

Blatt 2

Revision: 1.1

Strukturtests – C1-Coverage

In dieser Aufgabenserie stehen Strukturtests zur Codeüberdeckung gemäß **Decision Coverage (C1)** im Vordergrund. Wir nutzen dazu Programmquellen als Testling, die parallel zum Aufgabenblatt auf der TA1-Webseite zur Verfügung gestellt sind. Die Quellen stammen in leicht modifizierter Form aus einer von Siemens Corporate Research publizierten Bibliothek von Beispielprogrammen, die als “Benchmarks” zur Prüfung der Wirksamkeit von Testverfahren dienen. Die vollständige Bibliothek ist unter <http://www.cc.gatech.edu/aristotle/Tools/subjects> verfügbar. Zu jedem Programm wurde eine Serie von **Mutanten** erstellt, die typische Programmierfehler enthalten.

Die für dieses Aufgabenblatt verwendete Software besteht aus drei Funktionen, welche den Austausch eines Textpatterns $p1$ gegen ein Muster $p2$ ermöglichen:

```
int getpat(char* arg, char* pat);
```

erhält das zu suchende Textpattern in `arg` und schreibt es in aufbereiteter Form nach `pat`.

```
bool getsub(char* arg, char* sub);
```

erhält den Substitutionsstring in `arg`, kappt ungültige führende und endende Zeichen und schreibt den bereinigten String nach `sub`.

```
void subline(char *lin, char *pat, char *sub, char *resStr);
```

erhält eine Textzeile `lin`, das Pattern `pat` und den Substitutionsstring `sub` als Eingabe und schreibt den modifizierten String, in welchem alle Vorkommnisse von `pat` in `lin` durch `sub` ersetzt wurden, in den Ergebnisstring `resStr`.

Das die 3 Funktionen verwendende Hauptprogramm muss die Strings in der erforderlichen Größe bereitstellen.

Aufgabe 1: Software-Integrationstest

Realisieren Sie einen SW-Integrationstest mit dem Testwerkzeug RT-Tester in der folgenden Form:

- Eine Abstrakte Maschine.
- Deklaration der Strings `pat`, `sub`, `resStr` als globale Variablen geeigneter Größe.
- Für jeden Testfall wird die Aufruffolge

```
getpat(arg1, pat);  
getsub(arg2, sub);  
subline(lin,pat,sub,resStr);
```

mit den Testfall-spezifischen Argumenten `arg1`, `arg2`, `pat`, `sub`, `lin` ausgeführt.

- Am Ende jeder Testfallausführung wird die korrekte Wirkungsweise mittels `@rttAssert()` geprüft.
- Entwerfen Sie die Eingangsdaten mit dem Ziel, eine **Zweigüberdeckung (C1)** zu erreichen.

Aufgabe 2: Modultests

Führen Sie ergänzende Modultests durch, um während des SW-Integrationstests schwer erreichbare Zweige in speziellen Modultestsituationen zu überdecken. Prüfen Sie auch hier wieder das Verhalten des Testlings gegen geeignet definierte erwartete Resultate.

Dokumentieren Sie bei beiden Aufgaben die Testfälle innerhalb der Testprozeduren (*.rts Dateien) im RT-Tester Format.

Fertigen Sie eine Abschlussdokumentation an (RT-Tester Standard-Dokumentationsformat).

Der zur Verfügung gestellte Quellcode enthält einen Fehler. Wenn Ihre Testfälle diesen aufdecken, werden 20 Sonderpunkte vergeben – die maximal erreichbare Punktzahl ist damit 120.

Abgabe: Bis Montag, 27. November 2006, 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!