt gemäß Ausdruck (5.3):

$$A (B \vee C) = 1 \quad (5.5)$$

Konkretes Beispiel nach (4.1) bis (4.6):

$$S_1 \vee S_2 \vee S_3 = \quad (6.0)$$

$$\text{Kunde} \vee \text{Lieferant} \vee \text{Gewerblicher_Arbeitnehmer} = \quad (6.1)$$

$$a_1 a_2 a_3 a_4 a_5 \vee a_1 a_2 a_4 a_6 \vee a_2 a_3 a_7 a_8 a_9 a_{10} a_{11} a_{12} = 1 \quad (6.2)$$

$$\text{Durch Ausklammern von } a_2 a_3 \text{ erhält man folgenden Ausdruck:} \quad (6.3)$$

$$a_2 a_3 (a_1 a_4 a_5 \vee a_6 \vee a_7 a_8 a_9 a_{10} a_{11} a_{12}) = 1$$

Eine Zuweisung der Attribute zu den entstandenen generalisierten / spezialisierten Objekttypen ergibt:

$$\text{Externer Partner} = E = a_2 a_3 \quad (6.4)$$

$$\text{Kunde} = K = a_1 a_4 a_5 \quad (6.5)$$

$$\text{Lieferant} = L = a_6 \quad (6.6)$$

$$\text{Gewerbl. Arbeitnehmer} = G = a_7 a_8 a_9 a_{10} a_{11} a_{12} \quad (6.7)$$

Der Boolesche Ausdruck (6.3) kann damit wie folgt geschrieben werden:

$$E (K \vee L \vee G) = 1 \quad (6.8)$$

Die generalisierten / spezialisierten Objekttypen enthalten jetzt nur noch ihre spezifischen Attribute. Für zusätzlich entstehende Teilterme (Konjunktionen) sind treffende Bezeichnungen zu einzuführen. So ist für die beiden Attribute a_2 und a_3 , die durch den Generalisierungsvorgang als eigener Objekttyp isoliert wurden, eine Objekttypenbezeichnung (z.B. „Externer Partner“) einzuführen. Bei Bedarf können zusätzlich neue, identifizierende Attribute eingeführt werden, z.B. die „Externe Partner“-Nummer a_0 für den Objekttyp E . Dieses Attribut kann einfach den bereits in diesem Objekttyp befindlichen Attributen konjunktiv hinzugefügt werden: $E = a_0 a_2 a_3$. Da es sich beim genannten Beispiel multiplikativ mit allen in den Klammern befindlichen Attributen verknüpft, wird es jedem spezialisierten Objekttyp vererbt.

Durch die Spezialisierungen werden Teilmengen von Attributen gebildet. Jeder Teilmenge (spezialisierter Objekttyp) wird ein Symbol (ein Name) zugeteilt, unter dem sie künftig explizit genannt werden kann. Damit entfällt die ansonsten notwendige dynamische Bildung dieser Teilmengen (Schritt- und/oder Vereinigungsmengen) als dynamische Datensichten während des Bearbeitungsprozesses auf Instanzebene. Die während des Bearbeitungsprozesses dynamisch gebildeten Teilmengen sind anonym (ohne Namen) und nicht dauerhaft (nicht persistent). Der Testaufwand ist erheblich. Durch die Generalisierung / Spezialisierung wird die Komplexität der Bearbeitungsprozesse wesentlich reduziert. Der formale Gehalt wird in das statische Datenmodell verlagert.

Für alle zweckorientierten Datensichten S des untersuchten Systems gilt:

$$S = \bigvee_{k=1}^p S_k = \bigvee_{k=1}^p \bigwedge_{i \in J_k} a_i = 1 \quad (6.9)$$

wobei p die Anzahl aller zweckorientierten Datensichten im untersuchten System ist.

Als Beispiel der Entwicklung von (6.9) sei der Ausdruck (5.2) mit $J_1 = \{1,2,3,4\}$ und $J_2 = \{1,2,5\}$ und $p = 2$ angeführt:

$$S = \bigvee_{k=1}^2 S_k = S_1 \vee S_2 = \bigwedge_{i \in J_1} a_i \vee \bigwedge_{i \in J_2} a_i = (a_1 \wedge a_2 \wedge a_3 \wedge a_4) \vee (a_1 \wedge a_2 \wedge a_5) = 1 \quad (6.10)$$

1.5 Graphische Interpretation der Boolesch algebraischen Klammernaussdrücke

Die Umsetzung eines Klammernaussdrucks in eine graphische Interpretation ist wie folgt: Jede „Klammer auf“ begründet eine Spezialisierung. Die dazu paarige Klammer in der gleichen Ebene beendet die Spezialisierung in dieser Ebene. Alle Booleschen Variablen (stellvertretend für die Attribute) innerhalb einer Klammer gelten bezüglich der Booleschen Variablen außerhalb der Klammer als spezialisiert. Die Variablen außerhalb einer Klammer gelten bezüglich der Booleschen Variablen innerhalb der Klammer als generalisiert.

Die Vorgehensweise ist nicht an eine der bekannten Datenmodellierungsmethoden (z.B. ERM [Sche90], SERM [Fes93], OOA nach Coad and Yourdon [CoYo90]) gebunden. Die Ergebnisse können in jeder Methode interpretiert werden. Durch die präzise Ausformulierung wird dem „semantischen Bedarf“ an Generalisierungen / Spezialisierungen der unterschiedlichen Methoden Rechnung getragen. Inwieweit die jeweiligen syntaktischen Ausdrucksformen der Methoden die von dieser Vorgehensweise gelieferten Inhalte zu transportieren vermögen, ist von Methode zu Methode verschieden.

Die Darstellungen in dieser Beschreibung erfolgen stellvertretend immer in der Notation der OOA nach Coad and Yourdon, die um die ermittelten Boolesch algebraischen Ausdrücke zur Kontrolle der referenziellen Integritäten zwischen den generalisierten / spezialisierten Objekttypen erweitert wurde.

Für den Klammerausdruck des abstrakten Beispiels (5.5) als Objekttypen- bzw. (5.3) Attributausdruck mit den Attributen aus Tabelle 3 ergibt sich folgende Darstellung:

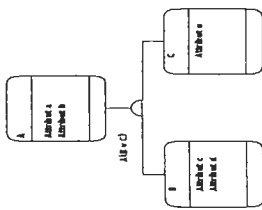


Abbildung 1: Graphische Umsetzung des Ausdrucks $A \vee B \vee C$

Der Klammerausdruck des konkreten Beispiels (6.8) als Objekttypen- bzw. (6.3) als Attributausdruck mit den Attributen aus Tabelle 4 führt zu folgender Darstellung:

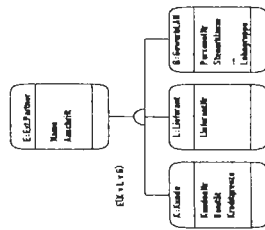


Abbildung 2: Graphische Umsetzung des konkreten Beispiels (6.8)

Die Elemente der Spezialisierung (untergeordnete Ebene) sind mit dem Element der Generalisierung (übergeordnete Ebene) konjunktiv (multiplikativ) verknüpft (vertikale Verknüpfung), z.B. $E(K \vee L \vee G)$. Die Elemente der Spezialisierung (der untergeordneten Ebene) sind untereinander disjunktiv verknüpft (horizontale Verknüpfung), z.B. $E(K \vee L \vee G)$. Nach einer „Klammer auf“ oder vor einer „Klammer zu“ in der gleichen Klammerebene muß mindestens eine Boolesche Variable stehen, da jede „Klammer auf“ eine Spezialisierung bedeutet. Überschüssige bzw. mehrfache Klammern, wie z.B. $a_1(((a_2)a_3))$, stören bei der Umsetzung eines Ausdrucks in das graphische Modell und sind deshalb wegzulassen.

