Übung zur Vorlesung

Automatisierte Logik und Programmierung I

Prof. Chr. Kreitz

Universität Potsdam, Theoretische Informatik — Wintersemester 2008/09

Blatt 2 — Abgabetermin: 12.11.08 nach der Übung

Das Übungsblatt soll dazu dienen, sich mit dem analytischen Sequenzenkalkül der Refinement Logic und den zugehörigen Themen Substitution, Korrektheit und Gleichheit vertraut zu machen.
Die Aufgaben 2.2-d, 2.2-e und 2.3-c sind schreibintensiv und leichter mit Nuprl zu bearbeiten

---

Aufgabe 2.1 (Substitution)

Führen Sie die folgenden Substitutionen durch:

2.1–a $(\exists y{:}T.P(x,y,z) \Rightarrow \forall z{:}T.Q(x,y,z))[f(a)/x]$

2.1–b $(\exists y{:}T.P(x,y,z) \Rightarrow \forall z{:}T.Q(x,y,z))[f(a)/z]$

2.1–c $(\exists y{:}T.P(x,y,z) \Rightarrow \forall z{:}T.Q(x,y,z))[f(a)/y]$

2.1–d $(\exists y{:}T.P(x,y,z) \Rightarrow \forall z{:}T.Q(x,y,z))[f(z)/x]$

2.1–e $(\exists y{:}T.P(x,y,z) \Rightarrow \forall z{:}T.Q(x,y,z))[f(y)/z]$

2.1–f $(\exists y{:}T.P(x,y,z) \Rightarrow \forall z{:}T.Q(x,y,z))[f(x)/z]$

Aufgabe 2.2 (Refinement Logic)

Beweisen Sie die folgenden Formeln mit Hilfe des Kalküls der Refinement Logic. Beachten Sie, daß manche Beweise nur mit Hilfe der `magic`–Regel an geeigneter Stelle zu führen sind:

2.2–a $(\neg A \vee \neg B) \;\wedge\; A \wedge B \;\Rightarrow\; \forall x{:}T_1.\exists y{:}T_2.\,Q(f(y,g(f(y,g(x))))) \;\vee\; \neg P(f(x,g(y)))$

2.2–b $\neg(\forall x{:}T.P(x)) \;\Rightarrow\; \exists x{:}T.\neg P(x)$

2.2–c $\forall 0{:}\mathbb{N}.(\forall n,m{:}\mathbb{N}.\geq(n,m) \Rightarrow \geq(s(n),m)) \;\wedge\; \geq(s(0),0) \;\Rightarrow\; \geq(s(s(0)),0)$

2.2–d $\quad \forall a,b,c{:}Kiste.Auf(a,b) \wedge Auf(b,c) \wedge Rot(c) \wedge \neg Rot(a)$
$\Rightarrow\; \exists x,y{:}Kiste.Auf(y,x) \wedge \neg Rot(y) \wedge Rot(x)$

2.2–e $\quad (\forall x{:}Mensch.\exists y{:}Mensch.Vater(y,x))$
$\wedge\; (\forall x,y,z{:}Mensch.Vater(y,x) \wedge Vater(z,y) \;\Rightarrow\; Gro\text{ß}Vater(z,x))$
$\Rightarrow\; \forall x{:}Mensch.\exists y{:}Mensch.Gro\text{ß}Vater(y,x)$

Aufgabe 2.3 (Gleichheit)

Beweisen Sie folgende Formeln mit den Regeln der Refinement Logic:

2.3–a $\vdash\; \forall n,m{:}\mathbb{N}.\;\; m=n \;\Rightarrow\; s(n)=s(m)$

2.3–b $\vdash\; \forall a{:}O.\; (g(g(g(a)))=a \;\wedge\; g(g(g(g(g(a)))))=a) \;\Rightarrow\; g(a)=a$

2.3–c $\vdash\; \forall 0{:}\mathbb{N}. \qquad (\forall n,m{:}\mathbb{N}.\;\; plus(n$
$\Rightarrow\quad plus(\, s(s(0)),s(s(s(0)))\,) = s(s(s(s(s(0)))))$

Aufgabe 2.4 (Korrektheit von Regeln)

