Aufgabe 1 **1+1+2=4 T**

Prädikatenlogik

a) Geben Sie zu der Formel

$$F = (P(x,y) \wedge \forall x(\neg(Q(x) \vee \forall y R(x,y))))$$

die bereinigte Pränex-Normalform an.

b) Suchen Sie eine Skolem-Normalform SkNF(F) zu der Formel

$$F = \exists u \forall x \forall y \exists v P(x, f(y,u,v), u, g(y), z).$$

c) Beweisen Sie, daß sich aus der folgenden Klauselmenge nicht die leere Klausel ableiten läßt.

$$\{\{P(x), \neg Q(y)\}, \{\neg P(a), Q(b)\}\}$$

Lösung zu Aufgabe 1

a) x und y kommen frei und gebunden vor, also gebundene Umbenennung:

$$F \equiv G := (P(x,y) \wedge \forall u(\neg(Q(u) \vee \forall v R(u,v))))$$

Nun die Quantoren vorziehen:

$$G \equiv H := \forall u \exists v(P(x,y) \wedge (\neg Q(u) \wedge \neg R(u,v)))$$

fertig!

b)

$$\text{SkNF}(F) = \forall x \forall y P(x, f(y,a,h(x,y)), a, g(h(x,y)), z)$$

c) Die Matrix der zugehörigen Formel sieht folgendermaßen aus:

$$F^* = ((P(x) \vee \neg Q(y)) \wedge (\neg P(a) \vee Q(b)))$$

Diese Formel ist offensichtlich erfüllbar! Z.B.: $I_{\mathcal{A}}(P) = I_{\mathcal{A}}(Q) = U_{\mathcal{A}}$

Aufgabe 2 **2+2+4=8 T**

Formalisieren in Prädikatenlogik

Gegeben Sei die folgende Teil-Interpretation einer Formel:

$$\begin{aligned} I_{\mathcal{A}}(P) &= \{(x,x) | x \in U_{\mathcal{A}}\} \\ I_{\mathcal{A}}(Q) &= A \subset U_{\mathcal{A}} \\ I_{\mathcal{A}}(R) &= B \subset U_{\mathcal{A}} := A \backslash \{x\} \\ I_{\mathcal{A}}(f) &= f \text{ ist eine Abbildung } f : A \to B \end{aligned}$$

Geben Sie PL-Formelnan , deren Interpretationen die folgenden Aussagen implizieren:

a) f ist injektiv.

b) f ist surjektiv.

c) A enthält unendlich viele Elemente.

Lösung zu Aufgabe 2

a)
$$F = \forall x \forall y(((Q(x) \wedge Q(y)) \wedge (\neg P(f(x), f(y)) \vee P(x, y))))$$

b)
$$G = \forall y \exists x(((R(y) \wedge Q(x)) \wedge P(y, f(x))))$$

c)
$$H = ((F \wedge G) \wedge Q(x)$$

Aufgabe 3 **1+1+1+1+2=6 T**

O-Kalkül

Gelten die folgenden Aussagen? Beweisen Sie Ihre Antwort!

a) $O(n) = O(\sqrt{n})$.

b) $O(\log_2 n) = O(\log_3 n)$.

c) $O(af(n)) = O(f(n))$.

d) $O(2^n) = O(3^n)$.

e) $O(f) \cdot O(g) = O(f \cdot g)$

Lösung zu Aufgabe 3

a)
$$n \leq c \cdot \sqrt{n} \Leftrightarrow n \leq c^2$$

Also Nein!

b) ja, denn
$$\log_2 n = \frac{\log_3 n}{\log_3 2}$$

c) Ja, für $c = \frac{1}{a}$, $n_0 = 1$.

d)
$$3^n \leq 2^n \Leftrightarrow n \leq \log_{3/2} c$$

Nein!

