

# Inkrementelle Lokalisierung funktionaler Eigenschaften

Thomas Eisenbarth, Rainer Koschke, Daniel Simon

[www.bauhaus-stuttgart.de](http://www.bauhaus-stuttgart.de)

{[eisenbts](mailto:eisenbts@informatik.uni-stuttgart.de),[koschke](mailto:koschke@informatik.uni-stuttgart.de),[simondl](mailto:simondl@informatik.uni-stuttgart.de)}@informatik.uni-stuttgart.de

## 1. Einführung

Bevor eine Funktionalität eines Software-Systems geändert werden kann, müssen die Routinen im Quellcode identifiziert werden, die diese Funktionalität implementieren. In früheren Arbeiten zum Problem der Lokalisierung bestimmter funktionaler Eigenschaften im Quellcode stellten wir ein Verfahren vor, das sowohl dynamische als auch statische Analysen sowie die Begriffsanalyse verknüpft [1].

Abbildung 1 zeigt eine Übersicht über den Prozess. Zunächst wird der statische Aufrufgraph aus dem Programm extrahiert. Durch Instrumentierung des Quellcodes mittels Compiler und anschließender mehrfacher Ausführung des Programmes anhand verschiedener Anwendungsszenarien werden dann durch einen Profiler die ausgeführten Routinen ermittelt. **Anwendungsszenarien** beschreiben relevante Verwendungsweisen des Systems, z.B. auch in Form von Testfällen. Dadurch ergibt sich die **Aufruftabelle**, die beschreibt, welche Routinen für welche Anwendungsszenarien ausgeführt wurden. Diese Tabelle stellt eine binäre Relation zwischen Anwendungsszenarien und Routinen dar und kann somit durch die Begriffsanalyse untersucht werden [2]. Hierzu werden die Anwendungsszenarien als Objekte und die Routinen als Attribute im Sinne der Begriffsanalyse betrachtet. Die Begriffsanalyse liefert die folgenden Informationen:

- alle Routinen, die von mehreren Anwendungsszenarien gemeinsam benötigt werden,
- alle Anwendungsszenarien, zu denen Routinen gemeinsam beitragen
- und die Spezifität einer Routine bezüglich der Anwendungsszenarien.

Die Spezifität einer Routine definiert sich durch die Menge der verschiedenen Anwendungsszenarien, zu denen eine Routine beiträgt. Wird eine Routine nur für ein Anwendungsszenario benutzt, so ist sie spezifisch für dieses Anwendungsszenario. Wird sie hingegen für alle Szenarien benutzt, ist sie allgemein. Im Begriffsverband lässt sich die Spezifität an der Position des Begriffs ablesen, in dem die Routine zum ersten Mal auftritt: Je näher die Routine dem Bottom-Element des Verbandes ist, desto allgemeiner ist sie.

Die Aufruftabelle stellt eine Relation zwischen Anwendungsszenarien und Routinen dar, so dass

der Begriffsverband entsprechend beschreibt, wie Anwendungsszenarien und Routinen zusammenhängen. Der Wartungsingenieur ist jedoch weniger an den Anwendungsszenarien selbst interessiert, sondern daran, wo bestimmte funktionale Eigenschaften im Quellcode implementiert werden. Falls ein Anwendungsszenario genau eine Funktionalität abdeckt, kann man durch das eben beschriebene Verfahren diese Zuordnung identifizieren. In der Regel sind Anwendungsszenarien jedoch komplexer: Sie verwenden tatsächlich mehrere funktionale Eigenschaften. Ist man z.B. bei einem graphischen Editor daran interessiert, welche Routinen für das Löschen eines Kreises zuständig sind, so muss dazu das System erst gestartet, ein Kreis gezeichnet, anschließend gelöscht und schließlich das System wieder beendet werden. Für einen solchen Programmablauf werden somit nicht nur die Routinen für das Löschen selbst aufgezeichnet, sondern auch für alle anderen genannten Schritte.