Beweisen Sie die Korrektheit der folgenden Regeln der Refinement Logic:

2.4–a `orE`

2.4–b `impE`

2.4–c `exE`

Lösung 2.1 *Ziel dieser Aufgabe ist es, aufgrund der Entscheidung, welche Auftreten der Variablen frei bzw. gebunden sind, den jeweils vorliegenden Fall für die Substitutionsregeln zu erkennen:*

2.1–a
$$\begin{aligned} & (\exists y{:}T.P(x,y,z) \Rightarrow \forall z{:}T.Q(x,y,z))[f(a)/x] \\ = \; & \exists y{:}T.(P(x,y,z))[f(a)/x] \Rightarrow (\forall z{:}T.Q(x,y,z))[f(a)/x] \\ = \; & \exists y{:}T.P(f(a),y,z) \Rightarrow \forall z{:}T.(Q(x,y,z))[f(a)/x] \\ = \; & \exists y{:}T.P(f(a),y,z) \Rightarrow \forall z{:}T.Q(f(a),y,z) \end{aligned}$$

2.1–b
$$\begin{aligned} & (\exists y{:}T.P(x,y,z) \Rightarrow \forall z{:}T.Q(x,y,z))[f(a)/z] \\ = \; & \exists y{:}T.(P(x,y,z))[f(a)/z] \Rightarrow (\forall z{:}T.Q(x,y,z))[f(a)/z] \\ = \; & \exists y{:}T.P(x,y,f(a)) \Rightarrow \forall z{:}T.Q(x,y,z) \end{aligned}$$

2.1–c
$$\begin{aligned} & (\exists y{:}T.P(x,y,z) \Rightarrow \forall z{:}T.Q(x,y,z))[f(a)/y] \\ = \; & \exists y{:}T.P(x,y,z) \Rightarrow \forall z{:}T.Q(x,y,z) \end{aligned}$$

2.1–d
$$\begin{aligned} & (\exists y{:}T.P(x,y,z) \Rightarrow \forall z{:}T.Q(x,y,z))[f(z)/x] \\ = \; & \exists y{:}T.(P(x,y,z))[f(z)/x] \Rightarrow (\forall z{:}T.Q(x,y,z))[f(z)/x] \\ = \; & \exists y{:}T.P(f(z),y,z) \Rightarrow (\forall z'{:}T.(Q(x,y,z))[z'/z])[f(z)/x] \\ = \; & \exists y{:}T.P(f(z),y,z) \Rightarrow \forall z'{:}T.(Q(x,y,z'))[f(z)/x] \\ = \; & \exists y{:}T.P(f(z),y,z) \Rightarrow \forall z'{:}T.Q(f(z),y,z') \end{aligned}$$

2.1–e
$$\begin{aligned} & (\exists y{:}T.P(x,y,z) \Rightarrow \forall z{:}T.Q(x,y,z))[f(y)/z] \\ = \; & (\exists y'{:}T.(P(x,y,z) \Rightarrow \forall z{:}T.Q(x,y,z))[y'/y])[f(y)/z] \\ = \; & (\exists y'{:}T.(P(x,y,z))[y'/y] \Rightarrow (\forall z{:}T.Q(x,y,z))[y'/y])[f(y)/z] \\ = \; & (\exists y'{:}T.P(x,y',z) \Rightarrow \forall (z{:}T.Q(x,y,z))[y'/y])[f(y)/z] \\ = \; & (\exists y'{:}T.P(x,y',z) \Rightarrow \forall z{:}T.Q(x,y',z))[f(y)/z] \\ = \; & \exists y'{:}T.(P(x,y',z))[f(y)/z] \Rightarrow (\forall z{:}T.Q(x,y',z))[f(y)/z] \\ = \; & \exists y'{:}T.P(x,y',f(y)) \Rightarrow \forall z{:}T.Q(x,y',z) \end{aligned}$$

2.1–f
$$\begin{aligned} & (\exists y{:}T.P(x,y,z) \Rightarrow \forall z{:}T.Q(x,y,z))[f(x)/z] \\ = \; & \exists y{:}T.(P(x,y,z))[f(x)/z] \Rightarrow (\forall z{:}T.Q(x,y,z))[f(x)/z] \\ = \; & \exists y{:}T.P(x,y,f(x)) \Rightarrow \forall z{:}T.Q(x,y,z) \end{aligned}$$

Lösung 2.2

Ziel dieser Aufgabe ist es, sich mit formalen Beweisen in der Refinement Logic auseinanderzusetzen. Außerdem soll verstärkt ein Gefühl für das Beweisen in voller Prädikatenlogik entwickelt werden. Mit Hilfe der soeben geübten Substitution sollten die Quantorregeln beherrschbar sein.