1.6 Absorptionsfall (Verschmelzungsfall)

Am folgenden Beispiel wird die Absorption [Bar77] und ihre Interpretation in dieser Vorgehensweise behandelt. Folgende zweckorientierte Datensichten S_1 und S_2 mit ihren Attributen mögen vorliegen:

$S_1 = a_1, S_2 = a_1 a_2$ (7.1)

$S_1 \vee S_2 =$ (7.2)

$a_1 \vee a_1 a_2 = 1$ (7.3)

Die Anwendung des Distributivgesetzes ergibt folgenden Ausdruck:

$a_1 (1 \vee a_2) = 1$ (7.4)

Nach den Regeln der Booleschen Algebra wäre der Ausdruck in der Klammer immer erfüllt (immer = 1). Er könnte deshalb weggelassen werden, er wird absorbiert. Das Ergebnis wäre

$a_1 = 1$ (7.5)

Bei dieser Vorgehensweise darf der Klammerausdruck nicht entfernt werden. Die „1“ in der Klammer erfüllt hier die Funktion eines Platzhalters. Dies bedeutet, a_1 kann auch allein ohne a_2 in einem Objekttyp auftreten, was leicht dem Ausdruck (7.3) entnommen werden kann.

Es ergeben sich die beiden generalisierten / spezialisierten Objekttypen A mit a_1 und B mit a_1, a_2 , wobei sich die Objekttypschreibweise wie folgt ergibt:

$A (1 \vee B) = 1$

Die graphische Darstellung ist wie folgt:

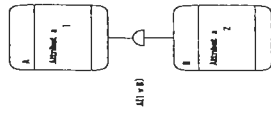


Abbildung 3: Graphische Umsetzung der Absorption

Folgendes konkrete Beispiel zeigt ebenfalls diesen Sachverhalt. Die Attribute eines Barzählers sind eine Untermenge der Attribute eines Kunden. Der Barzahler benötigt weder das Attribut Bonität noch das Attribut Kreditgrenze. Ansonsten besitzt er dieselben Attribute wie der Kunde:

Lfd. Nr.	Attribute (Boolesche Variable)	Zweckorientierte Datensichten			
		Kunde	Lieferant	Gewerblicher Arbeitnehmer	Barzahler
		S_1	S_2	S_3	S_4
1	Ext. PartnerNr	+	+	+	+
2	KundenNr	+	+	+	+
3	Name	+	+	+	+
4	Anschrift	+	+	+	+
5	Bonität	+			
6	Kreditgrenze	+			
7	LieferantNr		+		
8	PersonalNr			+	
9	Steuerklasse			+	
10	Finanzamt			+	
11	Aberkennung			+	
12	Stundensatz			+	
13	Lohngruppe			+	

Tabelle 6: Attributtabelle mit den zweckorientierten Datensichten „Kunde“, „Lieferant“, „Gewerblicher Arbeitnehmer“ und „Barzahler“

Die Ausformulierung dieser Tabelle in die disjunktive Normalform ergibt folgenden Ausdruck:

$$S_1 \vee S_2 \vee S_3 \vee S_4 = \quad (8.0)$$

$$\text{Kunde} \vee \text{Lieferant} \vee \text{Gewerblicher Arbeitnehmer} \vee \text{Barzähler} = \quad (8.1)$$

$$\text{In der verkürzten Form ergibt dies:} \quad (8.2)$$

$$a_0 a_1 a_2 a_3 a_4 a_5 \vee a_6 a_7 a_8 a_9 a_{10} a_{11} a_{12} \vee a_0 a_1 a_2 a_3 a_4 = 1 \quad (8.2)$$

Durch Ausklammern von $a_0 a_2 a_3$ aus Ausdruck (8.2) ergibt sich:

$$a_0 a_2 a_3 (a_1 a_4 a_5 \vee a_6 \vee a_7 a_8 a_9 a_{10} a_{11} a_{12} \vee a_1) = 1 \quad (8.3)$$

Durch Ausklammern von a_1 aus Ausdruck (8.3) ergibt sich:

$$a_0 a_2 a_3 (a_1 (1 \vee a_4 a_5) \vee a_6 \vee a_7 a_8 a_9 a_{10} a_{11} a_{12}) = 1 \quad (8.4)$$

Die Attribute können entsprechenden Objekttypen zugeordnet werden:

- Externer Partner = $E = a_0 a_2 a_3$ (8.5)
- Kunde = $K = a_1$ (8.6)
- Kreditor = $R = a_4 a_5$ (8.7)
- Lieferant = $L = a_6$ (8.8)
- Gewerblicher Arbeitnehmer = $G = a_7 a_8 a_9 a_{10} a_{11} a_{12}$ (8.9)

Dies ergibt für Ausdruck (8.4) folgende Objekttypenanzuordnung:

$$E (K (1 \vee R) \vee L \vee G) = 1 \quad (8.10)$$

Die graphische Darstellung ergibt folgendes Bild:

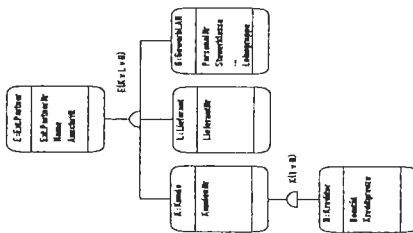


Abbildung 4: Graphische Umsetzung des mehrstufigen Ausdrucks (8.10) mit Absorption

Damit kann ein Barzähler mit seiner Untermenge an Attributen durch das System repräsentiert werden.

1.7 Sich gegenseitig ausschließende Attribute (Disjunktion)

Die explizite Abwesenheit eines Attributes in einer zweckorientierten Datensicht wird formal dadurch zum Ausdruck gebracht, daß die dieses Attribut verkörpernde Boolesche Variable negiert und der Negation der

„1“ zugewiesen wird. Die negierte Variable wird hier mit einer Unterstrichung dargestellt: \bar{a}_i . Der einer disjunktiven Normalform zugewiesene Funktionswert von „1“ bleibt erhalten. Die folgende Wahrheitstafel zeigt die Interpretation:

Boolesche Variable	\bar{a}_1	\bar{a}_2
Wertzuweisung „Attribut existiert“	1	0
Wertzuweisung „Attribut existiert nicht“	0	1

Tabelle 7: Wahrheitstafel bei Booleschen - Variablen „Logisch Nicht“

In einer Zuordnungsmatrix von Attributen zu zweckorientierten Datensichten wird die explizite Abwesenheit eines Attributes durch ein „-“ - Zeichen in der jeweiligen Zeile / Spalte signalisiert. Die „Logisch-Nicht“ - Variablen gehen nicht mit in das Datenmodell ein. Zwei Attribute können in zwei unterschiedlichen zweckorientierten Datensichten wie folgt existieren und sich wechselseitig beeinflussen:

LfdNr	Zweckorientierte Datensichten		Bemerkung
	S_1	S_2	
1	+	+	Attribut kommt in S_1 und in S_2 gleichzeitig vor
2	+	-	Attribut kommt nur in S_1 vor, unabhängig von S_2
3	-	+	Attribut kommt nur in S_2 vor, unabhängig von S_1
4	+	-	Attribut kommt in S_1 vor, nicht aber in S_2
5	-	+	Attribut kommt in S_2 vor, nicht aber in S_1

Tabelle 8: Tabelle sich wechselseitig beeinflussender Attribute in zweckorientierten Datensichten

Zu lfd. Nr. 1 in Tabelle 8:

Das Attribut a_1 kommt immer in beiden zweckorientierten Datensichten S_1 und S_2 vor. Das Attribut a_2 wird in diesem Fall bezüglich der beiden zweckorientierten Datensichten immer vor die Klammer gezogen d.h. generalisiert werden können.

Zu lfd. Nr. 2 und 3 in Tabelle 8:

Es kann eine Instanz (Ausprägung) der zweckorientierten Datensicht S_1 mit dem Attribut a_2 existieren und/oder es kann eine Instanz (Ausprägung) der zweckorientierten Datensicht S_2 mit dem Attribut a_3 existieren. Befinden sich beide Attribute über ein drittes generalisiertes Attribut (z.B. a_1) gemeinsam in einer Klammer, gelten sie als zu diesem Attribut spezialisiert: $a_2 a_3 \vee a_1 a_3$ mit a_1 = Geschäftspartner, a_2 = Kunde und a_3 = Lieferant. Ein Geschäftspartner kann entweder Kunde oder Lieferant oder beides gleichzeitig sein.