Dieser Artikel beschreibt, wie man die eigentliche Frage des Wartungsingenieurs nach den Routinen, die bestimmte funktionale Eigenschaften implementieren, beantworten kann. Dazu ist es unter Umständen notwendig, ein diskriminierendes Anwendungsszenario hinzuzufügen, das ursprünglich nicht berücksichtigt wurde. So könnte man für das genannte Beispiel ein Anwendungsszenario konstruieren, das den graphischen Editor startet, einen Kreis zeichnet und anschließend den Editor beendet, ohne den Kreis zu löschen. In diesem Falle ergibt sich ein neuer Begriffsverband, der die Routinen für das Löschen des Kreises identifiziert. Der Wartungsingenieur, der den ersten Begriffsverband ohne das neue Anwendungsszenario bereits betrachtet und verstanden hat, sollte sich im neuen Begriffsverband unter Einbeziehung des zusätzlichen Anwendungsszenarios möglichst schnell wieder zurechtfinden. Dieser Artikel beschreibt hierzu eine einfache Möglichkeit, die sich aus der Betrachtung so genannter Teilkontexte ergibt [2].

In Abschnitt 2 wird vorgestellt, wie man die funktionalen Eigenschaften im Begriffsverband für Szenarien und Routinen finden kann. Abschnitt 3 diskutiert die inkrementelle Erweiterung von Begriffsverbänden.

Dieser Artikel setzt Kenntnisse in der Begriffsanalyse voraus. Eine allgemeine und umfassende Beschreibung der Begriffsanalyse wird im Buch von

Abbildung 1. Prozess zur Lokalisierung von Merkmalen.

Ganter und Wille gegeben [2]. Eine kurze, für das Verständnis dieses Artikels ausreichende Einführung findet man z.B. in unseren vorausgehenden Arbeiten [1]. Eine kurze Beschreibung der Terminologie aus der Begriffsanalyse, die im Folgenden verwendet wird, befindet sich im Anhang.

## 2. Von Anwendungsszenarien zu funktionalen Eigenschaften

Der formale Kontext, auf den die Begriffsanalyse in unserem Ansatz angewandt wird, ist das Tripel  $(S,R,A)$ , wobei  $S$  die Menge der Anwendungsszenarien,  $R$  die Menge der Routinen und  $A$  die Aufruftabelle darstellen. Der resultierende Begriffsverband gewährt somit Einblicke in den Zusammenhang von  $S$  und  $R$ .

Wie bereits diskutiert, decken viele Anwendungsszenarien jedoch mehrere funktionale Eigenschaften eines Systems ab. Somit sollte ein Anwendungsszenario besser als eine Menge funktionaler Eigenschaften aufgefasst werden. (Dieser Modellierung liegt allerdings die vereinfachende Annahme zugrunde, dass die Anzahl und die Reihenfolge der benutzten funktionalen Eigenschaften eines Anwendungsszenarios keine Rolle spielen.)

Ein Anwendungsszenario kann anhand dieser Modellierung formal als eine Menge  $s = \{f_1, f_2, \dots, f_m\}$  beschrieben werden, wobei die Elemente  $f_n \in F$  mit  $1 \leq n \leq m$  relevante funktionale Eigenschaften aus einer Menge  $F$  darstellen.

Von Interesse für den Wartungsingenieur ist primär der Zusammenhang von  $F$  und  $R$  und weniger der Zusammenhang von  $S$  und  $R$ , wie er sich in Form der Aufruftabelle als Ergebnis der dynamischen Analyse als Relation zwischen  $S$  und  $R$  ergibt. Mit dem zusätzlichen Wissen, wie sich Anwendungsszenarien aus funktionalen Eigenschaften zusammensetzen, kann allerdings auch ein Zusammenhang zwischen  $F$  und  $R$  hergestellt werden. Die Information, welche funktionalen Eigenschaften durch ein Anwendungsszenario abgedeckt werden, muss durch den Wartungsingenieur beigesteuert werden. Die anschließende Herstellung des Zusammenhangs zwischen  $F$  und  $R$  lässt sich dann automa-

tisieren.

	$f_1$	$f_2$	$f_3$	$r_1$	$r_2$	$r_3$	$r_4$	$r_5$	$r_6$	$r_7$
$s_1$	X		X	X			X		X	X
$s_2$	X	X			X		X	X		X
$s_3$		X	X			X		X	X	X

Tabelle 1: Aufruftabelle.