2.2–a $(\neg A \vee \neg B) \;\wedge\; A \wedge B \;\Rightarrow\; \forall x{:}T_1.\exists y{:}T_2.\,Q(f(y,g(f(y,g(x))))) \;\vee\; \neg P(f(x,g(y)))$:
Von der rechten Seite sollte man sich nicht abschrecken lassen, da die linke bereits widersprüchlich ist. Damit kann man alles, also auch solche Mammutkonstruktionen folgern, auf die es im Beweis jedoch nicht ankommt.

Der Sequenzenbeweis ist, bis auf die komplexere rechte Seite identisch mit dem aus Übung 1.4b

2.2–b `¬(∀x:T.P(x)) ⇒ ∃x:T.¬P(x)`:
Die Aussage ist intuitionistisch nicht gültig. Man muß also davon ausgehen, die `magic`-Regel verwenden zu müssen.

```
 ⊢ ¬(∀x:T. P x) ⇒ (∃x:T. ¬(P x))                                    BY impI
 \
  ¬(∀x:T. P x) ⊢ (∃x:T. ¬(P x))      BY cut "(∃x:T. ¬(P x)) ∨¬(∃x:T. ¬(P x))"
  |\
  | ¬(∀x:T. P x) ⊢ (∃x:T. ¬(P x)) ∨¬(∃x:T. ¬(P x))                  BY magic
  \
   ¬(∀x:T. P x), (∃x:T. ¬(P x)) ∨¬(∃x:T. ¬(P x)) ⊢ (∃x:T. ¬(P x))    BY orE 2
  |\
  |  ¬(∀x:T. P x), ∃x:T. ¬(P x) ⊢ (∃x:T. ¬(P x))              BY hypothesis 2
   \
    ¬(∀x:T. P x), ¬(∃x:T. ¬(P x)) ⊢ (∃x:T. ¬(P x))                   BY notE 1
    \
     ¬(∀x:T. P x), ¬(∃x:T. ¬(P x)) ⊢ ∀x:T. P x                        BY allI
     \
      ¬(∀x:T. P x), ¬(∃x:T. ¬(P x)), x:T ⊢ P x
                                BY cut P x ∨ ¬(P x)" THENL [magic; orE 4]
      \
       ¬(∀x:T. P x), ¬(∃x:T. ¬(P x)), x:T, ¬(P x) ⊢ P x              BY notE 2
       \
        ¬(∀x:T. P x), ¬(∃x:T. ¬(P x)), x:T, ¬(P x) ⊢ ∃x:T. ¬(P x)  BY exI "x"
        \
         ¬(∀x:T. P x), ¬(∃x:T. ¬(P x)), x:T, ¬(P x) ⊢ ¬(P x)  BY hypothesis 4
```

2.2–c `∀0:ℕ.(∀n,m:ℕ.≥(n,m)⇒ ≥(s(n),m))∧≥(s(0),0)⇒ ≥(s(s(0)),0)`:

Der folgende Beweis wurde direkt mit Nuprl geführt und enthält daher auch Deklarationen von Prädikaten. Hypothesen werden nur aufgeführt, wenn sie neu sind

```
 ⊢ ∀0:ℕ.  (∀n:ℕ.  ∀m:ℕ.  ≥ n m ⇒ ≥ (s n) m) ∧ ≥ (s 0) 0 ⇒ ≥ (s (s 0)) 0
 |                                                                   BY allI
 \
  1.  N: 𝕌{1}
  2.  Ge:  ℙ{1}
  3.  s:  N → N
  4.  0:ℕ
   ⊢ (∀n:ℕ.∀m:ℕ.  ≥ n m ⇒ ≥ (s n) m) ∧ ≥ (s 0) 0 ⇒ ≥ (s (s 0)) 0  BY impI
  \
   5.(∀n:ℕ.  ∀m:ℕ.  ≥ n m ⇒ ≥ (s n) m) ∧ ≥ (s 0) 0
    ⊢ ≥ (s (s 0)) 0                                                BY andE 5
   \
    5.  ∀n:ℕ.  ∀m:ℕ.  ≥ n m ⇒ ≥ (s n) m
    6.  ≥ (s 0) 0
     ⊢ ≥ (s (s 0)) 0                                         BY allE 5 ’s 0’
    \
     7.  ∀m:ℕ.  ≥ (s 0) m ⇒ ≥ (s (s 0)) m
      ⊢ ≥ (s (s 0)) 0                                          BY allE 7 ’0’
     \
      8.  ≥ (s 0) 0 ⇒ ≥ (s (s 0)) 0
       ⊢ ≥ (s (s 0)) 0                                              BY impE 8
      |\
      | ⊢ ≥ (s 0) 0                                           BY hypothesis 6
       \
        8.  ≥ (s (s 0)) 0
         ⊢ ≥ (s (s 0)) 0                                      BY hypothesis 8
```

2.2–d $\forall$`a,b,c:Kiste.Auf(a,b)`$\wedge$`Auf(b,c)`$\wedge$`Rot(c)` $\wedge$ $\neg$`Rot(a)`
$\Rightarrow$ $\exists$`x,y:Kiste.Auf(y,x)` $\wedge$ $\neg$`Rot(y)`$\wedge$`Rot(x):`

Der Beweis wird schwieriger als man ursprünglich annimmt, da es in der Formel für `Auf` kein Transitivitätsaxiom gibt.

```
⊢ ∀a:Kiste.  ∀b:Kiste.  ∀c:Kiste
    Auf a b ∧ Auf b c ∧ Rot c ∧ ¬(Rot a)
           ⇒ (∃x:Kiste.∃y:Kiste.  Auf y x ∧ ¬(Rot y) ∧ Rot x)
 |                                             BY allI THEN allI THEN allI
 \
  4.  a:Kiste, 5.  b:Kiste, 6.  c:Kiste
   ⊢ Auf a b ∧ Auf b c ∧ Rot c ∧ ¬(Rot a)
           ⇒ (∃x:Kiste.  ∃y:Kiste.  Auf y x ∧ ¬(Rot y) ∧ Rot x)      BY impI
  \
   7.  Auf a b ∧ Auf b c ∧ Rot c ∧ ¬(Rot a)
    ⊢ ∃x:Kiste.  ∃y:Kiste.  Auf y x ∧ ¬(Rot y) ∧ Rot x
    |                                   BY andE 7 THEN andE 8 THEN andE 9
    \
    7.  Auf a b,  8.  Auf b c,  9.  Rot c,  10.  ¬(Rot a)
    ⊢ ∃x:Kiste.  ∃y:Kiste.  Auf y x ∧ ¬(Rot y) ∧ Rot x
     |                         BY cut "Rot b ∨ ¬(Rot b)" THENL [magic; orE 11]
     |\
     | 11.  Rot b
     |  ⊢ ∃x:Kiste.  ∃y:Kiste.  Auf y x ∧ ¬(Rot y) ∧ Rot x
     |                                            BY exI 'b' THEN exI 'a'
     | \
     |  ⊢ Auf a b ∧ ¬(Rot a) ∧ Rot b                                BY andI
     |   |\
     |   | ⊢ Auf a b                                       BY hypothesis 7
     |    \
     |     ⊢ ¬(Rot a) ∧ Rot b                                       BY andI
     |      |\
     |      | ⊢ ¬(Rot a)                                  BY hypothesis 10
     |       \
     |        ⊢ Rot b                                     BY hypothesis 11
      \
       11.  ¬(Rot b)
        ⊢ ∃x:Kiste.  ∃y:Kiste.  Auf y x ∧ ¬(Rot y) ∧ Rot x
       |                                          BY exI 'c' THEN exI 'b'
       \
        ⊢ Auf b c ∧ ¬(Rot b) ∧ Rot c                                BY andI
         |\
         | ⊢ Auf b c                                       BY hypothesis 8
          \
           ⊢ ¬(Rot b) ∧ Rot c                                       BY andI
            |\
            | ⊢ ¬(Rot b)                                  BY hypothesis 11
             \
              ⊢ Rot c                                      BY hypothesis 9
```

