Universität Karlsruhe

Informatik 1 WS 00/01

Institut für Telematik, Forschungsgruppe C&M
Prof. Dr. S. Abeck, R. Scholderer

Musterlösungen zum Übungsblatt 9

Lösung

1. Auswertungsstrategien für Termersetzungssysteme

Im folgenden werden die verwendeten Regeln als Hinweise in Klammern angegeben:

Teilaufgabe 1: Auswertung nach **E**:

add(succ(succ(zero)), succ(zero)) $\stackrel{?}{=}$ zero
→{(A1) mit *x*, *y* := succ(zero), succ(zero)}
succ(add(succ(zero), succ(zero))) $\stackrel{?}{=}$ zero
→{(Al) mit *x*, *y* := zero, succ(zero)}
succ(succ(add(zero, succ(zero)))) $\stackrel{?}{=}$ zero
→{(A0) mit *y* := succ(zero)}
succ(succ(succ(zero))) $\stackrel{?}{=}$ zero
→{(G2) mit, *x* := succ(succ(zero))}
false

Auswertung nach **L**:

add(succ(succ(zero)), succ(zero)) $\stackrel{?}{=}$ zero
→{(A1) mit *x*, *y* := succ(zero), succ(zero)}
succ(add(succ(zero), succ(zero))) $\stackrel{?}{=}$ zero
→{(G2) mit *x* := add(succ(zero), succ(zero))}
false

Beobachtung: Die Auswertung nach **L** ist hier kürzer, da der Wert von ". $\stackrel{?}{=}$ zero" bereits feststeht, sobald im Term der linken Seite ein succ außen steht.

Teilaufgabe 2: Auswertung nach **E**:

db(add(succ(zero), zero))
→{(Al) mit *x*, *y* := zero, zero}
db(succ(add(zero, zero)))

→{(A0) mit *y* := zero}
db(succ(zero))
→{(D) mit *x* := succ(zero)}
add(succ(zero), succ(zero))
→{(A1) mit *x*, *y* := zero, succ(zero))
succ(add(zero, succ(zero)))
→{(A0) mit *y* := succ(zero)}
succ(succ(zero))

Eine Auswertung nach L:

db(add(succ(zero), zero))
→{(D) mit *x* := add(succ(zero), zero)}
add(add(succ(zero), zero), add(succ(zero), zero))
→{(Al) mit *x*, *y* := zero, zero}
add(succ(add(zero, zero)), add(succ(zero), zero))
→{(Al) mit *x*, *y* := add(zero, zero), add(succ(zero), zero)}
succ(add(add(zero, zero), add(succ(zero), zero)))
→{(A0) mit *y* := zero}
succ(add(zero, add(succ(zero), zero)))
→{(A0) mit *y* := add(succ(zero), zero)}
succ(add(succ(zero), zero))
→{(A1) mit *x*, *y* := zero, zero}
succ(succ(add(zero, zero)))
→{(A0) mit *y* := zero}
succ(succ(zero))

Beobachtung: Hier ist die Auswertung nach **E** kürzer, da das Argument von db ausgewertet wird, bevor es über die Regel (D) an zwei Stellen kopiert wird.

Lösung

2. Termersetzungssysteme

a)

add(pred(succ(zero)), add(zero, succ(zero)) $\stackrel{?}{=}$ zero	(P)
add(zero, add(zero, succ(zero)) $\stackrel{?}{=}$ zero	(A0)
add(zero, succ(zero)) $\stackrel{?}{=}$ zero	(A0)
succ(zero) $\stackrel{?}{=}$ zero	(G2)
false	

b)

pred(succ(succ(add(succ(zero), zero)))) $\stackrel{?}{=}$ succ(zero)	(P)

succ(add(succ(zero), zero)) $\stackrel{?}{=}$ succ(zero)	(A1)
succ(succ(add(zero, zero))) $\stackrel{?}{=}$ succ(zero)	(A0)
succ(succ(zero)) $\stackrel{?}{=}$ succ(zero)	(G3)
succ(zero) $\stackrel{?}{=}$ zero	(G2)
false	

Lösung

3. Theoreme der Aussagenlogik

a) Informell bedeutet {false}K t, dass aus etwas Falschem alles, also Wahres und Falsches, abgeleitet werden kann.

