

Automatisierte Logik und Programmierung

Prof. Chr. Kreitz

Universität Potsdam, Theoretische Informatik — Wintersemester 2016/17

Blatt 2 — Abgabetermin: 10.11.2016

Das zweite Übungsblatt soll dazu dienen, Erfahrungen mit der Entwicklung von Beweisen in Refinement Logik und prädikatenlogischer Evidenz zu sammeln und sich mit Fragen der Korrektheit von Kalkülen auseinanderzusetzen. Wir werden mögliche Lösungen zu Beginn der Veranstaltung am 10.11.2016 besprechen

Aufgabe 2.1 (Aussagenlogisches Beweisen in Refinement Logik)

Konstruieren Sie für die folgenden aussagenlogischen Formeln einen Beweis in Refinement Logik und extrahieren Sie aus dem Beweis einen Evidenzterm. Wenn es nicht möglich ist, einen Beweis zu entwickeln, geben Sie eine kurze Erklärung, warum der Beweis nicht vollendet werden kann.

2.1-a $(P \Rightarrow Q) \Rightarrow (\neg Q \Rightarrow \neg P)$:

2.1-b $(P \Rightarrow Q) \Rightarrow (\neg P \Rightarrow \neg Q)$:

2.1-c $\neg(P \vee Q) \Rightarrow (\neg P \wedge \neg Q)$:

2.1-d $(\neg P \wedge \neg Q) \Rightarrow \neg(P \vee Q)$:

2.1-e $\neg(\neg P \wedge \neg Q) \Rightarrow (P \vee Q)$:

2.1-f $\neg\neg P \Rightarrow P$:

2.1-g $\neg P \vee P$:

2.1-h $\neg(P \vee \neg P)$:

2.1-i $\neg\neg(P \vee \neg P)$:

2.1-j $(P \vee (Q \wedge R)) \Rightarrow (P \vee Q) \wedge (P \vee R)$:

Aufgabe 2.2 (Evidenz in Prädikatenlogik)

Bestimmen Sie mögliche Evidenzterme für die folgenden prädikatenlogischen Formeln. Wenn es nicht möglich ist, Evidenz zu konstruieren, geben Sie eine kurze Begründung oder ein Gegenbeispiel für die Gültigkeit der Formel an.

2.2-a $((\forall x)Px \wedge (\forall x)Qx) \Rightarrow ((\forall x)(Px \wedge Qx))$:

2.2-b $((\forall x)Px \vee (\forall x)Qx) \Rightarrow ((\forall x)(Px \vee Qx))$:

2.2-c $((\forall x)(Px \vee Qx)) \Rightarrow ((\forall x)Px \vee (\forall x)Qx)$:

2.2-d $((\exists x)(Px \wedge Qx)) \Rightarrow ((\exists x)Px \wedge (\exists x)Qx)$:

2.2-e $((\exists x)Px \wedge (\exists x)Qx) \Rightarrow ((\exists x)(Px \wedge Qx))$:

2.2-f $((\exists x)Px \vee (\exists x)Qx) \Rightarrow ((\exists x)(Px \vee Qx))$:

2.2-g $((\exists x)(Px \vee Qx)) \Rightarrow ((\exists x)Px \vee (\exists x)Qx)$:

2.2-h $(\exists x)(Px \Rightarrow (\forall y)Py)$:

2.2-i $(\forall x)((\forall y)Py \Rightarrow Px)$:

2.2-j $(\exists x)((\exists y)Py \Rightarrow Px)$:

$$2.2\text{-k} \quad \neg((\exists x)Px) \Rightarrow ((\forall x)((\exists y)Py \Rightarrow Px)):$$

$$2.2\text{-l} \quad ((\exists x)Px) \Rightarrow ((\forall x)(Px \Rightarrow Qx) \Rightarrow ((\exists y)Qy)):$$

$$2.2\text{-m} \quad \neg((\exists x)Px) \Rightarrow ((\forall x)\neg(Px)):$$

$$2.2\text{-n} \quad ((\forall x)\neg(Px)) \Rightarrow \neg((\exists x)Px):$$

$$2.2\text{-o} \quad ((\exists x)Px) \Rightarrow \neg((\forall x)\neg(Px)):$$

$$2.2\text{-p} \quad \neg((\forall x)Px) \Rightarrow ((\exists x)\neg(Px)):$$

Aufgabe 2.3 (Beweise in Prädikatenlogik)

Konstruieren Sie für die Formeln aus der vorhergehenden Aufgabe einen Beweis in Refinement Logik und extrahieren Sie aus dem Beweis einen Evidenzterm. Vergleichen Sie den extrahierten Term mit dem “von Hand” konstruierten. Wenn es nicht möglich ist, einen Beweis zu entwickeln, geben Sie eine kurze Erklärung, warum der Beweis nicht vollendet werden kann.

