Fachhochschule Wedel
Prof. Dr. Uwe Schmidt 30. Januar 2001

Aufgaben zur Klausur **Grundlagen der Programmierung** im WS 2000/01 (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, b, )`$\}$. Der Automat soll alle Zeichenreihen erkennen, in denen vollständig geklammerte Ausrücke vorkommen, wobei als Klammerschachtelungstiefe höchstens 3 erlaubt ist. Innerhalb von Klammern sind nur b's erlaubt, außerhalb nur a's
Beispiele: `(((b)))` , `aaa` und `()()` werden akzeptiert, `a(b` , `bbb` und `((((()b)b)b)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:

Berechnen Sie für die folgenden Ausdrücke die konjunktive Form.

Die konjunktive Form erlaubt Negation nur vor Variablen, Variablen und negierte Variablen dürfen mit $\vee$ verknüpft werden, die so geformten Ausdrücke dürfen mit $\wedge$ verknüpft werden.

Ergebnisse:

1. $\neg(a \vee (b \wedge c))$

   ....

2. $a \wedge (b \vee c)$

   ....

3. $a \oplus b$

   ....

4. $a \Rightarrow (b \Rightarrow c)$

   ....

5. $a \Leftrightarrow b \Rightarrow c$

   ....

Aufgabe 3:

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:

$pol(p)$
: für einen Politiker

$ubs(p)$
: für eine unbestechliche Person, als Gegenteil von unbestechlich gilt hier auch korrupt

$ewg(p)$
: für eine Person, die ihr Ehrenwort gibt

Die Aussagen über Personen

1. $\exists\, p \in Personen \bullet (pol(p) \wedge ubs(p)) \Rightarrow ewg(p)$
2. $\exists\, p \in Personen \bullet (pol(p) \wedge ubs(p)) \wedge ewg(p)$
3. $\forall\, p \in Personen \bullet (pol(p) \wedge ubs(p)) \Rightarrow \neg ubs(p)$
4. $\forall\, p \in Personen \bullet ubs(p) \Rightarrow (pol(p) \wedge \neg ewg(p))$
5. $\forall\, p \in Personen \bullet \neg ewg(p) \Rightarrow (pol(p) \vee \neg ubs(p))$
6. $\forall\, p \in Personen \bullet ewg(p) \vee pol(p) \vee \neg ubs(p)$
7. $\exists\, p \in Personen \bullet (\neg pol(p) \wedge \neg ubs(p)) \wedge \neg ewg(p)$
8. $\exists\, p \in Personen \bullet \neg(pol(p) \vee ubs(p) \vee ewg(p))$
9. $\forall\, p \in Personen \bullet (pol(p) \wedge ubs(p)) \Rightarrow ubs(p)$
10. $\forall\, p \in Personen \bullet pol(p) \Rightarrow ubs(p) \vee \neg ewg(p)$
11. $\forall\, p \in Personen \bullet pol(p) \Rightarrow ubs(p) \oplus \neg ewg(p)$

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 sind keine Politiker, sind korrupt und geben ihr Ehrenwort nicht.

   ....

2. Es gibt unbestechliche Personen, die ihr Ehrenwort nicht geben, und trotzdem Politiker sind.

   ....

3. Alle unbestechlichen Politiker sind korrupt.

   ....

4. falsch

   ....

5. Alle Politiker sind entweder unbestechlich oder geben ihr Ehrenwort.

   ....

6. Alle unbestechlichen Personen sind Politiker und geben ihr Ehrenwort.

   ....

7. Alle Personen sind korrupt oder keine Politiker.

   ....

8. Alle, die ihr Ehrenwort geben, sind korrupt oder Politiker.

   ....

9. wahr

   ....

10. Es gibt keine unbestechlichen Politiker.

   ....