Zu lfd. Nr. 4 und 5 in Tabelle 8:

Falls die zweckorientierte Datensicht S_1 mit dem Attribut a_4 instanziiert (ausgeprägt) ist, darf nicht gleichzeitig die zweckorientierte Datensicht S_2 mit dem Attribut a_5 instanziiert (ausgeprägt) sein und umgekehrt. Diese gegenseitigen Ausschlüsse treten paarweise auf. Dazu müssen beide zweckorientierten Datensichten gemeinsame Attribute (z.B. a_1) besitzen, die vor die Klammer gezogen werden können. Dann ergibt sich nach der Durchführung der Generalisierung / Spezialisierung ein exklusiver Ausschuß (Disjunktion) der spezialisierten Objekttypen: $a_1 a_2 a_3 \vee a_1 a_4 a_5$ mit a_1 = Geschäftspartner, a_2 = natürliche Person (Privatperson) und a_3 = juristische Person (Gesellschaft). Ein Geschäftspartner kann entweder eine natürliche Person oder eine juristische Person, niemals aber beides gleichzeitig sein.

Abstraktes Beispiel:

$$S_1 = a_1 a_2 a_3, S_2 = a_1 a_4 a_5 \quad (9.1)$$

Inhaltliche Aussage: Wenn eine Instanzierung der zweckorientierten Datensicht S_1 existiert, darf gleichzeitig keine Instanzierung der zweckorientierten Datensicht S_2 mit der gleichen Ausprägung des Attributes a_1 existieren und umgekehrt. Die folgende Wahrheitstafel zeigt die vollständige Enumeration der Booleschen Va-

Konkretes Beispiel eines gegenseitigen Ausschlusses von Attributen (Tabelle 4 erweitert):

Lfd. Nr.	Attribute (Boolesche Variable)	Attribute (verkürzt)	Zweckorientierte Datensichten			
			Kunde	Lieferant	Gewerblicher Arbeitnehmer	Angestellter
	1	2	S ₁	S ₂	S ₃	S ₄
1	Ext.PartnerNr	a ₀	+	+	+	+
2	KundenNr	a ₁	+	+	+	+
3	Name	a ₂	+	+	+	+
4	Anschrift	a ₃	+	+	+	+
5	Bonität	a ₄	+	+	+	+
6	Kreditgrenze	a ₅	+	+	+	+
7	LieferantNr	a ₆	+	+	+	+
8	PersonalNr	a ₇	+	+	+	+
9	Steuerklasse	a ₈	+	+	+	+
10	Finanzamt	a ₉	+	+	+	+
11	Abteilung	a ₁₀	+	+	+	+
12	Stundensatz	a ₁₁	+	+	+	+
13	Lohngruppe	a ₁₂	+	+	+	+
14	Monatsgehalt	a ₁₃	+	+	+	+
15	Gehaltsgruppe	a ₁₄	+	+	+	+

Tabelle 10: Attributtabelle mit den zweckorientierten Datensichten „Kunde“, „Lieferant“, „Gewerblicher Arbeitnehmer“ und „Angestellter“ mit sich gegenseitig ausschließenden Attributen

Das „-“ Zeichen in der zweckorientierten Datensicht „Angestellter“ (S₄, Spalte 6) bedeutet, daß das Attribut Stundensatz a₁₁ (Zeile 12) beim einem Angestellten nicht vorkommen darf. Gleiches gilt für das Attribut Lohngruppe a₁₂ (Zeile 13). Beide dürfen nur bei der zweckorientierten Datensicht „Gewerblicher Arbeitnehmer“ (S₃, Spalte 5) vorkommen. Analog dazu dürfen die beiden Attribute Monatsgehalt a₁₃ (Zeile 14) und Gehaltsgruppe a₁₄ (Zeile 15) nicht in der zweckorientierten Datensicht „Gewerblicher Arbeitnehmer“ (S₃, Spalte 5) vorkommen. Sie dürfen nur in der zweckorientierten Datensicht „Angestellter“ (S₄, Spalte 6) vorkommen.

Die disjunktive Normalform aus Tabelle 10 ergibt sich wie folgt:

$$S_1 \vee S_2 \vee S_3 \vee S_4 = \quad (10.0)$$

$$\text{Kunde} \vee \text{Lieferant} \vee \text{Gewerbliche_Arbeitnehmer} \vee \text{Angestellter} = \quad (10.1)$$

$$a_0 \cdot a_1 \cdot a_2 \cdot a_3 \cdot a_4 \cdot a_5 \vee a_0 \cdot a_2 \cdot a_3 \cdot a_7 \cdot a_8 \cdot a_9 \cdot a_{10} \cdot a_{11} \cdot a_{12} \cdot a_{13} \cdot a_{14} \vee a_0 \cdot a_2 \cdot a_3 \cdot a_7 \cdot a_8 \cdot a_9 \cdot a_{10} \cdot a_{11} \cdot a_{12} \cdot a_{13} \cdot a_{14} = 1 \quad (10.2)$$

Durch Ausklammern von a₀ · a₂ · a₃ (10.3) und von a₇ · a₈ · a₉ · a₁₀ (10.4) erhält man folgende Ausdrücke:

$$a_0 \cdot a_2 \cdot a_3 \cdot (a_1 \cdot a_4 \cdot a_5 \vee a_6 \vee a_7 \cdot a_8 \cdot a_9 \cdot a_{10} \cdot a_{11} \cdot a_{12} \cdot a_{13} \cdot a_{14}) = 1 \quad (10.3)$$

$$a_0 \cdot a_2 \cdot a_3 \cdot (a_1 \cdot a_4 \cdot a_5 \vee a_6 \vee a_7 \cdot a_8 \cdot a_9 \cdot a_{10} \cdot (a_{11} \cdot a_{12} \cdot a_{13} \cdot a_{14} \vee a_{11} \cdot a_{12} \cdot a_{13} \cdot a_{14})) = 1 \quad (10.4)$$

in „Exklusiv Oder“- Darstellung:

$$a_0 \cdot a_2 \cdot a_3 \cdot (a_1 \cdot a_4 \cdot a_5 \vee a_6 \vee a_7 \cdot a_8 \cdot a_9 \cdot a_{10} \cdot (a_{11} \cdot a_{12} \cdot a_{13} \cdot a_{14} \vee a_{11} \cdot a_{12} \cdot a_{13} \cdot a_{14})) = 1 \quad (10.5)$$

oder in Objekttypennotation:

- Externer Partner = E = a₀ · a₂ · a₃ (10.6)
- Kunde = K = a₁ · a₄ · a₅ (10.7)
- Lieferant = L = a₆ (10.8)
- Mitarbeiter = M = a₇ · a₈ · a₉ · a₁₀ (10.9)
- Gewerblicher Arbeitnehmer = G = a₁₁ · a₁₂ (10.10)
- Angestellter = A = a₁₃ · a₁₄ (10.11)

bin für die beiden zweckorientierten Datensichten S₁ und S₂ aus (9.1) und deren Wertzuweisungen S₁ = f(a₁, a₂, a₃) bzw. S₂ = f(a₁, a₂, a₃):

LfdNr	a ₁	a ₂	a ₃	S ₁	S ₂	f(a ₁ , a ₂ , a ₃)	f(a ₁ , a ₂ , a ₃)
1	0	0	1	0	0	0 * 1 * 0	0
2	0	0	1	0	0	0 * 1 * 1	0
3	0	1	0	0	0	0 * 0 * 0	0
4	0	1	0	0	0	0 * 0 * 1	0
5	1	0	1	1	0	1 * 1 * 0	0
6	1	0	1	1	0	1 * 1 * 1	1
7	1	1	0	1	1	1 * 1 * 1	1
8	1	1	0	1	0	1 * 1 * 0	0

Tabelle 9: Wahrheitstafel der sich gegenseitig ausschließenden Attribute a₂ und a₃

Wie der Wahrheitstafel zu entnehmen ist, kann eine Instanz zur zweckorientierten Datensicht S₁ genau dann existieren, wenn die Attribute a₁ und a₂ vorhanden sind und gleichzeitig Attribut a₃ nicht vorhanden ist: S₁ = f(a₁, a₂, a₃) = 1 in Zeile 7. Ebenso existiert eine Instanz zur zweckorientierten Datensicht S₂ genau dann, wenn die Attribute a₁ und a₃ vorhanden sind und gleichzeitig Attribut a₂ nicht vorhanden ist: S₂ = f(a₁, a₂, a₃) = 1 in Zeile 6.

