

7.3.5 Eigenschaften von Rechten

Wir stellen hier einige Eigenschaften und modifizierte Definitionen für die Auswertung von Rechten zusammen, die wir später benutzen wollen.

7.3.5.1 Zusammengefasste Klassennamenauswertung

Während wir die Expandierung der Klassenhierarchie und der Klassen zu Objektmengen aus systematischen Gründen begrifflich getrennt hatten, fassen wir sie nachfolgend aus technischen Gründen zusammen, um Klassennamen in einem Schritt auswerten zu können. Dies ist zur Formulierung und für die Beweise einiger Sätze günstiger.

Wir verwenden dafür die Funktionen covered_objects^* . Diese Funktionen liefern die Menge der Objekte, die von einer Handlungskomponente überdeckt werden.

Definition 5.12: Sei $ss \in S \cup SC$, $so \in O \cup OC$, $sg \in G \cup GC$, $t \in TAG$.

$\text{covered_objects}^*_t(ss) :=$

if $\text{is_object}(ss)$ then $\{ss\}$ else $\text{members}(\text{subclasses}_{S,t}(ss))$ fi

$\text{covered_objects}^*_t(so) :=$

if $\text{is_object}(so)$ then $\{so\}$ else $\text{members}(\text{subclasses}_{O,t}(so))$ fi

$\text{covered_objects}^*_t(sg) :=$

if $\text{is_object}(sg)$ then $\{sg\}$ else $\text{members}(\text{subclasses}_{G,t}(sg))$ fi

Beispiel 5.13: $\text{covered_objects}^*_{\text{Verbot}}(\text{Arzt}) =$
 $\text{members}(\{\text{Arzt}, \text{Krankenschwester}, \text{Zivildienstleistender}\}) =$

$mc_S(\text{Arzt}) \quad mc_S(\text{Krankenschwester}) \quad mc_S(\text{Zivildienstleistender}) =$
 $\{\text{john, jane, karin, thomas}\}$

Analog definieren wir:

Definition 5.14: Sei $sr=(t,p,sa)$ ein spezifiziertes Recht mit
 $sa=(ss,so,sg) \quad (S \subseteq SC) \times (O \subseteq OC) \times (G \subseteq GC)$, $t \in \text{TAG}$, $p \in \text{PRIO}$, $SR' \subseteq SR$.
 $elementary_actions^*(t,ss,so,sg) := \{(es, eo, eg) \in EA \mid es \in covered_objects^*_t(ss)$
 $\quad eo \in covered_objects^*_t(so)$
 $\quad eg \in covered_objects^*_t(sg)\}$
 $elementary_rights^*(t,p,sa) := \{(t,p,ea) \in ER \mid ea \in elementary_actions^*(t,sa)\}$

Außerdem benutzen wir folgende Abkürzungen:

$elementary_rights^*(SR') :=$
 $elementary_rights(hierarchyfree_rights(SR')) =$
 $\bigcup_{sr \in SR'} elementary_rights^*(sr)$

ist die Menge aller elementaren Rechte, die aus einer Menge von spezifizierten Rechten SR' abgeleitet werden können und

$explicit_rights^*(SR') :=$
 $explicit_rights(elementary_rights(hierarchyfree_rights(SR')))$

ist die entsprechende Menge aller expliziten Rechte.

7.3.5.2 Überdeckte Klassen

Hier und in den folgenden Unterkapiteln stellen wir einige Eigenschaften für die Auswertung von Rechten

zusammen, die wir später benutzen wollen.

Da sich die Eigenschaften der Rechte für die verschiedenen Kategorien Subjekte, Operationen und Granule nur in der Behandlung der Klassenhierarchie unterscheiden, behandeln wir die Eigenschaften weitgehend kategorienunabhängig und unterscheiden nur zwischen gleich- und gegenläufiger Hierarchiebehandlung; wo sich eine Abhängigkeit von Kategorien nicht vermeiden ließ, benutzen wir zur Vereinfachung der Notation Granule stellvertretend für die gleichläufige Hierarchie und Subjekte stellvertretend für die gegenläufige Hierarchie.

In unserem Standardbeispiel benutzen wir stets die gleichläufige Hierarchiebehandlung für Granule und die gegenläufige Hierarchiebehandlung für Subjekte und Operationen.

Bemerkung 5.15: ER SR

Lemma 5.16: Sei c eine Klasse.

