

7.4 Unvollständige und widersprüchliche Rechtespezifikationen

Lässt man als explizite Spezifikation nur die Angabe von Erlaubnissen zu, so erhält man implizit nur eine weitere Möglichkeit, nämlich die Abwesenheit von Erlaubnissen, die man üblicherweise als Verbot versteht:

(elem.) Handlung wird durch ein spezifiziertes Recht erlaubt	Erlaubnis
(elem.) Handlung wird nicht durch spezifiziertes Recht erlaubt	Verbot

Spezifikation expliziter Erlaubnisse

Diese Situation ändert sich, wenn eine Spezifikation nicht nur explizite Erlaubnisse, sondern auch explizite Verbote enthält.

Dann kann es sowohl widersprüchliche Spezifikationen geben, d.h. es gibt eine Handlung, für die sowohl eine Erlaubnis als auch ein Verbot spezifiziert ist, als auch unvollständige Spezifikationen, d.h. es gibt eine Handlung, für die weder eine Erlaubnis noch ein Verbot spezifiziert ist.

Das nachfolgende Schema macht diese Situation transparent.

	(elem.) Handlung wird durch ein spezifiziertes Recht verboten	(elem.) Handlung wird nicht durch spezifiziertes Recht verboten
(elem.) Handlung wird durch ein spezifiziertes Recht erlaubt	Konflikt	Erlaubnis
(elem.) Handlung wird nicht durch ein spezifiziertes Recht erlaubt	Verbot	Egal (Don't Care)

Spezifikation expliziter Erlaubnisse und Verbote

Insbesondere hat jetzt die Abwesenheit von Rechten für eine Handlung eine andere Bedeutung bekommen: Statt eines Verbots ist die Bedeutung jetzt 'egal'.

7.4.1 Unvollständige Rechtespezifikationen: Don't Cares

Unvollständige Spezifikationen sind weder schlecht noch ungewöhnlich, sondern im Gegenteil, es sind genau diese unvollständigen Spezifikationen, die uns die Freiheitsgrade für eine Implementierung auf niedrigerer Ebene liefern.

Es ist allerdings eine gute Gewohnheit, zur Entwurfszeit zu überprüfen, ob die Menge aller Handlungen ohne Spezifikation wirklich diejenige ist, die bzgl. Erlaubnissen und Verboten irrelevant ist.

Auf der niedrigsten Implementierungsebene werden allerdings don't care-Handlungen i.a. genauso behandelt wie (implizite) Verbote.

7.4.2 Widersprüchliche Rechtespezifikationen: Konflikte zwischen Rechten

Es ist nicht sehr wünschenswert, eine Spezifikation zu haben, in der dieselbe Handlung sowohl erlaubt als auch verboten ist.

Dies ist offensichtlich eine **widersprüchliche Spezifikation** oder ein **Konflikt**.

Wenn man aber sowohl Erlaubnisse als auch Verbote in der Spezifikation zulässt, so können solche Konflikte natürlich auftreten.

Daher muss eine Rechtespezifikation bei Rechteinitialisierung und nach Rechteänderungen auf solche Konflikte überprüft werden und gegebenenfalls der zuständige Rechteverwalter mit einer verständlichen Fehlermeldung informiert werden.

7.4.2.1 Aktuelle Konflikte und Basiskonflikte

Zunächst betrachten wir explizite Rechte. Zwei explizite Rechte stehen offenbar dann miteinander in Konflikt, wenn sie beide die gleiche elementare Handlung betreffen, aber invertierte Rechtekennungen haben.

Diese Konflikte nennen wir **aktuelle Konflikte**:

Definition: Seien $xr=(t,ea)$ und $xr'=(t',ea')$ zwei explizite Rechte.

actual_x_conflict(xr,xr') : $ea=ea' \quad t=\neg t'$

Bei elementaren Rechten ist die Situation etwas anders.

Wenn zwei elementare Rechte miteinander in Konflikt stehen, müssen die beiden Rechte ebenfalls die gleiche elementare Handlung betreffen und invertierte Rechtekennungen haben.

Aber hier lösen verschiedene Prioritäten einen Konflikt zwischen elementaren Rechten dadurch auf, dass bei der Auswertung der Prioritäten nur die Rechte mit einer –bzgl. der elementaren Handlung- maximalen Priorität berücksichtigt werden.