Die weitere Behandlung der Attribute ist wie folgt:

$$a_1 \cdot a_2 \cdot a_3 \vee a_1 \cdot \bar{a}_2 \cdot a_3 = 1 \quad (9.2)$$

$$a_1 \cdot (a_2 \cdot a_3 \vee \bar{a}_2 \cdot a_3) = 1 \quad (9.3)$$

oder in „Exklusiv Oder Darstellung“:

$$a_1 \cdot (a_2 \times a_3) = 1 \quad (9.4)$$

Das Zeichen „x“ steht für „Exklusiv Oder“.

Weist man den Booleschen Variablen (9.3) bzw. (9.4) die Objekttypen A = a₁, B = a₂ und C = a₃ zu, dann erhält man in Objekttypennotation folgenden Ausdruck:

$$A \cdot (B \bar{C} \vee \bar{B} C) = 1 \quad (9.5)$$

oder

$$A \cdot (B \times C) = 1 \quad (9.6)$$

Dieser Ausdruck kann graphisch wie folgt interpretiert werden:

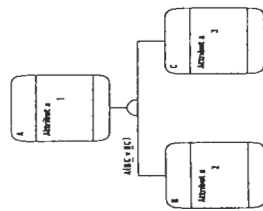


Abbildung 5: Graphische Umsetzung des gegenseitigen Ausschlusses von Objekttypen (disjunkte Objekttypen)

$$E(K \vee L \vee M) \vee (AG \vee AG) = 1$$

$$(10.12) \text{ oder}$$

$$E(K \vee L \vee M) \vee (G \vee A) = 1$$

$$(10.13)$$

Die graphische Interpretation von (10.13) bzw (10.5) ergibt folgende Darstellung

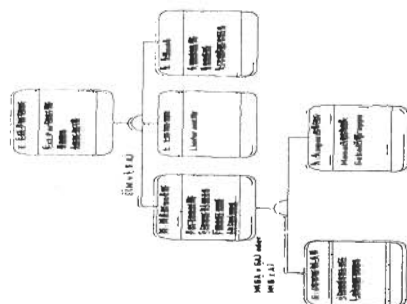


Abbildung 6: Graphische Darstellung des mehrstufigen konkreten und disjunkten Beispiels (10.13)

Die beiden Objekttypen G (Gewerblicher Arbeitnehmer) und A (Angestellter) sind zueinander disjunkt. Die Disjunktheit wird durch den Booleschen Ausdruck $M(\overline{AG} \vee AG)$ bzw. $M(G \times A)$ ausgedrückt.

1.8 Alternative graphische Darstellung einer Konjunktion

Die graphische Interpretation multiplikativer Verknüpfungen von Variablen außerhalb von Klammern mit den Variablen innerhalb von Klammern durch das Setzen von Spezialisierungssymbolen gilt auch für jede einzelne Konjunktion. Eine Konjunktion, die graphisch alternativ dargestellt werden soll, ist bereits generalisiert / spezialisiert und ihre Attribute (Boolesche Variable) z.B. a_1, a_2, a_3 sind zunächst einem Objekttyp (z.B. A) zugewiesen. Die einzelnen Variablen einer Konjunktion sind nichts anderes als multiplikative Verknüpfungen der Form: $a_1, a_2, a_3 = a_1(a_2, a_3) = a_1(a_2(a_3)) = \dots$ u.s.w.. Sie können für bestimmte Zwecke graphisch wie Disjunktionen interpretiert werden, mit dem Unterschied, daß dann spezialisierte Objekttypen der gleichen Ebene untereinander nicht disjunktiv sondern konjunktiv verbunden sind.

In den folgenden Darstellungen werden beispielhaft die drei Variablen a_1, a_2 und a_3 kombiniert bzw. variiert. Die Ausdrücke sind identisch. Dies gilt auch für die daraus abgeleiteten graphischen Interpretationen. Jede Konjunktion kann auf diese Weise dargestellt werden. Damit besitzt der Entwickler die Möglichkeit, bei Bedarf jederzeit konjunktive Strukturen graphisch in einfachere Terme (wieder Konjunktionen) zu zerlegen und in das Datenmodell zu integrieren. Die Darstellungen sind bezüglich der Möglichkeiten, drei Variablen zu kombinieren bzw. zu variieren, nur beispielhaft und deshalb nicht vollständig. Die drei Booleschen Variablen a_1, a_2, a_3 können wie folgt einfacheren Objekttypen zugewiesen werden:

LfdNr	Attributvarianation	Objekttyp- / Attributzuordnung	Objekttypenvarianation
1	a_1, a_2, a_3	$A' = a_1, a_2, a_3$	A'
2	$(a_1, a_2), a_3$	$A = a_1, B = a_2, a_3$	$A(B)$
3	$a_1, (a_2, a_3)$	$A = a_1, a_2, B = a_3$	$A(B)$
4	a_1, a_2, a_3	$A = a_1, B = a_2, C = a_3$	$A(B, C)$
5	$a_1, (a_2, a_3)$	$A = a_1, B = a_2, C = a_3$	$A(B, C)$
6	a_1, a_2, a_3	$A = a_1, B = a_2, a_3$	$A(B) \text{ wie } 2$
7

Die graphischen Interpretationen zu Tabelle 11 sind wie folgt:

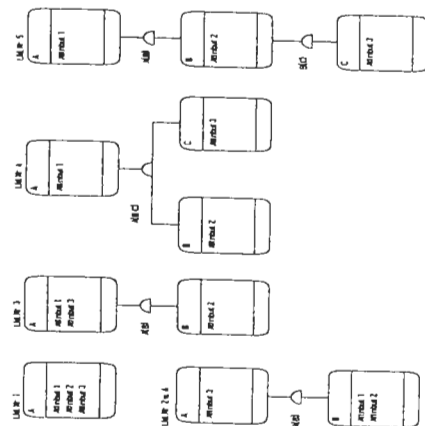


Abbildung 7: Beispiele der alternativen graphischen Darstellung der Konjunktion $A' = a_1, a_2, a_3$

Die Elemente der Spezialisierung (untergeordnete Ebene) sind mit dem Element der Generalisierung (übergeordnete Ebene) konjunktiv (multiplikativ) verknüpft (vertikale Verknüpfung), z.B. $A(B, C)$. Die Elemente der Spezialisierung (der untergeordneten Ebene) sind untereinander ebenfalls konjunktiv verknüpft (horizontale Verknüpfung), z.B. $A(B, C)$. Es liegt hier der Spezialfall der Disjunktion mit $B = 1$ und $C = 1$ vor. $A(B \vee C) = A(B, C)$. Dies gilt über alle Ebenen eines mehrstufigen Systems hinweg.

Die multiplikative Verknüpfung erzwingt, daß die jeweiligen Attribute bzw. die sie vertretenden Objekttypen jeweils alle oder gar nicht vorhanden sind. Ist ein Attribut nicht vorhanden (nicht belegt, =0), dann wird das gesamte Produkt (die Konjunktion) zu Null, d.h. diese Ausprägung des Objekttyps und alle konjunktiv mit diesem Objekttyp verbundenen Objekttypen gibt es nicht.

Graphisch ist die Potenzmenge aller Attribute ohne die leere Menge in einer Konjunktion zur getrennten Darstellung als Objekttypen zugelassen. Es sollte jedoch ein Minimum an Spezialisierungen (an spezialisierten Objekttypen) angestrebt werden.

Eine Menge dieser Potenzmenge kann jede mögliche Variation annehmen. Alle Darstellungsformen sind inhaltlich identisch anzusehen. Sie genügen dem Kommutativgesetz der Booleschen Algebra für die Logische Multiplikation.

