Übungen zur Vorlesung
Testautomatisierung 1
Wintersemester 2006/2007

AG BS
Jan Peleska, Tobias
Hartmann

Blatt 4

Revision: 1.1

W-Methode

Folgende Spezifikation in Form eines Mealy-Automaten beschreibt eine Anwendung, die aus einem C-Programmtext alle Kommentare erkennt und ausgibt:

- Die Eingabemenge ist $I = \{*, /, \phi\}$. Dabei abstrahiert ϕ ein beliebiges Zeichen, welches nicht in $\{*, /\}$ enthalten ist.
- Die Ausgabemenge besteht aus den Aktionen $O = \{\mathsf{ignore}, \mathsf{reset}, \mathsf{cat}, \mathsf{truncate}, \mathsf{print}\}$, die einen in der Implementierung zu manipulierenden Ausgabepuffer B betreffen. Diese Aktionen haben folgende Interpretation:
  - ignore: Das eingelesene Zeichen wird ignoriert – der Ausgabepuffer B bleibt unverändert.
  - reset: B wird auf den leeren Puffer zurückgesetzt.
  - cat: Das aktuell eingelesene Zeichen wird hinten an B angehängt.
  - truncate: Das *letzte* Zeichen in B wird gelöscht.
  - print: Der Pufferinhalt wird ausgegeben.
- Die Implementierung soll in Form eines C-Programms erfolgen, welches von `stdin` liest und den Pufferinhalt bei Bedarf nach `stdout` schreibt.

Der zu dieser Aufgabenstellung gehörige Mealy-Automat ist in Abb. 1 Mealy spezifiziert. Da "/" ein Eingabesymbol ist, wird als Trennungssymbol zwischen Eingabe und Aktion ":" gewählt.

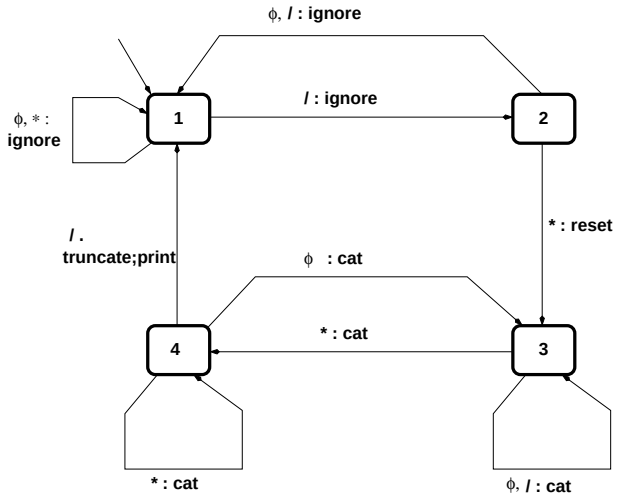

Abbildung 1: Mealy-Automat zur oben beschriebenen Anwendung.

Aufgabe 1: Spezifikationsbasierter Test gegen Mealy-Automaten

Zur oben beschriebenen Spezifikation wurde ein C-Programm entwickelt (`printcomments.c`, hängt neben dem Aufgabenblatt im Netz), welches mit einem Fehler infiziert ist.

Einbindung in RT-Tester Umgebung für Modultests: Binden Sie den Testling folgendermaßen in die RT-Tester Umgebung ein:

1. Als System Under Test soll Funktion

   ```
   void mealyAutomaton(int c)
   ```

   mit den Funktionen

   ```
   int handleState1/2/3/4(int c)
   ```

   dienen.

2. Funktion `int input(void)` ist durch einen RT-Tester Stub zu ersetzen, dessen Verhalten durch das automatische Testdatengenerierungsverfahren (s.u.) mittels `@DEFINE`-Sektion definiert wird.

3. Funktion `void print(void)` ist durch einen RT-Tester Stub zu ersetzen, der durch eine `@GLOBAL` Sektion und eine `@ASSERT`-Sektion definiert wird: Die `@GLOBAL:` Sektion enthält einen Array von Strings, die nacheinander die erwarteten Resultate für die einzelnen Testfälle enthalten. Die `@ASSERT`-Sektion prüft bei jedem Aufruf mittels `memcmp()` den Inhalt von `buffer` im Testling gegen das aktuelle Array-Element (= Referenzstring).

4. Um die Aufrufe der Aktionen truncate(), reset(), ... in der Testumgebung erkennen und auf Korrektheit prüfen zu können, wird der Code des SUT *instrumentiert*: Jede Aktion macht einen zusätzlichen Stub-Aufruf

   ```
   void action(int actionId)
   ```

   bei dem durch die `actionId` identifiziert wird, welche Aktion im SUT getriggert wurde. Dieser Stub wird wieder bei der Testgenerierung generiert: Er enthält einen Array von `actionId`'s, in der Reihenfolge, in welcher diese bei korrektem SUT-Verhalten erscheinen müssten.

Implementierung der W-Methode: Implementieren Sie die W-Methode in C++, so dass sie sich auf den Test der oben beschriebenen Applikation anwenden lässt:

1. Entwickeln Sie Klassen zum Modellieren von Mealy-Automaten:
   (a) State
   (b) Transition
   (c) Input
   (d) Action

2. Instanzieren Sie den oben beschriebenen Automaten aus diesen Klassen.

3. Entwickeln Sie eine Methode zur Generierung der *Transition Cover* T aus dem instanzierten Automatenmodell.

4. Entwickeln Sie eine Methode zur Generierung des *Characterization Sets* W.

5. Generieren Sie aus T und W alle Testfälle unter der Annahme, dass die Implementierung nicht mehr Zustände als die Spezifikation enthält (was de facto nicht richtig ist!). Für Inputs ϕ verwenden Sie zufällig generierte Zeichen $\neq *, /$.

6. Testen Sie das SUT und decken Sie dabei den injizierten Fehler auf!

Abgabe: Bis Montag, 05. Februar 2007, 12:00.

Geben Sie alle Aufgabenlösungen *elektronisch* (mailto:hartmann@informatik.uni-bremen.de) ab. Benutzen Sie bitte Subject `[TA1]` in der EMail.

In allen Dokumenten und Dateien die Namen aller Gruppenmitglieder nicht vergessen!