Betrachten wir dazu die Aufrufrelation in Tabelle 1. Neben den aufgerufenen Routinen  $r_1$  bis  $r_7$  pro Szenario enthält diese Tabelle auch noch die Zerlegung von Szenarien in funktionale Eigenschaften (grauer Bereich). Szenario  $s_1$  benutzt die funktionalen Eigenschaften  $f_1$  und  $f_2$ , Szenario  $s_2$  benutzt  $f_1$  und  $f_2$  und  $s_3$  benutzt  $f_2$  und  $f_3$ . Der Begriffsverband für diese Aufruftabelle ist in Abbildung 2 abgebildet.

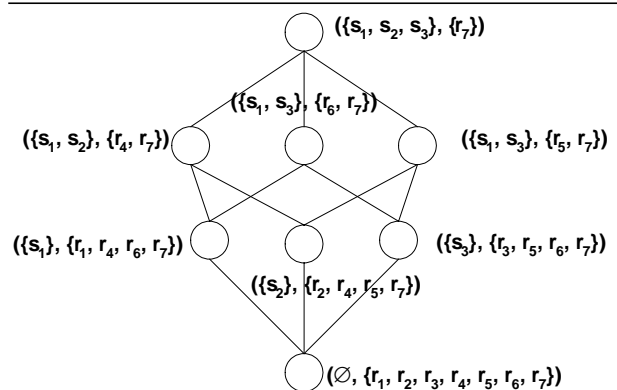


Abbildung 2. Begriffsverband für Tabelle 1.

Die Routinen, die speziell für die funktionale Eigenschaft  $f_1$  sind, sollten in der Schnittmenge der ausgeführten Routinen der beiden Szenarien  $s_1$  und  $s_2$  zu erwarten sein, da  $f_1$  sowohl für  $s_1$  als auch für  $s_2$  benutzt wird. Die Schnittmenge der Routinen für  $s_1$  und  $s_2$  sind im Begriffsverband als Supremum  $\gamma(s_1) \wedge \gamma(s_2)$  zu identifizieren. In Abbildung 2 ist dies der Begriff  $\{\{s_1, s_2\}, \{r_4, r_7\}\}$ . Da  $s_1$  und  $s_2$  sonst keine gemeinsame funktionale Eigenschaft haben, sind die Routinen  $r_4$  und  $r_7$  nur für  $f_1$  besonders relevant. Allerdings ist festzuhalten, dass  $r_7$  auch in anderen Szenarien benutzt wird. Tatsächlich wird  $r_7$  sogar in allen Szenarien benutzt, so dass  $r_7$  nicht wirklich als

spezifisch betrachtet werden kann. Die Routine  $r_4$  hingegen wird nur in Szenarien benutzt, in denen  $f_1$  verwendet wird. Wir können daraus schließen, dass  $r_4$  spezifisch für  $f_1$  ist,  $r_7$  jedoch nicht. Da es sonst keine weiteren Szenarien gibt, in deren Schnittmenge die gemeinsame funktionale Eigenschaft  $f_1$  vorkommt (was nicht notwendigerweise der Fall sein muss), ist  $r_4$  auch die einzige Routine, die spezifisch für  $f_1$  ist.

Routinen, die prinzipiell etwas mit  $f_1$  zu tun haben könnten, jedoch nicht als wirklich spezifisch betrachtet werden können, sind jene Routinen, die für Szenarien ausgeführt werden, in denen neben  $f_1$  noch weitere funktionale Eigenschaften benutzt werden. In unserem Beispiel betrachten wir dazu die Routinen, die für  $s_1$  oder  $s_2$  ausgeführt werden. Routinen in Begriffsinhalten der Begriffe, in deren Begriffsumfang  $s_1$  oder  $s_2$  enthalten sind, sind somit potenziell relevant für  $f_1$ . Im Beispiel sind dies neben den bereits ausgemachten Routinen  $r_4$  und  $r_7$  noch zusätzlich  $r_1$ ,  $r_2$ ,  $r_5$  und  $r_6$ . Die Routine  $r_3$  hingegen wird nur im Szenario  $s_3$  ausgeführt, das  $f_1$  nicht enthält. Für die Routinen lassen sich somit bezüglich  $f_1$  insgesamt die folgenden Kategorien identifizieren:

- **Kategorie 1:**  $r_4$  ist speziell zu  $f_1$ , weil es in allen Szenarien aufgerufen wird, in denen  $f_1$  benutzt wird, und nur in diesen;
- **Kategorie 2:**  $r_7$  ist relevant für  $f_1$ , aber sehr allgemein, weil es in allen Szenarien aufgerufen wird, die  $f_1$  und andere funktionale Eigenschaften benutzen, und darüber hinaus auch in Szenarien, in denen  $f_1$  nicht benutzt wird;
- **Kategorie 3:**  $r_1$  und  $r_2$  werden nur in solchen Szenarien aufgerufen, die  $f_1$  benutzen; sie sind aber weniger speziell als  $r_4$ , weil sie nicht in allen Szenarien ausgeführt werden, in denen  $f_1$  benutzt wird; inwiefern  $r_1$  und  $r_2$  jedoch spezieller oder allgemeiner als  $r_7$  sind, lässt nicht ohne Weiteres entscheiden; einerseits werden sie in allen Szenarien aufgerufen, die  $f_1$  und andere funktionale Eigenschaften benutzen, und nicht in Szenarien, die  $f_1$  nicht benutzen; andererseits wird  $r_7$  in *allen* Szenarien aufgerufen, in denen  $f_1$  benutzt wird;
- **Kategorie 4:**  $r_5$  und  $r_6$  werden in Szenarien aufgerufen, die  $f_1$  benutzen; allerdings werden sie auch in solchen Szenarien aufgerufen, in denen  $f_1$  nicht benutzt wird; damit sind sie wahrscheinlich weniger relevant als  $r_1$  und  $r_2$ , die nur in Szenarien aufgerufen werden, in denen  $f_1$  benutzt wird;  $r_5$  und  $r_6$  sind auch weniger speziell als  $r_7$ , weil sie nicht in allen Szenarien aufgerufen werden, in denen  $f_1$  benutzt wird;
- **Kategorie 5:**  $r_3$  ist irrelevant für  $f_1$ , weil es nur in solchen Szenarien ausgeführt wird, die  $f_1$  nicht enthalten.

Diese Sachverhalte lassen sich leichter in der kondensierten Darstellung des Begriffsverbandes erken-

nen. Betrachten wir dazu die semi-kondensierte Darstellung des Begriffsverbandes in Abbildung 3 für die Aufrufrelation aus Tabelle 1. Jede Routine wird hier genau einem Begriff zugeordnet. Die Knoten sind außerdem mit den funktionalen Eigenschaften annotiert: Umfasst der Begriffsumfang eines Begriffes  $B$  die Szenarien  $\{s_1, \dots, s_n\}$ , dann ergeben sich die funktionalen Eigenschaften als die Schnittmenge der enthaltenen Szenarien:

$$\bigcap_{1 \leq i \leq n} s_i$$

Dies ist die Menge gemeinsamer funktionaler Eigenschaften der Szenarien im Begriffsumfang von  $B$ . Routinen des Begriffsinhaltes von  $B$  tragen zu allen gemeinsamen funktionalen Eigenschaften dieser Szenarien bei.

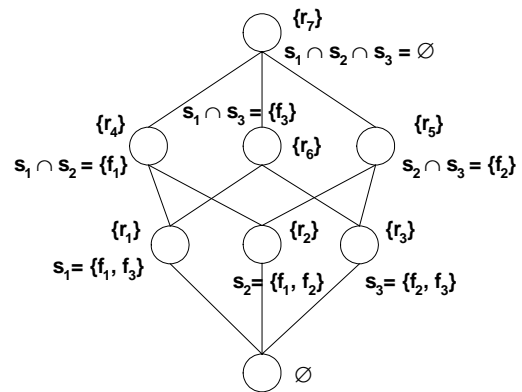


Abbildung 3. Semi-kondensierter Begriffsverband.