2. Hauptgeneralisierung / -spezialisierung

2.1 Bestimmung der Wurzelobjekttypen bei Generalisierungen / Spezialisierungen

Erweitert man die Tabelle 10 um die zweckorientierte Datensicht „Barzahler“ und um die Attribute Bankleitzahl, Bankkonto-Nr., Debitoren-Nr. und Kreditoren-Nr., dann ergibt sich folgende Tabelle mit den jeweiligen Zuweisungen dieser Attribute zu den zweckorientierten Datensichten. Gleichzeitig wurde der gegenseitige Ausschluß von Attributen der zweckorientierten Datensichten „Lieferant“ und „Gewerblicher Arbeitnehmer“ bzw. „Angestellter“ vorgenommen. Dies bedeutet, daß eine bestimmte Instanz eines „Externen Partners“ nicht gleichzeitig ein „Lieferant“ und ein „Gewerblicher Arbeitnehmer“ bzw. „Angestellter“ sein kann und umgekehrt. Bestimmt man die Mächtigkeit der Attribute über alle zweckorientierten Datensichten (systemweit) hinweg, dann erhält man die am häufigsten vorkommenden Attribute (Spalte 8):

Lfd. Nr.	Attribute (Boolesche Variable)	Attribute (verkürzt)	Zweckorientierte Datensichten				Bar-zahler	Attribut Mächt.
			Kunde	Lieferant	Gewerbl. Arbeitnehmer	Angestellter		
1	Ext.PartnerNr	a ₀	S ₁	S ₂	S ₃	S ₄	S ₅	8
2	KundenNr	a ₁	+	+	+	+	+	5
3	Name	a ₂	+	+	+	+	+	2
4	Anschrift	a ₃	+	+	+	+	+	5
5	Bonität	a ₄	+	+	+	+	+	1
6	Kreditgrenze	a ₅	+	+	+	+	+	1
7	LieferantNr	a ₆	+	+	+	+	+	1
8	PersonalNr	a ₇	-	-	+	+	+	2
9	Steuerklasse	a ₈	-	-	+	+	+	2
10	Finanzamt	a ₉	-	-	+	+	+	2
11	Abteilung	a ₁₀	-	-	+	+	+	1
12	Stundensatz	a ₁₁	-	-	+	+	+	1
13	Lohngruppe	a ₁₂	-	-	+	+	+	1
14	Monatsgehalt	a ₁₃	-	-	+	+	+	1
15	Gehaltsgruppe	a ₁₄	-	-	+	+	+	1
16	Bankleitzahl	a ₁₅	+	+	+	+	+	4
17	Bankkonto	a ₁₆	+	+	+	+	+	4
18	KreditorenNr	a ₁₇	+	+	+	+	+	1
19	DebitorenNr	a ₁₈	+	+	+	+	+	3

Tabelle 12: Attributabelle mit den zweckorientierten Datensichten „Kunde“, „Lieferant“, „Gewerbliche Arbeitnehmer“, „Angestellter“ und „Barzahler“, erweitert um die Mächtigkeit der Attribute in Spalte 8

Die daraus entwickelte disjunktive Normalform stellt sich wie folgt dar:

$$S_1 \vee S_2 \vee S_3 \vee S_4 \vee S_5 = \text{Kunde} \vee \text{Lieferant} \vee \text{Gewerbl. Arbeitnehmer} \vee \text{Angestellter} \vee \text{Barzahler} = 1 \quad (11.1)$$

oder

$$a_{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} \vee a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} \vee a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} \vee a_{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} = 1 \quad (11.2)$$

Um möglichst viele Attribute zu zentralisieren, d.h. in der sich ergebenden Hierarchie möglichst weit nach oben zu legen, empfiehlt es sich, die Attribute, die zuerst generalisiert werden sollen, nach ihrer Mächtigkeit bezüglich ihres Vorkommens auszuwählen. Die Mächtigkeit der Attribute ist in Tabelle 12 in Spalte 8 eingetragen. Wie der Tabelle zu entnehmen ist, bieten sich die drei Attribute a₀, a₂ und a₃ mit einer Mächtigkeit von jeweils 5 zur ersten Generalisierung / Spezialisierung an. Sie kommen in allen zweckorientierten Datensichten (S₁ bis S₅) vor und sind deshalb in alle künftigen spezialisierten Objekttypen zu vererben.

Durch Ausklammern von a₂a₃ erhält man folgenden Ausdruck:

$$a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} \vee a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} \vee a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} \vee a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} = 1 \quad (11.3)$$

Als nächste Attribute, mit je einer Mächtigkeit von 4, bieten sich die beiden Attribute a₁₅ und a₁₆ zum Ausklammern an. Sie werden deshalb in der sich ergebenden Hierarchie an zweiter Stelle von oben angestodet sein.

$$a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} \vee a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} \vee a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} \vee a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} = 1 \quad (11.4)$$

Danach bietet sich das Attribut a₁₈ zum Ausklammern an, das eine Mächtigkeit von 3 besitzt.

$$a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} \vee a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} \vee a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} \vee a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} = 1 \quad (11.5)$$

Weiterhin können die Attribute a₇, a₈, a₉ und a₁₀ mit je einer Mächtigkeit von 2 ausklammert werden.

$$a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} \vee a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} \vee a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} \vee a_{0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18} = 1 \quad (11.6)$$

Die Attribute können zu Objekttypen zusammengefaßt werden:

- Externer Partner = E = a₀a₂a₃ (11.7)
- Bank = B = a₁a₅a₁₆ (11.8)
- Kunde = K = a₁ (11.9)
- Kreditör = R = a₁a₄a₅a₁₇ (11.10)
- Debitör = D = a₁₈ (11.11)
- Lieferant = L = a₆ (11.12)
- Mitarbeiter = M = a₇a₈a₉a₁₀ (11.13)
- Gewerblicher Arbeitnehmer = G = a₁₁a₁₂ (11.14)
- Angestellter = A = a₁₃a₁₄ (11.15)

Wird der Ausdruck (11.6) in Objekttypennotation dargestellt, ergibt sich:

$$E(B(R \vee D(L \underline{A} \underline{G} \underline{M} \vee \underline{L} M(G \underline{A} \vee \underline{G} A))) \vee K) = 1 \quad (11.16)$$

oder

$$E(B(R \vee D(L \times M(G \times A))) \vee K) = 1 \quad (11.17)$$

2.2 Vernetzte Generalisierungs- / Spezialisierungsstrukturen (Mehrfachvererbung)

Wie in Ausdruck (11.6) zu erkennen ist, befindet sich das Attribut a₁ (= Kunden-Nr.) an zwei Stellen im Gesamtausdruck. Dieses Attribut, mit einer Mächtigkeit von 2, darf nach dem Distributivgesetz der Booleschen Algebra nicht ausklammert werden. Die beiden Vorkommen des Attributes befinden sich nicht direkt in einer Klammerebene.

Unbehandelt käme das Attribut a₁ in zwei verschiedenen Objekttypen vor und wäre damit redundant. Der vorliegende Ausdruck ist nicht mehr als strenge, baumartige Hierarchie zu interpretieren. Vielmehr liegt jetzt eine netzartige Form vor, wenn man versucht, das mehrfach vorkommende Attribut nur an einer Stelle im Baum einzusetzen und es mit den entsprechenden Stellen, an denen es vorkommt, zu verbinden. An dieser Stelle endet die Hauptgeneralisierung / -spezialisierung. Falls eine Vernetzung des Systems durch mehrfach in den „Blättern“ einer baumartigen Hierarchie vorkommende Boolesche Variable vorgenommen werden muß, ist eine Nebengeneralisierung / -spezialisierung durchzuführen.

Bei objektorientierten Modellierungsmethoden entspricht diese Vernetzung der Mehrfachvererbung von Attributen und Methoden (Prozessen, Services). In der Literatur werden solche Konstruktionen auch als hintereinandergeschichtete Typkonstruktionen [Hohe93] bezeichnet.