- (a) $\text{subclasses}_{\text{gegen},t}(c) = \text{if } t=\text{Erlaubnis} \text{ then } \text{subclasses}(c) \text{ else } \text{superclasses}(c) \text{ fi}$
- (b) $\text{subclasses}_{\text{gleich},t}(c) = \text{subclasses}(c)$

Beweis: (a):

$$\begin{aligned} & \text{subclasses}_{\text{gegen},t}(c) \\ = & \\ & \text{if } t=\text{Erlaubnis} \text{ then } \text{subclasses}_{\text{gegen},\text{Erlaubnis}}(c) \\ & \text{else } \text{subclasses}_{\text{gegen},\text{Verbot}}(c) \text{ fi} \\ = & (\text{Bemerkung 5.2(c)}) \end{aligned}$$

if t=Erlaubnis then subclasses_{gegen,Erlaubnis}(c)
 else superclasses_{gegen,Erlaubnis}(c) fi

=(Bemerkung 5.3)

if t=Erlaubnis then subclasses(c) else superclasses(c) fi

(b):

subclasses_{gleich,t}(c)

=

if t=Erlaubnis then subclasses_{gleich,Erlaubnis}(c)
 else subclasses_{gleich,Verbot}(c) fi

=(Bemerkung 5.2(a))

if t=Erlaubnis then subclasses_{gleich,Erlaubnis}(c)
 else subclasses_{gleich, Erlaubnis} (c) fi

=(Bemerkung 5.3)

subclasses(c)

•

Lemma 5.17: Sei c eine Klasse, o ein Objekt (der gleichen Kategorie) und t eine Rechtekennung. Dann gilt:

- (a) classes(o) subclasses_{gegen,t}(c) \emptyset
 c' classes(o): if t=Erlaubnis then c' c else c' c fi
- (b) classes(o) subclasses_{gleich,t}(c) \emptyset c' classes(o): c' c

Beweis: (a):

classes(o) subclasses_{gegen,t}(c) \emptyset

$c' \text{ classes}(o): c' \text{ subclasses}_{\text{gegen},t}(c)$

(Lemma 5.16(a))

$c' \text{ classes}(o):$

if $t=\text{Erlaubnis}$ then $c' \text{ subclasses}(c)$ else $c' \text{ superclasses}(c)$ fi

(Bemerkung 3.9)

$c' \text{ classes}(o):$ if $t=\text{Erlaubnis}$ then $c' \text{ } c$ else $c' \text{ } c$ fi

(b):

$\text{classes}(o) \text{ subclasses}_{\text{gleich},t}(c) \emptyset$

$c' \text{ classes}(o): c' \text{ subclasses}_{\text{gleich},t}(c)$

(Lemma 5.16(b))

$c' \text{ classes}(o): c' \text{ subclasses}(c)$

(Bemerkung 3.9)

$c' \text{ classes}(o): c' \text{ } c$

Korollar 5.18:

Gehört ein Objekt o zu höchstens einer Klasse, also $\|\text{classes}(o)\| = 1$, so gilt:

(a) $\text{classes}(o) \text{ subclasses}_{\text{gegen},t}(c) \emptyset$

if $t=\text{Erlaubnis}$ then $\text{classes}(o) \text{ } c$ else $\text{classes}(o) \text{ } c$ fi

(b) $\text{classes}(o) \text{ subclasses}_{\text{gleich},t}(c) \emptyset \quad \text{classes}(o) \text{ } c$

Beweis:

5.18 (a) und (b) folgen dann unmittelbar aus Lemma 5.17 (a) und (b). •

Aus Lemma 5.16 folgt unmittelbar:

Lemma 5.19: Seien c, c' Klassen (derselben Kategorie) und t, t' Rechtekennungen. Dann gilt:

$$\begin{aligned} \text{subclasses}_{\text{gleich},t}(c) \quad \text{subclasses}_{\text{gleich},t'}(c') \quad \emptyset \\ \text{subclasses}(c) \quad \text{subclasses}(c') \quad \emptyset \end{aligned}$$

Lemma 5.20:

Seien c, c' Klassen (derselben Kategorie) und t, t' Rechtekennungen. Dann gilt:

$$\text{subclasses}_{\text{gegen},t}(c) \quad \text{subclasses}_{\text{gegen},t'}(c') \quad \emptyset$$

if $t = \text{Erlaubnis}$ then if $t' = \text{Erlaubnis}$ then $cl \quad SC: cl \quad c \quad cl \quad c'$
 else $c' \quad c \quad fi$
else if $t' = \text{Erlaubnis}$ then $c \quad c'$
 else $cl \quad SC: cl \quad c \quad cl \quad c' \quad fi \quad fi$

Beweis: $\text{subclasses}_{\text{gegen},t}(c) \quad \text{subclasses}_{\text{gegen},t'}(c') \quad \emptyset$

