

# Probeklausur zur Vorlesung Softwarekonstruktion – Musterlösung

24.01.2018

Aufgabe	1	2	3	4	5	$\Sigma$
Punkte	12,5	12,5	15	20	20	80
erreicht	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

1. Prüfer: \_\_\_\_\_

2. Prüfer: \_\_\_\_\_

In der Klausur sind insgesamt **80 Punkte** erreichbar.  
Zum Bestehen sind mindestens **40 Punkte (50 %)** erforderlich.

***Wir wünschen viel Erfolg!***

Nach der Korrektur wird Ihre Note unter einem dreistelligen Pseudonym-Code veröffentlicht, den Sie von der Aufsicht erhalten. Tragen Sie diesen Code hier ein:

<input type="text"/>	<input type="text"/>	<input type="text"/>
----------------------	----------------------	----------------------

**Hinweis:** Wir als Klausurveranstalter sind organisatorisch nicht dazu in der Lage, vor bzw. während der Klausur zu überprüfen, ob Teilnehmerinnen und Teilnehmer dazu berechtigt sind, die Klausur mitzuschreiben bzw. ob sie ordnungsgemäß bei der jeweils zuständigen Stelle angemeldet sind. Daher gilt folgendes:

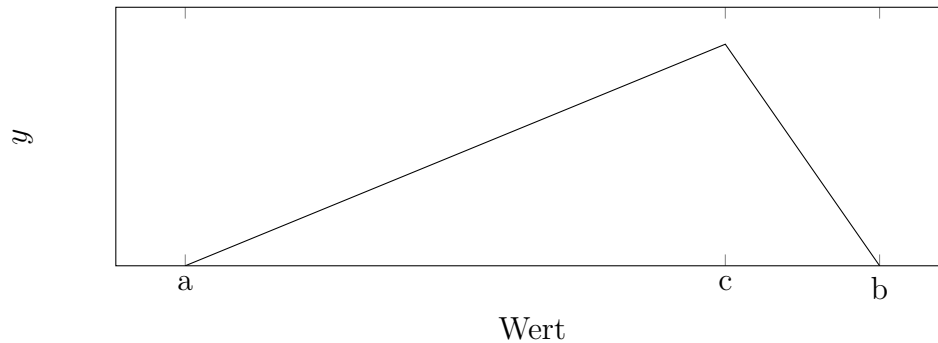
*Durch die Teilnahme an der Klausur erkennt der Teilnehmer bzw. die Teilnehmerin an, dass diese unter Vorbehalt stattfindet. Genauer: Die Anerkennung der bei der Klausur erzielten Note hängt von der jeweils zuständigen Stelle ab und ist nicht automatisch durch die Teilnahme an der Klausur gegeben.*

# Hinweise & Organisatorisches

- Als Hilfsmittel sind ausschließlich ein beidseitig beschriebener A4-Klausurzettel und ein nicht programmierbarer Taschenrechner zugelassen. Der Klausurzettel wird mit der Klausur eingesammelt.
- Legen Sie Ihren Studierendenausweis bzw. eine Immatrikulationsbescheinigung und einen Lichtbildausweis sichtbar vor sich hin, um eine störungsarme Prüfung zu ermöglichen.
- Während der Ausweiskontrolle werden Pseudonym-Codes ausgeteilt. Tragen Sie Ihren Code auf dem Deckblatt ein, damit Sie Ihr Ergebnis so früh wie möglich erfahren können. Verwenden Sie *nur* den ausgegebenen Code. Eigene Codes werden *nicht* berücksichtigt.
- Die Klausur ist zunächst auf Vollständigkeit zu prüfen.
- Jede Seite der Klausur inkl. Deckblatt und Zusatzblättern sowie Ihr mitgebrachter Klausurzettel müssen *leserlich* mit Namen, Vornamen und Matrikelnummer versehen werden.
- Trennen Sie die Seiten der Klausur nicht voneinander. Sollte sich die Heftung lösen, melden Sie sich per Handzeichen bei der Klausuraufsicht.
- Verwenden Sie zur Bearbeitung der Klausur dokumentenechte Stifte in den Farben blau oder schwarz.
- Bearbeitungen mit Bleistift oder anderen nicht dokumentenechten Stiften werden *nicht* berücksichtigt.
- Sonstige Hilfsmittel oder eigenes Papier sind nicht zur Verwendung zugelassen. Die Verwendung von Tipp-Ex, Tintenkiller, Korrektur-Rollern und vergleichbaren Utensilien führt zur Bewertung der betroffenen (Teil-)Aufgaben mit 0 Punkten.
- Lösungen sind eindeutig kenntlich zu machen. Streichen Sie gegebenenfalls nicht zu wertende Antworten in deutlich erkennbarer Form. Die Abgabe mehrerer Lösungen für eine Aufgabe führt zur Bewertung der zugehörigen (Teil-)Aufgaben mit 0 Punkten.
- Zusatzblätter können Sie bei Bedarf von der Klausuraufsicht erhalten. Melden Sie sich dazu per Handzeichen.
- Bleiben Sie nach der Klausur auf Ihrem Platz sitzen, bis die Klausuraufsicht alle Klausuren und Klausurzettel eingesammelt hat *und* Ihnen mitteilt, dass Sie aufstehen und den Raum verlassen dürfen.

