
Aufgaben zur Übergangsprüfung **Grundlagen der Programmierung** im WS 96/97 (WI)

Zeit: 60 Minuten

erlaubte Hilfsmittel: keine

Bitte tragen Sie Ihre Antworten und fertigen Lösungen ausschließlich an den gekennzeichneten Stellen in das Aufgabenblatt ein. Ist ihre Lösung wesentlich umfangreicher, so überprüfen Sie bitte nochmals Ihren Lösungsweg.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 6 Seiten

Aufgabe 1:

Beweisen Sie durch Transformation, daß die folgende Formel ein Satz der Aussagenlogik ist. Begründen Sie die einzelnen Beweisschritte.

$$(x \Rightarrow y) \wedge \neg(x \oplus z) \Rightarrow (\neg y \Rightarrow \neg z)$$

Nutzen Sie diese Seite für die Kladde, die nächste Seite für die fertige Lösung.

$$(x \Rightarrow y) \wedge \neg(x \oplus z) \Rightarrow (\neg y \Rightarrow \neg z)$$

\Leftrightarrow Begründung :

.....

\Leftrightarrow Begründung :

.....

\Leftrightarrow Begründung :

.....

\Leftrightarrow Begründung :

.....

\Leftrightarrow Begründung :

.....

\Leftrightarrow Begründung :

.....

\Leftrightarrow Begründung :

.....

\Leftrightarrow Begründung :

.....

\Leftrightarrow Begründung :

.....

Aufgabe 2:

Analysieren Sie die folgenden Aussagen. Dabei ist die Grundmenge, über die Aussagen gemacht wird, die Menge aller PCs, hier mit PCs bezeichnet, diese ist nicht leer.

Es werden folgende einstellige Elementaraussagen verwendet:

$vonFibosGebaut(pc)$

für einen PC, der von der Firma *Fibos* gebaut wurde

$preiswert(pc)$

für einen preiswerten PC, nicht preiswerte PCs sind teuer

$fehleranfällig(pc)$

für einen fehleranfälligen PC, nicht fehleranfällige PCs sind zuverlässig

Die Aussagen über PCs als prädikatenlogische Formeln

1. $\forall pc \in PCs \bullet vonFibosGebaut(pc) \Rightarrow (\neg preiswert(pc) \wedge fehleranfällig(pc))$
2. $\forall pc \in PCs \bullet vonFibosGebaut(pc) \Rightarrow preiswert(pc) \wedge fehleranfällig(pc)$
3. $\forall pc \in PCs \bullet preiswert(pc) \Rightarrow (vonFibosGebaut(pc) \wedge fehleranfällig(pc))$
4. $\forall pc \in PCs \bullet (\neg fehleranfällig(pc) \vee \neg preiswert(pc)) \Rightarrow \neg vonFibosGebaut(pc)$
5. $\forall pc \in PCs \bullet \neg vonFibosGebaut(pc) \vee (preiswert(pc) \wedge \neg fehleranfällig(pc))$
6. $\forall pc \in PCs \bullet fehleranfällig(pc) \Rightarrow (vonFibosGebaut(pc) \wedge \neg fehleranfällig(pc))$
7. $\forall pc \in PCs \bullet (\neg vonFibosGebaut(pc) \vee \neg preiswert(pc)) \vee fehleranfällig(pc)$
8. $\exists pc \in PCs \bullet preiswert(pc) \Rightarrow (vonFibosGebaut(pc) \wedge fehleranfällig(pc))$
9. $\exists pc \in PCs \bullet vonFibosGebaut(pc) \wedge (preiswert(pc) \wedge fehleranfällig(pc))$
10. $\exists pc \in PCs \bullet (vonFibosGebaut(pc) \wedge fehleranfällig(pc)) \Rightarrow preiswert(pc)$
11. $\exists pc \in PCs \bullet (\neg vonFibosGebaut(pc) \wedge preiswert(pc)) \Rightarrow \neg fehleranfällig(pc)$
12. $\exists pc \in PCs \bullet preiswert(pc) \Rightarrow \neg (vonFibosGebaut(pc) \wedge \neg preiswert(pc))$

Geben sie für die folgenden Aussagen die Nummer(n) von **gleichwertigen** Formeln an, Mehrfachnennungen sind möglich, gibt es keine Formel tragen Sie 0 an die vorgesehene Stelle ein.

1. Es gibt preiswerte, aber fehleranfällige Fibos PCs.

.....

2. Von Fibos gebaute PCs sind immer preiswert und zuverlässig.

.....

3. Von Fibos gebaute PCs sind immer preiswert und fehleranfällig.

.....

4. Von Fibos gebaute PCs sind weder preiswert noch zuverlässig.

.....

5. Alle preiswerten PCs sind von Fibos gebaute, fehleranfällige PCs.

.....

6. Es gibt teure oder von Fibos gebaute fehleranfällige PCs.

.....

7. Es gibt Fibos PCs, die sind preiswert und zuverlässig.

.....

8. Alle von Fibos gebauten preiswerten PCs sind fehleranfällig.

.....

9. Wahr.

.....

10. Falsch.

.....

Aufgabe 3:

Transformieren Sie die folgende Funktion in eine gleichwertige, die mit einer Schleife arbeitet und keine weiteren Funktionsaufrufe enthält.

```
fact(n :  $\mathbb{N}_0$ ; r :  $\mathbb{N}_0$ ) :  $\mathbb{N}_0$ 
  if n ≤ 1
  then
    r
  else
    if n mod 2 = 0
    then
      fact(n div 2, r + 1)
    else
      fact(3 * n - 1, r + 2)
```

fact(*n* : \mathbb{N}_0 ; *r* : \mathbb{N}_0) : \mathbb{N}_0

.....

.....

.....

.....

.....

.....

.....

.....
