Aufgaben zum Übungskomplex Algorithmenspezifikation und Verifikation

Aufgabe 1. Bestimmen Sie die schwächste Vorbedingung $P = wp(A, Q)$. Dabei sei A ein Algorithmenstück, Q Nachbedingung sowie $x, y, z \in \mathbf{R}$, $n \in \mathbf{N}$:

a) wp(`x:=2*a+3`, $\{x = 13\}$)

b) wp(`x:=x+y`, $\{x > 2 \cdot y\}$)

c) wp(`x:=(x-y)*(x+y)`, $\{x + x \cdot y \neq 0\}$)

d) wp(
```
x:=abs(x);
y:=abs(y);
z:=abs(x-2*y)-abs(y-2*x),
```
$\{z < x - y\}$)

e) wp(`x:=sqrt(x*x+y*y)`, $\{x < \sqrt{2} \cdot y\}$)

f) wp(`z:=x*y; x:=x div 2; y:=2*y` , $\{z = x \cdot y\}$)

g) wp(`repeat x:=x+1; y:=y+2; until x=z`, $\{y = z\}$)

Aufgabe 2. Es gilt die gleiche Aufgabenstellung wie in der ersten Aufgabe. Allerdings sind nun die Aufgaben etwas schwieriger.

a) wp(`x:=sqrt(x*x-y*y)`, $\{x < \sqrt{2} \cdot y\}$)

b) wp(
```
a:=n mod 30;
if (a+2) mod 5 = 0 then a:=a+4;
if a<18 then a:=a+12,
```
{a ist Primzahl})

c) wp(
```
z:=(y-x)/2;
while x<y do begin x:=x+(y-x)/2; z=z+1; endwhile,
```
$\{z > 10\}$)

d) wp(
```
z:=x-y;
y:=z;
z:=x+y;
x:=z;
if x<y then y:=z else z:=y,
```
$\{x = y\}$)

e) wp(
```
z:=2*x;
x:=z;
while x<y do x:=x+z,
```
$\{x = y\}$)

Aufgabe 3. Bestimmen Sie für folgende Vorbedingung und gegebenen Algorithmus die Nachbedingung:
Vorbedingung: $P = \{\ (a = a_0) \wedge (b = b_0)\ \}$
Algorithmus: `t:=a; a:=b; b:=t;`

Aufgabe 4. Was leistet der folgende Algorithmus für ganze Zahlen $x, y > 0$? Geben Sie Vor- und Nachbedingung formal an. Weisen Sie die Korrektheit des Algorithmus formal nach.

```
z:=x+y;
while z mod 2 = 0 do
   z:=z div 2;
if  (z-y) mod 2 = 0 then py:=true;
                    else py:=false;
```

Aufgabe 5. Gegeben seien die beiden ganzen Zahlen p und x sowie $n \in \mathbf{N}$. Folgende Nachbedingung sei zu erfüllen:

$$Q = \{\, p = \exp(x, n) \,\} \text{ mit } \exp = \begin{cases} 1 & : \quad n = 0 \\ x \cdot \exp(x, n-1) & : \quad n \geq 1 \end{cases}$$

Entwerfen Sie einen Algorithmus zur Lösung der Aufgabe und geben Sie die Schleifeninvariante und die Steuervariable an. Nutzen Sie nur die in der 3. Vorlesung gegebenen Algorithmenbausteine.

Aufgabe 6. Es ist ein Algorithmus zu finden, der das folgende Problem löst:

> Gegeben seien zwei ganze Zahlen $a, b \geq 0$. Gesucht ist der Betrag der Differenz der beiden Zahlen. Dieser soll der größeren der beiden Zahlen zugewiesen werden. Die Funktion `abs()` steht nicht zur Verfügung.

Ist die Aufgabenstellung ausreichend genau spezifiziert? Geben sie zu Ihrem Algorithmus Vor- und Nachbedingung an und beweisen Sie seine Korrektheit.

Zusatzaufgabe 1. Geben Sie sinvolle Vor- und Nachbedingungen für x und x_{newton} an und bestimmen Sie damit die Schleifeninvariante und Steuervariable in folgendem Algorithmenstück (Tip: Näherungsverfahren zur Nullstellenbestimmung bei Polynomen):

```
repeat
   x:=x_newton;
   x_newton:=x-(5*x*x*x-3*x*x+5*x+3)/(15*x*x-6*x+5);
   abweichung:=abs(x_newton-x);
until abweichung<0.001;
```

Zusatzaufgabe 2. Schreiben Sie ein Programm, das folgendes leistet:
Eingabe: Ein Polynom Π_n vom Grad n und eine ganze Zahl $a > 0$
Ausgabe: Die a-te Ableitung von Π_n
Tip: Überlegen Sie sich eine für die Verarbeitung möglichst günstige interne Darstellung und Eingabe des Polynoms. Beeinflußt der Grad n den Ablauf des Programms? Sind für n Einschränkungen zu treffen? Was geschieht bei $a > n + 1$?

Hinweise, Fragen, Anregungen bitte an: `lst@informatik.tu-chemnitz.de`