Um nun die Routinen für eine funktionale Eigenschaft  $f$  den oben genannten Kategorien zuzuordnen, kann man wie folgt vorgehen (vergleiche Abbildung 4). Ausgehend von allen Begriffen  $B_i = (S, R)$ , für die gilt (es könnte mehrere geben):

$$\bigcap_{s \in S} s = \{f\}$$

teilt man die Routinen folgendermaßen ein;

- **Kategorie 1:** alle Routinen  $r$ , für die gilt:  $\mu(r) = B_i$  für ein  $i$
- **Kategorie 2:** alle Routinen  $r$ , für die gilt:  $\mu(r) = B$  mit  $B_i < B$  für ein  $i$
- **Kategorie 3:** alle Routinen  $r$ , für die gilt:  $\mu(r) = B$  mit  $B < B_i$  für ein  $i$
- **Kategorie 4:** alle Routinen  $r$ , für die gilt:  $r$  ist im Begriffsinhalt von  $B < B_i$  für ein  $i$  enthalten, wobei  $B_i$  und  $\mu(r)$  unvergleichlich sind
- **Kategorie 5:** alle sonstigen Routinen

Innerhalb der Klassen 2, 3 und 4 sind weitere Abstufungen möglich anhand der Entfernung des Begriffes  $B$  von  $B_i$ , genauer anhand der Menge funktionaler Eigenschaften, zu denen die Routine außer  $f$  noch beiträgt bzw. nicht beiträgt.

Sollte es kein  $B_i$  mit der oben genannten Eigenschaft geben, so müssen die Szenarien weiter verfei-

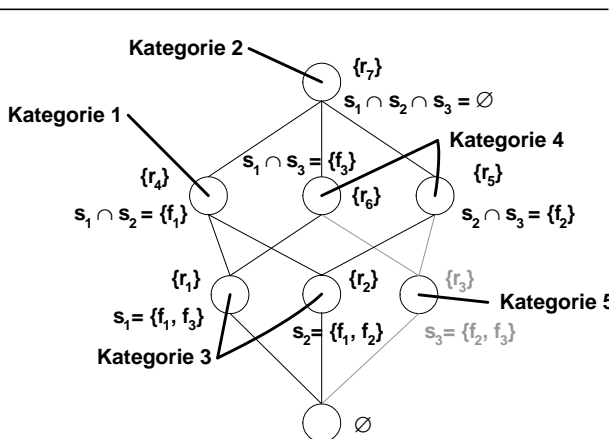


Abbildung 4. Kategorisierte Routinen.

nernt werden, um f herauszufaktorisieren.

### 3. Inkrementeller Aufbau

Werden neue Anwendungsszenarien hinzugefügt, ändert sich der Begriffsverband. Neue Anwendungsszenarien sind z.B. dann notwendig, wenn man Anwendungsszenarien weiter unterscheiden möchte, um auf die eigentlich relevanten funktionalen Eigenschaften des Systems zu schließen. Neue Anwendungsszenarien werden auch dann vorteilhaft sein, wenn man versucht, das System schrittweise zu verstehen. Hierbei beginnt man zunächst mit einer kleinen Menge relevanter Anwendungsszenarien, um eine überschaubare Menge funktionaler Eigenschaften des Systems zu lokalisieren und zu verstehen. Sobald man diesen Teil des Systems verstanden hat, kann man weitere funktionale Eigenschaften hinzunehmen. Würden alle funktionalen Eigenschaften auf einmal betrachtet, ergäbe sich ein sehr großer und damit unter Umständen viel zu detaillierter Begriffsverband.

Das bisherige Verständnis des Wartungsingenieurs vom Begriffsverband sollte bei einer inkrementellen Vorgehensweise aber erhalten bleiben, indem der alte Begriffsverband auf den neuen automatisch abgebildet wird. Eine geeignete Abbildung liefert ein Satz von Ganter und Wille über so genannte Teilkontexte [2]. Bevor wir zu diesem Satz kommen, folgt die Beschreibung eines Teilkontexts:

Ist  $(G, M, I)$  ein formaler Kontext und sind  $H \subseteq G$  und  $N \subseteq M$ , so wird  $(H, N, I \cap (H \times N))$  ein **Teilkontext** von  $(G, M, I)$  genannt.

