Technische Universität München
Fakultät für Informatik
Prof. Dr. M. Broy

WS 2001
Übungsblatt 6
22. November 2001

Übungen zu Einführung in die Informatik I

Aufgabe 22 Prädikatenlogische Ausdrücke über den natürlichen Zahlen

Über den natürlichen Zahlen gelte die Beziehung $T(x,y)$, wenn gilt: x ist Teiler von y. Definieren Sie mit Hilfe der Prädikatenlogik:

a) y ist eine gerade Zahl

b) y ist Primzahl

c) x und y sind teilerfremd.

d) Eine natürliche Zahl z ist größter gemeinsamer Teiler von x und y.

Aufgabe 23 Formulierung prädikatenlogischer Ausdrücke

Folgende Tatsachen sind gegeben:

- Jeder Drache ist glücklich, wenn alle seine Kinder fliegen können.
- Grüne Drachen können fliegen.
- Ein Drache ist grün, wenn er Kind mindestens eines grünen Drachen ist.

(**Hinweis:** Drachen haben, wie alle Reptilien, zwei Eltern.)

a) Drücken Sie die angegebenen Tatsachen als prädikatenlogische Formeln aus. Verwenden Sie dabei die Prädikate *Ki(x, y)* für „*x* ist Kind von *y*“, *Fl(x)* für „*x* kann fliegen“, *Gl(x)* für „*x* ist glücklich“, und *Gr(x)* für „*x* ist grün“.

b) Begründen Sie informell: Alle grünen Drachen sind glücklich!

c) Zeigen Sie formal, dass alle grünen Drachen glücklich sind!

Aufgabe 24 Interpretation von prädikatenlogischen Formeln

Sei A eine Rechenstruktur mit Signatur $\Sigma = (S, F)$ und $\mathbf{m} \in S$ eine Sorte. Seien $p, q : \mathbf{m} \to \mathbf{bool}$ einstellige Prädikate und

$$t = (\exists\, \mathbf{m}\, x : \; p(x)) \wedge (\exists\, \mathbf{m}\, x : \; q(x)) \text{ und } s = \exists\, \mathbf{m}\, x : (p(x) \wedge q(x))$$

Formeln über p, q. Geben Sie jeweils eine Rechenstruktur A, eine Sorte $\mathbf{m}$ und eine Spezifikation der Prädikate p, q an, so dass

a) die Formeln t und s beide gültig sind.

b) die Formeln t und s nicht beide gültig sind.

Aufgabe 25 Ableitungen von prädikatenlogischen Formeln

Überlegen Sie für die nachfolgenden prädikatenlogischen Formeln, ob diese für beliebige Rechenstrukturen gelten. Geben Sie dann jeweils entweder eine Ableitung mit Hilfe der Ableitungsregeln der Prädikatenlogik oder eine Rechenstruktur als Gegenbeispiel an. Die zugrundeliegende Signatur sei Σ, **m** und **n** seien Sorten in Σ und t sei eine prädikatenlogische Formel über Σ.

a) $(\forall\, \mathbf{bool}\, x : t) \Rightarrow (\exists\, \mathbf{bool}\, x : t)$

b) $(\forall\, \mathbf{m}\, x : t) \Rightarrow (\exists\, \mathbf{m}\, x : t)$

c) $(\forall\, \mathbf{m}\, x : \forall\, \mathbf{n}\, y : t) \Rightarrow (\forall\, \mathbf{n}\, y : \forall\, \mathbf{m}\, x : t)$

d) $(\forall\, \mathbf{n}\, y : \exists\, \mathbf{m}\, x : t) \Rightarrow (\exists\, \mathbf{m}\, x : \forall\, \mathbf{n}\, y : t)$

e) **Freie Zusatzaufgabe (schwierig!):** $(\forall\, \mathbf{n}\, y : \exists\, \mathbf{m}\, x : t) \Leftarrow (\exists\, \mathbf{m}\, x : \forall\, \mathbf{n}\, y : t)$

Neben den in der Vorlesung eingeführten Ableitungsregeln (1) - (4) der Pradikatenlogik können noch folgende zwei Regeln verwendet werden, die sich durch Übertragung von Regeln der Aussagenlogik auf die Prädikatenlogik ergeben. Dabei bezeichnet Σ eine beliebige Signatur, H eine beliebige Menge von prädikatenlogischen Formeln über Σ, sowie t, t_1 und t_2 beliebige prädikatenlogische Formeln über Σ.
(0) Es gilt: $(\Sigma, H \cup \{t\}) \vdash t$
(5) Gilt $(\Sigma, H \cup \{t_1\}) \vdash t_2$, dann gilt: $(\Sigma, H) \vdash t_1 \Rightarrow t_2$

Aufgabe 26 (P) Die Rechenstruktur der Punkte im Koordinatensystem

In dieser Aufgabe greifen wir die aus Aufgabe 18 bekannte Rechenstruktur **K** der Punkte im kartesischen Koordinatensystem auf. Die dort genannten Funktionen werden spezifiziert und sollen von Ihnen in Gofer implementiert werden.

Spezifikation der Rechenstruktur **K**: Der Sorte **bool** ist die Trägermenge $\mathbb{B}^\perp$, der Sorte **int** die Trägermenge $\mathbb{Z}^\perp$ und der Sorte **point** die Trägermenge $\mathbb{Z}^\perp \times \mathbb{Z}^\perp$ zugeordnet. Die Funktionen der Rechenstruktur **INT** sind wie in der Rechenstruktur **INT** spezifiziert, die Funktionen {make, projX, projY, moveto} wie folgt:

- make(x, y): **fct** make = **(int, int) point**; erzeugt ein Element der Sorte **point**.
- projX(p) / projY(p): **fct** projX = **(point) int**; projiziert auf die erste (zweite) Komponente des Punktes p.
- moveto(p, dx, dy): **fct** moveto = **(point, int, int) point**; gibt einen Punkt zurück, dessen erste (zweite) Komponente die Summe der ersten (zweiten) Komponente von p mit dx (dy) ist.

a) Geben Sie eine geeignete Spezifikation zu `make, projX, projY` und `moveto` an!

b) Implementieren Sie den Typ **point** (keine Verwendung von Typsynonymen), sowie die Funktionen `make, projX, projY` und `moveto` in Gofer! Testen Sie Ihre Implementierung durch einen Aufruf der Funktion `moveto`!