Aufgabe 2.4 (Korrektheit der Refinement Logik)

Zeigen Sie, daß die prädikatenlogische Refinement Logik korrekt ist.

- 2.4-a Beweisen Sie, daß die Regeln `impliesR`, `impliesL`, `orL` und `exL` korrekt sind, und nehmen Sie im folgenden an, daß alle anderen Regeln ebenfalls als korrekt bewiesen sind.
- 2.4-b Beweisen Sie durch Induktion über Beweisbäume: “Wenn alle Regeln eines Kalküls korrekt sind, dann haben alle Theoreme gültige Initialsequenzen als Wurzeln”
- 2.4-c Beweisen Sie: “Eine Formel C ist gültig, wenn die Initialsequenz $\vdash C$ gültig ist”
- 2.4-d Zeigen Sie: “Eine Formel C ist gültig, wenn $\vdash C$ in Refinement Logik beweisbar ist”

Lösung 2.1

Lösungen werden interaktiv mit Nuprl vorgeführt

Lösung 2.2

2.2-a $((\forall x)Px \wedge (\forall x)Qx) \Rightarrow ((\forall x)(Px \wedge Qx))$: The evidence is $\lambda z. (\lambda x. (z_1 x, z_2 x))$

2.2-b $((\forall x)Px \vee (\forall x)Qx) \Rightarrow ((\forall x)(Px \vee Qx))$:
The evidence is $\lambda z. (\lambda x. (\text{case } z \text{ of } \text{inl}(f) \rightarrow \text{inl}(f x) \mid \text{inr}(g) \rightarrow \text{inr}(g x)))$

2.2-c $((\forall x)(Px \vee Qx)) \Rightarrow ((\forall x)Px \vee (\forall x)Qx)$: This formula is not valid.
A possible counterexample is Px being $\text{odd}(x)$ and Qx being $\text{even}(x)$.

2.2-d $((\exists x)(Px \wedge Qx)) \Rightarrow ((\exists x)Px \wedge (\exists x)Qx)$: The evidence is $\lambda z. ((z_1, z_{21}), (z_1, z_{22}))$

2.2-e $((\exists x)Px \wedge (\exists x)Qx) \Rightarrow ((\exists x)(Px \wedge Qx))$: This formula is not valid.
A possible counterexample is Px being $\text{odd}(x)$ and Qx being $\text{even}(x)$.

2.2-f $((\exists x)Px \vee (\exists x)Qx) \Rightarrow ((\exists x)(Px \vee Qx))$:
The evidence is $\lambda z. (\text{case } z \text{ of } \text{inl}(x) \rightarrow (x_1, \text{inl}(x_2)) \mid \text{inr}(y) \rightarrow (y_1, \text{inr}(y_2)))$

2.2-g $((\exists x)(Px \vee Qx)) \Rightarrow ((\exists x)Px \vee (\exists x)Qx)$:
The evidence is $\lambda z. (\text{case } z_2 \text{ of } \text{inl}(x) \rightarrow \text{inl}(z_1, x) \mid \text{inr}(y) \rightarrow \text{inr}(z_1, y))$

2.2-h $(\exists x)(Px \Rightarrow (\forall y)Py)$: This is classically true. Either we can pick an x with $\neg(Px)$ or $(\forall y)Py$
The evidence must be a pair (a, f) where $a:\mathbb{U}$ and f is a function that takes evidence p for Pa as input and generates a function $g : (x:\mathbb{U} \rightarrow [Px])$. There is no way to construct the generic “evidence function” g solely on the basis of the evidence for a specific Pa .