In unserer Anwendung der Begriffsanalyse können nur weitere Szenarien hinzukommen, die Menge der deklarierten Routinen  $R$  ist jedoch fest. Somit betrachten wir im Folgenden lediglich Teilkontexte der folgenden Art:

$(S, R, A \cap (S \times R))$  ist ein Teilkontext von  $(S', R, A')$  mit  $S \subseteq S'$ . In Worten bedeutet dies, dass wir in der Tabelle A für jedes neue Anwendungsszenario eine neue Zeile hinzufügen (nicht jedoch Spalten).

Dadurch gelangen wir zu einem neuen formalen Kontext  $(S', R, A')$ , dessen Relation  $A'$  eine Erweiterung der Relation  $A$  darstellt (vergleiche Tabelle 2, in der  $S_5$  und  $S_6$  neu hinzukommen; der zugehörige Begriffsverband, wenn nur die ersten vier Szenarien betrachtet werden, findet sich in Abbildung 5).

	$R_1$	$R_2$	$R_3$	$R_4$
$S_1$			X	
$S_2$		X	X	
$S_3$	X	X		
$S_4$	X			
$S_5$	X		X	X
$S_6$	X	X		

Tabelle 2: Tabelle für Teilkontext.

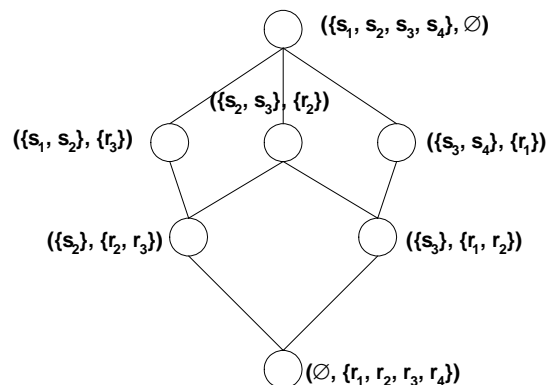


Abbildung 5. Begriffsverband für Tabelle 2.

Für solche Teilkontexte gilt der folgende Satz [2]:

Ist  $S \subseteq S'$ , so ist jeder Begriffsinhalt von  $(S, R, A \cap (S \times R))$  ein Begriffsinhalt von  $(S', R, A')$ .

Das heißt, jeder Begriffsinhalt im Teilkontext wird auch im erweiterten Kontext auftreten. Anschaulich wird dies in der Aufruftabelle dadurch, dass sich die Einträge der existierenden Zeilen durch Hinzufügung von weiteren Zeilen nicht ändern und sich somit die maximalen Rechtecke nur vertikal ausdehnen können.

Dies gilt nicht notwendigerweise für Begriffsumfänge. Das zusätzliche Szenario  $S_6$  in Tabelle 2 etwa führt dazu, dass der ursprüngliche Begriff  $(\{S_3\}, \{R_1, R_2\})$  zum Begriff  $(\{S_3, S_6\}, \{R_1, R_2\})$  ausgedehnt werden kann. Während der Begriffsinhalt gleich bleibt, ändert sich der Begriffsumfang.

Aus dem Satz über den Erhalt der Begriffsumfänge für Teilkontexte, die sich lediglich in der Objektmenge unterscheiden, folgt eine einfache Abbildung der Begriffe aus dem Teilkontext auf den erweiterten Kontext dergestalt:

$$(A, B) \rightarrow (\sigma(B), B)$$

Durch diese Abbildung lassen sich alle Begriffe

des Teilkontexts im erweiterten Kontext auffinden. Es können höchstens neue Begriffe hinzukommen. In einer Visualisierung des Begriffsverbandes als gerichteter azyklischer Graph lassen sich somit die neu hinzugekommenen Begriffe entsprechend einfärben und die Begriffe des alten Verbandes (z.B. durch Nummerierung) in der Visualisierung des erweiterten Verbandes auf den Teilkontext abbilden.

Diese Abbildung ist eine  $\vee$ -treue Ordnungseinkbettung, so dass auch die partielle Ordnung im erweiterten Begriffsverband vollständig erhalten bleibt. In diesem Sinne stellt der Begriffsverband für den erweiterten Kontext eine Verfeinerung des Begriffsverbandes für den Teilkontext dar.

