

---

Gruppe: \_\_\_\_\_

Übung 2

Name: \_\_\_\_\_

**Systemtheorie 1**

Matr.Nr.: \_\_\_\_\_

Wintersemester 2006/2007

Erfolg: \_\_\_\_\_

Abgabe 09.11.2006 10:30

---

Institut für Formale Modelle und Verifikation, Dr. Toni Jussila, Dipl.-Ing. Robert Brummayer

### Aufgabe 5

Gegeben seien die regulären Ausdrücke  $I = (cs \mid sc)(cs \mid sc \mid ssc)^*$  und  $S = cs((s \mid c)^2 \mid sss)^*$ .

(5a) Zeichnen Sie einen EA  $A_1$  der  $I$  entspricht. Hinweis:  $A_1$  muss nicht vollständig sein.

(5b) Zeichnen Sie eine unvollständige und eine vollständige Variante eines nicht-deterministischen EA  $A_2$  der  $S$  entspricht.

(5c) Zeichnen Sie den komplementären Automaten der unvollständigen Variante von  $A_2$ . Akzeptiert dieser die komplementäre Sprache? Wenn nein, dann geben Sie ein Gegenbeispiel an.

(5d) Zeichnen Sie den komplementären Automaten der vollständigen Variante von  $A_2$ . Akzeptiert dieser die komplementäre Sprache? Wenn nein, dann geben Sie ein Gegenbeispiel an.

### Aufgabe 6

Prüfen Sie mit Hilfe des Konformitätstests für endliche Automaten, ob  $A_1$  konform zur vollständigen Variante von  $A_2$  ist. Wenn nein, dann geben Sie ein Gegenbeispiel an.

Hinweis: Der Test kann on-the-fly durchgeführt werden. Es muss nicht der gesamte resultierende Produktautomat konstruiert werden.

### Aufgabe 7

Gegeben sei folgendes Code-Fragment:

```
...
0: assert (i <= n);
1: lock ();
   do {
2:   if (i == n)
3:     unlock ();
4:   i++;
5: } while (i <= n);
6: ...
```

Weiters sei die Spezifikation der erlaubten Lock- und Unlock-Muster  $S = (lu)^*$  gegeben.

(7a) Führen Sie eine totale Abstraktion, wie in der Vorlesung gezeigt, durch, sodass das resultierende Programm nur noch die Kontrollstrukturen sowie die Lock- und Unlock-Operationen enthält. Zeichnen Sie das zugehörige LTS mit  $\varepsilon$ -Übergängen. Erfüllt das LTS die Spezifikation? Wenn nein, dann geben Sie ein Gegenbeispiel an.

(7b) Verfeinern Sie das Programm aus 7a, indem Sie das Prädikat  $b \Leftrightarrow i \leq n$  einführen. Zeichnen Sie das zugehörige LTS mit  $\varepsilon$ -Übergängen. Erfüllt das LTS die Spezifikation? Wenn nein, dann geben Sie ein Gegenbeispiel an.

(7c) Verfeinern Sie das Programm aus 7a, indem Sie die Prädikate  $b_1 \Leftrightarrow i \leq n$  und  $b_2 \Leftrightarrow i == n$  einführen. Zeichnen Sie das zugehörige LTS mit  $\varepsilon$ -Übergängen. Erfüllt das LTS die Spezifikation? Wenn nein, dann geben Sie ein Gegenbeispiel an.

Hinweis: Nicht-Determinismus im Programm-Code können Sie mit Hilfe des Symbols \* ausdrücken. Dieses Symbol stellt die nicht-deterministische Auswahl zwischen *true* und *false* dar.

### Aufgabe 8

Gegeben Sei das LTS  $A = (S, I, \Sigma, T)$  mit  $S = \{1, 2, 3, 4\}$ ,  $I = \{1\}$ ,  $\Sigma = \{a, b\}$  und  $T = \{(1, a, 2), (1, a, 3), (2, a, 2), (2, b, 1), (3, a, 2), (3, b, 3), (3, b, 4)\}$ .

(8a) Gilt  $1 \lesssim 2$  in  $A$ ?

(8b) Ist die Relation  $\lesssim = \emptyset$  im Allgemeinen eine Simulation? Wenn nein, dann geben Sie ein minimales Gegenbeispiel an.

(8c) Ist die Relation  $\lesssim = S \times S$  im Allgemeinen eine Simulation? Wenn nein, dann geben Sie ein minimales Gegenbeispiel an.