2.2–e

```
      (∀x:Mensch.∃y:Mensch.Vater(y,x)
   ∧   ∀x,y,z:Mensch.Vater(y,x)∧Vater(z,y) ⇒ GroßVater(z,x))
   ⇒   ∀x:Mensch.∃y:Mensch.GroßVater(y,x):
```

```
⊢ (∀x:Mensch.  ∃y:Mensch.  Vater y x)
| ∧ (∀x:Mensch.∀y:Mensch.∀z:Mensch.  Vater y x ∧ Vater z y ⇒ GrossVater z x)
| ⇒ (∀x:Mensch.  ∃y:Mensch.  GrossVater y x)                          BY impI
\
 4.  (∀x:Mensch.  ∃y:Mensch.  Vater y x)
  ∧ (∀x:Mensch.∀y:Mensch.∀z:Mensch.  Vater y x ∧ Vater z y ⇒ GrossVater z x)
  ⊢ ∀x:Mensch.  ∃y:Mensch.  GrossVater y x                           BY andE 4
  \
   4.  ∀x:Mensch.  ∃y:Mensch.  Vater y x
   5.  ∀x:Mensch.∀y:Mensch.∀z:Mensch.  Vater y x ∧ Vater z y ⇒ GrossVater z x
    ⊢ ∀x:Mensch.  ∃y:Mensch.  GrossVater y x                           BY allI
    \
     6.  x:  Mensch
      ⊢ ∃y:Mensch.  GrossVater y x                               BY allE 4 ’x’
      \
       7.  ∃y:Mensch.  Vater y x
        ⊢ ∃y:Mensch.  GrossVater y x                                  BY exE 7
        \
         7.  y:  Mensch
         8.  Vater y x
          ⊢ ∃y:Mensch.  GrossVater y x                             BY allE 5’x’
          \
           9.  ∀y:Mensch.  ∀z:Mensch.  Vater y x ∧ Vater z y ⇒ GrossVater z x
            ⊢ ∃y:Mensch.  GrossVater y x                           BY allE 9 ’y’
            \
             10.  ∀z:Mensch.  Vater y x ∧ Vater z y ⇒ GrossVater z x
              ⊢ ∃y:Mensch.  GrossVater y x                         BY allE 4 ’y’
              \
               11.  ∃y0:Mensch.  Vater y0 y
                ⊢ ∃y:Mensch.  GrossVater y x                            BY exE 11
                \
                 11.  y0:  Mensch
                 12.  Vater y0 y
                  ⊢ ∃y:Mensch.  GrossVater y x                    BY allE 10 ’y0’
                  \
                   13.  Vater y x ∧ Vater y0 y ⇒ GrossVater y0 x
                    ⊢ ∃y:Mensch.  GrossVater y x                       BY impE 13
                    |\
                    | ⊢ Vater y x ∧ Vater y0 y                            BY andI
                    | |\
                    | | ⊢ Vater y x                                BY hypothesis 8
                    |   \
                    |    ⊢ Vater y0 y                             BY hypothesis 12
                     \
                      13.  GrossVater y0 x
                       ⊢ ∃y:Mensch.  GrossVater y x                    BY exI ’y0’
                       \
                        ⊢ GrossVater y0 x                         BY hypothesis 13
```

Lösung 2.3 *Ziel dieser Aufgabe ist es, außer der vertieften Auseinandersetzung mit dem prädikatenlogischen Sequenzenkalkül, ein gewisses Gefühl für den Umgang mit Gleichheit zu bekommen. Die dazu ausgewählten Beispiele dürften dabei von ausreichendem Umfang sein. . .*

Die etwas eigenartige Hypothesennumerierung in den folgenden NuPRL–Ableitungen beruht darauf, daß die Typisierungshypothesen für Bereiche, Prädikate und Funktionen der Übersicht halber weggelassen wurden:

2.3–a ⊢ ∀n,m:ℕ. m=n ⇒ s(n)=s(m):

```
⊢ ∀n:ℕ.  ∀m:ℕ.  m = n ⇒ s n = s m                         BY allI THEN allI
\
 3.  n:  ℕ
 4.  m:  ℕ
  ⊢ m = n ⇒ s n = s m                                               BY impI
  \
  5.  m = n
  ⊢ s n = s m                                        BY substitution 'm = n'
   |\
   | ⊢ m = n                                                  BY hypothesis 5
    \
    ⊢ s n = s n                                                BY reflexivity
```

2.3–b ⊢ ∀a:O. (g(g(g(a)))=a ∧ g(g(g(g(g(a)))))=a) ⇒ g(a)=a:

```
⊢ ∀a:O. g (g (g a)) = a ∧ g (g (g (g (g a)))) = a ⇒ g a = a          BY allI
\
 3.  a:  O ⊢ g (g (g a)) = a ∧ g (g (g (g (g a)))) = a ⇒ g a = a     BY impI
 \
  4.  g (g (g a)) = a ∧ g (g (g (g (g a)))) = a ⊢ g a = a          BY andE 4
  \
   4.  g (g (g a)) = a
   5.  g (g (g (g (g a)))) = a
    ⊢ g a = a                  BY substitution 'a = g (g (g (g (g a))))'
    |\
    | ⊢ a = g (g (g (g (g a))))                                 BY symmetry
    |  \
    |   ⊢ g (g (g (g (g a)))) = a                            BY hypothesis 5
    \
     ⊢ g (g (g (g (g (g a))))) = g (g (g (g (g a))))
     |                     BY substitution 'g (g (g (g (g (g a))))) = a'
     |\
     | ⊢ g (g (g (g (g (g a))))) = a        BY substitution 'g (g (g a)) = a'
     | |\
     | | ⊢ g (g (g a)) = a                                   BY hypothesis 4
     |  \
     |   ⊢ g (g (g a)) = a                                   BY hypothesis 4
      \
      ⊢ a = g (g (g (g (g a))))                                 BY symmetry
       \
        ⊢ g (g (g (g (g a)))) = a                            BY hypothesis 5
```

```
2.3–c  ⊢  ∀0:ℕ.  ( ∀n,m:ℕ.  plus(n,s(m))=s(plus(n,m)) ∧ plus(n,0)=n )
                 ⇒          plus( s(s(0)),s(s(s(0))) ) = s(s(s(s(s(0))))):

      ⊢ ∀0:ℕ
      |   (∀n:ℕ.  ∀m:ℕ.  plus n (s m) = s (plus n m) ∧ plus n 0 = n)
      |   ⇒ plus (s(s0)) (s (s(s0))) = s (s (s (s(s0))))                  BY allI
      \
       4.  0:  ℕ
        ⊢ (∀n:ℕ.  ∀m:ℕ.  plus n (s m) = s (plus n m) ∧ plus n 0 = n)
       |  ⇒ plus (s(s0)) (s (s(s0))) = s (s (s (s(s0))))                  BY impI
       \
        5.  ∀n:ℕ.  ∀m:ℕ.  plus n (s m) = s (plus n m) ∧ plus n 0 = n
         ⊢ plus (s(s0)) (s (s(s0))) = s (s (s (s(s0))))           BY allE 5 's(s0)'
         \
          6.  ∀m:ℕ. plus (s(s0)) (s m) = s (plus (s(s0)) m) ∧ plus (s(s0)) 0 = s(s0)
          ⊢ plus (s(s0)) (s (s(s0))) = s (s (s (s(s0))))           BY allE 6 's(s0)'
          \
           7.  plus (s(s0)) (s (s(s0))) = s (plus (s(s0)) (s(s0)))
                        ∧  plus (s(s0)) 0 = s(s0)
            ⊢ plus (s(s0)) (s (s(s0))) = s (s (s (s(s0))))                BY andE 7
            \
             7.  plus (s(s0)) (s (s(s0))) = s (plus (s(s0)) (s(s0)))
             8.  plus (s(s0)) 0 = s(s0)
              ⊢ plus (s(s0)) (s (s(s0))) = s (s (s (s(s0))))
             | BY substitution  'plus (s(s0)) (s (s(s0))) = s (plus (s(s0)) (s(s0)))'
             |\
             | ⊢ plus (s(s0)) (s (s(s0))) = s (plus (s(s0)) (s(s0)))  BY hypothesis 7
              \
               ⊢ s (plus (s(s0)) (s(s0))) = s (s (s (s(s0))))         BY allE 6 's0'
               \
                9.  plus (s(s0)) (s(s0)) = s (plus (s(s0)) (s0)) ∧  plus (s(s0)) 0 = s(s0)
                 ⊢ s (plus (s(s0)) (s(s0))) = s (s (s (s(s0))))              BY andE 9
                 \
                  9.  plus (s(s0)) (s(s0)) = s (plus (s(s0)) (s0))
                  10.  plus (s(s0)) 0 = s(s0)
                   ⊢ s (plus (s(s0)) (s(s0))) = s (s (s (s(s0))))
                  |     BY substitution 'plus (s(s0)) (s(s0)) = s (plus (s(s0)) (s0))'
                  |\
                  | ⊢ plus (s(s0)) (s(s0)) = s (plus (s(s0)) (s0))     BY hypothesis 9
                   \
                    ⊢ s (s (plus (s(s0)) (s0))) = s (s (s (s(s0))))      BY allE 6 '0'
                    \
                     11.  plus (s(s0)) (s0) = s (plus (s(s0)) 0) ∧  plus (s(s0)) 0 = s(s0)
                      ⊢ s (s (plus (s(s0)) (s0))) = s (s (s (s(s0))))       BY andE 11
                      \
                       11.  plus (s(s0)) (s0) = s (plus (s(s0)) 0)
                       12.  plus (s(s0)) 0 = s(s0)
                        ⊢ s (s (plus (s(s0)) (s0))) = s (s (s (s(s0))))
                       |          BY substitution 'plus (s(s0)) (s0) = s (plus (s (s0)
                       |\
                       | ⊢ plus (s(s0)) (s0) = s (plus (s(s0)) 0)      BY hypothesis 11
                        \
                         ⊢ s (s (s (plus (s(s0)) 0))) = s (s (s (s(s0))))
                         |                    BY substitution 'plus (s(s0)) 0 = s(s0)'
                         |\
                         | ⊢ plus (s(s0)) 0 = s(s0)                    BY hypothesis 12
                          \
                           ⊢ s (s (s (s(s0)))) = s (s (s (s(s0))))       BY reflexivity
```