e)

$$\begin{array}{rl}
„\subseteq“: & \text{Sei } f_1 \in O(f) \text{ und } g_1 \in O(g) \\
\Rightarrow & \exists c_f \exists n_f \forall n(n > n_f \rightarrow 0 \leq f_1(n) \leq cf(n) \text{ und } \exists c_g \exists n_g \forall n(n > n_g \rightarrow 0 \leq g_1(n) \leq cg(n)) \\
\Rightarrow & \text{für alle } n \leq \max n_f n_g \text{ gilt: } 0 \leq f_1(n) g_1(n) \leq c_f c_g f(n) g(n) \\
„\supseteq“: & \text{Sei } h \in O(fg) \\
\Rightarrow & \text{es gibt } c > 0, n_0 \text{ mit } 0 \leq h(n) \leq cf(n)g(n) \text{ für alle } n > n_0. \\
 & \text{Sei nun } f_1(n) = \dfrac{h(n)}{g(n)} \text{ und } g_1(n) = \dfrac{h(n)}{f(n)}, \\
\Rightarrow & f_1 \in O(f) \text{ und } g_1 \in O(g) \\
\Rightarrow & f_1 \cdot g_1 \in O(f) \cdot O(g) \\
 & \text{Außerdem ist } f_1 \cdot g_1 = \dfrac{h^2}{g \cdot f} \leq \dfrac{1}{c} \cdot h
\end{array}$$

Aufgabe 4 **1+2+1+3=7 R**

Haskell, Analyse von Rekurrenzen, Generierende Funktionen

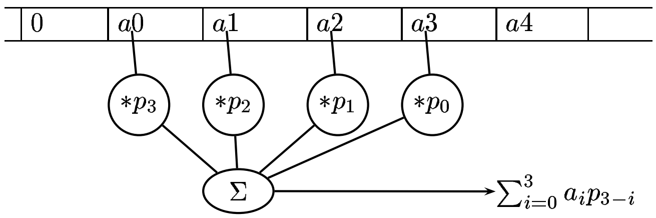
Das Ziel dieser Aufgabe ist es, Ihnen eine Reihe nützlicher Funktionen zur Analyse von Rekurrenzen zur Verfügung zu stellen. Geben Sie bei jeder Funktion, die Sie implementieren auch eine gültige Signatur der Funktion an.

b) Eine unendliche Liste von Zahlen `[a0,a1,a2,a3,..]` in Haskell soll die formale Potenzreihe $\sum_i a_i x^i$ repräsentieren. Implementieren Sie in Haskell die Funktion `cauchy`, die das Cauchy-Produkt

$$\left(\sum_i a_i x^i\right)\left(\sum_i b_i x^i\right) = \sum_n \sum_{i+j=n} a_i b_j x^n$$

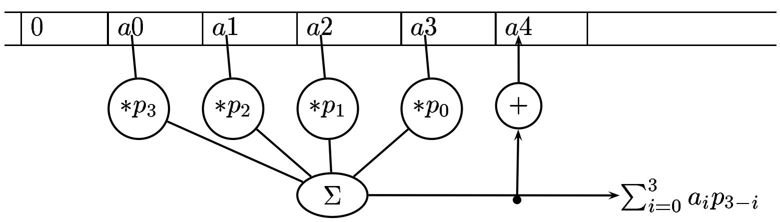
zweier Reihen berechnet.

c) Eine Multiplikation mit einem Polynom (also einer Reihe, die fast nur Nullen enthält) kann einfach über ein Schieberegister implementiert werden: Z.B. für $p(x) = p_0x^0 + p_1x^1 + p_2x^2 + p_3x^3$ kann die Liste der Koeffizienten von $p(x)\sum_i a_i x^i$ mit folgendem Schieberegister berechnet werden:



Implementieren Sie eine Haskell-Funktion `falte`, welche die Multiplikation einer Reihe mit einem Polynom berechnet.

d) Wird das Schieberegister zusätzlich rückgekoppelt, so berechnet es die Division durch $1 - x \cdot p(x)$, wobei $p(x)$ das Rückkopplungspolynom ist. Die folgende Skizze zeigt das Schieberegister aus Teilaufgabe c) mit der beschriebenen Rückkopplung.



Implementieren Sie nun die Division als Haskell-Funktion `dividiere`.

Hinweise:

- Zur Lösung dieser Aufgabe können die folgenden Haskell-Funktionen nützlich sein:

```
take,drop         :: Int → [a] → [a]
splitAt           :: Int → [a] → ([a],[a])
repeat,replicate  :: a → [a]
zipWith           :: (a → b → c) → [a] → [b] → [c]
sum,product       :: Num a ⇒ [a] → a
reverse           :: [a] → [a]
```

- Es ist nicht notwendig den Datentyp `Schieberegister` zu implementieren!