2.2-i $(\forall x)((\forall y)Py \Rightarrow Px)$: The evidence is $\lambda x. (\lambda f. f x)$

2.2-j $(\exists x)((\exists y)Py \Rightarrow Px)$: This is classically true. we select x to be the y
The evidence must be a pair (a, f) where $a:\mathbb{U}$ and f is a function that takes a pair $(b, p) : (y:\mathbb{U} \times ([Py]))$ as input and generates evidence for Pa . The only way to construct f is to make sure that a matches its input b – then the evidence for Pa would be the second input component. There is no way to do this without knowing the input.

2.2-k $\neg((\exists x)Px) \Rightarrow ((\forall x)((\exists y)Py \Rightarrow Px))$: The evidence is $\lambda f. (\lambda x. (\lambda z. \text{any}(f z)))$

2.2-l $((\exists x)Px) \Rightarrow ((\forall x)(Px \Rightarrow Qx) \Rightarrow ((\exists y)Qy))$: The evidence is $\lambda z. (\lambda f. (z_1, (f z_1) z_2))$

2.2-m $\neg((\exists x)Px) \Rightarrow ((\forall x)\neg(Px))$:
The evidence must be a function that takes as input a function $f : (x:\mathbb{U} \times [Px]) \rightarrow \{\}$ and produces a function $g : (x:\mathbb{U} \rightarrow ([Px] \rightarrow \{\}))$. To construct g we take an input $x:\mathbb{U}$ and evidence $p : [Px]$ and create the element of $\{\}$ by applying f to the pair (x, p) . Thus the overall evidence is $\lambda f. (\lambda x. (\lambda p. f(x, p)))$ In a sense this is dependent currying - compare evidence

2.2-n $((\forall x)\neg(Px)) \Rightarrow \neg((\exists x)Px)$: The evidence is $\lambda f. (\lambda z. (f z_1) z_2)$

2.2-o $((\exists x)Px) \Rightarrow \neg((\forall x)\neg(Px))$: The evidence is $\lambda z. (\lambda f. (f z_1) z_2)$

2.2-p $\neg((\forall x)Px) \Rightarrow ((\exists x)\neg(Px))$:
The evidence must be a function that takes as input a function $f : (x:\mathbb{U} \rightarrow [Px]) \rightarrow \{\}$ and produces a pair $z : (x:\mathbb{U} \times ([Px] \rightarrow \{\}))$. To construct g we have to find an element $a:\mathbb{U}$ and evidence p for Pa . There is no uniform way to construct these two evidences solely from f .

Lösung 2.3

2.3-a $((\forall x)(Px \wedge Qx)) \Rightarrow ((\forall x)Px \wedge (\forall x)Qx)$:

$\vdash ((\forall x)(Px \wedge Qx)) \Rightarrow ((\forall x)Px \wedge (\forall x)Qx)$	by impliesR
1 $(\forall x)(Px \wedge Qx) \vdash ((\forall x)Px \wedge (\forall x)Qx)$	by andR
1.1 $(\forall x)(Px \wedge Qx) \vdash (\forall x)Px$	by allR
1.1.1 $(\forall x)(Px \wedge Qx) \vdash Pa$	by allL a
1.1.1.1 $(\forall x)(Px \wedge Qx), Pa \wedge Qa \vdash Pa$	by andL
1.1.1.1.1 $(\forall x)(Px \wedge Qx), Pa, Qa \vdash Pa$	by axiom
1.2 $(\forall x)(Px \wedge Qx) \vdash (\forall x)Qx$	by allR
1.2.1 $(\forall x)(Px \wedge Qx) \vdash Qa$	by allL a
1.2.1.1 $(\forall x)(Px \wedge Qx), Pa \wedge Qa \vdash Qa$	by andL
1.2.1.1.1 $(\forall x)(Px \wedge Qx), Pa, Qa \vdash Qa$	by axiom

The evidence extracted from this proof is $\lambda f. (\lambda x. (f x)_1, \lambda x. (f x)_2)$ 2.3-b $((\forall x)Px \wedge (\forall x)Qx) \Rightarrow ((\forall x)(Px \wedge Qx))$:

$\vdash ((\forall x)Px \wedge (\forall x)Qx) \Rightarrow ((\forall x)(Px \wedge Qx))$	by impliesR
1 $(\forall x)Px \wedge (\forall x)Qx \vdash (\forall x)(Px \wedge Qx)$	by andL
1.1 $(\forall x)Px, (\forall x)Qx \vdash (\forall x)(Px \wedge Qx)$	by allR
1.1.1 $(\forall x)Px, (\forall x)Qx \vdash Pa \wedge Qa$	by andR
1.1.1.1 $(\forall x)Px, (\forall x)Qx \vdash Pa$	by allL a
1.1.1.1.1 $(\forall x)Px, (\forall x)Qx, Pa \vdash Pa$	by axiom
1.1.1.1.2 $(\forall x)Px, (\forall x)Qx \vdash Qa$	by allL a
1.1.1.2.1 $(\forall x)Px, (\forall x)Qx, Qa \vdash Qa$	by axiom

2.3-c $((\forall x)Px \vee (\forall x)Qx) \Rightarrow ((\forall x)(Px \vee Qx))$:

$\vdash ((\forall x)Px \vee (\forall x)Qx) \Rightarrow ((\forall x)(Px \vee Qx))$	by impliesR
1 $(\forall x)Px \vee (\forall x)Qx \vdash (\forall x)(Px \vee Qx)$	by orL
1.1 $(\forall x)Px \vee (\forall x)Qx \vdash Pa \vee Qa$	by allR
1.1.1 $(\forall x)Px \vdash Pa \vee Qa$	by allL a
1.1.1.1 $Pa \vdash Pa \vee Qa$	by orR1
1.1.1.1.1 $Pa \vdash Pa$	by axiom
1.1.2 $(\forall x)Qx \vdash Pa \vee Qa$	by
1.1.2.1 $Qa \vdash Pa \vee Qa$	by orR2
1.1.2.1.1 $Qa \vdash Qa$	by axiom

2.3-d $((\forall x)(Px \vee Qx)) \Rightarrow ((\forall x)Px \vee (\forall x)Qx)$: A proof attempt will get stuck

$\vdash ((\forall x)(Px \vee Qx)) \Rightarrow ((\forall x)Px \vee (\forall x)Qx)$	by impliesR
1 $(\forall x)(Px \vee Qx) \vdash ((\forall x)Px \vee (\forall x)Qx)$	by ???

At this point we have to prove either $(\forall x)Px$ or $(\forall x)Qx$ but there is no way to prove that.2.3-e $((\exists x)(Px \wedge Qx)) \Rightarrow ((\exists x)Px \wedge (\exists x)Qx)$:

$\vdash ((\exists x)(Px \wedge Qx)) \Rightarrow ((\exists x)Px \wedge (\exists x)Qx)$	by impliesR
1 $(\exists x)(Px \wedge Qx) \vdash ((\exists x)Px \wedge (\exists x)Qx)$	by exL
1.1 $Pa \wedge Qa \vdash ((\exists x)Px \wedge (\exists x)Qx)$	by andL
1.1.1 $Pa, Qa \vdash ((\exists x)Px \wedge (\exists x)Qx)$	by andR
1.1.1.1 $Pa, Qa \vdash (\exists x)Px$	by exR a
1.1.1.1.1 $Pa, Qa \vdash Pa$	by axiom
1.1.1.1.2 $Pa, Qa \vdash (\exists x)Qx$	by exR a
1.1.1.2.1 $Pa, Qa \vdash Qa$	by axiom

2.3-f $((\exists x)Px \wedge (\exists x)Qx) \Rightarrow ((\exists x)(Px \wedge Qx))$: Here is a proof attempt

$\vdash ((\exists x)Px \wedge (\exists x)Qx) \Rightarrow ((\exists x)(Px \wedge Qx))$	by impliesR
1 $(\exists x)Px \wedge (\exists x)Qx \vdash (\exists x)(Px \wedge Qx)$	by andL
1.1 $(\exists x)Px, (\exists x)Qx \vdash (\exists x)(Px \wedge Qx)$	by exL
1.1.1 $Pa, (\exists x)Qx \vdash (\exists x)(Px \wedge Qx)$	by exL
1.1.1.1 $Pa, Qb \vdash (\exists x)(Px \wedge Qx)$	by ???

The proof gets stuck because in the second application of `exL` we will have to use a *new* parameter instead of using *a* again.

