Aufgabe 1 [Σ 12] Punkte

a) Schritt 1:

$g_0 = 3 \cdot g_{-1} + 2 = 2 \neq 1$

$\Rightarrow g_n = 3 \cdot g_{n-1} + 2 - [n = 0]$

Schritt 2:

$$\begin{aligned} G(z) &= \sum_n g_n z^n \\ &= 3 \cdot \sum_n g_{n-1} z^n + 2 \cdot \sum_n z^n - 1 \\ &= 3 \cdot zG(z) + \frac{2}{1-z} - 1 \end{aligned}$$

Schritt 3:

$G(z) \cdot (1 - 3z) = \frac{2}{1-z} - 1$

$\Rightarrow G(z) = \frac{2}{(1-z)(1-3z)} - \frac{1}{1-3z}$

Schritt 4:

$\frac{2}{(1-z)(1-3z)} = \frac{A}{1-z} + \frac{B}{1-3z}$

$\Rightarrow \quad 2 = A(1-3z) + B(1-z)$

$z = \frac{1}{3}$: $\quad 2 = B(1 - \frac{1}{3}) = \frac{2}{3}B \ \Rightarrow B = 3$

$z = 1$: $\quad 2 = A(1-3) \ \Rightarrow A = -1$

Also:

$$\begin{aligned} G(z) &= \frac{3}{1-3z} - \frac{1}{1-z} - \frac{1}{1-3z} \\ &= \frac{2}{1-3z} - \frac{2}{1-z} \\ &= \sum_n (2 \cdot 3^n - 1) z^n \end{aligned}$$

$\Rightarrow g_n = 2 \cdot 3^n - 1$

b)
1. g(n) = 0.
2. Für die Rekurrenz gilt: $g_0 = 1$, $\frac{g_n}{g_{n-1}} = c$, also $g_n = c \cdot g_{n-1}$ und somit ist die geschlossene Form $g(n) = c^n$.

Aufgabe 2 [Σ 13] Punkte

a) Da k fest ist, ist der Aufwand allein durch n bestimmt. Im schlimmsten Fall ist $n = k^j - 1$. Für $n = 0$ wird keine Multiplikation durchgeführt. Ist sonst n durch k teilbar, so wird n durch k geteilt, andernfalls wird solange multipliziert bis wieder `(n 'mod' k) == 0` eintritt. Im schlimmsten Fall sind das $k - 1$ Multiplikationen. Die Anzahl der Multiplikationen ist in $\Theta(\log_k n)$. Die Rekurrenz ist $g_0 = 1$, $g_n = k - 1 + g_{\lceil n \div k \rceil}$.

b) Es wird hier nur eine Division durchgeführt. Also ist die Anzahl in $\Omega(1)$.

c)
- I.H. `potenz x j k=` x^j gelte für `j<=n`
- I.A. `n == 0: potenz x 0 k = 1 ==` x^0
- I.S. Teilt `k (n+1)`, so ist `potenz x (n+1) k = pot k (potenz x ((n+1)/k) k)` und nach I.H. gilt somit `potenz x (n+1) k = pot k (`$x^{(n+1)/k}$`) =` $(x^{(n+1)/k})^k = x^{n+1}$.

  Ist hingegen `k` kein Teiler von `n+1`, so ist `potenz x (n+1) k = x * potenz x n k` und nach I.H. ist also `potenz x (n+1) k = x *` $x^n = x^{n+1}$.

Aufgabe 3 [Σ 12] Punkte

a) Teile und Herrsche: Basisfall: Hat A die Größe 1, so teste ob $A[1] = y$, sonst: Teile A in zwei Hälften und überprüfe ob y im linken Teil ist. Wenn ja, suche dort weiter, wenn nein suche im rechten Teil weiter. Führe dies rekursiv weiter.

Für den Aufwand werden die Vergleiche gezählt. Die Rekurrenz ist:

$$g_0 = 0, \;\; g_n = g_{n \div 2} + 1.$$

Der Aufwand ist somit in $\Theta(\log n)$.

b) Paßt x nicht an die Position von y, so muß eine neue Position gefunden werden (eine Suche nach x), Danach müssen im schlimmsten Fall alle Elemente um eine Position verschoben werden. Der Aufwand ist somit in $\Theta(n)$

c) Für die Hashfunktion h und die kollidierenden Schlüssel x und y gilt: $h(x) \neq h(y)$, obwohl x schon an Position $h(y)$ eingefügt ist.

Beispiel: $h(x) = x \mod 15$, $s(i) = i + 1$. Füge nacheinaner ein: 1, 16, 2. Dann ist 2 Kollision zweiter Ordnung.

Aufgabe 4 [Σ 12] Punkte

a) Ein Unifikator ist eine Folge (Hintereinanderausführung) von Substitutionen auf einer Klausel, welche Variablen derart durch Terme ersetzen, daß die resultierende Klausel einelementig ist.

b) Behauptung stimmt nicht. Beweis durch Angabe eines Gegenbeispiels:

Für die Klausel $\{P(x), P(y)\}$ ist sowohl $[x/y]$ als auch $[y/x]$ ein allgemeinster Unifikator.

c) Angenommen $F, \neg F$ sind erfüllbarkeitsäquivalent, aber F ist unerfüllbar. Also ist $\neg F$ Tautologie. Folglich sind F und $\neg F$ nicht erfüllbarkeitsäquivalent, denn $\neg F$ besitzt Modelle, F aber nicht.

Die Umkehrung stimmt nicht, denn falls F Tautologie ist, ist F erfüllbar, aber $\neg F$ besitzt kein Modell.

d)

$$\forall x \forall y \forall z (((P(x,y) \wedge P(y,z))) \rightarrow P(x,z))$$

Aufgabe 5 [Σ 10] Punkte

Nr.	Behauptung	richtig	falsch
1.	xRy gdw. $x \in \Theta(y)$ ist eine Äquivalenzrelation.	X	
2.	Selection-Sort braucht weniger Vertauschungen als Merge-Sort.	X	
3.	Für zwei prädikatenlogische Formeln F und G gilt: Ist $F \rightarrow G$ eine Tautologie, so sind F und G erfüllbarkeitsäquivalent.		X
4.	Besitzt ein Graph eine transitive Kante, so ist er kein Baum.	X	
5.	Die Höhe eines Suchbaums mit n Schlüsseln liegt in $O(\log n)$.		X
6.	$O(n^2) + \Theta(n) = O(n^2)$	X	
7.	Die Suche des Wortes “gefunden” in beliebigen Texten der Länge n ist mit dem Algorithmus von Knuth, Morris und Pratt mit $O(n)$ Vergleichen möglich.	X	
8.	Löschen des größten Elements in einem Heap verursacht konstanten Aufwand.		X
9.	Wenn zwei PL-Formeln F, G keine Modelle besitzen, so sind sie kongruent.	X	
10.	In der objektorientierter Programmierung gilt: Jede Klasse kennt die von ihr abgeleiteten Objekte.		X