Um ein Attribut, das in unterschiedlichen Konjunktionen vorkommt, aus diesen herausgenommen werden muß, damit es nur einmal im Netz vorkommt, ist die alternative Darstellung konjunktiv verbundener Attribute, wie in Punkt 1.8 beschrieben, zu benutzen.

Mit dieser Darstellungsform läßt sich der in (11.6) enthaltene Ausdruck R = a₁a₄a₅a₁₇ zerlegen. Das Attribut a₁ kann damit in der graphischen Darstellung wie folgt isoliert werden. Es wird in die beiden übergeordneten generalisierenden Objekttypen vererbt:

R wird ersetzt durch R' = (a₄a₅a₁₇) und K = (a₁). Damit ändert sich der Ausdruck (11.16) in Objekttypennotation wie folgt:

$$E(B(K'R' \vee D(L \underline{A} \underline{G} \underline{M} \vee \underline{L} M(G \underline{A} \vee \underline{G} A))) \vee K) = 1 \quad (11.18)$$

siert / spezialisiert. Sie stellt nur einen zusammengefaßten Objekttyp dar, der durch den Generalisierungs- / Spezialisierungsprozeß noch zu vertiefen ist. Das Datenmodell wurde mit SQL realisiert.

Gegeben waren folgende Attribute mit den dazugehörigen zweckorientierten Datensichten aus der Kundenrechnungssposition. Die Attribute sind nur stellvertretend ausgewählt worden und deshalb nicht vollzählig:

Lfd. Nr.	Attribute (Boolesche Variable)	Attribute (verkürzt)	Artikel	Artikel-zuschlag	Liefer-zuschlag	Kunden-auftrags-zuschlag	Mächt-Attribut
1	Rechnungspositions-Nr.	a ₁	+	+	+	+	4
2	Erlöskonto-Nr.	a ₂	+	+	+	+	4
3	Kundenauftrags-Nr.	a ₃	+	+	+	+	4
4	Lieferschein-Nr.	a ₄	+	+	+	+	3
5	Lieferscheinspositions-Nr.	a ₅	+	+	+	+	2
6	Zuschlags-Nr.	a ₆	+	+	+	+	3
7	Berechnungsmenge	a ₇	+	+	+	+	2
8	Einzelpreis je Mengeneinh.	a ₈	+	+	+	+	2
9	Zuschlagsverrech. Kennz.	a ₉	+	+	+	+	1
10	Zuschlagsbetrag einmalig	a ₁₀	+	+	+	+	2

Tabelle 13: Attributabelle „Rechnungsposition“ mit den zweckorientierten Datensichten „Artikel“, „Artikelzuschlag“, „Lieferzuschlag“ und „Kundenauftragszuschlag“

Aus Vereinfachungsgründen wurde in dieser Abhandlung auf eine Darstellung verzichtet, in der alle nicht belegten Attribute sich gegenseitig ausschließen.

Aus dieser Tabelle läßt sich folgende disjunktive Normalform ableiten:

$$S_1 \vee S_2 \vee S_3 \vee S_4 = a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 \vee a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 \vee a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 \vee a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 = 1 \quad (12.1)$$

In diesem Ausdruck kommen die beiden Booleschen Variablen a₁ und a₂ viermal vor. Sie sind deshalb zunächst auszuklammern:

$$a_1 a_2 (a_3 a_4 a_5 a_6 a_7 a_8 a_9 \vee a_3 a_4 a_5 a_6 a_7 a_8 a_9 \vee a_3 a_4 a_5 a_6 a_7 a_8 a_9 \vee a_3 a_4 a_5 a_6 a_7 a_8 a_9) = 1 \quad (12.2)$$

Danach könnte entweder die Boolesche Variable a₃ oder die Variable a₄ ausgeklammert werden. Beide kommen dreimal vor. Zunächst soll einmal die Variable a₄ ausgeklammert werden:

$$a_1 a_2 (a_3 (a_5 a_6 a_7 a_8 \vee a_5 a_6 a_7 a_8 a_9 \vee a_5 a_6 a_7 a_8 a_9 \vee a_5 a_6 a_7 a_8 a_9) \vee a_3 a_4 a_5 a_6 a_7 a_8 a_9) = 1 \quad (12.3)$$

Die Variable a₅ ließe sich jetzt nur noch in der inneren Klammer ausklammern. In dieser Vorgehensweise ist dies nicht erlaubt, da sich dieselbe Variable nochmals in einer anderen Klammernebene befindet. Damit besteht nur noch die Möglichkeit, in der inneren Klammer die Variablen a₆, a₇ und a₈ auszuklammern:

$$a_1 a_2 (a_3 (a_5 a_6 a_7 a_8 (1 \vee a_9 a_9) \vee a_9 a_9) \vee a_3 a_4 a_5 a_6 a_7 a_8 a_9) = 1 \quad (12.4)$$

Eine Zuweisung der Attribute zu Objekttypen ergibt:

$$A = a_1 a_2, D = a_4, L = a_5 a_6 a_7 a_8, K = a_6 a_7 a_8, M = a_3 a_4 a_5 a_6 a_7 a_8 a_9$$

Der Ausdruck (12.4) läßt sich damit auch wie folgt darstellen:

$$A (D ((1 \vee L) \vee K) \vee M) = 1 \quad (12.5)$$

$$(11.19)$$

$$E (B (K \cdot R' \vee D (L \times M (G \times A)) \vee K)) = 1$$

Dieser Ausdruck läßt sich wie folgt graphisch darstellen:

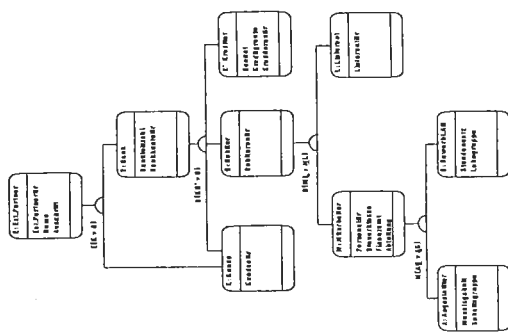


Abbildung 8: Vernetzte graphische Darstellung des Ausdrucks (11.19)

Diese Mehrfachvererbung ist intuitiv vorgenommen worden, in dem die in Punkt 1.8 beschriebene Vorgehensweise benutzt wurde. Die beiden Objekttypen KR' sind multiplikativ miteinander verknüpft und können, falls sie vorkommen, immer nur gemeinsam vorkommen.

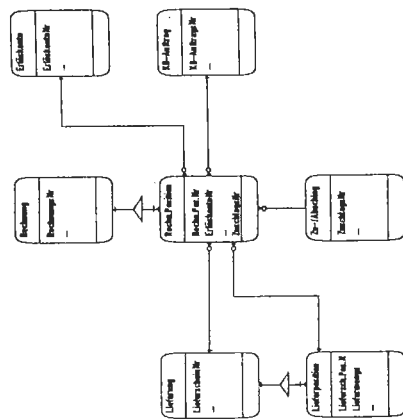


Abbildung 9: Datenmodellausschnitt der Kundenrechnungssposition und deren „referentielle Umgebung“

Ein weiteres Beispiel soll den Sachverhalt nochmals verdeutlichen. Abbildung 9 zeigt einen Ausschnitt aus dem Datenmodell eines Gießereinnehmens. Es zeigt den verteilten Entitätstyp [ScN690] Rechnungsposition und seine „referentielle Umgebung“. Die Rechnungsposition ist grau hinterlegt und lokal noch nicht generalisiert.