2.3-g $((\exists x)Px \vee (\exists x)Qx) \Rightarrow ((\exists x)(Px \vee Qx))$:

$\vdash ((\exists x)Px \vee (\exists x)Qx) \Rightarrow ((\exists x)(Px \vee Qx))$	by impliesR
1 $(\exists x)Px \vee (\exists x)Qx \vdash (\exists x)(Px \vee Qx)$	by orL
1.1 $(\exists x)Px \vdash (\exists x)(Px \vee Qx)$	by exL
1.1.1 $Pa \vdash (\exists x)(Px \vee Qx)$	by exR <i>a</i>
1.1.1.1 $Pa \vdash Pa \vee Qa$	by orR1
1.1.1.1.1 $Pa \vdash Pa$	by axiom
1.2 $(\exists x)Qx \vdash (\exists x)(Px \vee Qx)$	by exL
1.2.1 $Qa \vdash (\exists x)(Px \vee Qx)$	by exR <i>a</i>
1.2.1.1 $Qa \vdash Pa \vee Qa$	by orR2
1.2.1.1.1 $Qa \vdash Qa$	by axiom

2.3-h $((\exists x)(Px \vee Qx)) \Rightarrow ((\exists x)Px \vee (\exists x)Qx)$:

$\vdash ((\exists x)(Px \vee Qx)) \Rightarrow ((\exists x)Px \vee (\exists x)Qx)$	by impliesR
1 $(\exists x)(Px \vee Qx) \vdash (\exists x)Px \vee (\exists x)Qx$	by exL
1.1 $Pa \vee Qa \vdash (\exists x)Px \vee (\exists x)Qx$	by orL
1.1.1 $Pa \vdash (\exists x)Px \vee (\exists x)Qx$	by orR1
1.1.1.1 $Pa \vdash (\exists x)Px$	by exR <i>a</i>
1.1.1.1.1 $Pa \vdash Pa$	by axiom
1.1.2 $Pa \vdash (\exists x)Px \vee (\exists x)Qx$	by orR1
1.1.2.1 $Pa \vdash (\exists x)Px$	by exR <i>a</i>
1.1.2.1.1 $Pa \vdash Pa$	by axiom

2.3-i $(\exists x)(Px \Rightarrow (\forall y)Py)$: Here is a proof attempt

$\vdash (\exists x)(Px \Rightarrow (\forall y)Py)$	by exR <i>a</i>
1 $\vdash Pa \Rightarrow (\forall y)Py$	by impliesR
1.1 $Pa \vdash (\forall y)Py$	by allR
1.1.1 $Pa \vdash Pb$	by ???

The proof gets stuck because in the application of `allR` we will have to use a *new* parameter instead of using *a* again.

2.3-j $(\forall x)((\forall y)Py \Rightarrow Px)$:

$\vdash (\forall x)((\forall y)Py \Rightarrow Px)$	by allR
1 $\vdash (\forall y)Py \Rightarrow Pa$	by impliesR
1.1 $(\forall y)Py \vdash Pa$	by allL <i>a</i>
1.1.1 $Pa \vdash Pa$	by axiom

2.3-k $(\exists x)((\exists y)Py \Rightarrow Px)$:

This is classically true. we select *x* to be the *y*

Here is a proof attempt

$\vdash (\exists x)((\exists y)Py \Rightarrow Px)$	by exR <i>a</i>
1 $\vdash (\exists y)Py \Rightarrow Pa$	by impliesR
1.1 $(\exists y)Py \vdash Pa$	by exL
1.1.1 $Pb \vdash Pa$	by ???

The proof gets stuck because in the application of `exL` we will have to use a *new* parameter instead of using *a* again.

2.3-l $\neg((\exists x)Px) \Rightarrow ((\forall x)((\exists y)Py) \Rightarrow Px)$:

$\vdash \neg((\exists x)Px) \Rightarrow ((\forall x)((\exists y)Py) \Rightarrow Px)$	by impliesR
1 $\neg((\exists x)Px) \vdash (\forall x)((\exists y)Py) \Rightarrow Px$	by allR
1.1 $\neg((\exists x)Px) \vdash ((\exists y)Py) \Rightarrow Pa$	by impliesR
1.1.1 $\neg((\exists x)Px), (\exists y)Py \vdash Pa$	by notL
1.1.1.1 $\neg((\exists x)Px), (\exists y)Py \vdash (\exists x)Px$	by axiom

2.3-m $((\exists x)Px) \Rightarrow ((\forall x)(Px \Rightarrow Qx) \Rightarrow ((\exists y)Qy))$:

