Prof. Dr. Lutz Schröder
Lehrstuhl Informatik 8 (Theoretische Informatik)
Universität Erlangen-Nürnberg

Bachelorprüfung
Grundlagen der Logik in der Informatik
— Probeklausur

Aufgabe 1: Aussagenlogische Konsequenz und Beweise

a) Zeigen oder widerlegen Sie mittels Wahrheitstafeln, dass folgende logische Konsequenzen gelten. Im negativen Fall muss die Wahrheitstafel nicht vollständig angegeben werden, sondern es reicht, wenn sie ein Gegenbeispiel enthält, das aber deutlich markiert werden muss (d.h. eine unkommentierte Wahrheitstafel ohne deutlich identifiziertes Gegenbeispiel gilt nicht als Lösung).

i) $((A \wedge B) \to C) \models (A \to (B \to C))$;

ii) $((A \to B) \land (B \to C) \land (C \to \neg A)) \models \perp$.

b) Man betrachte das folgende Beweisskript in Coq:

```
1  Require Import Classical.
2
3  Parameters p q r : Prop.
4
5  Theorem Q1 : (~p -> ~q) -> (q -> p).
6  Proof.
7  intro A.
8  intro B.
9  apply NNPP.
10 intro C.
11 assert (~q).
12 apply A.
13 exact C.
14 contradiction.
15 Qed.
```

Geben Sie für jeden der Schritte 8—13 das jeweils aktuelle Unterziel (subgoal) und Annahmen samt Labels jeweils *nach* Durchführung des Schritts an, indem Sie die umseitige Tabelle vervollständigen. (*Achtung*: es kann weitere Unterziele geben, die gewissermaßen im Stapel unter dem aktuellen liegen; anzugeben ist zur Ersparnis von Schreibarbeit aber eben jeweils nur das aktuelle.). Die Antworten für Schritte 6–7 sind als Beispiele schon eingetragen.

Hinweis: Zur Vermeidung übermäßiger Schreibarbeit können Sie auf schon einmal angegebene Annahmen mit Label später einfach über ihren Label verweisen. Man erinnere sich, dass Taktiken immer auf das erste von mehreren Unterzielen angewendet werden. Durch `assert` wird ein neues *erstes* (also aktuelles und im Stapel oben liegendes) Unterziel erzeugt, das nach seinem Beweis als Annahme auftaucht, ggf. mit dem per `as` vorgegebenen Namen.

Schritt	Annahmen	Aktuelles Unterziel
6	—	`(~p -> ~q) -> (q -> p)`
7	`A : ~p -> ~q`	`q -> p`
8		
9		
10		
11		
12		
13		
14	—	`No more subgoals.`

Aufgabe 2: Formalisierung in Prädikatenlogik

Wir betrachten eine naive Mengenlehre mit einem zweistelligen Prädikat $\in$ in Infixschreibweise und einer einstelligen Funktion p. Hierbei nennen wir die Elemente des Grundbereichs *Mengen*; wir lesen $p(x)$ als die Potenzmenge von x, und $x \in y$ wie üblich als „x ist Element von y“. Formalisieren Sie folgende Aussagen als prädikatenlogische Formeln erster Stufe über der Signatur $\{p/1, \in/2\}$:

a) x ist Teilmenge von y.

b) Jede Menge ist durch ihre Elemente eindeutig bestimmt.

c) Die Potenzmenge einer Menge enthält gerade die Teilmengen dieser Menge.

d) Jede nichtleere Menge enthält ein Element, mit dem sie keine Elemente gemeinsam hat.

e) Jede Menge ist die große Vereinigung ihrer höchstens einelementigen Teilmengen.

Hinweis: Es sollen explizit keine neuen Signatursymbole (wie etwa „große Vereinigung“ oder „aus x gebildete einelementige Menge“) eingeführt werden. Man muss also gegebenenfalls statt eines Begriffs die ihn definierende Eigenschaft verwenden; Sie dürfen aber Abkürzungen für Formeln einführen. Nicht erklärte Begriffe wie „Teilmenge“ und „Vereinigung“ sind in ihrer üblichen Bedeutung zu verstehen.

Hinweis: Wir akzeptieren im Moment als Schreibweise für Quantoren sowohl $\forall x.\, \phi$ als auch $\forall x(\phi)$.