**Aufgabe 1 (Schätzen)****12,5 P.**

Betrachten Sie folgende Darstellung einer Verteilung:



Beantworten Sie nun folgende Fragen:

- (a) Welcher Typ Verteilung wird in der Grafik abgebildet? 2,5 P.
- (b) Welche Information wird durch die  $y$ -Achse dargestellt? 2,5 P.

Das Gewicht eines aktive genutzten Flugzeuges setzt sich aus zwei Komponenten zusammen: Leergewicht und Nutzlast. Beide lassen sich wie folgt abschätzen:

Komponente	minimal	mittel	maximal
Leergewicht	238 T	271 T	298 T
Nutzlast	121 T	250 T	313 T

Verwenden Sie nun zwei Ansätze, um Erwartungswert  $E$  und Standardabweichung  $S$  des Gewichtes des Flugzeuges unter der 3-Punkt-Schätzung zu ermitteln:

- (c) Bestimmen Sie zunächst  $E$  und  $S$  für jede Komponente und kombinieren Sie die Ergebnisse mit einem geeigneten Verfahren. 2,5 P.
- (d) Summieren Sie die Schätzwerte für alle Komponenten und bestimmen Sie  $E$  und  $S$  der Summen. 2,5 P.

Geben Sie in beiden Fällen Ihren Rechenweg an und entscheiden Sie:

- (e) Welcher der Ansätze ist gegenüber dem anderen vorzuziehen? Begründen Sie Ihre Antwort. 2,5 P.

**Musterlösung 1**

- (a) Die dargestellte Kurve gehört zur Dreieckverteilung.
- (b) Die Wahrscheinlichkeit, dass der tatsächliche Wert dem Wert auf der  $x$ -Achse entspricht.
- (c) L = Leergewicht, N = Nutzlast, F = Flugzeug.

$$E_L(x) = \frac{a + 4c + b}{6} = \frac{1620}{6} \text{ T} = 270 \text{ T}$$

$$S_L(x) = \frac{b - a}{6} = \frac{60}{6} \text{ T} = 10 \text{ T}$$

$$E_N(x) = \frac{a + 4c + b}{6} = \frac{1434}{6} \text{ T} = 239 \text{ T}$$

$$S_N(x) = \frac{b - a}{6} = \frac{192}{6} \text{ T} = 32 \text{ T}$$

$$E_F(x) = E_L(x) + E_N(x) = (270 + 239) \text{ T} = 509 \text{ T}$$

$$S_F(x) = \sqrt{E_L(x)^2 + E_N(x)^2} = \sqrt{10^2 + 32^2} \text{ T} = \sqrt{1124} \text{ T} \approx 33,5 \text{ T}$$

(d)

$$a = 359 \text{ T}$$

$$c = 521 \text{ T}$$

$$b = 611 \text{ T}$$

$$E_F(x) = \frac{a + 4c + b}{6} = \frac{3054}{6} \text{ T} = 509 \text{ T}$$

$$S_F(x) = \frac{b - a}{6} = \frac{252}{6} \text{ T} = 42 \text{ T}$$

- (e) Komponentenweises Vorgehen ist vorzuziehen. Da die Standardabweichung geringer ist, ist bei komponentenweiser Analyse das Ergebnis genauer als bei einer einzigen Gesamtschätzung.