Für die Visualisierung heißt dies, dass auch hier nur neue Kanten hinzukommen können. Die relative Position der Begriffe im Teilkontext bleibt somit erhalten. Ein passendes automatisches Graph-Lay-outing könnte die relative Positionen der Knoten für den Teilkontext erhalten und die neuen Knoten mit möglichst wenig Überschneidungen im erweiterten Verband platzieren. Auf diese Weise kann der Betrachter leicht den Verband für den Teilkontext im Verband für den erweiterten Kontext wiederfinden. Abbildung 6 zeigt den Begriffsverband für den erweiterten Kontext, der alle sechs Szenarien aus Tabelle 2 einschließt.

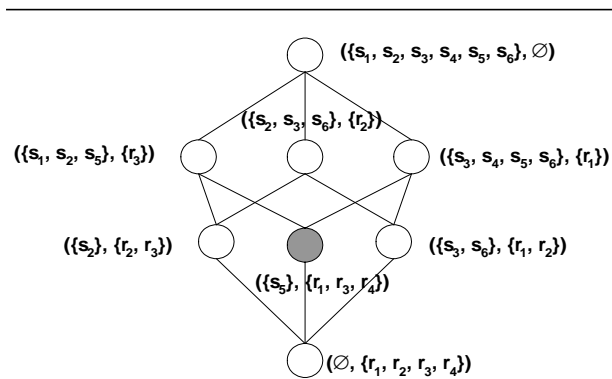


Abbildung 6. Begriffsverband für den erweiterten Kontext.

## Referenzen

- [1] Thomas Eisenbarth, Rainer Koschke, Daniel Simon. 'Aiding Program Comprehension by Static and Dynamic Feature Analysis'. In *Proceedings of the International Conference on Software Maintenance*, IEEE Computer Society Press, 2002.
- [2] Bernhard Ganter, Rudolf Wille. *Formale Begriffsanalyse*. Springer Verlag 1996.

## Anhang: Terminologie der Begriffsanalyse

Die Begriffsanalyse basiert auf einer **binären Relation**  $R$  zwischen einer **Menge von Objekten**  $O$  und einer **Menge von Attributen**  $A$ , d.h.  $R \subseteq O \times A$ .

Das Tripel  $\mathcal{C} = (O, A, R)$  wird **formaler Kontext** genannt. Für eine Menge von Objekten  $O \subseteq O$  ist die

Menge **gemeinsamer Attribute**  $\sigma$  definiert als:

$$\sigma(O) = \{a \in A \mid \forall (o \in O)(o, a) \in R\}$$

Analog, ist die Menge **gemeinsamer Objekte**  $\tau$  für eine Menge von Attributen  $A \subseteq A$  definiert als:

$$\tau(A) = \{o \in O \mid \forall (a \in A)(o, a) \in R\}$$

Das Paar  $(O, A)$  wird **Begriff** genannt, wenn  $A = \sigma(O) \wedge O = \tau(A)$  gilt. Für einen Begriff  $B = (O, A)$ , wird  $O$  als der **Begriffsumfang** und  $A$  als der **Begriffsinhalt** bezeichnet.

Für die Relation in Tabelle 3 ergeben sich die Begriffe in Tabelle 4.

	a <sub>1</sub>	a <sub>2</sub>	a <sub>3</sub>	a <sub>4</sub>	a <sub>5</sub>	a <sub>6</sub>	a <sub>7</sub>	a <sub>8</sub>
o <sub>1</sub>	X	X						
o <sub>2</sub>			X	X	X			
o <sub>3</sub>			X	X		X	X	X
o <sub>4</sub>			X	X	X	X	X	X

Tabelle 3: Beispielrelation.