Es bleibt nur dann ein Konflikt übrig, wenn die Prioritäten gleich oder unvergleichbar sind.

Diese Konflikte nennen wir **Basiskonflikte**:

Definition: Seien $er=(t,p,ea)$ und $er'=(t',p',ea')$ zwei elementare Rechte.

basic_e_conflict(er,er') : $ea=ea' \quad t=\neg t' \quad \neg(p < p') \quad \neg(p' < p)$

Bei linearen Prioritätsordnungen kann es dann nur Konflikte zwischen Rechten mit gleicher Priorität geben.

Korollar: Seien $er=(t,p,ea)$ und $er'=(t',p',ea')$ zwei elementare Rechte und sei PRIO eine linear geordnete Menge.

basic_e_conflict(er,er') $ea=ea' \quad t=\neg t' \quad p=p'$.

Der Begriff des Basiskonflikts ist enger verknüpft mit Entwurfsentscheidungen und darüberhinaus technisch einfacher als der Begriff des aktuellen Konflikts.

Beispiel: Die Spezifikation der zwei Rechte

$sr_1 = (\text{Verbot}, 60, \text{hendrik}, \text{transplantieren}, \text{herz})$ und
 $sr_2 = (\text{Erlaubnis}, 60, \text{Chirurg}, \text{transplantieren}, \text{herz})$

erzeugt einen aktuellen Konflikt zwischen den expliziten Rechten

$(\text{Verbot}, \text{hendrik}, \text{transplantieren}, \text{herz})$ und
 $(\text{Erlaubnis}, \text{hendrik}, \text{transplantieren}, \text{herz})$,

sofern es kein spezifiziertes Recht gibt mit höherer Priorität, das die elementare Handlung $(\text{hendrik}, \text{transplantieren}, \text{herz})$ überdeckt.

Gibt es zusätzlich z.B. das spezifizierte Recht

$sr_3 = (\text{Erlaubnis}, 70, \text{Chirurg}, \text{Med. Operation}, \text{herz})$,

so verschwindet der aktuelle Konflikt, allerdings bleibt ein Basiskonflikt zwischen den elementare Rechten

$(\text{Verbot}, 60, \text{hendrik}, \text{transplantieren}, \text{herz})$ und
 $(\text{Erlaubnis}, 60, \text{hendrik}, \text{transplantieren}, \text{herz})$

bestehen.

Dieser latente Basiskonflikt wird zu einem aktuellen Konflikt, wenn sr_3 gelöscht wird.

Wir unterscheiden also zwei verschiedene Konfliktbegriffe:

Aktuelle Konflikte sind diejenigen Konflikte, die unmittelbar wirksam sind.

Basiskonflikte schließen außerdem diejenigen Konflikte ein, die unterdrückt werden durch ein weiteres Recht mit höherer Priorität.

Ein **latenter Konflikt** ist dann ein nicht-aktueller Basiskonflikt.

Natürlich muss man sich entscheiden, ob man nur die aktuellen oder auch latente Konflikte erkennen will.

Obwohl es für die Rechteausswertung ausreicht, sich auf die Erkennung von aktuellen Rechtskonflikten zu beschränken, ist es aus Gründen des konsistenten Rechteentwurfs wünschenswert, bereits alle Basiskonflikte zu enttarnen.

Vielleicht ist eine flexible Strategie am sinnvollsten:

Aktuelle Konflikte führen zu einem Fehler und latente Konflikte zu einer Warnung, so dass der Rechteevertwarter benachrichtigt ist, aber nicht gezwungen wird, unmittelbar zu reagieren.

7.4.2.2 Konflikterkennung bei spezifizierten Rechten

Aus Effizienzgründen ziehen wir Konfliktbegriffe vor, die auf der Ebene der spezifizierten Rechte definiert sind. Damit können wir eine Menge von spezifizierten Rechten unmittelbar auf Konflikte überprüfen.

Die grundlegende Idee für unsere nachfolgenden Definitionen ist die, dass eine Menge von spezifizierten, hierarchiefreien oder elementaren Rechten in einem (aktuellen bzw. Basis-) Konflikt stehen, wenn die Menge der daraus erzeugten (expliziten bzw. elementaren) Rechte einen (aktuellen bzw. Basis-) Konflikt enthält.

