Übungen zur Vorlesung
Testautomatisierung 1
Wintersemester 2006/2007

AG BS
Jan Peleska, Tobias Hartmann

Blatt 3

Revision: 1.2

Strukturtests – MC/DC-Coverage

Aufgabe 1: Modultests

Für das in Serie 2 verwendete Programm `replace.c` sollen Strukturtests der Funktion

```
int makepat(char *arg, int start, char delim, char *pat);
```

mit dem Ziel **Modified Condition/Decision (MC/DC) Coverage** angefertigt werden.
Um das Vorgehen zu systematisieren, gehen Sie dabei folgendermaßen vor:

1. Erzeugen Sie den Kontrollflussgraphen der Funktion, indem Sie diese mit

   ```
   g++ -g -fdump-tree-cfg ...
   ```

   übersetzen. Hierzu benötigen Sie einen g++ 4.0.x. Der Übersetzungslauf mit diesen Optionen erzeugt eine "*.cfg"-Datei. Beachten Sie dabei, dass im Kontrollflussgraph einige Kanten fehlen ($a \wedge b$ mit $a = false \wedge b = true$, bzw. $a \vee b$ mit $a = true \wedge b = false$), die für die MC/DC-Coverage relevant sind. Ergänzen Sie daher die "*.cfg"-Datei manuell um diese relevanten Kanten.

2. Visualisieren Sie den Kontrollflussgraphen mit Hilfe von Graphviz (dot) (Postscript-Ausgabe ist am einfachsten – html-Ausgabe macht mehr Mühe und ist nicht erforderlich, ist aber besser zu verwenden).

3. Konstruieren Sie die für MC/DC-Coverage erforderlichen Testdaten manuell, aber systematisch auf Basis der Kontrollflussgraphen. Beschreiben Sie für jeden Eingabevektor, welche Kanten des Kontrollflussgraphen mit diesen Daten überdeckt werden.

4. Versuchen Sie, für die aus dem Test resultierenden Datenbelegungen der zusammengesetzten Bedingungen – also beispielsweise

   ```
   while ((!done) && (arg[i] != delim) && (arg[i] != ENDSTR))
   ```

   mit den verschiedenen erforderlichen Abbruchkriterien

   ```
   1. (!done) == TRUE && (arg[i] != delim) == TRUE && (arg[i] != ENDSTR) == FALSE
   2. (!done) == TRUE && (arg[i] != delim) == FALSE && (arg[i] != ENDSTR) == TRUE
   3. (!done) == FALSE && (arg[i] != delim) == TRUE && (arg[i] != ENDSTR) == TRUE
   ```

   und Fortsetzungsbedingung

   ```
   4. (!done) == TRUE && (arg[i] != delim) == TRUE && (arg[i] != ENDSTR) == TRUE
   ```

   – zu erläutern, welche *funktionalen* Eigenschaften durch diese Auswertebedingung geprüft werden. Welche Kombinationen sind nur Robustheitstests – also ohne funktionalen "Beitrag"?

5. Transformieren Sie `makepat()` in eine äquivalente C-Funktion `makepat1()`, so dass die Anweisungsüberdeckung von `makepat1()` bereits die MC/DC-Coverage impliziert.

6. Führen Sie die Tests mit RT-Tester für `makepat()` und `makepat1()` mit denselben Testvektoren durch und weisen Sie die erzielte Überdeckung mittels `gcov` nach.

Abgabe: Bis Montag, 10. Januar 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!