C <sub>1</sub>	(({o <sub>1</sub> , o <sub>2</sub> , o <sub>3</sub> , o <sub>4</sub> }, $\emptyset$ )
C <sub>2</sub>	(({o <sub>2</sub> , o <sub>3</sub> , o <sub>4</sub> }, {a <sub>3</sub> , a <sub>4</sub> })
C <sub>3</sub>	(({o <sub>1</sub> }, {a <sub>1</sub> , a <sub>2</sub> })
C <sub>4</sub>	(({o <sub>2</sub> , o <sub>4</sub> }, {a <sub>3</sub> , a <sub>4</sub> , a <sub>5</sub> })
C <sub>5</sub>	(({o <sub>3</sub> , o <sub>4</sub> }, {a <sub>3</sub> , a <sub>4</sub> , a <sub>6</sub> , a <sub>7</sub> , a <sub>8</sub> })
C <sub>6</sub>	(({o <sub>4</sub> }, {a <sub>3</sub> , a <sub>4</sub> , a <sub>5</sub> , a <sub>6</sub> , a <sub>7</sub> , a <sub>8</sub> })
C <sub>7</sub>	( $\emptyset$ , {a <sub>1</sub> , a <sub>2</sub> , a <sub>3</sub> , a <sub>4</sub> , a <sub>5</sub> , a <sub>6</sub> , a <sub>7</sub> , a <sub>8</sub> })

Tabelle 4: Begriffe für die Relation aus Tabelle 3.

Die Menge aller Begriffe bildet eine **partielle Ordnung** mit:

$$(O_1, A_1) \leq (O_2, A_2) \Leftrightarrow O_1 \subseteq O_2$$

oder äquivalent mit

$$(O_1, A_1) \leq (O_2, A_2) \Leftrightarrow A_1 \supseteq A_2.$$

Falls  $c_1 \leq c_2$  gilt, dann wird  $c_1$  ein **Unterbegriff** von  $c_2$  und  $c_2$  wird der **Oberbegriff**  $c_1$  genannt.

Sei  $\mathcal{L}$  die Menge aller Begriffe eines formalen Kontexts wie folgt:

$$\mathcal{L}(\mathcal{C}) = \{(O, A) \in 2^O \times 2^A \mid A = \sigma(O) \wedge O = \tau(A)\}$$

$\mathcal{L}$  und die partielle Ordnung  $\leq$  bilden einen vollständigen Verband, den so genannten **Begriffsverband**.

Das **Infimum** zweier Begriffe in diesem Verband ergibt sich wie folgt:

$$(O_1, A_1) \wedge (O_2, A_2) = (O_1 \cap O_2, \sigma(O_1 \cap O_2))$$

Das **Supremum** ergibt sich entsprechend:

$$(O_1, A_1) \vee (O_2, A_2) = (\tau(A_1 \cap A_2), A_1 \cap A_2).$$

Für die Relation in Tabelle 3 ergibt sich der

Begriffsverband in Abbildung 7.

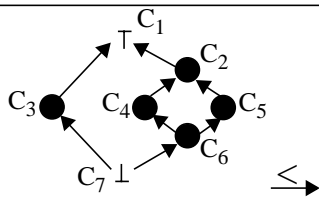


Abbildung 7. Begriffsverband für Tabelle 3.

Der Begriffsverband kann in einer **kondensierten Form** repräsentiert werden, indem man Attribute nur dem allgemeinsten Begriff zuordnet, in dessen Begriffsinhalt das Attribut vorkommt. Analog kann man Objekte dem speziellsten Begriff zuordnen, in dessen Begriffsumfang das Objekt vorkommt. Diese Begriffe lassen sich wie folgt identifizieren:

$$\mu(a) = \bigvee \{c \in L(C) \mid a \in \text{Begriffsinhalt}(c)\}$$

$$\gamma(o) = \bigwedge \{c \in L(C) \mid o \in \text{Begriffsumfang}(c)\}$$

Die kondensierte Darstellung des Begriffsverbands aus Abbildung 7 wird in Abbildung 8 wiedergegeben. In der **semi-kondensierten Darstellung** wird die vereinfachende Markierung entweder nur für Objekte oder nur für Attribute vorgenommen.

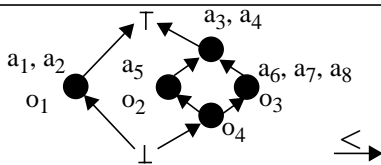


Abbildung 8. Kondensierte Darstellung von Abbildung 7.