Definition: Sei $ER' \subseteq ER$ eine Menge elementarer Rechte und $er, er' \in ER'$.

actual_e_conflict(er, er', ER') : $er, er' \in \text{maximal_priority}(ER')$ und
 $\text{actual_x_conflict}(\text{forget_priority}(er), \text{forget_priority}(er'))$

Definition: Sei $HR' \subseteq HR$ eine Menge hierarchiefreier Rechte und $hr, hr' \in HR'$.

actual_h_conflict(hr, hr', HR') :
 $er \in \text{elementary_rights}(hr)$: $er' \in \text{elementary_rights}(hr')$:
 $\text{actual_e_conflict}(er, er', \text{elementary_rights}(HR'))$

Definition: Sei $SR' \subseteq SR$ eine Menge spezifizierter Rechte und $sr, sr' \in SR'$.

actual_s_conflict(sr, sr', SR') :
 $hr \in \text{hierarchyfree_rights}(sr)$: $hr' \in \text{hierarchyfree_rights}(sr')$:
 $\text{actual_h_conflict}(hr, hr', \text{hierarchyfree_rights}(SR'))$

Die letzten beiden Definitionen kann man vorteilhaft zusammenfassen:

Definition: Sei $SR' \subseteq SR$ eine Menge spezifizierter Rechte und $sr, sr' \in SR'$.

actual_s_conflict(sr, sr', SR') :
 $er \in \text{elementary_rights}^*(sr)$: $er' \in \text{elementary_rights}^*(sr')$:
 $\text{actual_e_conflict}(er, er', \text{elementary_rights}^*(SR'))$

Analog definiert man die Basiskonflikte.

Ein Basiskonflikt hängt dabei nur von den beiden betroffenen Rechten ab, nicht von der Gesamtheit der für eine Anwendung geltenden Rechte:

Definition: Seien hr und hr' zwei hierarchiefreie Rechte.

basic_h_conflict(hr, hr') :

er elementary_rights(hr): er' elementary_rights(hr'): basic_e_conflict(er, er')

Definition: Seien sr und sr' zwei spezifizierte Rechte.

basic_s_conflict(sr, sr') :

hr hierarchyfree_rights(sr): hr' hierarchyfree_rights(sr'): basic_h_conflict(hr, hr')

Auch hier lassen sich beide Definitionen vorteilhaft zusammenfassen:

Definition: Seien sr und sr' zwei spezifizierte Rechte.

basic_s_conflict(sr, sr') :

er elementary_rights*(sr): er' elementary_rights*(sr'): basic_e_conflict(er, er')

Für unsere verschiedenen Konfliktbegriffe benutzen wir
–basierend auf der jeweiligen Definition des Konfliktes zwischen zwei Rechten $\text{conflict}(r,r')$ –
auch wie folgt abgeleitete Notationen für Konflikte einer Rechtemenge $\text{conflict}(R)$ und für die zu einem Recht
in Konflikt stehende Rechtemenge $\text{conflict}(r',R)$:

Sei **conflict**(r,r') eine der obigen Konfliktdefinitionen.

conflict(R) := $\{r,r' \mid R: \text{conflict}(r,r')\}$
Insbesondere gilt dann: $\text{conflict}(r,r') \subseteq \text{conflict}(\{r,r'\})$.

conflict(r',R) := $\{r \mid R \mid \text{conflict}(r,r')\}$

Die Konfliktrelation ist offensichtlich irreflexiv, symmetrisch und nicht transitiv.

In den nachfolgenden Kapiteln beschreiben wir effizient berechenbare Charakterisierungen von Konflikten
(insbesondere zwischen spezifizierten Rechten).

7.4.2.3 Charakterisierungen von aktuellen Konflikten

Einen aktuellen Konflikt zwischen zwei elementaren Rechten können wir wie folgt charakterisieren:

Lemma: Sei $ER' \subseteq ER$ eine Menge elementarer Rechte und $er=(t,p,ea)$, $er'=(t',p',ea') \in ER'$. Dann gilt:
 $actual_e_conflict(er,er',ER')$

$$\begin{aligned} & (t,p,ea), (\neg t,p',ea) \text{ maximal_priority}(ER') \\ & ea=ea' \quad t=\neg t' \quad (p=p' \vee p, p' \text{ unvergleichbar}) \\ & er''=(t'',p'',ea) \in ER': \neg(p''>p) \quad \neg(p''>p') \end{aligned}$$