(Lemma 5.16)

if $t = \text{Erlaubnis}$ then
 if $t' = \text{Erlaubnis}$ then $\text{subclasses}(c) \quad \text{subclasses}(c') \quad \emptyset$
 else $\text{subclasses}(c) \quad \text{superclasses}(c') \quad \emptyset \quad fi$
else if $t' = \text{Erlaubnis}$ then $\text{superclasses}(c) \quad \text{subclasses}(c') \quad \emptyset$

else superclasses(c) superclasses(c') \emptyset fi fi

if t=Erlaubnis then

if t'=Erlaubnis then

cl SC: cl subclasses(c) subclasses(c')

else cl SC: cl subclasses(c) superclasses(c') fi

else if t'=Erlaubnis then cl SC: cl superclasses(c) subclasses(c')

else cl SC: cl superclasses(c) superclasses(c') fi fi

(Bem. 3.9b/Bem. 3.9a)

if t=Erlaubnis then if t'=Erlaubnis then cl SC: cl c cl c'

else cl SC: cl c cl c' fi

else if t'=Erlaubnis then cl SC: cl c cl c'

else cl SC: cl c cl c' fi fi

(Transitivität von)

if t=Erlaubnis then if t'=Erlaubnis then cl SC: cl c cl c'

else c' c fi

else if t'=Erlaubnis then c c'

else cl SC: cl c cl c' fi fi

•

Für unterschiedliche Rechtekennungen folgt unmittelbar:

Korollar 5.21:

Seien c, c' Klassen (derselben Kategorie) und t eine Rechtekennung. Dann gilt:

$$\text{subclasses}_{\text{gegen},t}(c) \cap \text{subclasses}_{\text{gegen},\neg t}(c') = \emptyset$$

if t=Erlaubnis then c' c else c c' fi

Lemma 5.22:

Seien c, c' Klassen (derselben Kategorie) und t eine Rechtekennung. Dann gilt:

$c' \text{ subclasses}_{,t}(c) \quad \text{subclasses}_{,t}(c') \quad \text{subclasses}_{,t}(c)$

Beweis: Wir unterscheiden zunächst gleich- und gegenläufige Hierarchiebehandlung:

Fall 1: gleichläufige Hierarchiebehandlung: Zu zeigen ist:

$c' \text{ subclasses}_{\text{gleich},t}(c)$
 $\text{subclasses}_{\text{gleich},t}(c') \quad \text{subclasses}_{\text{gleich},t}(c),$

oder nach Definition von $\text{subclasses}_{\text{gleich}}$ äquivalent:

$c' \text{ subclasses}(c) \quad (c^* \text{ subclasses}(c') \quad c^* \text{ subclasses}(c))$

" " folgt aus der Transitivität,

" " aus der Reflexivität ($c' \text{ subclasses}(c')$, setze c' für c*) der Ordnung.

Fall 2: gegenläufige Hierarchiebehandlung: Zu zeigen ist:

$c' \text{ subclasses}_{\text{gegen},t}(c)$
 $\text{subclasses}_{\text{gegen},t}(c') \quad \text{subclasses}_{\text{gegen},t}(c)$

Fall 2a: t=Erlaubnis: Zu zeigen ist:

$c' \text{ subclasses}_{\text{gegen}, \text{Erlaubnis}}(c)$

$\text{subclasses}_{\text{gegen, Erlaubnis}}(c')$ $\text{subclasses}_{\text{gegen, Erlaubnis}}(c)$,
 oder nach Definition von $\text{subclasses}_{\text{gegen}}$ äquivalent:

$$c' \text{ subclasses}(c) \quad (c^* \text{ subclasses}(c') \quad c^* \text{ subclasses}(c))$$

Unter Fall 1 schon bewiesen.

Fall 2b: t=Verbot: Zu zeigen ist:

$$c' \text{ subclasses}_{\text{gegen, Verbot}}(c) \quad \text{subclasses}_{\text{gegen, Verbot}}(c') \quad \text{subclasses}_{\text{gegen, Verbot}}(c),$$

oder nach Definition von $\text{subclasses}_{\text{gegen}}$ äquivalent:

$$c' \text{ superclasses}(c) \quad (c^* \text{ superclasses}(c') \quad c^* \text{ superclasses}(c))$$

" " folgt aus der Transitivität,

" " aus der Reflexivität ($c' \text{ superclasses}(c')$, setze c' für c^*) der Ordnung. •

Lemma 5.23: Seien i, i' Objekte oder Klassen (derselben Kategorie) und t, t' Rechtekennungen. Dann gilt:

$$\text{covered_items}_{,t}(i) \quad \text{covered_items}_{,t'}(i') \quad \emptyset$$

$$\text{if is_object}(i) \text{ then is_object}(i') \quad i=i'$$

$$\text{else } \neg \text{is_object}(i') \quad \text{subclasses}_{,t}(i) \quad \text{subclasses}_{,t'}(i') \quad \emptyset \text{ fi}$$