Lösung 2.4 *Ziel dieser Aufgabe ist es, die Korrektheit der Regeln für den prädikatenlogischen Sequenzenkalkül einzusehen. Dazu soll an einigen Beispielen der Zusammenhang zwischen Regeln und Semantik exemplarisch gezeigt werden:*

2.4–a `orE`:

Annahmen

1. $\Gamma = A_1, \ldots A_n$ und $\Delta = B_1, \ldots B_m$ sind beliebige Folgen von n bzw. m Formeln mit $m, n \in \mathbb{N}_0$.
2. A, B, C sind beliebige Formeln.
3. ι ist eine beliebige Interpretationsfunktion.
4. $\iota(A_1) = \cdots = \iota(A_n) = \iota(B_1) = \cdots = \iota(B_m) =$ wahr.
5. $\iota(\Gamma, A, \Delta \vdash C) =$ wahr.
6. $\iota(\Gamma, B, \Delta \vdash C) =$ wahr.
7. $\iota(A \vee B) =$ wahr.

Beweis

Mit Definition 2.2.13 (Skript, S. 32) folgt aus 1. bis 6. daß $\iota(C) =$ wahr gilt, wann immer Annahme 4. und $\iota(A) =$ wahr gilt und wann immer Annahme 4. und $\iota(B) =$ wahr gilt.

Nach Definition 2.2.8 (Skript, S. 28) folgt nun aber aus Annahme 7. daß *entweder* $\iota(A) =$ wahr *oder* $\iota(B) =$ wahr gilt. Unter Hinzunahme von Annahme 4. folgt nach dem letzten Schluß somit $\iota(C) =$ wahr.