Beweis: $actual_e_conflict(er,er',ER')$

$$\text{(Def. 6.5)} \quad er,er' \text{ maximal_priority}(ER')$$

$$actual_x_conflict(\text{forget_priority}(er), \text{forget_priority}(er'))$$

$$\text{(Def. 5.10 (forget_priority))} \quad er,er' \text{ maximal_priority}(ER') \quad actual_x_conflict((t,ea), (t',ea'))$$

$$\text{(Def. 6.1)} \quad er,er' \text{ maximal_priority}(ER') \quad ea=ea' \quad t=\neg t'$$

(Daraus folgt unmittelbar die erste Behauptung.)

$$ea=ea' \quad t=\neg t' \quad er''=(t'',p'',ea) \in ER': \neg(p''>p) \quad \neg(p''>p')$$

Da (t,p,ea) und $(t',p',ea) \in ER'$, folgt insbesondere: $\neg(p'>p) \wedge \neg(p>p')$, also $p=p'$ oder p, p' unvergleichbar. Wir erhalten also insgesamt:

$$\begin{aligned} & ea=ea' \quad t=\neg t' \quad (p=p' \vee p, p' \text{ unvergleichbar}) \\ & er''=(t'',p'',ea) \in ER': \neg(p''>p) \quad \neg(p''>p') \end{aligned} \quad \bullet$$

Wir erhalten damit auch formal:
 Jeder aktuelle Konflikt ist ein Basiskonflikt.

Für linear geordnete Prioritäten vereinfacht sich die Konfliktcharakterisierung:

Korollar: Sei $ER' \subseteq ER$ eine Menge elementarer Rechte und $er=(t,p,ea)$, $er'=(t',p',ea') \in ER'$ und sei $PRIO$ eine linear geordnete Menge. Dann gilt:

$$\text{actual_e_conflict}(er,er',ER') \\
 (t,p,ea),(\neg t,p,ea) \text{ maximal_priority}(ER') \\
 ea=ea' \quad t=\neg t' \quad p=p' \quad er''=(t'',p'',ea') \quad ER': \neg(p''>p)$$

Beweis: Da die Menge der Prioritäten jetzt linear geordnet ist, kann der Fall "p, p' unvergleichbar" nicht vorkommen. Es gilt also $p=p'$. Damit vereinfacht sich das Lemma zu dem Korollar. •

Für spezifizierte Rechte erhalten wir die folgende Charakterisierung eines aktuellen Konflikts:

Lemma: Sei $SR' \subseteq SR$ eine Menge spezifizierter Rechte und $sr=(t,p,sa)$, $sr'=(t',p',sa') \in SR'$. Dann gilt:

$$\text{actual_s_conflict}(sr,sr',SR') \\
 t=\neg t' \quad (p=p' \quad p, p' \text{ unvergleichbar}) \\
 ea \in \text{elementary_actions}^*(t,sa) \quad \text{elementary_actions}^*(t',sa'): \\
 sr''=(t'',p'',sa'') \quad SR': \\
 ea \in \text{elementary_actions}^*(t'',sa'') \quad \neg(p''>p) \quad \neg(p''>p')$$

Beweis: $\text{actual_s_conflict}(sr, sr', SR')$

(Def. 6.8)

er elementary_rights*(sr):

er' elementary_rights*(sr'):

$\text{actual_e_conflict}(er, er', \text{elementary_rights}^*(SR'))$

(t, p, ea) elementary_rights*((t, p, sa)):

(t', p', ea') elementary_rights*((t', p', sa')):

$\text{actual_e_conflict}((t, p, ea), (t', p', ea'), \text{elementary_rights}^*(SR'))$

(Def. 5.14 (elementary_rights*))

ea elementary_actions*(t.sa):

ea' elementary_actions*(t'.sa'):

$\text{actual_e_conflict}((t, p, ea), (t', p', ea'), \text{elementary_rights}^*(SR'))$

(Lemma 6.12)

ea elementary_actions*(t.sa):

ea' elementary_actions*(t'.sa'):

$ea=ea'$ $t=\neg t'$ ($p=p'$ p, p' unvergleichbar)

$er''=(t'', p'', ea)$ elementary_rights*(SR'): $\neg(p''>p)$ $\neg(p''>p')$

($ea=ea'$)

$t=\neg t'$ ($p=p'$ p, p' unvergleichbar)

ea elementary_actions*(t.sa) elementary_actions*(t'.sa'):