Tragen Sie Ihre Antworten zu a)–e) in die unten dafür vorgesehenen Boxen ein.

a)

b)

c)

d)

e)

Aufgabe 3: Unifikation

Wenden Sie den Unifikationsalgorithmus aus der Vorlesung an, um zu entscheiden, ob die Gleichung

$$f(g(x, h(z)), z) = f(g(h(z), h(z)), h(w))$$

(wobei x, y, z, w Variablen sowie f, g, h Funktionssymbole sind) unifizierbar ist und ggf. einen allgemeinsten Unifikator zu berechnen. (Achtung: es ist durchaus von Bedeutung, dass die beiden Seiten gemeinsame Variablen verwenden! Gefragt ist der korrekte Unifikationsalgorithmus gemäß Vorlesung, also mit Occurs Check. Wir bestehen hier anders als in der Vorlesung nicht auf den Notation $\doteq$ für zu lösende Gleichungen.) Gefordert sind nicht nur die letztendliche (*explizite!*) Antwort, sondern auch die einzelnen vom Algorithmus durchgeführten Umformungsschritte, unter Angabe der jeweils verwendeten Umformungsregel.

Aufgabe 4: Prädikatenlogische Resolution

a) Bringen Sie die Formel

$$\forall x.\, ((\forall y.\, R(x,y)) \vee \exists y.\, R(y,x)) \to \exists z.\, S(x,z) \wedge S(z,x)$$

(die Formel selbst, nicht ihre Negation!) in pränexe Normalform, sodann in Skolemform und schließlich in Skolemform mit quantorenfreiem Anteil in CNF (also praktisch in Klauselform, nur zur Zeitersparnis ohne den Schreibweisenwechsel). Tragen Sie die jeweils angegebenen Zwischenergebnisse in die unten dafür vorgesehenen Boxen ein; oberhalb der Boxen ist Platz für eventuell benötigte Zwischenrechnungen. *Hinweis:* Hier und im Weiteren wird die im aktuellen Semester verwendete Syntax eingesetzt; insbesondere erstreckt sich der Geltungsbereich eines Quantors immer so weit wie möglich nach rechts.

Kodierung von $\to$ durch $\neg$ und $\vee$:

Negationsnormalform:

Pränexe Normalform:

Skolemform:

Skolemform mit quantorenfreiem Anteil in konjunktiver Normalform:

b) Zeigen Sie die Unerfüllbarkeit der aus den unten in Kästchen stehenden Klauseln bestehenden Klauselmenge, indem Sie aus den gegebenen Klauseln mittels prädikatenlogischer Resolution die leere Klausel herleiten. Zeichnen Sie dazu in jedem Schritt Verbindungslinien von den beiden zu resolvierenden Klauseln zur neu hergeleiteten Resolvente. Annotieren Sie eine der Linien mit der jeweils verwendeten Substitution. Rechnen Sie vorher auf Schmierpapier, bevor Sie Ihre Lösung unten eintragen. In den Klauseln sind a und c Konstanten, während x, y, z Variablen sind. *Hinweis:* Es hilft, sich das Argument vorab informell durch eine Zeichnung klarzumachen.

$\boxed{R(x, f(g(x)))}$ $\boxed{\neg R(x, y), \neg R(y, z), R(x, z)}$ $\boxed{R(f(x), a)}$ $\boxed{\neg R(c, a)}$

$\square$

Aufgabe 5: Formale Deduktion

Geben Sie eine formale Herleitung der Formel

$$\forall y.\, \exists x.\, P(x, y)$$

aus

$$\exists x.\, \forall y.\, P(x, y)$$

im System natürlichen Schließens gemäß Vorlesung an. Dabei ist P ein Prädikat. *Achtung:* Es darf *nur* natürliches Schließen verwendet werden; *nicht* zulässig sind insbesondere Umformungen gemäß bekannter logischer Äquivalenzen (diese dürfen natürlich, wenn nötig, im System nachvollzogen werden, was sich aber meist eher nicht empfiehlt).

Aufgabe 6: Induktion

Sei $\Sigma = \{mult/2, zero/0, one/0\}$, und sei $\mathfrak{M}$ ein Σ-Modell, so dass $M = \mathbb{N}$, $\mathfrak{M}[\![zero]\!] = 0$, $\mathfrak{M}[\![one]\!] = 1$ und für alle $x, y \in M$

$$\mathfrak{M}[\![mult]\!](x, y) = x * y.$$

Zeigen Sie durch Induktion über E, dass für jeden geschlossenen Term E (d.h. E enthält keine freien Variablen, formal: $FV(E) = \emptyset$) gilt

$$\mathfrak{M}[\![E]\!] \in \{0, 1\}.$$

Die Aufgaben sind untereinander gleich gewichtet; die Klausur ist mit der Hälfte der Punkte in jedem Fall bestanden.

Achtung: Selbstverständlich müssen alle Schritte vollständig durchgeführt werden.

Hinweis: Die Aufgabe ist als Induktionsbeweis gesehen absichtlich sehr einfach. Selbstverständlich müssen dennoch alle Schritte vollständig durchgeführt werden. Teil der Aufgabe ist unter anderem auch die Verwendung korrekter Notation für die Semantik der Logik erster Stufe, auf die also besonders geachtet werden sollte. Elementare Fakten über Arithmetik dürfen vorausgesetzt werden.