**Aufgabe 2 (Die Methode der kritischen Pfade)****12,5 P.**

Der Bau eines Wohnhauses lässt sich in folgende Arbeitsschritte unterteilen, die teilweise voneinander abhängen:

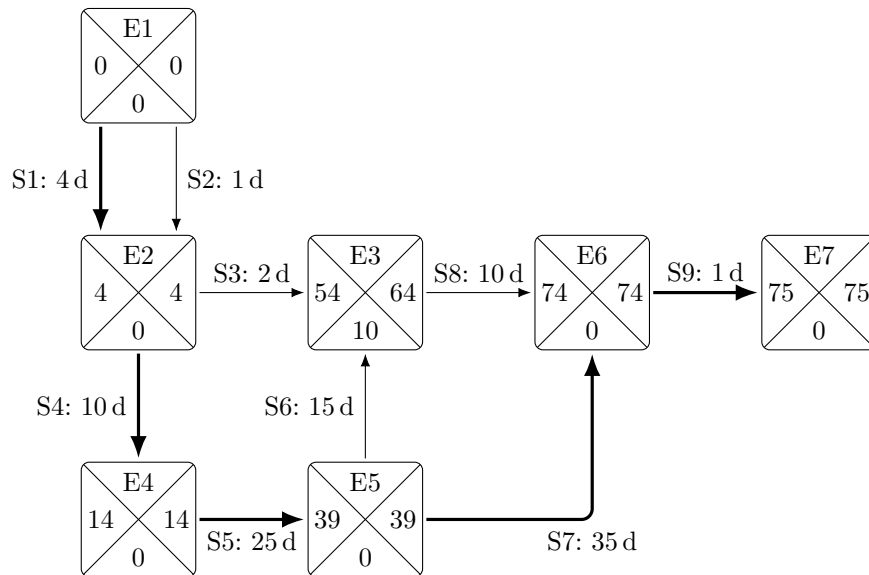
Schritt	Aufwand	Beginnt nach
<b>S1:</b> Baustelle absichern	4 Tage	–/–
<b>S2:</b> Bauamt benachrichtigen	1 Tag	–/–
<b>S3:</b> Leitungsanschlüsse verlegen	2 Tage	S1, S2
<b>S4:</b> Fundament gießen	10 Tage	S1, S2
<b>S5:</b> Hauswand errichten	25 Tage	S4
<b>S6:</b> Decken einziehen	15 Tage	S5
<b>S7:</b> Dach errichten	35 Tage	S5
<b>S8:</b> Wasser- und Elektroinstallation	10 Tage	S3, S6
<b>S9:</b> Schlüsselübergabe	1 Tag	S7, S8

Bestimmen Sie nun die minimale Bauzeit und die kritischen Arbeitsschritte. Gehen Sie dazu wie folgt vor:

- (a) Zeichnen Sie ein CPM-Netz, welches den Ablauf des Bauprojektes bis nach der Schlüsselübergabe visualisiert. Bestimmen Sie mithilfe des Netzes die minimale Dauer des Bauprojektes. 10 P.
- (b) Identifizieren Sie sämtliche kritischen Pfade durch das Netz und kennzeichnen Sie diese. 2,5 P.

## Musterlösung 2

(a)



Die Bauzeit beträgt also mindestens 75 Tage.

(b) Der kritische Pfad ist in fett dargestellt.