$er''=(t'', p'', ea)$ elementary_rights*(SR'): $\neg(p''>p)$ $\neg(p''>p')$

(Def. 5.14 (elementary_rights*))

$t = \neg t' \quad (p = p' \quad p, p' \text{ unvergleichbar})$

$ea \text{ elementary_actions}^*(t, sa) \quad \text{elementary_actions}^*(t', sa')$:

$sr'' = (t'', p'', sa'') \quad SR'$:

$ea \text{ elementary_actions}^*(t'', sa'') \quad \neg(p'' > p) \quad \neg(p'' > p')$ •

Korollar: Sei $SR' \subseteq SR$ eine Menge spezifizierter Rechte und $sr = (t, p, sa)$, $sr' = (t', p', sa') \in SR'$ und sei $PRIO$ eine linear geordnete Menge. Dann gilt:

$actual_s_conflict(sr, sr', SR')$

$t = \neg t' \quad p = p'$

$ea \text{ elementary_actions}^*(t, sa) \quad \text{elementary_actions}^*(t', sa')$:

$sr'' = (t'', p'', sa'') \in SR'$: $ea \text{ elementary_actions}^*(t'', sa'') \quad \neg(p'' > p)$

Beweis: Wie obiges Korollar 6.13. •

7.4.2.4 Charakterisierungen von Basiskonflikten

Basiskonflikte sind konzeptuell einfacher und daher auch einfacher zu erkennen:

Lemma: Seien $sr = (t, p, sa)$ und $sr' = (t', p', sa')$ spezifizierte Rechte.

Dann gilt:

$basic_s_conflict(sr, sr')$

$t = \neg t' \quad \neg(p < p') \quad \neg(p' < p)$

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

Beweis: $\text{basic_s_conflict}(sr, sr')$

(Def. 6.11)

er elementary_rights*(sr):
 er' elementary_rights*(sr'): $\text{basic_e_conflict}(er, er')$

(t, p, ea) elementary_rights*(t, p, sa):
 (t', p', ea') elementary_rights*(t', p', sa):
 $\text{basic_e_conflict}((t, p, ea), (t', p', ea'))$

(Def. 5.14 (elementary_rights*))

ea elementary_actions*(t, sa):
 ea' elementary_actions*(t', sa):
 $\text{basic_e_conflict}((t, p, ea), (t', p', ea'))$

(Def. 6.2)

ea elementary_actions*(t, sa):
 ea' elementary_actions*(t', sa):
 $ea=ea' \quad t=\neg t' \quad \neg(p < p') \quad \neg(p' < p)$

($ea=ea'$)

$t=\neg t' \quad \neg(p < p') \quad \neg(p' < p)$
 ea elementary_actions*(t, sa) elementary_actions*(t', sa')

$t=\neg t' \quad \neg(p < p') \quad \neg(p' < p)$
elementary_actions*(t, sa) elementary_actions*(t', sa') \emptyset •

Für lineare Prioritätsordnungen gilt dann: Wir haben einen Basiskonflikt zwischen zwei spezifizierten Rechten, wenn beide spezifizierte Rechte eine gemeinsame Handlung überdecken, die gleiche Priorität, aber invertierte Rechtekennungen haben.

Korollar: Seien $sr=(t,p,sa)$ und $sr'=(t',p',sa')$ spezifizierte Rechte und sei PRIO eine linear geordnete Menge. Dann gilt:

$basic_s_conflict(sr, sr')$

$t=\neg t' \quad p=p' \quad elementary_actions^*(t,sa) \quad elementary_actions^*(t',sa') \quad \emptyset$

Basiskonflikte können effizient charakterisiert werden durch das folgende

Theorem: Sei $SR' \subseteq SR$ eine Menge von spezifizierten Rechten und $sr'=(t',p',ss',so',sg') \in SR'$. Seien alle Klassen nicht-leer und alle Objekte gehören genau einer Klasse an. Dann ist