Zunächst soll die Hilfsbehauptung (**) bewiesen werden:
Aus der leeren Axiomenmenge (∅) ist unmittelbar nur t ∨¬t (Tertium non datur) ableitbar (1. Ableitungsregel der Aussagenlogik)

∅ K t ∨ ¬t	Tertium non datur
∅ K (t ∨ t) ∨ ¬t	Idempotenz Disjunktion
∅ K t ∨ (t ∨ ¬t)	Assoziativität Disjunktion
∅ K (t ∨ ¬t) ∨ t	Kommutativität Disjunktion
∅ K ¬(¬(t ∨ ¬t)) ∨ t	Involutionsgesetz
∅ K ¬(¬true) ∨ t	Komplementgesetz
∅ K ¬false ∨ t	¬true = false folgt aus Neutralitäts- und Komplementgesetzen (***)
∅ K false ⇒ t	Definition Implikation
∅ ∪ {false} K t	Lemma der Vorlesung
{false} K t	∅ ∪ M = M (Neutralitätsgesetz der Mengenalgebra)

Beweis der Beh: ¬true = false. (***)
Zu zeigen ist, daß false das Komplement von true also ¬true ist. Dies beweist man, indem man zeigt, dass die Komplementgesetze gelten.
1. true E false = false (Neutralitätsgesetz)
2. true ∨ false = true (Neutralitätsgesetz)

Nun soll die Behauptung (*) des Satzes gezeigt werden:

Nach Voraussetzung gilt:
H K false
Wegen (**) aber auch für beliebiges t:
{false} K t

Dann gilt aber auch:
$H \cup \{false\}$ K t für beliebiges t.

Nach einem Satz der Vorlesung dürfen aus H bereits abgeleitete Aussagen s zur Ableitung weiterer Aussagen **über H (nicht nur über HÈ{s})** hinzugenommen werden.
$\rightarrow$ Behauptung

b)
Es sei im folgenden $I_\beta[t]$ eine Interpretationsfunktion, die einer Aussage t mit freien Identifikatoren $x_1,...,x_n$ einen beliebigen Belegungsvektor $\beta=(\beta_1,...,\beta_n)$, $\beta_i \in \{L,O\}$ aus der booleschen Algebra **B** zuordnet.
β sei stets ein beliebiger Belegungsvektor. $I_\beta[t]$ sei definiert wie in Kap. 3.2 Folie 12. Insbesondere ist $I_\beta[t]$ also linear: $I_\beta[f(t_1, ...,t_n)] = f(I_\beta[t_1], ..., I_\beta[t_n])$.

1. Zunächst soll die Korrektheit der Theorie mit der leeren Axiomenmenge bewiesen werden:
Aufgrund der ersten Ableitungsregel der Aussagenlogik lässt sich aus der leeren Axiomenmenge nur der Tertium non datur $t \vee \neg t$ ableiten.
Zu zeigen: $I_\beta[t\vee \neg t]=L$ für alle $\beta \in \{L,O\}^2$ und beliebige Aussagen t. Hier ist die Korrektheitseigenschaft für t nicht unbedingt vorausgesetzt. Es müssen hier also auch Aussagen t mit $I_\beta[t] = O$ berücksichtigt werden.

<u>Fall $I_\beta[t_1] = O$</u>
$I_\beta[t\vee \neg t]= I_\beta[t] \vee \neg I_\beta[t] = O \vee \neg O = L$

<u>Fall $I_\beta[t_1] = L$</u>
$I_\beta[t\vee \neg t]= I_\beta[t] \vee \neg I_\beta[t] = L \vee \neg L = L$

2. Nun sollen nicht leere, jedoch nach Voraussetzung konsistente Axiomenmengen (false ist nicht ableitbar) betrachtet werden.
Hier können zur Ableitung neben dem Tertium non datur nur der Modus Ponens und die Gleichheitsgesetze der booleschen Algebra angewendet werden.

Für den Tertium non datur verläuft der Beweis für die nicht leere Axiomenmenge analog wie in 1., wobei hier nur Aussagen t mit $I_\beta[t] = L$ berücksichtigt werden müssen. (Axiomenmenge ist konsistent.)
Für die beiden anderen Ableitungsregeln verbleibt die Korrektheitseigenschaft zu zeigen:

Um den Modus Ponens anzuwenden, müssen wenigstens zwei beliebige Aussagen der Form t_1 und $\neg t_1 \vee t_2$ in der Axiomenmenge enthalten sein. Nach Voraussetzung gilt für diese:
$I_\beta[t_1] = L$ und $I_\beta[\neg t_1 \vee t_2] = L$ für beliebige Belegungen β
$\rightarrow L = I_\beta[\neg t_1 \vee t_2] = \neg I_\beta[t_1] \vee I_\beta[t_2] = \neg L \vee I_\beta[t_2]$
$\rightarrow O \vee I_\beta[t_2] = L$
$\rightarrow I_\beta[t_2] = L$ für beliebige Belegungen β, was zu zeigen war, da der Modus Ponens besagt:
$\{\neg t_1 \vee t_2, t_2\}$ K t_2

