Schriftliche Abiturprüfung 2010
Fach: Informatik
Lösungs- und Bewertungsvorschlag - **nur für die Hand des Lehrers**

<table>
<tr><th>Teilaufg</th><th>Aufgabe 1</th><th>Pkt</th></tr>
<tr><td>1.1.1</td><td>

$$\begin{aligned} f(3) &= 2 \cdot f(2) + 1 \\ &= 2 \cdot (2 \cdot f(1) + 1) + 1 \\ &= 2 \cdot (2 \cdot (2 \cdot f(0) + 1) + 1) + 1 \\ &= 2 \cdot (2 \cdot (2 \cdot 1 + 1) + 1) + 1 \\ &= 2 \cdot (2 \cdot 3 + 1) + 1 \\ &= 2 \cdot 7 + 1 \\ &= 15 \end{aligned}$$

</td><td>3</td></tr>
<tr><td>1.1.2</td><td>

Lösung in Pascal/Delphi:

a) gemäß der rekursiven Definition:

```
function f(n:integer):integer;
begin
  if n=0 then result:=1
  else result:=2*f(n-1)+1
end;
```

</td><td>3</td></tr>
<tr><td></td><td>

b) iterative Lösung:

```
function f(n:integer):integer;
var i,erg:integer;
begin
  erg:=1;i:=1;
  while i<=n do begin
    erg:=2*erg+1;
    i:=i+1;
  end;
  result:=erg
end;
```

</td><td>3,5</td></tr>
<tr><td>1.2.1</td><td>

Klassendefinition:

```
Set = class
  private  Elemente    : array[1..100] of integer;
           Anzahl      : integer;
  public   constructor create;
           procedure insert(Zahl:integer);
           procedure delete(Element:integer);
           function has(Zahl:integer):boolean;
           procedure clear;
           function isEmpty():boolean;
           function isSubsetOf(Menge:Set):boolean;
end;
```

</td><td>4</td></tr>
<tr><td>1.2.2</td><td>

Methode insert:

```
procedure Set.insert(Zahl:integer);
begin
  if not has(Zahl)
  then begin
    Anzahl:=Anzahl+1;
    Elemente[Anzahl]:=Zahl;
  end;
end;
```

</td><td>3</td></tr>
</table>

Methode isSubsetOf: (3,5)

```
function Set.isSubsetOf(Menge: Set):boolean;
var i:integer;
begin if isEmpty
      then result:=true
      else begin
        result:=true;
        for i:=1 to Anzahl do
          if not Menge.has(Elemente[i]) then result:= false;
      end;
end;
```

Teilaufg.	Aufgabe 2	Pkt
2.1	Codierung des Pfades: 33233322111103	1
2.2	w1: 1+1+2+2+3+3+2 = 14 Übertragungsfehler! w2: 1+1+1+2+2+3+2+3+3+3+3 = 24 Kein Übertragungsfehler, wird akzeptiert.	2
2.3	(siehe Code unten)	4

```
function obAkzeptiert(Zeichenkette: string): boolean;
var i, pruefsumme: integer;
begin
  pruefsumme:=0;
  for i:= 1 to length(Zeichenkette) do
        case Zeichenkette [i] of
          '1' : pruefsumme:= pruefsumme+1;
          '2' : pruefsumme:= pruefsumme+2;
          '3' : pruefsumme:= pruefsumme+3;
        end;
  result:=(pruefsumme mod 4 = 0)
end;
```

2.4.1 (2 Pkt)

Es gibt einen Pfad vom Anfangszustand Z_0 in den Endzustand Z_1, dessen Kantenbeschriftungen das Wort 1112233 ergeben, zum Beispiel:

$Z_0 \xrightarrow{\mathbf{1}} Z_0 \xrightarrow{\mathbf{1}} Z_0 \xrightarrow{\mathbf{1}} Z_2 \xrightarrow{\mathbf{2}} Z_2 \xrightarrow{\mathbf{2}} Z_2 \xrightarrow{\mathbf{3}} Z_1 \xrightarrow{\mathbf{3}} Z_1$

Die Zeichenkette wird akzeptiert.

2.4.2 (1 Pkt)

Im Zustand Z_0 gibt es beim Lesen des Zeichens 1 drei verschiedene mögliche Folgezustände, nämlich Z_0, Z_1 und Z_2.