Folgende Darstellung zeigt diesen Ausdruck in graphischer Form, wobei aber die mehrfach vorkommenden Attribute a_3 (Zuschlagsnummer) und a_0 (Zuschlagsbetrag einmalig) in den Objekttypen L, K und M hingegen nicht vorkommen. Da das Attribut a_0 (Zuschlagsnummer) in den Objekttypen K, L und M vorkommt, ist für jeden Objekttyp eine Beziehung zum Entitätstyp (Scheitel) ZU- / ABSCHLAG vorhanden:

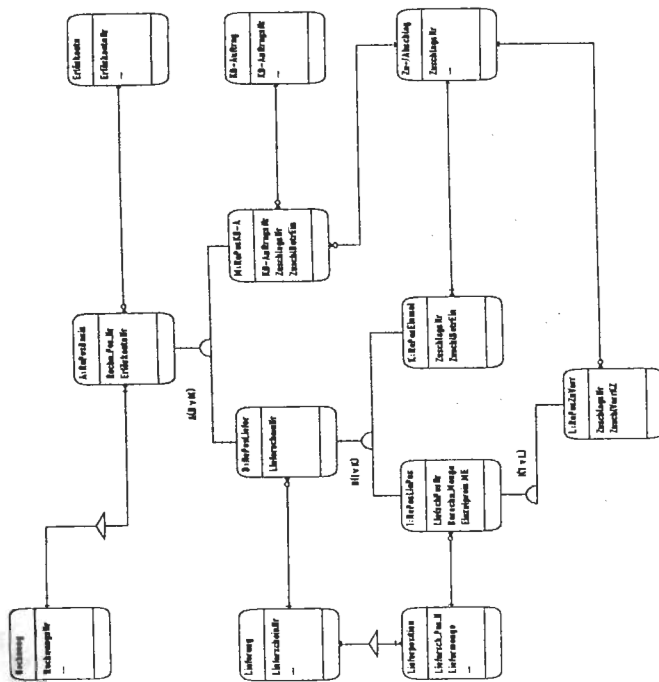


Abbildung 10: Graphische Darstellung der Rechnungsposition Ausdruck (12.5) Zerlegt man die Objekttypen $L = a_2 a_3$, $K = a_2 a_1 a_0$ und $M = a_1 a_2 a_1 a_0$ nochmals nach den Regeln der alternativen graphischen Darstellung einer Konjunktion (Punkt 1.8), dann kann eine Vernetzung abgeleitet werden:

$$L = (a_2)(a_3), K = (a_2)(a_1 a_0) \text{ und } M = (a_1)(a_2)(a_1 a_0)$$

$$L \text{ wird ersetzt durch } B = a_2 \text{ und } H = a_1, K \text{ wird ersetzt durch } B = a_2 \text{ und } C = a_1 a_0 \text{ und } M \text{ wird ersetzt durch } E = a_1, B = a_2 \text{ und } C = a_1 a_0$$

Der Ausdruck (12.5) ändert sich damit in:

$$A(D((1 \vee BH) \vee BC) \vee BCE) = 1 \quad (12.5.1)$$

Daraus kann die folgende vernetzte graphische Darstellung abgeleitet werden. Wie zu erkennen ist, besteht zwischen dem Entitätstyp ZU- / ABSCHLAG jetzt nur noch eine Beziehung zum spezialisierten Objekttyp B über das Attribut a_2 (Zuschlagsnummer). Dieses Attribut wird den anderen generalisierten / spezialisierten Objekttypen vererbt:

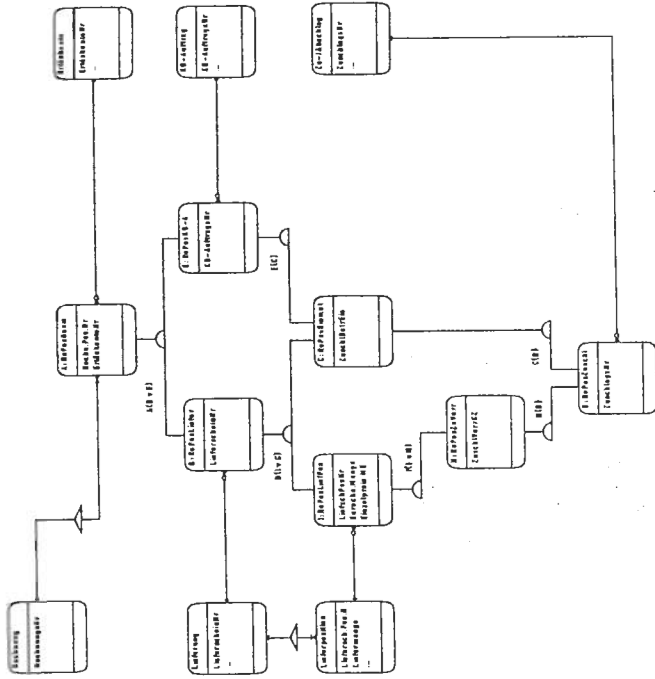


Abbildung 11: Vernetzte graphische Darstellung der Rechnungsposition Ausdruck (12.5) mit Mehrfachvererbung

Gegenüber der Darstellung in Abbildung 10 sind die referenziellen Beziehungen aus den Umgebungsobjekttypen jetzt genau den generalisierten / spezialisierten Objekttypen zuzuordnen, in denen sich die entsprechenden Fremdschlüssel befinden.

2.3 Elementare Attribute (Variable)

Boolesche Variable, die nicht vor Klammern gezogen wurden, gelten als elementar. In (12.4) sind dies die Variablen a_1, a_2, a_3 und $a_1 a_0$. Sie bilden die „Blätter“ in der baumartigen Hierarchie. Der Ausdruck (12.4) ist das Ergebnis der Hauptgeneralisierung / -spezialisierung. In der Nebengeneralisierung / -spezialisierung können die beiden in Ausdruck (12.4) redundant vorkommenden Variablen a_2 und $a_1 a_0$ weiter behandelt werden, damit nur sie einmal vorkommen aber mehrfach vererbt werden können. Die Nebengeneralisierung / -spezialisierung kann in einfachen Fällen intuitiv vorgenommen werden, indem die zu vernetzten Terme mit Hilfe der in Punkt 1.8 beschriebenen alternativen Darstellung konjunktiv verbundener Attribute zerlegt werden. Eine formale Vorgehensweise zur Nebengeneralisierung / -spezialisierung, die zur Mehrfachvererbung führen, wird in Punkt 5 dieser Abhandlung beschrieben.

Mehrfach vorkommende Boolesche Variable können in verschiedenen Ebenen einer Hauptgeneralisierung / Spezialisierung vorkommen. Zur weiteren Behandlung in einer Nebengeneralisierung / -spezialisierung dürfen sie nur in den „Blättern“ und nicht in den „Zweigen“ (Knoten) des Baumes einer Hauptgeneralisierung / -spezialisierung vorkommen. Falls sie in Knoten des Generalisierungs / -spezialisierungsbaumes vorkommen, dann ist ihre Vernetzung mehrdeutig. Damit kann die Kontrolle der referenziellen Integrität über zwei benachbarte Ebenen nicht mehr durch einen Booleschen Ausdruck gewährleistet werden, in dem sich nur Objekttypen aus den zwei direkt benachbarten Ebenen befinden.

Um mehrdeutige Knoten in vernetzten Datenmodellen der Hauptgeneralisierung / -spezialisierung zu vermeiden, ist folgende Regel zu beachten:

Zerlegt man die Objekttypen $G = a_4, a_5, a_8$, $F = a_4, a_5, a_8, a_9$ und $D = a_4$ nochmals nach den Regeln einer alternativen graphischen Darstellung einer Konjunktion (Punkt 1.8), dann kann eine Vernetzung abgeleitet werden:

$$G = (a_4)(a_5, a_8), F = (a_4)(a_5, a_8, a_9), D = (a_4)$$

G wird ersetzt durch $D = (a_4)$ und I = (a_5, a_8) und F wird ersetzt durch $D = (a_4)$, I = (a_5, a_8, a_9) und H = (a_9) .

