
Aufgaben zur Klausur **Grundlagen der Programmierung** im SS 2000 (WI v303, II v303, MI v403, MI 71)

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:

Konstruieren Sie einen endlichen Automaten $A = (I, Q, \delta, q_0, F)$ mit dem Eingabealphabet $I = \{ (,), [,], a \}$. Der Automat soll alle Zeichenreihen erkennen, in denen die Klammern korrekt verwendet werden, d.h. die Klammern treten nur paarig auf und die runden und eckigen Klammern überschneiden sich nicht. Die runden Klammern dürfen aber nicht geschachtelt werden, das Gleiche gilt für die eckigen Klammern. Beispiele: $([aaaa]a)a$, $a[]aa$ und $()a()$ werden akzeptiert, $a(a , ([a])$ und $((a))a[[]]$ werden nicht akzeptiert.

Versuchen Sie die Anzahl der Zustände möglichst klein zu halten.

Die Zustandsmenge Q :

.....

Der Startzustand q_0 :

.....

Die Endzustandsmenge F :

.....

Die Übergangstabelle δ als Grafik (Zustands-Übergangs-Diagramm):

Aufgabe 2:

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

Es werden folgende einstellige Elementaraussagen verwendet:

$kbMM(p)$

für eine Person, die beim MediaMarkt kauft, also einen MediaMarkt-Kunden

$clever(p)$

für eine clevere Person, als Gegenteil von clever gilt hier blöd

$sW(p)$

für eine Person, die die Werbung des MediaMarktes schlecht findet, das Gegenteil von schlechter Werbung ist gute Werbung

Die Aussagen über Personen

1. $\exists p \in \text{Personen} \bullet (kbMM(p) \wedge clever(p)) \Rightarrow sW(p)$
2. $\exists p \in \text{Personen} \bullet (kbMM(p) \wedge clever(p)) \wedge sW(p)$
3. $\forall p \in \text{Personen} \bullet (kbMM(p) \wedge clever(p)) \Rightarrow \neg clever(p)$
4. $\forall p \in \text{Personen} \bullet clever(p) \Rightarrow (kbMM(p) \wedge \neg sW(p))$
5. $\forall p \in \text{Personen} \bullet \neg sW(p) \Rightarrow (kbMM(p) \vee \neg clever(p))$
6. $\forall p \in \text{Personen} \bullet sW(p) \vee kbMM(p) \vee \neg clever(p)$
7. $\exists p \in \text{Personen} \bullet (\neg kbMM(p) \wedge \neg clever(p)) \wedge \neg sW(p)$
8. $\exists p \in \text{Personen} \bullet \neg(kbMM(p) \vee clever(p) \vee sW(p))$
9. $\forall p \in \text{Personen} \bullet (kbMM(p) \wedge clever(p)) \Rightarrow clever(p)$
10. $\forall p \in \text{Personen} \bullet kbMM(p) \Rightarrow (clever(p)) \vee \neg sW(p)$
11. $\forall p \in \text{Personen} \bullet kbMM(p) \Rightarrow (clever(p)) \oplus \neg sW(p)$

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. Manche Personen finden die MediaMarkt–Werbung gut und kaufen trotzdem nicht beim MediaMarkt. Diese Personen sind nicht clever.
.....
2. Es gibt clevere Personen, die die MediaMarkt–Werbung schlecht finden und trotzdem beim MediaMarkt kaufen.
.....
3. Alle cleveren Personen, die beim MediaMarkt kaufen sind blöd.
.....
4. falsch
.....
5. Alle MediaMarkt–Kunden sind entweder clever oder finden die MediaMarkt–Werbung gut.
.....
6. Alle cleveren Personen sind MediaMarkt–Kunden und finden die MediaMarkt–Werbung gut.
.....
7. Alle Personen sind blöd oder keine MediaMarkt–Kunden.
.....
8. Alle, die die MediaMarkt–Werbung gut finden, sind blöd oder kaufen beim MediaMarkt.
.....
9. wahr
.....
10. Es gibt keine cleveren MediaMarkt–Kunden.
.....

Aufgabe 3:

Gegeben sei die folgende rekursive Funktion

```
f(x : N1) : Z
  if x = 1
  then 0
  else
    if x mod 3 = 0
    then f(x div 3) - 1
    else f(2 * x + 1) + 1
```

Transformieren Sie diese Funktion auf systematische Art und Weise in eine gleichwertige Funktion, die mit einer **while** Schleife arbeitet.

Die Transformation erfordert einen Zwischenschritt, führen Sie diesen in Kladde durch.

Die vollständig transformierte Funktion:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....