**Aufgabe 3 (Architektur)****15 P.**

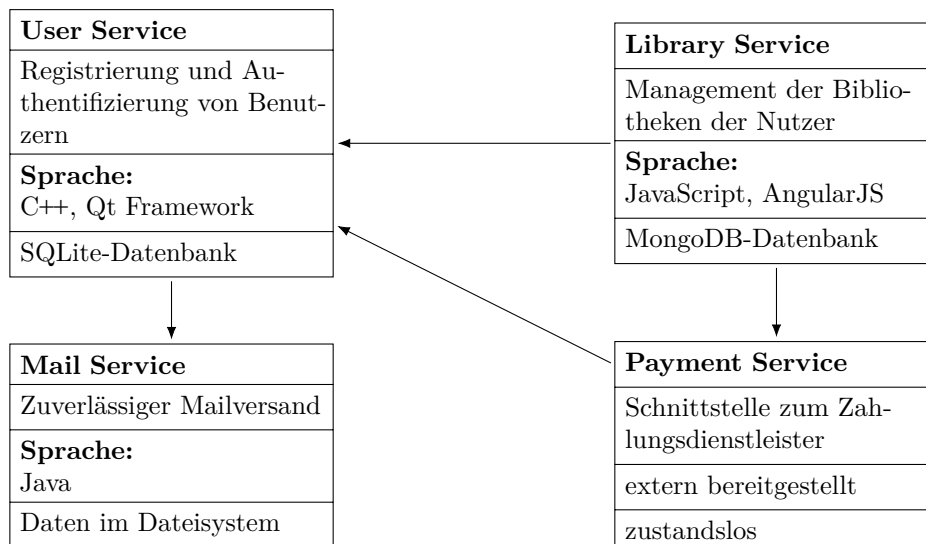
Beantworten Sie zunächst folgende Frage:

- (a) Folgende Architekturmuster werden genutzt, um die Skalierbarkeit von Systemen zu erhöhen: 9 P.

- Sharding
- Load Balancing
- Vertikale Dekomposition (Partition)

Erläutern Sie für jedes Muster in eigenen Worten, wie es das Prinzip der Lastteilung realisiert.

Betrachten Sie nun folgende Architekturskizze für einen Teil eines e-Book-Stores:



- (b) Welches aus der Vorlesung bekannte Architekturmuster wurde genutzt, um den Dienst zu entwickeln? Nennen Sie ein charakteristisches Merkmal des Musters, das sich in der Skizze widerspiegelt. 6 P.

### Musterlösung 3

- (a) Sharding dupliziert die gesamte Anwendung auf mehrere Systeme. Nutzer werden entsprechend einer Eigenschaft der Nutzer (z. B. Name) auf die Instanzen verteilt.

Load Balancing geht analog vor, verteilt die Nutzer aber so, dass die Last möglichst gleich auf alle Systeme verteilt wird.

Vertikale Dekomposition trennt die Anwendung in kleinere, unabhängige Teile auf, die aber jeweils allen Nutzern zur Verfügung stehen. Diese Teile können auf mehrere Systeme verteilt werden.

- (b) Microservices. Die Anwendungen und Daten sind dezentral organisiert; jeder Dienst hält seine eigenen Daten vor. Heterogene Programmiersprachen und Frameworks deuten an, dass die Dienste unabhängig entwickelt werden.



**Aufgabe 4 (Die Domäne der Ausdrücke)****20 P.**

Betrachten Sie folgende Sprache für boolesche Ausdrücke:

$$\begin{aligned}
 b &\in \mathcal{BE} \text{ (Boolesche Ausdrücke)} \\
 b &::= v \mid \mathbf{true} \mid \mathbf{false} \mid \neg b_1 \mid (b_1 \circ b_2) \\
 v &\in X \text{ Variable} \\
 b_1, b_2 &\in \mathcal{BE} \\
 \circ &\in \{\wedge, \vee\}
 \end{aligned}$$

Umgangssprachlich sind boolesche Ausdrücke also:

- Operationen, d. h. die Verknüpfung zweier boolescher Ausdrücke mit  $\wedge$  und  $\vee$ ,
- die Negation eines boolescher Ausdrucks durch ein vorgestelltes  $\neg$ ,
- die Literale **true** und **false** und
- Variablen.

Ziel ist es nun, eine domänenspezifische Sprache zu entwickeln, um diese Ausdrücke darstellen zu können. Gehen Sie dazu wie folgt vor:

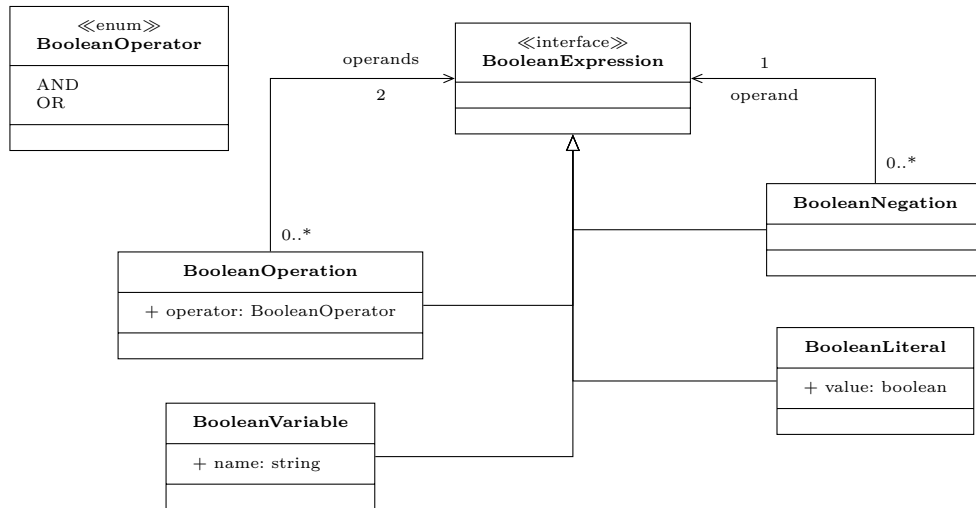
- (a) Stellen Sie ein UML-Klassendiagramm auf, dass die einzelnen Komponenten eines Ausdrucks als Klassen modelliert und ihre Zusammenhänge wiedergibt. Achten Sie darauf, die Syntax von UML, insbesondere Multiplizitäten, korrekt umzusetzen! (*abstrakte Syntax*) 10 P.
- (b) Beschreiben Sie nun eine eigene grafische domänenspezifische Sprache (keine bereits bekannten Sprachen), die Ihrem Klassendiagramm entspricht. (*konkrete Syntax*) 5 P.
- (c) Stellen Sie in Ihrer Sprache nun den Ausdruck 5 P.

$$(x \wedge y) \vee (\neg(y \wedge z) \wedge \mathbf{true})$$

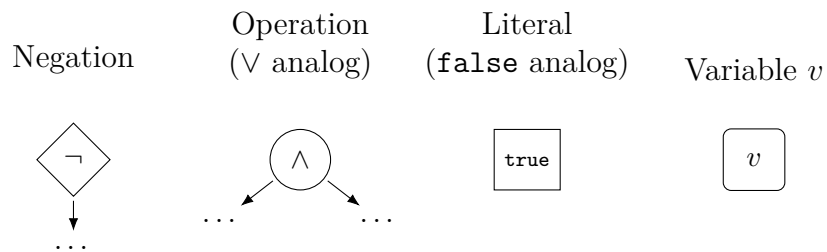
dar.

## Musterlösung 4

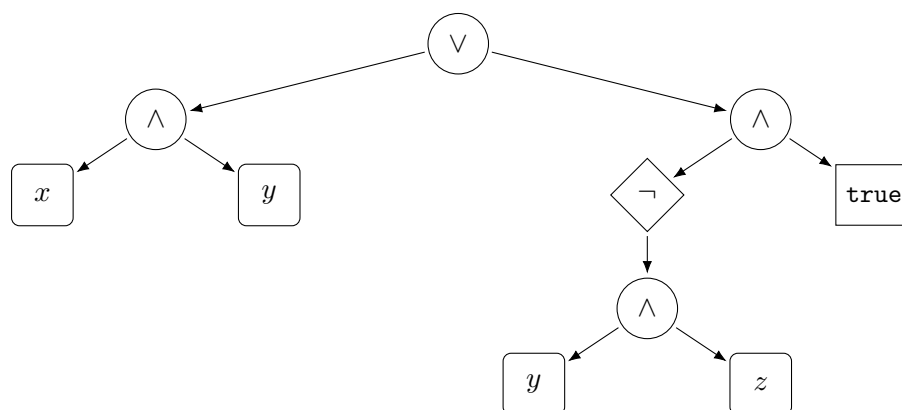
(a)



(b) Nutze eine Syntax ähnlich der von Syntaxbäumen:



(c)



**Aufgabe 5 (Symbolische Ausführung)****20 P.**

Beantworten Sie zunächst folgende Fragen:

- (a) Erklären Sie in eigenen Worten, wann eine Analyse ein falsch-negatives Ergebnis zurückliefert. 2,5 P.
- (b) Nennen Sie für ein Beispiel für eine solche Analyse. 2,5 P.