Damit wird der Ausdruck (12.8) durch folgenden Ausdruck ersetzt:

$$A(D \vee B(DH \vee C(D \vee E))) = 1 \tag{12.9}$$

Daraus kann die folgende vernetzte graphische Darstellung abgeleitet werden:

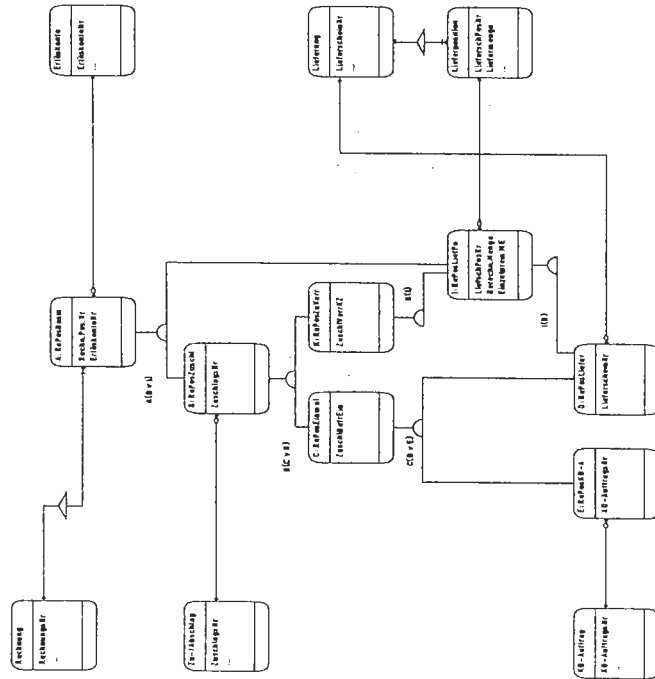


Abbildung 13: Vernetzte graphische Darstellung der Rechnungspositionsausdrucks (12.8) mit Mehrfachvererbung

Die beiden Ausdrücke (12.4) und (12.7) sind semantisch völlig gleichwertig. Die Anzahl der Objekttypen ist gleich. Wie jedoch die beiden graphischen Darstellungen zeigen, hat die Darstellung der Abbildung 13 eine Generalisierung / Spezialisierung weniger als die Darstellung der Abbildung 11. Deshalb wäre der Ausdruck (12.7) vorzuziehen, da er bei gleichem Inhalt weniger Elemente zur Repräsentation benötigt.

Eine weitere Entscheidungsmöglichkeit, wenn mehrere gültige Lösungen bei Generalisierungen / Spezialisierungen vorliegen, wäre die Auswahl der Alternative, bei der die häufigsten Instanzierungen zur Generalisierungs- / Spezialisierungswurzel hin zu erwarten sind. Damit werden immer minimale Instanzierungen bzw. -baume ausgeprägt. Dies hätte zur Folge, daß der Zeitverbrauch für Verarbeitungsprozesse insgesamt ein Minimum annehmen würde. Dasselbe gilt für die Anzahl der dann tatsächlich ausgeprägten Instanzen.

Kommt eine Boolesche Variable öfters in verschiedenen Stufen eines Ausdrucks der Hauptgeneralisierung / -spezialisierung vor, dann darf sie lokal nicht ausgeklammert werden. Eine mehrfach vorkommende Boolesche Variable muß elementar bleiben. Damit wird sichergestellt, daß mehrfach vorkommende Boolesche Variable sich nur in den „Blättern“ einer baumartigen Hierarchie der Hauptgeneralisierung / -spezialisierung befinden. Sie sind in einer Nebengeneralisierung / -spezialisierung weiter zu behandeln. Auf den Ausdruck (12.3) bezogen bedeutet dies: Die in der inneren Klammer zweimal vorkommende Boolesche Variable a_6 darf nicht ausgeklammert werden, da sie nochmals in der äußeren Klammer (letzter Term $a_6 a_6 a_{10}$) vorkommt.

2.4 Mehrere gültige Lösungen bei Generalisierungen / Spezialisierungen

Wird in Ausdruck (12.2) nicht die Variable a_4 , sondern die Variable a_6 ausgeklammert, was formal völlig gleichwertig ist, dann ergibt sich folgender Ausdruck:

$$a_1 a_2 (a_4, a_5, a_7, a_8 \vee a_6 (a_4, a_5, a_7, a_8, a_9 \vee a_4, a_{10} \vee a_3, a_{10})) = 1 \tag{12.6}$$

Danach ist die Variable a_{10} in der inneren Klammer auszuklammern. Die Variable a_4 kann nicht ausgeklammert werden, da sie in einer anderen Klammernebene nochmals elementar vorkommt:

$$a_1 a_2 (a_4, a_5, a_7, a_8 \vee a_6 (a_4, a_5, a_7, a_8, a_9 \vee a_{10} (a_4 \vee a_3))) = 1 \tag{12.7}$$

Eine Zuweisung der Attribute aus Ausdruck (12.7) zu Objekttypen ergibt:

$$A = a_1, G = a_4, a_5, a_7, a_8, B = a_6, F = a_4, a_5, a_7, a_8, a_9, C = a_{10}, D = a_4, E = a_3$$

Der Ausdruck (12.7) läßt sich damit auch wie folgt darstellen:

$$A(G \vee B(F \vee C(D \vee E))) = 1 \tag{12.8}$$

Folgende Darstellung zeigt diesen Ausdruck in graphischer Form, wobei hier die mehrfach vorkommenden Attribute a_4 , a_5 , a_7 und a_8 in den Objekttypen G, F und D hingenommen werden müßten.

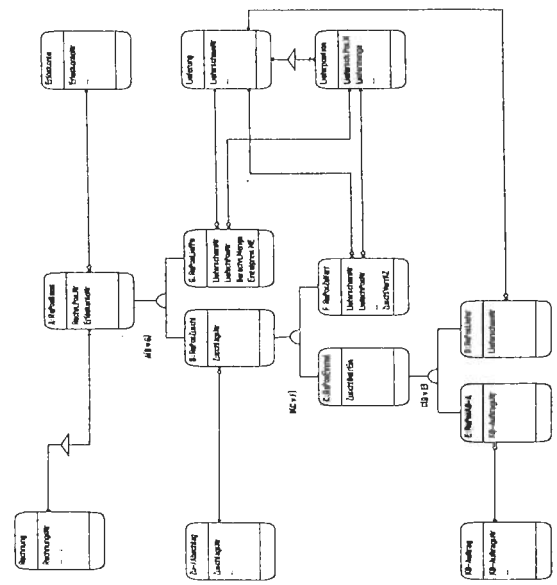


Abbildung 12: Graphische Darstellung Rechnungspositionsausdrucks (12.8)

Die Vorgehensweise stellt sicher, daß eine Instanzierung von generalisierten / spezialisierten Objekttypen immer von der jeweiligen Generalisierungswurzel aus lückenlos bis zum jeweiligen letzten spezialisierten Objekttypen (Blatt) erfolgt. Dies ist z.B. in der objektorientierten Systementwicklung Voraussetzung für die Vererbung von Attributen und Operationen (Methoden, Services). Für jeden direkten generalisierten / spezialisierten Objekttypnachbar ist die referentielle Integrität durch einen Boolesch algebraischen Ausdruck gesichert.

- 3 Instanzierungstypen (in Vorbereitung)
- 4 Systematisierte und integrierte Datenmodell- und Instanzenwartung (in Vorbereitung)
- 5 Nebengeneralisierung / -spezialisierung (in Vorbereitung)

Literaturverzeichnis:

- [Bart77] Bartsch, H.-J., Taschenbuch Mathematischer Formeln, Leipzig, 1977
- [CoYo90] Coad, P., Yourdon, E., Object-Oriented Analysis, Englewood Cliffs, New Jersey, 1990
- [FeS93] Ferstl, O. K., Sinz, E. J., Grundlagen der Wirtschaftsinformatik, Band 1, München, 1993
- [Hohe93] Hohenstein, U., Formale Semantik eines erweiterten Entity-Relationship-Modells, Leipzig, 1993
- [InDu89] Informatik Duden, Mannheim, 1989
- [Mar88] Marty, R., Von der Subroutinechnik zu Klassenhierarchien - Eine schrittweise Hinführung zu objektorientierter Programmierung, Zürich-Irchel, 1988
- [Sche90] Scheer, A.-W., Wirtschaftsinformatik, Heidelberg, 1990
- [ScN690] Schönthaler, F., Németh, T., Software-Entwicklungswerkzeuge: Methodische Grundlagen, Stuttgart, 1990

Die gesamte Vorgehensweise wird etwa Mitte Dezember 1995 als Buch im VDE-Verlag, Offenbach, unter dem Titel „Objektorientierte Analyse - mit Hilfe der Booleschen Algebra“, Autor Gisbert Englmeier, erscheinen.