Auch für die Anwendung der 3. Ableitungsregel (Gleichheitsgesetze), darf die Axiomenmenge nicht leer sein.
Für die enthaltenen Aussagen t gilt nach Vorraussetzung wieder $I_\beta[t] = L$.
Sei nun also t_1 eine beliebige Aussage aus der Axiomenmenge mit $I_\beta[t_1] = L$. Falls eine weitere Aussage t_2 mit der 3.Ableitungsregel abgeleitet werden kann, so muss für t_2 gelten:

$t_2 = t_1$ nach Gesetzen der booleschen Algebra
$\rightarrow I_\beta[t_2] = I_\beta[t_1] = L$, da sich t_2 in t_1 syntaktisch, ohne Änderung der Semantik umformen lässt.

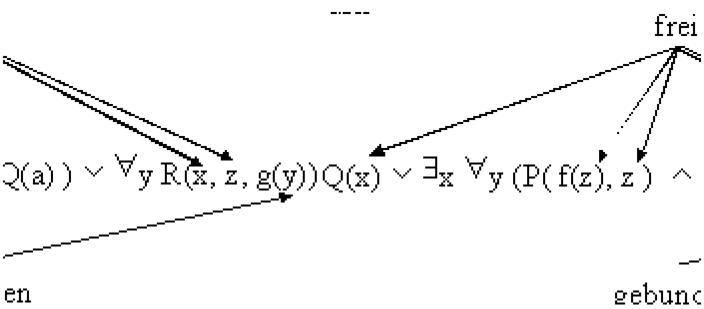
® Aus den 3 Teilbeweisen folgt die Behauptung

Lösung

4. Prädikatenlogische Formel

Freie Variablen:
Diese Variablen werden durch keinen All- oder Existenzquantor gebunden.
- x ist frei in Q(x) und in R(x,z,g(y))
- z ist in der gesamten Formel frei

Gebundene Variablen:
- y in R(x,z,g(y)) wird über den Allquantor $\forall$y gebunden



Lösung

5. Modelle

Teilaufgabe 1:

Bemerkung: Im Folgenden sind $\leftrightarrow$, $\rightarrow$ die Symbole für "genau dann, wenn" und "daraus folgt" in der verwendeten mathematischen Metasprache. $\Leftrightarrow$ und $\Rightarrow$ bezeichnen dagegen die Symbole für Implikation und Äquivalenz in der für die Prädikate verwendeten booleschen Algebra.

$F = \exists x\, \exists y\, \exists z\, (P(x,y,z) \wedge P(x,z,y) \wedge P(z,y,x))$

1. Aus P(x,y,z) folgt $x < y \leq z$ (1)
2. Aus P(x,z,y) folgt $x < z \leq y$ (2)
3. Aus P(z,y,x) folgt $z < y \leq x$ (3)

Mit (1), (2) und (3) ergibt sich:
$\exists x\, \exists y\, \exists z\, (x < y \leq z) \wedge (x < z \leq y) \wedge (z < y \leq x) \leftrightarrow$ (zusammenfassen von (1) und (2))
$\exists x\, \exists y\, \exists z\, (x < y \wedge y=z) \wedge (z < y \leq x) \leftrightarrow \exists x\, \exists y\, \exists z\, (x < y) \wedge (y = z) \wedge (z < y \leq x) \rightarrow \exists x\, \exists y\, \exists z\, (y = z) \wedge (z < y)$, was offensichtlich einen Widerspruch darstellt.
® A ist kein Modell für F
Eine alternative Darstellung:

Teilaufgabe 2:

$F = \exists x \, \exists y \, \exists z \, (P(x, y) \wedge P(z, y) \wedge P(x, z) \wedge \neg P(z,x))$

1. $P(x,y) \rightarrow x \subseteq y$
2. $P(z,y) \rightarrow z \subseteq y$
3. $P(x,z) \rightarrow x \subseteq z$
4. $\neg P(z,x) \rightarrow z$ ° x

Aus 3. und 4. folgt (5.) $z \supset x$ bzw. $x \subset z$
Nun folgt aus 2. und 5.: $x \subset z \subseteq y$ (*)
Aus 3. folgt (*) offensichtlich, da $x \subset z$ erst recht gilt, wenn $x \subseteq z$ gilt. Wegen der Transitivität der Teilmengenbeziehung folgt aus 1. $x \subseteq y$ auch (*), da auch $x \subset y$ in (*) gilt.
® A ist ein Modell für F