Gegeben sei nun folgendes Programm mit Eingabevariablen  $i, j$  in While-Syntax:

```

[k := 1]1;
if [j > i + 1]2 then
  [k := 0]3
else
  while [i < j]4 do
    [k := k × i]5;
    [i := i + 1]6
[skip]7;

```

- (c) Werten Sie das Programm mittels symbolischer Ausführung vollständig aus. Nutzen Sie  $\{i \mapsto I, j \mapsto J\}$  als initiale Belegung der Variablen. Stellen Sie Ihre Ergebnisse als Baum dar. Denken Sie daran, nach jedem Blatt des Baumes SAT und ein Modell oder UNSAT zu vermerken! 10 P.

Betrachten Sie zuletzt folgendes Programm:

```

[k := 1]1;
while [i < j]4 do
  [k := k × i]5;
  [i := i + 1]6
[skip]7;

```

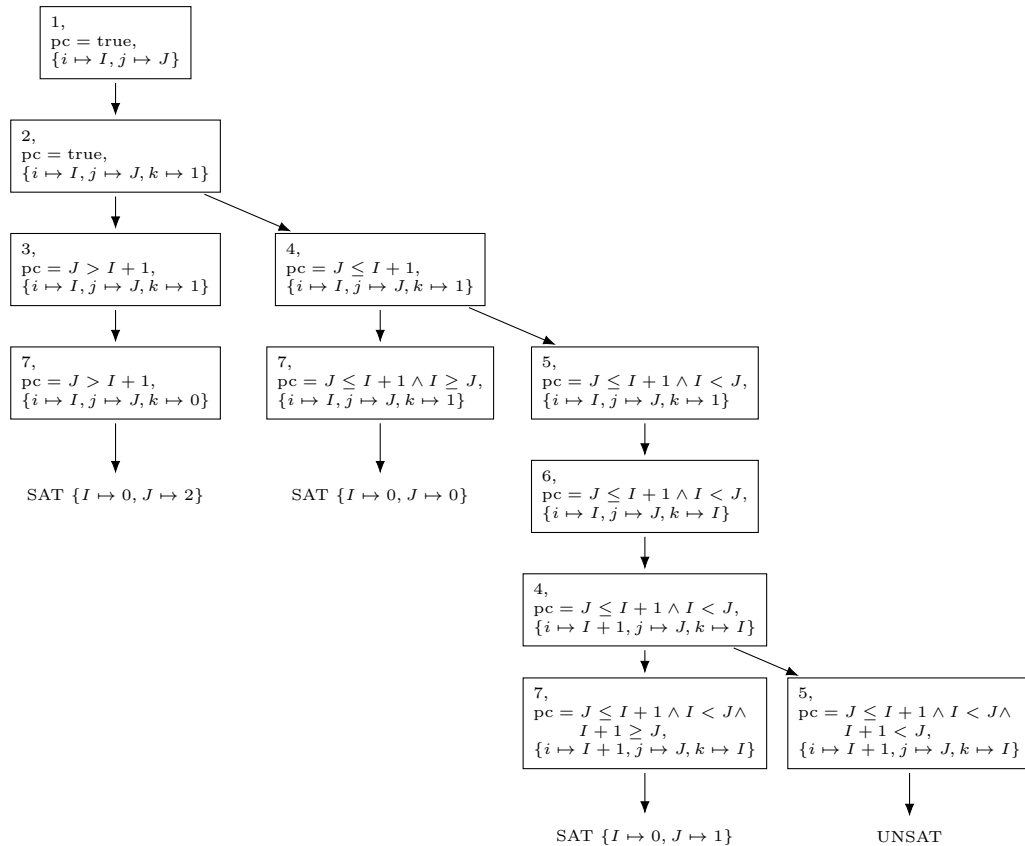
- (d) Ist es möglich, mittels symbolischer Ausführung zu verifizieren, ob die Invariante 5 P.

$$j < 0 \implies k \neq 0$$

während der gesamten Ausführung des gezeigten Programms gilt? Begründen Sie Ihre Antwort.

**Musterlösung 5**

- (a) Ein falsch-negatives Ergebnis entsteht, wenn eine Analyse nur eine Untermenge des tatsächlichen Zustandsraums betrachtet und daher einen Defekt übersieht.
- (b) Unit Tests.
- (c)



- (d) Nein, da die Invariante zwar gilt, aber die symbolische Ausführung nicht terminiert, weil das Terminieren der While-Schleife nicht bewiesen werden kann.