Beweis:

$$\text{covered_items}_{,t}(i) \quad \text{covered_items}_{,t'}(i') \quad \emptyset$$

$$\text{hs covered_items}_{,t}(i) \quad \text{hs covered_items}_{,t'}(i')$$

(Def. 5.4)

```
hs S SC:  
if is_object(i) then hs=i else hs subclassest(i) fi  
if is_object(i') then hs=i' else hs subclassest(i') fi
```

```
hs S SC:  
if is_object(i)  
then if is_object(i') then hs=i hs=i'  
      else hs=i hs subclassest(i') fi  
else if is_object(i') then hs subclassest(i) hs=i'  
      else hs subclassest(i) hs subclassest(i') fi fi
```

```
hs S SC:  
if is_object(i)  
then if is_object(i') then hs=i=i'  
      else hs=i subclassest(i') fi  
else if is_object(i') then hs=i' subclassest(i)  
      else hs subclassest(i) subclassest(i') fi fi
```

```
hs S SC:  
if is_object(i)  
then if is_object(i') then hs=i=i'  
      else false fi
```

```

else if is_object(i') then false
      else hs subclassest(i) subclassest'(i') fi fi

hs S SC:
if is_object(i) then is_object(i')   hs S: hs=i=i'
else ¬ is_object(i')
      hs SC: hs subclassest(i) subclassest'(i') fi

if is_object(i) then is_object(i')   i=i'
else ¬ is_object(i') subclassest(i) subclassest'(i') ∅ fi

```

7.3.5.3 Durchschnitte überdeckter Handlungen und Objektmengen

In diesem Abschnitt betrachten wir die Eigenschaften von Durchschnitten überdeckter Handlungen. Wir benötigen diese später zur Berechnung von Konflikten.

Lemma 5.24: Seien i und i' Objekte oder Klassen einer Kategorie (also Subjekte, Operationen oder Granule) und t, t' Rechtekennungen. Seien alle Klassen dieser Kategorie nichtleer. Dann gilt:

- o $S: o$ covered_objects_t*(i) covered_objects_{t'}*(i')
 - if is_object(i')
 - then if is_object(i)
 - then $i'=i$
 - else i' members(subclasses_t(i)) fi
 - else if is_object(i)

```

then i members(subclassest(i'))
else members(subclassest(i'))
      members(subclassest(i))  ∅ fi fi

```

Beweis:

o S: o covered_objects*_t(i) covered_objects*_t(i')

o S: o covered_objects*_t(i) o covered_objects*_t(i')

(Def. 5.12)

```

o S: if is_object(i) then o {i} else o members(subclassest(i)) fi
if is_object(i') then o {i'} else o members(subclassest(i')) fi

```

(Fallunterscheidung: is_object/is_class)

```

o S: if is_object(i')
then if is_object(i)
then i'=o=i
else i'=o o members(subclassest(i)) fi
else if is_object(i)
then i=o o members(subclassest(i'))
else o members(subclassest(i')) o members(subclassest(i)) fi fi

```

(Vereinfachung, nicht-leere Klassen (für))

```

if is_object(i')
then if is_object(i)
then i'=i

```

```

    else i' members(subclassest(i)) fi
else if is_object(i)
    then i members(subclassest(i'))
    else members(subclassest(i')) members(subclassest(i)) ∅ fi fi •

```

Lemma 5.25: Seien i und i' Objekte oder Klassen einer Kategorie mit gleichläufiger Hierarchiebehandlung (in unserem Beispiel Granule) und t' eine Rechtekennung. Seien alle Klassen dieser Kategorie nichtleer und jedes Objekt dieser Kategorie gehöre zu nur einer Klasse. Dann gilt:

```

o G: o covered_objects*¬t'(i) covered_objects*t'(i')
    if is_object(i')
        then if is_object(i)
            then i'=i
            else classes(i') i fi
        else if is_object(i)
            then i' classes(i)
            else i'' GC: i'' i i'' i' fi fi

```

Beweis: Sei $t = \neg t'$.

```

o G: o covered_objects*t(i) covered_objects*t(i')

```

(Lemma 5.24)

```

    if is_object(i')
    then if is_object(i)
        then i'=i

```

```

    else i' members(subclassesgleich,t(i)) fi
else if is_object(i)
    then i members(subclassesgleich,t(i'))
    else members(subclassesgleich,t(i'))
        members(subclassesgleich,t(i))  ∅ fi fi

```

(Lemma 5.16(b))

```

if is_object(i')
    then if is_object(i)
        then i'=i
        else i' members(subclasses(i)) fi
    else if is_object(i)
        then i members(subclasses(i'))
        else members(subclasses(i'))  members(subclasses(i))  ∅ fi fi