$\vdash ((\exists x)Px) \Rightarrow ((\forall x)(Px \Rightarrow Qx) \Rightarrow ((\exists y)Qy))$	by impliesR
1 $(\exists x)Px \vdash (\forall x)(Px \Rightarrow Qx) \Rightarrow ((\exists y)Qy)$	by impliesR
1.1 $(\exists x)Px, (\forall x)(Px \Rightarrow Qx) \vdash (\exists y)Qy$	by exL
1.1.1 $Pa, (\forall x)(Px \Rightarrow Qx) \vdash (\exists y)Qy$	by allL a
1.1.1.1 $Pa, Pa \Rightarrow Qa \vdash (\exists y)Qy$	by exR a
1.1.1.1.1 $Pa, Pa \Rightarrow Qa \vdash Qa$	by impliesL
1.1.1.1.1.1 $Pa, Pa \Rightarrow Qa \vdash Pa$	by axiom
1.1.1.1.1.2 $Pa, Pa \Rightarrow Qa, Qa \vdash Qa$	by axiom

2.3-n $\neg((\exists x)Px) \Rightarrow ((\forall x)\neg(Px))$:

$\vdash \neg((\exists x)Px) \Rightarrow ((\forall x)\neg(Px))$	by impliesR
1 $\neg((\exists x)Px) \vdash (\forall x)\neg(Px)$	by allR
1.1 $\neg((\exists x)Px) \vdash \neg(Pa)$	by notR
1.1.1 $\neg((\exists x)Px), Pa \vdash f$	by notL
1.1.1.1 $\neg((\exists x)Px), Pa \vdash (\exists x)Px$	by exR a
1.1.1.1.1 $\neg((\exists x)Px), Pa \vdash Pa$	by axiom

2.3-o $((\forall x)\neg(Px)) \Rightarrow \neg((\exists x)Px)$:

$\vdash ((\forall x)\neg(Px)) \Rightarrow \neg((\exists x)Px)$	by impliesR
1 $(\forall x)\neg(Px) \vdash \neg((\exists x)Px)$	by notR
1.1 $(\forall x)\neg(Px), (\exists x)Px \vdash f$	by exL
1.1.1 $(\forall x)\neg(Px), Pa \vdash f$	by allL a
1.1.1.1 $\neg(Pa), Pa \vdash f$	by notL
1.1.1.1.1 $\neg(Pa), Pa \vdash Pa$	by axiom

2.3-p $((\exists x)Px) \Rightarrow \neg((\forall x)\neg(Px))$:

$\vdash ((\exists x)Px) \Rightarrow \neg((\forall x)\neg(Px))$	by impliesR
1 $(\exists x)Px \vdash \neg((\forall x)\neg(Px))$	by exL
1.1 $Pa \vdash \neg((\forall x)\neg(Px))$	by notR
1.1.1 $Pa, (\forall x)\neg(Px) \vdash f$	by allL a
1.1.1.1 $Pa, \neg(Pa) \vdash f$	by notL
1.1.1.1.1 $Pa, \neg(Pa) \vdash Pa$	by axiom

2.3-q $\neg((\forall x)Px) \Rightarrow ((\exists x)\neg(Px))$: Here is a proof attempt

$\vdash \neg((\forall x)Px) \Rightarrow ((\exists x)\neg(Px))$	by impliesR
1 $\vdash \neg((\forall x)Px) \vdash ((\exists x)\neg(Px))$	by ???

At this point we're stuck. If we apply **notL** we will lose the conclusion $((\exists x)\neg(Px))$ and have to prove $(\forall x)Px$, which clearly won't work. But there are no other proof rule that can be applied here.

Lösung 2.4

2.4-a Beweisen Sie, daß die Regeln **impliesR**, **impliesL**, **orL** und **exL** korrekt sind, und nehmen Sie im folgenden an, daß alle anderen Regeln ebenfalls als korrekt bewiesen sind.

Die Korrektheit einer Regel $r = (dec, val)$ war wie folgt definiert: Sind $S_i = H_i \vdash C_i$ Ergebnis der Anwendung von dec auf $S = H \vdash C$ und c_i Evidenzterme für S_i , dann ist $val(S, (S_i, c_i))$ Evidenzterm für S . Die konkreten Beweise folgen den Begründungen für die Konstruktion der Evidenzterme in den Regeln.