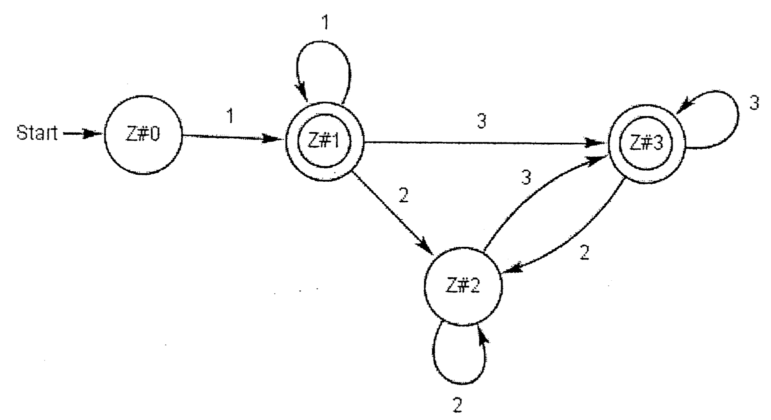
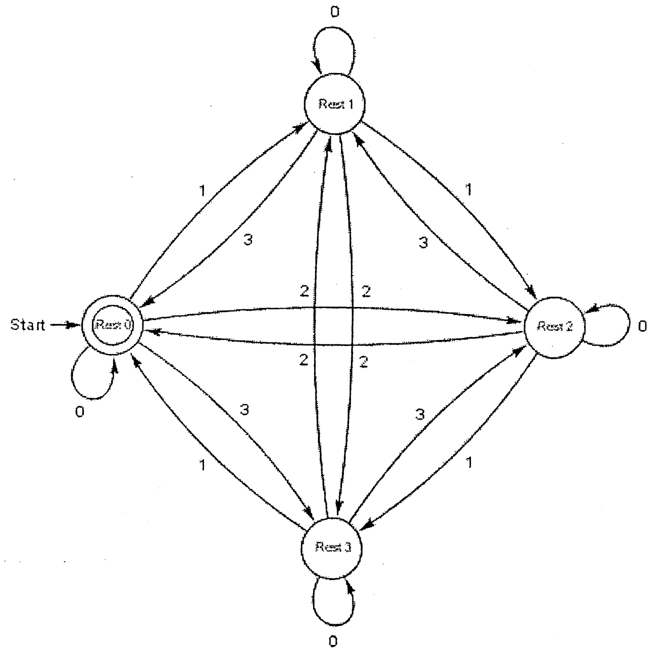
2.4.3 (6 Pkt)

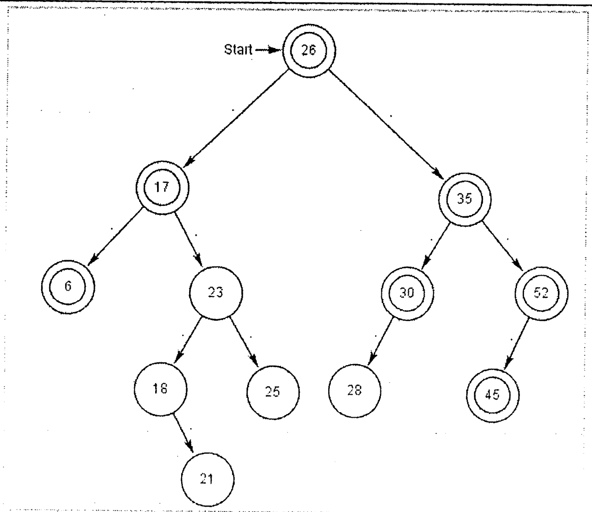
δ^*	1	2	3
$Z\#_0 := \{Z_0\}$	$\{Z_0, Z_1, Z_2\}$	{}	{}
$Z\#_1 := \{Z_0, Z_1, Z_2\}$	$\{Z_0, Z_1, Z_2\}$	$\{Z_2\}$	$\{Z_1\}$
$Z\#_2 := \{Z_2\}$	{}	$\{Z_2\}$	$\{Z_1\}$
$Z\#_3 := \{Z_1\}$	{}	$\{Z_2\}$	$\{Z_1\}$

(3)

Startzustand: $Z\#_0$ (0,5)
Endzustände: $Z\#_1$, $Z\#_3$ (0,5)

Schriftliche Abiturprüfung 2010
Fach: Informatik
Lösungs- und Bewertungsvorschlag - **nur für die Hand des Lehrers**

	Übergangsgraph: 	(2)
2.5		4

Teilaufg.	Aufgabe 3	Pkt
3.1.1	Postorder-Reihenfolge: 6 17 30 45 52 35 26	2
3.1.2		2,5

3.1.3	Um 200 Knoten anzuordnen, benötigt man wegen $2^7 < 200 < 2^8$ einen ausgeglichenen geordneten Binärbaum der Höhe 7. Beim Suchen werden deshalb maximal „Höhe + 1“, also 8, Knoten besucht.	2,5
3.2.1	Beim Start des Programmes stehen in den Speicherzellen 09, 10 und 11 die Zahlen 5, 1 und 0. Zum Inhalt von 11 wird nun solange der Inhalt von 09 addiert und anschließend der Inhalt von 09 um den in der Speicherzelle 10 stehenden Wert 1 dekrementiert, bis die Speicherzelle 09 den Inhalt 0 enthält. Dann steht in der Speicherzelle 11 der Wert der Summe 5 + 4 + 3 + 2 + 1, also 15. Durch „OUT 11“ wird am Ende des Programmlaufs der Wert 15 ausgegeben.	3
3.2.2	00 INM 15 01 INM 16 02 LDA 15 03 JNP 10 04 SUB 17 05 STA 15 06 LDA 16 07 ADD 18 08 STA 18 09 JMP 02 10 OUT 18 11 END 17 DEF 1 18 DEF 0	5
3.3	Gesucht ist ein d mit $e \cdot d = s \cdot (p-1)(q-1) + 1$ für irgend ein s. $(p-1) \cdot (q-1) = 16 \cdot 40 = 640$ Gesucht ist also ein d mit $21 \cdot d - s \cdot 640 = 1$ für irgend ein s. 640 : 21 = 30 Rest 10 21 : 10 = 2 Rest 1 D.h. also: $640 = 30 \cdot 21 + 10$ $21 = 2 \cdot 10 + 1$ Damit ergibt sich: $1 = 21 - 2 \cdot 10$ $= 21 - 2 \cdot (640 - 30 \cdot 21)$ $= 21 - 2 \cdot 640 + 60 \cdot 21$ $= 61 \cdot 21 - 2 \cdot 640$ Also ist d = 61. Das geheime Schlüsselpaar lautet: (61,697)	5