Lösung zu Aufgabe 4

a)

```
differenzen :: (a → a → a) → [a] → [a]
differenzen f [] = []
differenzen f (x) = []
differenzen f (x:y:xs) = (f x y): differenzen xs
```

b)

```
cauchy :: Num a ⇒ [a] → [a] → [a]
cauchy  = mkcauchy 0
 where mkcauchy :: Num a ⇒ Int → [a] → [a] → [a]
       mkcauchy i xs ys = (sum (zipWith (*) (reverse (take (i+1) xs)) ys)):
                          mkcauchy (i+1) xs ys
```

```
falte xs [] = []
falte xs ys = (sum (zipWith (*) (reverse xs) ys )):falte xs (tail ys)
```

d)

```
dividiere :: Num a => [a] -> [a] -> [a]
dividiere xs ys = res:dividiere xs (tail(fi++[res] ++la))
 where (fi,r:la) = splitAt n ys
       res       = r + (sum (zipWith (*) (reverse xs) ys ))
       n         = length xs
```

Aufgabe 5 **2 T**

Ein Inselparadies?

Die Insel Paradiso ist seit jeher mit glücklichen und zufriedenen Menschen bevölkert, die alle entweder blaue oder braune Augen besitzen. Leider existiert eine altes Ritual, das besagt, daß sich jeder, der eines Tages feststellt, daß er braune Augen hat, in der darauf folgenden Nacht umbringt. Da die Bewohner von Paradiso sich zwar diesem Ritual verpflichtet fühlen, aber gleichzeitig wissen, wie unsinnig es ist, sprechen sie nie über die Augenfarben ihrer Mitbewohner. Außerdem gibt es keine Spiegel oder andere Möglichkeiten, die eigene Augenfarbe festzustellen. Auf diese Weise konnte jahrhundertelang jeder Selbstmord verhindert werden.

Eines Tages strandet ein Seemann bei ihnen. Hilfreich und großzügig, wie es der Mentalität der Paradisianer entspricht, wird er wieder aufgepäppelt, und man baut ihm sogar ein neues Boot. Als er in See sticht, versammelt sich die gesamte Bevölkerung am Strand, um ihn zu verabschieden. Zu diesem Zeitpunkt gibt es exakt 50 Menschen mit braunen Augen und 50 mit blauen. Vom Boot aus, ruft der Seemann der Versammlung zu: „Es gibt bei Euch Leute mit braunen Augen!".

Wie groß ist die Bevölkerung von Paradiso in hundert Tagen? Gehen Sie davon aus, daß sich die Bevölkerungszahlen in diesem Zeitraum nicht auf natürliche Weise ändern. Begründen Sie Ihre Meinung.

Lösung zu Aufgabe 5

Obwohl eigentlich durch den Ausruf des Seemannes zunächst keine neue Information fließt, hat er doch katastrophale Auswirkungen. Dadurch, daß alle Inselbewohner gleichzeitig den Ausruf hören, können sie davon ausgehen, daß alle Personen an diesem Tag beginnen, ihre Augenfarbe zu ermitteln.

Induktion über die Anzahl n der Braunäugigen.

I.A.: n=1
Nach der Aussage des Seemans erfährt der einzige Braunäugige, daß es braune Augen gibt. Da er nur Blauäugige sieht, muß er braune Augen haben und bringt sich in der ersten Nacht um.

I.V.: Es gibt ein $n \in \mathbb{N}$, so daß, wenn auf der Insel n Braunäugige leben, sich diese in der n. Nacht das Leben nehmen.

I.S.: $n \rightarrow n+1$
Die $n+1$ Braunäugigen sehen jeweils n andere Braunäugige. Da sie davon ausgehen, daß sie selbst blaue Augen haben, erwarten sie nach I.H., daß sich diese in der n. Nacht umbringen. Dies tun sie aber nicht, da alle so denken. Deswegen wissen nun alle, daß es einen Braunäugigen mehr gibt, welcher nur sie selbst sein kann (da die anderen ja offensichtlich blaue Augen haben). Aus diesem Grund bringen sich alle Braunäugigen in der $n+1$. Nacht um.

Die Blauäugigen denken zwar auf die gleiche Weise, sehen aber einen Braunäugigen mehr. Dadurch verzögert sich der Massenselbstmord um einen Tag. Da sich aber die Braunäugigen dann schon umgebracht haben, wissen Sie, daß sie blaue Augen haben müssen.

Fazit: In Nacht 50 bringen sich alle Braunäugigen um, das heißt, die Bevölkerung beträgt nach 100 Tagen genau 50 Personen.