
Aufgaben zur Übergangsprüfung **Grundlagen der Programmierung** (Wiederholung) im
WS 2000/01 (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.

Sollten Unklarheiten oder Mehrdeutigkeiten bei der Aufgabenstellung auftreten, so notieren Sie bitte, wie Sie die Aufgabe interpretiert haben.

Viel Erfolg !

Diese Klausur besteht einschließlich dieses Deckblattes aus 5 Seiten

Aufgabe 1:

Transformieren Sie die folgende Funktion in eine gleichwertige Funktion, die mit einer Schleife arbeitet. Benutzen Sie hierzu Techniken aus der Vorlesung.

```
f(i : N0, j : N0) : N0
  if i = 0
  then 0
  else
    if i mod 2 = 0
    then f(i div 2, 2 * j)
    else j + f(i - 1, j)
```

Die vollständig transformierte Funktion:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Aufgabe 2:

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

Es werden folgende einstellige Elementaraussagen verwendet:

japanisch(a)

für ein Auto, das in Japan gebaut wurde

teuer(a)

für ein teures Auto, nicht teure Autos sind preiswert

zuverlässig(a)

für ein zuverlässiges Auto, alle nicht zuverlässigen Autos sind unzuverlässig

Die Aussagen über Autos als prädikatenlogische Formeln

1. $\exists a \in Autos \bullet \neg teuer(a) \Rightarrow (japanisch(a) \wedge \neg zuverlässig(a))$
2. $\exists a \in Autos \bullet japanisch(a) \wedge (\neg teuer(a) \wedge \neg zuverlässig(a))$
3. $\exists a \in Autos \bullet (japanisch(a) \wedge \neg zuverlässig(a)) \Rightarrow \neg teuer(a)$
4. $\exists a \in Autos \bullet (\neg japanisch(a) \wedge \neg teuer(a)) \Rightarrow zuverlässig(a)$
5. $\exists a \in Autos \bullet \neg teuer(a) \Rightarrow \neg(japanisch(a) \wedge teuer(a))$
6. $\forall a \in Autos \bullet japanisch(a) \Rightarrow (teuer(a) \wedge \neg zuverlässig(a))$
7. $\forall a \in Autos \bullet japanisch(a) \Rightarrow (\neg teuer(a) \wedge \neg zuverlässig(a))$
8. $\forall a \in Autos \bullet \neg teuer(a) \Rightarrow (japanisch(a) \wedge \neg zuverlässig(a))$
9. $\forall a \in Autos \bullet (zuverlässig(a) \vee teuer(a)) \Rightarrow \neg japanisch(a)$
10. $\forall a \in Autos \bullet \neg japanisch(a) \vee (\neg teuer(a) \wedge zuverlässig(a))$
11. $\forall a \in Autos \bullet \neg zuverlässig(a) \Rightarrow (japanisch(a) \wedge zuverlässig(a))$
12. $\forall a \in Autos \bullet (\neg japanisch(a) \vee \neg teuer(a)) \vee \neg zuverlässig(a)$

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 unzuverlässige japanische Autos.

.....

2. Japanische Autos sind immer preiswert und zuverlässig.

.....

3. Japanische Autos sind immer preiswert und unzuverlässig.

.....

4. Japanische Autos sind weder preiswert noch zuverlässig.

.....

5. Alle preiswerten Autos sind in Japan gebaute, unzuverlässige Autos.

.....

6. Es gibt teure Autos oder unzuverlässige japanische Autos.

.....

7. Es gibt japanische Autos, die sind nicht teuer aber zuverlässig.

.....

8. Alle preiswerten japanischen Autos sind unzuverlässig.

.....

9. Alle Autos sind zuverlässig.

.....

10. Wahr.

.....

Aufgabe 3:

Gegeben sei die folgende Testfunktion zum Überprüfen, ob eine natürliche Zahl n in Binärdarstellung eine gerade Anzahl von Bits besitzt.

```
evenParity( $n : \mathbb{N}_0$ ) : B
  if  $n = 0$ 
  then
    true
  else
    if  $n \bmod 2 = 0$ 
    then evenParity( $n \operatorname{div} 2$ )
    else  $\neg$ evenParity( $n \operatorname{div} 2$ )
```

Transformieren Sie diese Funktion in eine gleichwertige Funktion, die aber keine bedingten Ausdrücke mehr enthält, sondern neben den Arithmetikoperationen und den Gleichheitstests nur noch logische Operatoren enthält.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Wann ist in einer Programmiersprache diese Transformation erlaubt?

.....

.....