```

(nicht-leere Klassen, jedes Objekt nur eine Klasse, Korollar 3.14, Lemma 3.18(b))

```

if is_object(i')
    then if is_object(i)
        then i'=i
        else classes(i')  i fi
    else if is_object(i)
        then i' classes(i)
        else i'' GC: i'' i  i'' i' fi fi

```

•

Lemma 5.26: Seien i und i' Objekte oder Klassen einer Kategorie mit gegenläufiger Hierarchiebehandlung

(also wie in unserem Beispiel Subjekte bzw. Operationen) und t' eine Rechtekennung. Seien alle Klassen dieser Kategorie nichtleer und jedes Objekt dieser Kategorie gehöre zu nur einer Klasse. Dann gilt:

```

o S: o covered_objects* $\neg_{t'}$ (i) covered_objects* $_{t'}$ (i')
    if is_object(i')
    then if is_object(i)
         then i'=i
         else if t'=Erlaubnis then classes(i') i else classes(i') i fi fi
    else if is_object(i)
         then if t'=Erlaubnis then i' classes(i) else i' classes(i) fi
         else if t'=Erlaubnis then i' i else i' i fi fi fi

```

Beweis: Sei $\neg t'=t$.

```

o S: o covered_objects* $_{t'}$ (i) covered_objects* $_{t'}$ (i')

```

(Lemma 5.24)

```

    if is_object(i')
    then if is_object(i)
         then i'=i
         else i' members(subclassesgegen,t(i)) fi
    else if is_object(i)
         then i members(subclassesgegen,t'(i'))
         else members(subclassesgegen,t'(i'))
              members(subclassesgegen,t(i))  $\emptyset$  fi fi

```

(Lemma 5.16 (subclasses_{gegen}))

```

if is_object(i')
then  if is_object(i)
      then i'=i
      else i'
      members(if t=Erlaubnis then subclasses(i) else superclasses(i)) fi
else  if is_object(i)
      then i
      members(if t'=Erlaubnis then subclasses(i') else superclasses(i'))
      else members(if t'=Erlaubnis then subclasses(i') else superclasses(i'))
      members(if t=Erlaubnis then subclasses(i) else superclasses(i))  Ø      fi fi

```

($t = \neg t'$, Vertauschen des THEN- und ELSE-Zweiges)

```

if is_object(i')
then  if is_object(i)
      then i'=i
      else  i'
      members(if t'=Erlaubnis then superclasses(i) else subclasses(i)) fi
else  if is_object(i)
      then i
      members(if t'=Erlaubnis then subclasses(i') else superclasses(i'))
      else  members(if t'=Erlaubnis then subclasses(i') else superclasses(i'))
      members(if t'=Erlaubnis then superclasses(i) else subclasses(i))  Ø      fi fi

```

(Fallunterscheidung: $t/\neg t$)

```

if is_object(i')

```



```

then if is_object(i)
      then i'=i
      else if t'=Erlaubnis then i' = members(superclasses(i))
            else i' = members(subclasses(i)) fi fi
else if is_object(i)
      then if t'=Erlaubnis then i = members(subclasses(i'))
            else i = members(superclasses(i')) fi
      else if t'=Erlaubnis
            then members(subclasses(i')) = members(superclasses(i)) ∅
            else members(superclasses(i')) = members(subclasses(i)) ∅ fi fi

```

(nicht-leere Klassen, jedes Objekt nur eine Klasse, Korollar 3.12, 3.14, Lemma 3.15(c))

```

if is_object(i')
then if is_object(i)
      then i'=i
      else if t'=Erlaubnis then classes(i') = i else classes(i') = i fi fi
else if is_object(i)
      then if t'=Erlaubnis then i' = classes(i) else i' = classes(i) fi
      else if t'=Erlaubnis then i' = i else i' = i fi fi

```

Insgesamt erhalten wir dann folgendes

Theorem 5.27: Seien $sa=(ss,so,sg)$, $sa'=(ss',so',sg')$ Handlungen und t, t' Rechtekennungen mit $t'=\neg t$. Seien alle Klassen nichtleer und jedes Objekt gehöre zu nur einer Klasse. Die Hierarchien für Subjekt- und

