Teilaufgabe d) Anweisungs- vs. Zweig- Überdeckungstests (1 Punkt)

Warum sind Zweigüberdeckungstests für Kontrollflussgraphen, in denen alle Knoten vom Startknoten aus erreichbar sind, mindestens genau so stark wie Anweisungsüberdeckungstests für diese Kontrollflussgraphen?

Teilaufgabe e) Zweig- vs. Pfad- Überdeckungstests (1 Punkt)

In welchen Fällen ist ein Pfadüberdeckungstest stärker als ein Zweigüberdeckungstest?

1+1+1+1+1+1+1+1+1 = 8

Aufgabe 9.2 (5 Punkte)

Die Methode `solveKnapsack(int[] weights, int[] values, int bound)` löst ein Knappsackproblem mit den Gewichten `weights`, Werten `values` und Rucksackkapazität bound. Der Wert `weights[i]` repräsentiert das Gewicht des Gegenstands `i`. Analog repräsentiert der Wert `values[i]` den Wert des Gegenstands `i`.

```
   Optional<Integer> solveKnapsack(int[] weights, int[] values, int bound) {
01   if(weights.length != values.length || bound <= 0) {
02     return Optional.empty();
     }
03   int objects = weights.length;
     int[][] r = new int[objects + 1][bound + 1];
04   for(int i = objects - 1; i >= 0; i--) {
05     for(int j = 1; j <= bound; j++) {
06       if(weights[i] <= j) {
07         int valWithI = values[i] + r[i+1][j-weights[i]];
           int valWithoutI = r[i+1][j];
08         if(valWithI > valWithoutI) {
09           r[i][j] = valWithI;
           } else {
10           r[i][j] = valWithoutI;
           }
         } else {
11         r[i][j] = r[i+1][j];
         }
       }
     }
12   return Optional.of(r[0][bound]);
   }
```

Teilaufgabe a) Kontrollflussgraph erstellen (3 Punkte)

Konstruieren Sie einen Kontrollflussgraphen für die Methode `solveKnapsack`. Benutzen Sie die links vom Methodenrumpf angegebenen Nummern zur Beschriftung der zugehörigen Knoten im Kontrollflussgraphen. Nutzen Sie zur Konstruktion des Kontrollflussgraphen die nächste Seite.

Teilaufgabe b) Anweisungsüberdeckungstest (2 Punkte)

Nennen Sie eine repräsentative Eingabemenge mit höchstens drei verschiedenen Eingaben für einen Anweisungsüberdeckungstest der Methode `solveKnapsack` und geben Sie für

jede Eingabe die Reihenfolge der besuchten Knoten im Kontrollflussgraphen an. Für die Angabe der Array-Eingabewerte können sie die übliche Tupelschreibweise nutzen. Beispielsweise repräsentiert das Tupel (1, 2, 3) ein Array A mit A[0]=1, A[1]=2, A[2]=3. Das Tupel () repräsentiert ein leeres Array. Die angegebenen Arrays dürfen jeweils nicht mehr als zwei Elemente enthalten. Sie können zur Angabe der Eingaben folgendes Schema verwenden:

1. Eingabe

   weights = — (z.B. []) "eyd"

   values = — (z.B. [])

   bound = 0

   Reihenfolge der besuchten Knoten = 1 → 2

2. Eingabe

   weights = [2]

   values = [100]

   bound = 2

   Reihenfolge der besuchten Knoten = 1,3,4,5,6,11,5,6,7,8,9,5,4,12

2" ~~3~~. Eingabe

   weights = [2, 2]

   values = [100, 99]

   bound = 2

   Reihenfolge der besuchten Knoten =

   1,3,4,5,6,11,5,6,7,8,9,5,4,5,6,11,5,6,7,8,10,5,4,12

3. Eingabe

   weights = [1]

   values = [0]

   bound = 1

   Reihfg: 1,3,4,5,6,7,8,10,5,4,12

SWT Globalübung 09

9.1 c) Ang. keine Zweigüberdeckung (Annahme d. Segmentes)

⇒ entweder in 1 nicht zu 3 (Fall 1)
oder in 1 nicht zu 2 (-"- 2)
oder in 3 nicht zu 4 (-"- 3)
oder in 3 nicht zu 5 (-"- 4)

Angen. Fall 1: ⇒ 3 nie besucht (ersichtl. am Ergebnis)
2: ⇒ 2 -"- -"-
3: ⇒ 4 -"- -"-
4: ⇒ 5 -"- -"-

} Widerspruch zu Ausgangsvor. (Anweisungsüberdeckung liegt vor)

d) Wenn eine Zweigüberdeckung vorliegt, dann wurde jeder Zielknoten aller Zweige besucht.

Ang. es gibt einen Knoten w, der nicht besucht wird (d.h. keine Anweisungsüberd. vor)

$\Rightarrow$ (w nicht von Startknoten erreichbar $\vee$ $\exists$ Pfad zu w, der nicht benutzt wurde)

- Ang. w nicht erreichbar (Reach(v,o) = false)

  $\Rightarrow$ Widerspruch

- Ang. $\exists$ Pfad zu w ....

  $\Rightarrow$ $\exists$ Abzweig, der "nicht vollständig besucht" wurde, d.h. ein Zweig der nicht benutzt wurde existiert

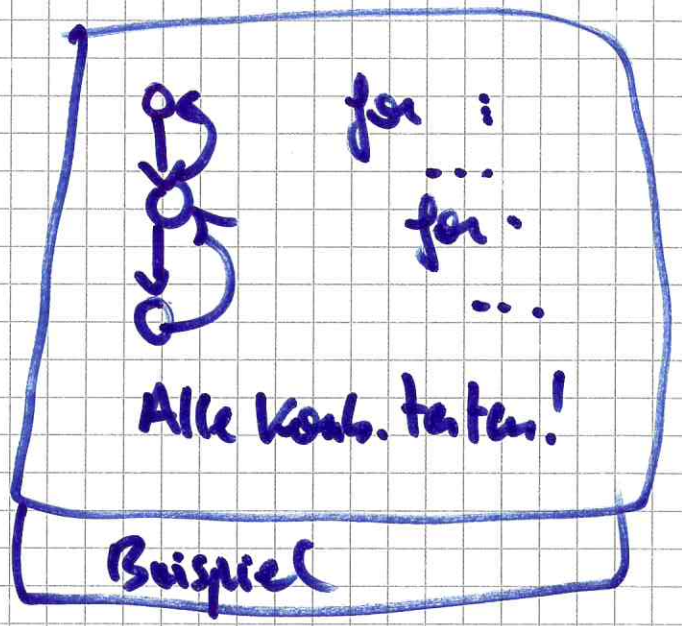
  Widerspruch

$\Rightarrow$ Annahme falsch

$\Rightarrow$ Anweisungsüberd. liegt vor

2) "Besser", wenn

- verschachtelte Zweige
- oder Schleifen vorliegen



9.2

e)