Damit ist also auch $\iota(\Gamma, A \vee B, \Delta \vdash C) =$ wahr.

Insgesamt folgt so aus $\iota(\Gamma, A, \Delta \vdash C) =$ wahr und $\iota(\Gamma, B, \Delta \vdash C) =$ wahr immer $\iota(\Gamma, A \vee B, \Delta \vdash C) =$ wahr, was zu zeigen war.

2.4–b `impE` :

Annahmen

1. $\Gamma = A_1, \ldots A_n$ und $\Delta = B_1, \ldots B_m$ sind beliebige Folgen von n bzw. m Formeln mit $m, n \in \mathbb{N}_0$.
2. A, B, C sind beliebige Formeln.
3. ι ist eine beliebige Interpretationsfunktion.
4. $\iota(A_1) = \cdots = \iota(A_n) = \iota(B_1) = \cdots = \iota(B_m) =$ wahr.
5. $\iota(\Gamma, A \Rightarrow B, \Delta \vdash A) =$ wahr.
6. $\iota(\Gamma, B, \Delta \vdash C) =$ wahr.
7. $\iota(A \Rightarrow B) =$ wahr.

Beweis

Unter den Annahmen 1. bis 4. folgt aus Annahme 5. nach Definition 2.2.13 (Skript, S. 32), daß $\iota(A) =$ wahr gilt

Zusammen mit Annahme 7. folgt dann nach Definition 2.2.8 (Skript, S. 28), daß somit $\iota(B) =$ wahr unter Annahme von 1. bis 5. gilt.

Damit folgt aber aus Annahme 6. nach Definition 2.2.13, daß $\iota(C) =$ wahr ist.

Insgesamt haben wir $\iota(C) =$ wahr und somit $\iota(\Gamma, A \Rightarrow B, \Delta \vdash C) =$ wahr unter Annahme von 1. bis 7., was zu zeigen war.

2.4–c `exE`:

Annahmen

1. $x \in \mathcal{V}$ ist eine beliebige Variable.
2. $T \in \mathcal{T}$ ist ein beliebiges Bereichssymbol.
3. $\Gamma = A_1, \ldots A_n$ und $\Delta = B_1, \ldots B_m$ sind beliebige Folgen von n bzw. m Formeln mit $m, n \in \mathbb{N}_0$, so daß x in keiner dieser Formeln frei vorkommt.
4. A, C sind beliebige Formeln, wobei x in C nicht frei vorkommt.
5. ι ist eine beliebige Interpretationsfunktion.
6. $\iota(A_1) = \cdots = \iota(A_n) = \iota(B_1) = \cdots = \iota(B_m) = \mathsf{wahr}$.
7. $\iota(\Gamma, A, \Delta \vdash C) = \mathsf{wahr}$.
8. $\iota(\exists x : T.A) = \mathsf{wahr}$.

Beweis

Nach Definition 2.2.8 (Skript, S. 28) folgt aus Annahme 8. daß es ein $u \in \iota(T)$ gibt, so daß $\iota_x^u(A) = \mathsf{wahr}$ gilt.

Da x weder in $A_1, \ldots A_n, B_1, \ldots B_m$ noch in C frei vorkommt, hängt die Wahrheit von $A_1, \ldots A_n, B_1, \ldots B_m$ und C nicht von $\iota(x)$ ab. Damit gilt unter den Annahmen 1. bis 7. nach Definition 2.2.13 (Skript, S. 32) und dem letzten Schluß (es gibt ein $u \in \iota(T)$ mit $\iota_x^u(A) = \mathsf{wahr}$), daß auch $\iota_x^u(C) = \mathsf{wahr}$ ist.

Unter Annahme von 1. bis 8. gilt also, daß $\iota(C) = \mathsf{wahr}$ gilt.

Insgesamt haben wir also nach Definition 2.2.13 gezeigt, daß $\iota(\Gamma, \exists x : T.A, \Delta \vdash C) = \mathsf{wahr}$ gilt, falls Annahmen 1. bis 5. und 7. gelten, was zu zeigen war.