Operationsklassen seien gegenläufig zu behandeln, die für Granulklassen gleichläufig. Dann gilt:
 $\text{elementary_actions}^*(t,sa) \quad \text{elementary_actions}^*(t',sa') \quad \emptyset$

```
if is_object(ss')
  then if is_object(ss) then ss'=ss
        else if t'=Erlaubnis then classes(ss') ss else classes(ss') ss fi fi
  else if is_object(ss)
        then if t'=Erlaubnis then ss' classes(ss) else ss' classes(ss) fi
              else if t'=Erlaubnis then ss' ss else ss' ss fi fi fi
```

```
if is_object(so')
  then if is_object(so)
        then so'=so
              else if t'=Erlaubnis then classes(so') so else classes(so') so fi fi
  else if is_object(so)
        then if t'=Erlaubnis then so' classes(so) else so' classes(so) fi
              else if t'=Erlaubnis then so' so else so' so fi fi fi
```

```
if is_object(sg')
  then if is_object(sg) then sg'=sg else classes(sg') sg fi
  else if is_object(sg) then sg' classes(sg)
        else sg'' GC : sg'' sg sg'' sg' fi fi
```

Beweis: elementary_actions*(t,sa) elementary_actions*(t',sa') \emptyset

ea EA: ea elementary_actions*(t,sa) ea elementary_actions*(t',sa')

(Def. 5.14 (elementary_actions*), sa=(ss,so,sg), sa'=(ss',so',sg'))

ea=(es,eo,eg) EA:

es covered_objects*_t(ss) eo covered_objects*_t(so)

eg covered_objects*_t(sg) es covered_objects*_t'(ss')

eo covered_objects*_t'(so') eg covered_objects*_t'(sg')

es S: es covered_objects*_t(ss) covered_objects*_t'(ss')

eo O: eo covered_objects*_t(so) covered_objects*_t'(so')

eg G: eg covered_objects*_t(sg) covered_objects*_t'(sg')

Unter der Voraussetzung, dass $t' = \neg t$ ist, erhalten wir:

für Subjekte:

es S: es covered_objects*_t(ss) covered_objects*_t'(ss')

(Lemma 5.26)

if is_object(ss')

then if is_object(ss)

then ss'=ss

else if t'=Erlaubnis then classes(ss') ss else classes(ss') ss fi fi

else if is_object(ss)

```

then if t'=Erlaubnis then ss' classes(ss) else ss' classes(ss) fi
else if t'=Erlaubnis then ss' ss else ss' ss fi fi fi

```

für Operationen:

```

eo O: eo covered_objects*_t(so) covered_objects*_t'(so')

```

(Lemma 5.26)

```

if is_object(so')
then if is_object(so)
     then so'=so
     else if t'=Erlaubnis then classes(so') so else classes(so') so fi fi
else if is_object(so)
     then if t'=Erlaubnis then so' classes(so) else so' classes(so) fi
     else if t'=Erlaubnis then so' so else so' so fi fi fi

```

für Granule:

```

eg G: eg covered_objects*_t(sg) covered_objects*_t'(sg')

```

(Lemma 5.25)

```

if is_object(sg')
then if is_object(sg)
     then sg'=sg
     else classes(sg') sg fi
else if is_object(sg)
     then sg' classes(sg)

```

else sg'' GC: sg'' sg sg'' sg' fi fi

Insgesamt ergibt sich damit der Satz. •

7.3.5.4 Teilmengen überdeckter Handlungen und Objektmengen

In diesem Abschnitt betrachten wir die Eigenschaften von Teilmengenbeziehung zwischen überdeckten Handlungen. Wir benötigen diese später zur Berechnung von Anfragen.

Lemma 5.28: Sei t eine Rechtekennung und seien i und i' Objekte oder Klassen einer Kategorie (also Subjekte, Operationen oder Granule), für die Rechte vergeben werden können. Enthalte weiterhin jede Klasse dieser Kategorie ein charakteristisches Objekt. Dann gilt:

```
covered_objects*_t(i')  covered_objects*_t(i)
  if is_object(i) then i'=i
    else if is_object(i')
      then subclasses_t(i)  classes(i')  Ø
      else i'  subclasses_t(i) fi fi
```

Beweis:

```
covered_objects*_t(i')  covered_objects*_t(i)
  (Def. 5.12)
    if is_object(i') then {i'} else members(subclasses_t(i')) fi
    if is_object(i) then {i} else members(subclasses_t(i)) fi
```

(Fallunterscheidung)

```

if is_object(i)
  then if is_object(i')
    then {i'} {i}
    else members(subclasses_t(i')) {i} 8 fi
  else if is_object(i')
    then {i'} members(subclasses_t(i))
    else members(subclasses_t(i')) members(subclasses_t(i)) 9 fi
fi

```

(Existenz charakteristischer Objekte: Bem. 3.8(b))

```

if is_object(i)
  then if is_object(i')
    then i'=i
    else false fi
  else if is_object(i')
    then c subclasses_t(i): i' mc(c)
    else subclasses_t(i') subclasses_t(i) fi fi

```

```

if is_object(i) then i'=i
else if is_object(i')
  then c subclasses_t(i): i' mc(c)
  else subclasses_t(i') subclasses_t(i) fi fi

```

⁸ Da jede Klasse ein charakteristisches Objekt `_Klassenname` enthält und da daher Rechte zwar an `i`, aber nicht an `_Klassenname` vergeben werden können, gilt `_Klassenname i` und daher ist die Bedingung unerfüllbar.