$basic_s_conflict((t',p',ss',so',sg'),SR') =$

$\{(t,p,ss,so,sg) \in SR' \mid t=\neg t' \quad \neg(p < p') \quad \neg(p' < p)$

if $is_object(ss')$

then if $is_object(ss)$ then $ss'=ss$

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

else if $is_object(ss)$

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

else if $t'=Erlaubnis$ then $ss' \subseteq ss$ else $ss' \subseteq ss$ fi fi fi

if $is_object(so')$

then if $is_object(so)$ then $so'=so$

else if $t'=Erlaubnis$ then $classes(so') \subseteq so$ else $classes(so') \subseteq so$ fi fi

else if $is_object(so)$

then if $t'=Erlaubnis$ then $so' \subseteq classes(so)$ else $so' \subseteq classes(so)$ fi

else if $t'=Erlaubnis$ then $so' \subseteq so$ else $so' \subseteq so$ fi fi fi

if $is_object(sg')$

then if $is_object(sg)$ then $sg'=sg$ else $classes(sg') \subseteq sg$ fi

else if $is_object(sg)$ then $sg' \subseteq classes(sg)$

else $sg'' \subseteq GC : sg'' \subseteq sg \quad sg'' \subseteq sg'$ fi fi }

Beweis: $\text{basic_s_conflict}(sr',SR')$

$\{sr \ SR' \mid \text{basic_s_conflict}(sr,sr')\}$

(Lemma 6.16)

$\{sr \ SR' \mid t=\neg t' \quad \neg(p<p') \quad \neg(p'<p)$
 $\quad ea \ \text{elementary_actions}^*(t,sa) \quad \text{elementary_actions}^*(t',sa')\}$

Insgesamt ergibt sich dann der Satz aus Satz 5.27. •

Im Falle einer linearen Prioritätsordnung vereinfacht sich die Bedingung an die Prioritäten wieder zu $p=p'$.

Beispiel:

Sei SR_1 die Menge der spezifizierten Rechte wie in Abschnitt 7.2.4 gegeben. Wir möchten das Recht sr' =(Erlaubnis, 20, catherine, Diagnose, Haut) hinzufügen.

```
basic_s_conflict((Erlaubnis, 20, catherine, Diagnose, Haut),SR1)
= {(t,p,ss,so,sg) SR1 | t = Verbot  p=20
    if is_object(ss) then ss=catherine else classes(catherine)  ss fi
    if is_object(so) then Diagnose  classes(so) else Diagnose  so fi
    if is_object(sg) then Haut  classes(sg)
        else ( sg'' GC : sg'' sg  sg'' Haut) fi }
= {(t,p,ss,so,sg) SR1 | t = Verbot  p=20
    if is_object(ss) then ss=catherine else Hautarzt  ss fi
    if is_object(so) then Diagnose=classes(so) else Diagnose=so fi
    if is_object(sg) then Haut=classes(sg) else Haut  sg fi }
= {(t,p,ss,so,sg) SR1 | t = Verbot  p=20
    [ss=catherine  ss {Hautarzt}]
    [classes(so) {Diagnose}  so {Diagnose}]
    [classes(sg) {Haut}
    sg {Haut, Kopf, Rumpf, Gliedmaßen, Körper}]}}
= ∅
```

Durch die Einfügung von sr' entsteht also kein Konflikt.

7.4.2.5 Unabhängige Teilbedingungen für Basiskonflikte

Wie man leicht aus den bisherigen Betrachtungen erkennt, setzt sich eine Bedingung für Konflikte aus drei Teilen zusammen:

- (1) Der erste Teil betrifft die Handlungen: diese Teilbedingung gibt an, ob zwei Rechte gemeinsam eine elementare Handlung betreffen.
Wir können diese Teilbedingung noch weiter zerlegen in Bedingungen an die einzelnen Handlungskategorien. Wir brauchen dann nur noch zwischen gleich- und gegenläufiger Klassenhierarchieinterpretation zu unterscheiden.
- (2) Der zweite Teil betrifft die Rechteerkennung: solange wir allerdings nur Erlaubnisse und Verbote betrachten, ist diese Teilbedingung recht einfach, nämlich $t = \neg t'$.
Dies kann sich jedoch ändern, wenn wir weitere Rechtekennungen (z.B. Pflichten und Freiheiten) hinzunehmen und es dann z.B. auch Konflikte zwischen Verboten und Pflichten geben kann.
- (3) Der dritte Teil betrifft die Prioritäten: betrachten wir nur lineare Prioritäten, so ist diese Teilbedingung wieder recht einfach, nämlich $p = p'$.
Wenn wir partielle Prioritätsordnungen verwenden, ändert sich diese Teilbedingung zu:
 $(\neg(p < p') \quad \neg(p' < p))$.

Wir haben unser Rechtssystem so konstruiert, dass bezüglich Basiskonflikten diese drei Teilbedingungen unabhängig voneinander sind.