Theoretische Informatik – Übung 10

SS 2019
Jochen Schmidt

Technische Hochschule Rosenheim

**Aufgaben 1a-c, 2, 3a/b, 4, 5 und 6 bitte vor der Übungsstunde zu Hause lösen.
Die Aufgaben 1d/e und 3c/d werden in der Übungsstunde bearbeitet.**

Aufgabe 1

Die Addition von zwei natürlichen Zahlen gehört zur Klasse der LOOP-berechenbaren Funktionen. Zeigen Sie, dass auch die folgenden Funktionen LOOP-berechenbar sind. Falls nötig, dürfen Sie die Additionsfunktion verwenden, ebenso IF … THEN. Das Ergebnis der Berechnung soll nach der Konvention für LOOP-Programme am Ende in der Variablen x_0 stehen.

a) die (modifizierte) Subtraktion von zwei Variablen x_1 und x_2.
b) die Multiplikation von zwei Variablen x_1 und x_2.
c) die Exponentialfunktion $\exp(x_1, x_2) = x_1^{x_2}$
d) die Fakultät n!, definiert durch:
n! = n (n – 1)!
0! = 1
e) die Berechnung der Fibonacci-Folge durch fib(n) mit
fib(n) = fib(n – 1) + fib(n – 2)
fib(0) = 0, fib(1) = 1
Die Fibonacci-Folge beginnt mit: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, …

Hinweis zu (d) und (e): Berechnen Sie die Funktionen entgegen der rekursiven Definition durch eine Iteration von unten nach oben. Der Parameter n wird in der Variablen x_1 übergeben.

Aufgabe 2

Zeigen Sie durch vollständige Induktion, dass alle Polynomfunktionen $p(x)$ der folgenden Form LOOP-berechenbar sind:

$$p(x) = a_0 + a_1x + a_2x^2 + a_3x^3 \ldots + a_nx^n \text{ mit } a_0, a_1, \ldots, a_n \in \mathbb{N}$$

Hinweise:

- Zeigen Sie zunächst, dass ein Polynom 0-ten Grades LOOP-berechenbar ist.
- Zeigen Sie anschließend, dass ausgehend von der Annahme, dass ein Polynom n-ten Grades LOOP-berechenbar durch ein Programm P ist, dies auch für ein Polynom vom Grad n + 1 gilt.
- Zur Verbesserung der Lesbarkeit können Sie beliebige Variablenbezeichner verwenden, das Endergebnis sollte aber in x_0 stehen.
- Falls nötig, dürfen Sie die Additions- und Multiplikationsfunktion verwenden, ebenso IF … THEN.

Aufgabe 3

Addition und Multiplikation von zwei natürlichen Zahlen gehören zur Klasse der primitiv-rekursiven Funktionen (und dürfen im Folgenden verwendet werden). Zeigen Sie, dass auch die folgenden Funktionen primitiv-rekursiv sind:

a) die Vorgängerfunktion:
   v(x) = x – 1 falls x ≥ 1
   v(x) = 0 falls x = 0
b) die modifizierte Differenz (wie bei LOOP-Programmen):
   f(m, n) = m – n falls m ≥ n
   f(m, n) = 0 sonst
c) die Exponentialfunktion exp(x_1, x_2) = $x_1^{x_2}$
d) die Fakultät n!, definiert durch:
   n! = n (n – 1)!
   0! = 1

$a(2,3) = a(1+1, 2+1) = a(1, a(1+1, 2)) = a(1, a(2,2))$
$= a(1, a(1+1, 1+1)) = a(1, a(1, a(1+1, 1))) = a(1, a(1, a(2,1)))$
$= a(1, a(1, a(1+1, 0+1))) = a(1, a(1, a(1, a(1+1, 0))))$
$= a(1, a(1, a(1, a(1,1)))) = a(1, a(1, a(1, a(0,1))))$
$= a(1, a(1, a(1, 2))) = a(1, a(1, a(0, a(1, 1))))$
$= a(1, a(1, 4)) = a(1, a(0, a(1, 3))) = a(1, a(0, a(0, a(1, 2))))$
$= a(1, a(0, a(0, a(0, a(1, 1))))) = a(1, 7) = 9$

Aufgabe 4

Eine in der theoretischen Informatik für die Betrachtung der Berechenbarkeit wichtige Funktion ist die Ackermann-Funktion a(x, y). Diese ist folgendermaßen rekursiv definiert:

a(0, y) = y + 1
a(x + 1, 0) = a(x, 1)
a(x + 1, y + 1) = a(x, a(x + 1, y))

a) Berechnen Sie: a(0,0), a(0,1), a(1,0) und a(1,1).
   1 2 2 3
b) Berechnen Sie a(2,3). Verwenden Sie dazu die Ergebnisse aus Aufgabe (a).
   Hinweis: Es ist günstig, sich an Stelle der geschachtelten Funktionen den Inhalt des Stacks aufzuschreiben, auf dem die Parameter abgelegt werden, also z.B.:
   statt: a(2, 3) = a(1, a(2, 2)) = a(1, a(1, a(2, 1))) = …
   übersichtlicher: a(2, 3) = 2, 3 = 1, 2, 2 = 1, 1, 2, 1 = …
   Hier entspricht die rechte Seite „oben" auf dem Stack. Es werden immer die beiden oberen Werte vom Stack genommen und entweder durch das Ergebnis ersetzt oder durch die Parameter zur Berechnung der nächsten Ackermann-Funktion.
c) Geben Sie für die folgenden rekursiv definierten Funktionen f_0, f_1 und f_2 äquivalente iterativ definierte (also nicht-rekursive) an:
   $f_0(x) = a(0, x) = x + 1$
   $f_1(x) = a(1, x) = x + 2$
   $f_2(x) = a(2, x) = 2x + 3$

   Gehen Sie dabei von den Ergebnissen in (a) und (b) aus.

   Beweisen Sie die Korrektheit Ihrer iterativen Formeln.
d) Ist die Ackermann-Funktion primitiv rekursiv oder µ-rekursiv?
e) Ist die Ackermann-Funktion LOOP-berechenbar? ja

2,0 = 3
2,1 = 5
2,2 = 7
9
11
15

Aufgabe 5

Implementieren Sie Funktionen zur Berechnung der Ackermann-Funktion in C oder Java und zwar

a) rekursiv (direkt ausgehend von der Definition der Funktion)
b) iterativ
   Hinweis: Implementieren Sie hierfür die Berechnung nach dem Stackprinzip wie in Aufgabe (1b) beschrieben

Aufgabe 6

Für a(3, y) gibt es eine geschlossene Darstellung: $a(3, y) = 2^{y+3} - 3$

$509, 13, 65533, 2^{65536} - 3, 2^{2^{65536}} - 3$

a) Berechnen Sie die Werte a(3, 6), a(4, 0), a(4, 1), a(4, 2), a(4, 3) und a(5, 0) $= 65533$
b) Prüfen Sie die Werte mit Ihrer C/Java-Implementierung. Funktioniert alles wie erwartet?