⁹ Nach Bemerkung 3.8(b) ist dies äquivalent zu: `covered_classes(t,i') covered_classes(t,i)`.

(Bemerkung 3.5)

```
if is_object(i) then i'=i
  else if is_object(i')
    then c subclassest(i): c classes(i')
    else subclassest(i') subclassest(i) fi fi
```

```
if is_object(i) then i'=i
  else if is_object(i')
    then subclassest(i) classes(i') ∅
    else subclassest(i') subclassest(i) fi fi
```

(Lemma 5.22)

```
if is_object(i) then i'=i
  else if is_object(i')
    then subclassest(i) classes(i') ∅
    else i' subclassest(i) fi fi
```

Lemma 5.29: Sei t eine Rechtekennung und seien i und i' Objekte oder Klassen einer Kategorie mit gegenläufiger Hierarchiebehandlung (also wie in unserem Beispiel für Subjekte und Operationen), für die Rechte vergeben werden können. Enthalte weiterhin jede Klasse dieser Kategorie ein charakteristisches Objekt. Dann gilt:

$\text{covered_objects}^*_t(i') \subseteq \text{covered_objects}^*_t(i)$

if is_object(i) then i'=i
 else if is_object(i')
 then subclasses_{gegen,t}(i) classes(i') \emptyset
 else if t=Erlaubnis then i' subclasses(i) else i' superclasses(i) fi fi fi

Beweis:

covered_objects*_t(i') covered_objects*_t(i)

(Lemma 5.28)

if is_object(i) then i'=i
 else if is_object(i')
 then subclasses_{gegen,t}(i) classes(i') \emptyset
 else i' subclasses_{gegen,t}(i) fi fi

(Lemma 5.16 (subclasses_{gegen}))

if is_object(i) then i'=i
 else if is_object(i')
 then subclasses_{gegen,t}(i) classes(i') \emptyset
 else if t=Erlaubnis then i' subclasses(i) else i' superclasses(i) fi fi fi

•

Korollar 5.30:

Gehört darüberhinaus jedes Objekt zu genau einer Klasse, so gilt sogar:

covered_objects*_t(i') covered_objects*_t(i)
 if is_object(i')
 then if t=Erlaubnis then i {i'} superclasses(classes(i'))
 else i {i'} subclasses(classes(i')) fi

else if t=Erlaubnis then i superclasses(i') else i subclasses(i') fi fi

Beweis: Unter der zusätzlichen Voraussetzung, dass jedes Objekt zu höchstens einer Klasse gehört, gilt mit Lemma 5.29:

```

if is_object(i) then i'=i
else if is_object(i')
  then subclassesgegen,t(i) classes(i') ∅
  else if t=Erlaubnis then i' subclasses(i) else i' superclasses(i) fi fi fi

```

(Korollar 5.18 (a))

```

if is_object(i) then i'=i
else if is_object(i')
  then if t=Erlaubnis then classes(i') i else classes(i') i fi
  else if t=Erlaubnis then i' subclasses(i) else i' superclasses(i) fi fi fi

```

(i'=i trifft nur zu, falls auch i' Objekt ist)

```

if is_object(i')
  then i {i'}
  if t=Erlaubnis then superclasses(classes(i')) else subclasses(classes(i')) fi
  else i if t=Erlaubnis then superclasses(i') else subclasses(i') fi fi

```

```

if is_object(i')
  then if t=Erlaubnis then i {i'} superclasses(classes(i'))
  else i {i'} subclasses(classes(i')) fi
  else if t=Erlaubnis then i superclasses(i') else i subclasses(i') fi fi

```

Lemma 5.31: Sei t eine Rechtekennung und seien i und i' Objekte oder Klassen einer Kategorie mit gleichläufiger Hierarchiebehandlung (also wie in unserem Beispiel für Granule), für die Rechte vergeben werden können. Enthalte weiterhin jede Klasse dieser Kategorie ein charakteristisches Objekt. Dann:

$\text{covered_objects}^*_t(i') \subseteq \text{covered_objects}^*_t(i)$
 if $\text{is_object}(i)$ then $i'=i$
 else if $\text{is_object}(i')$
 then $\text{subclasses}(i) \cap \text{classes}(i') = \emptyset$
 else $i' \in \text{subclasses}(i)$ fi fi

Beweis:

$\text{covered_objects}^*_t(i') \subseteq \text{covered_objects}^*_t(i)$
 (Lemma 5.28)
 if $\text{is_object}(i)$ then $i'=i$
 else if $\text{is_object}(i')$
 then $\text{subclasses}_{\text{gleich},t}(i) \cap \text{classes}(i') = \emptyset$
 else $i' \in \text{subclasses}_{\text{gleich},t}(i)$ fi fi

(Lemma 5.16 ($\text{subclasses}_{\text{gleich}}$))
 if $\text{is_object}(i)$ then $i'=i$
 else if $\text{is_object}(i')$
 then $\text{subclasses}(i) \cap \text{classes}(i') = \emptyset$
 else $i' \in \text{subclasses}(i)$ fi fi

Korollar 5.32: Falls zusätzlich jedes Objekt zu genau einer Klasse gehört:

```
covered_objects*_t(i')  covered_objects*_t(i)
  if is_object(i')
    then i {i'}  superclasses(classes(i'))
    else i  superclasses(i') fi
```

Beweis: Unter der zusätzlichen Voraussetzung, dass jedes Objekt zu höchstens einer Klasse gehört, gilt mit Lemma 5.31:

```
if is_object(i) then i'=i
  else if is_object(i')
    then subclasses(i)  classes(i')  ∅
    else i'  subclasses(i) fi fi
```

(Korollar 5.18(b))

```
if is_object(i) then i'=i
  else if is_object(i')
    then classes(i')  i
    else i'  subclasses(i) fi fi
```

(i'=i trifft nur zu, falls auch i' Objekt ist)

```
if is_object(i')
  then i {i'}  superclasses(classes(i'))
  else i  superclasses(i') fi
```

Damit erhalten wir folgendes

Theorem 5.33: Seien $sa=(ss,so,sg)$, $sa'=(ss',so',sg')$ Handlungen und t eine Rechteckennung. Seien alle Klassen nichtleer und jedes Objekt gehöre zu nur einer Klasse. Die Hierarchien für Subjekt- und Operationsklassen seien gegenläufig zu behandeln, die für Granulklassen gleichläufig. Dann gilt:

$elementary_actions^*(t,sa') \quad elementary_actions^*(t,sa)$

```

if is_object(ss')
then if t=Erlaubnis then ss {ss'} superclasses(classes(ss'))
      else ss {ss'} subclasses(classes(ss')) fi
else if t=Erlaubnis then ss superclasses(ss')
      else ss subclasses(ss') fi fi

```

```

if is_object(so')
then if t=Erlaubnis then so {so'} superclasses(classes(so'))
      else so {so'} subclasses(classes(so')) fi
else if t=Erlaubnis then so superclasses(so')
      else so subclasses(so') fi fi

```

```

if is_object(sg')
then sg {sg'} superclasses(classes(sg'))
else sg superclasses(sg') fi

```

Beweis:

$elementary_actions^*(t,sa') \quad elementary_actions^*(t,sa)$

ea elementary_actions*(t,sa) ea elementary_actions*(t,sa)

(Def. 5.14 (elementary_actions*), ea=(es,eo,eg), sa'=(ss',so',sg'), sa=(ss,so,sg),

Unabhängigkeit der Komponenten)

es covered_objects*_t(ss') es covered_objects*_t(ss)
eo covered_objects*_t(so') eo covered_objects*_t(so)
eg covered_objects*_t(sg') eg covered_objects*_t(sg)

covered_objects*_t(ss') covered_objects*_t(ss)
covered_objects*_t(so') covered_objects*_t(so)
covered_objects*_t(sg') covered_objects*_t(sg)

Nun benutzen wir die Eigenschaft, dass jedes Objekt zu höchstens einer Klasse gehört.

Für Subjekte gilt:

covered_objects*_t(ss') covered_objects*_t(ss)

(Korollar 5.30)

if is_object(ss')
then if t=Erlaubnis then ss { ss' } superclasses(classes(ss'))
 else ss { ss' } subclasses(classes(ss')) fi
else if t=Erlaubnis then ss superclasses(ss') else ss subclasses(ss') fi fi

Für Operationen gilt analog:

covered_objects*_t(so') covered_objects*_t(so)

(Korollar 5.30)

```
if is_object(so')
then if t=Erlaubnis then so {so'} superclasses(classes(so'))
      else so {so'} subclasses(classes(so')) fi
else if t=Erlaubnis then so superclasses(so') else so subclasses(so') fi fi
```

Für Granule gilt:

covered_objects*_t(sg') covered_objects*_t(sg)

(Korollar 5.32)

```
if is_object(sg')
then sg {sg'} superclasses(classes(sg'))
else sg superclasses(sg') fi
```

Insgesamt ergibt sich damit der Satz. •