Validierung

Simon Bischof, Jan Haag, Adrian Herrmann, Lin Jin, Tobias Schlumberger, Matthias Schnetz

Praxis der Softwareentwicklung
Projekt 3:
Automatisches Prüfen der Korrektheit von Programmen
Gruppe 1

WS 2011/2012

Inhaltsverzeichnis

1 GUI Testplan

1.1 Menubar

1.1.1 „File“ → „New“

1. Öffnen einer neuen Datei
   - Erwartetes Ereignis: Der Inhalt des Editors wird ohne zu speichern gelöscht, Breakpoints werden entfernt und die Konsolen geleert.
   - Status: BEHOBEN
     Breakpoints werden nicht entfernt, Konsole nicht geleert

1.1.2 „File“ → „Load“

1. Laden einer nichtexistenten Datei
   - Erwartetes Ereignis: Die Datei wird nicht geladen.
   - Status: OK
2. Laden einer Datei, die vom Programm erzeugt wurde oder einer txt-Datei
   - Erwartetes Ereignis: Die ausgewählte Datei wird in den Editor geladen.
   - Status: OK
3. Laden einer Datei, die nicht vom Programm erzeugt wurde und keine txt-Datei ist
   - Erwartetes Ereignis: Die Datei wird nicht geladen.
   - Status: FEHLSCHLAG
     Programm hängt sich auf

1.1.3 „File“ → „Save“

1. Speichern in einer nichtexistenten Datei
   - Erwartetes Ereignis: Die Datei wird erzeugt, der Inhalt des Editors darin gespreichert.
   - Status: OK
2. Speichern in einer vom Programm erzeugten Datei oder txt-Datei
   - Erwartetes Ereignis: Der Inhalt der Datei wird durch den des Editors ersetzt.
   - Status: OK
3. Speichern in einer Datei, die nicht vom Programm erzeugt wurde und keine txt-Datei ist
   - Erwartetes Ereignis: Die Datei wird nicht überschrieben.
   - Status: FEHLSCHLAG
     Die Datei wird überschrieben.

1.1.4 „File“ → „Exit“

1. Beenden des Programms
   - Erwartetes Ereignis: Das Programm wird sofort beendet.
   - Status: OK

1.1.5 „Edit“ → „Undo“

1. Rückgängigmachen des zuletzt eingetippten Zeichen
   - Erwartetes Ereignis: Das zuletzt eingetippte Zeichen wird gelöscht.
   - Status: OK
2. Beliebige Wiederholung von Punkt 1
   - Erwartetes Ereignis: Die zuletzt eingetippten Zeichen werden gelöscht.
   - Status: OK
3. Rückgängigmachen des zuletzt gelöschten Zeichen
   - Erwartetes Ereignis: Das zuletzt gelöschte Zeichen wird wieder hergestellt.
   - Status: BEHOBEN
     Das zuletzt gelöschte Zeichen wird nicht wieder hergestellt
4. Beliebige Wiederholung von Punkt 3
   - Erwartetes Ereignis: Die zuletzt gelöschten Zeichen werden wieder hergestellt.
   - Status: OK
5. Rückgängigmachen der letzten Paste-Aktion
   - Erwartetes Ereignis: Die zuletzt eingefügte Zeichenkette wird gelöscht.
   - Status: OK
6. Beliebige Wiederholung von Punkt 5
   - Erwartetes Ereignis: Die zuletzt eingefügten Zeichenketten werden gelöscht.
   - Status: OK
7. Rückgängigmachen der letzten Cut-Aktion
   - Erwartetes Ereignis: Die zuletzt gelöschte Zeichenkette wird wieder hergestellt.
   - Status: OK
8. Beliebige Wiederholung von Punkt 7
   - Erwartetes Ereignis: Die zuletzt gelöschten Zeichenketten werden wieder hergestellt.
   - Status: OK
9. Rückgängigmachen der Funktion „File“ → „New“
   - Erwartetes Ereignis: Der alte Inhalt des Editors wird wieder hergestellt.
   - Status: OK
10. Rückgängigmachen der Funktion „File“ → „Load“
    - Erwartetes Ereignis: Der alte Inhalt des Editors wird wieder hergestellt.
    - Status: OK
11. Undo, obwohl noch keine Aktion ausgeführt wurde
    - Erwartetes Ereignis: Es passiert nichts.
    - Status: OK

1.1.6 „Edit“ → „Redo“

1. Rückgängigmachen der letzten Undo-Aktion
   - Erwartetes Ereignis: Die rückgängig gemachte Aktion wird hergestellt.
   - Status: OK
2. Beliebige Wiederholung von Punkt 1
   - Erwartetes Ereignis: Die rückgängig gemachten Aktionen werden hergestellt.
   - Status: OK
3. Redo, obwohl noch kein Undo ausgeführt wurde
   - Erwartetes Ereignis: Es passiert nichts.
   - Status: OK

1.1.7 „Edit“ → „Cut“

1. Löschen der markierten Zeichenkette
   - Erwartetes Ereignis: Die markierte Zeichenkette wird gelöscht.
   - Status: OK
2. Cut ohne markierte Zeichenkette
   - Erwartetes Ereignis: Es passiert nichts.
   - Status: BEHOBEN
     Programm stürzt ab.

1.1.8 „Edit“ → „Copy“

1. Kopieren der markierten Zeichenkette
   - Erwartetes Ereignis: Die markierte Zeichenkette wird zum Kopieren gespeichert.
   - Status: OK
2. Beliebige Wiederholung von Punkt 1
   - Erwartetes Ereignis: Die zuletzt kopierte Zeichenkette wird gespeichert.
   - Status: OK
3. Copy ohne markierte Zeichenkette
   - Erwartetes Ereignis: Es passiert nichts.
   - Status: BEHOBEN
     Programm stürzt ab.

1.1.9 „Edit“ → „Paste“

1. Einfügen der aus dem Programm kopierten Zeichenkette
   - Erwartetes Ereignis: Die kopierte Zeichenkette wird im Editor eingefügt.
   - Status: OK
2. Einfügen der aus einem anderen Programm kopierten Zeichenkette
   - Erwartetes Ereignis: Die kopierte Zeichenkette wird im Editor eingefügt.
   - Status: OK

3. Beliebige Wiederholung von Punkt 1 oder 2
    - Erwartetes Ereignis: Die kopierte Zeichenkette wird jedes Mal im Editor eingefügt.
    - Status: OK

1.1.10 „Edit“ → „Settings“

1. Öffnen des Settingsfensters
    - Erwartetes Ereignis: Das Fenster zur Einstellung von Z3-Settings wird geöffnet.
    - Status: OK

1.1.11 „Run“ → „Random Test“

1. Öffnen des Randomtestfensters
    - Erwartetes Ereignis: Das Fenster für Randomtests wird geöffnet.
    - Status: OK

1.1.12 „Help“ → „Help“

1. Öffnen des Helpfensters und Anzeigen der Helpdokumentation
    - Erwartetes Ereignis: Die Helpdokumentation wird geöffnet.
    - Status: OK

1.1.13 „Help“ → „About“

1. Öffnen des Aboutfensters
    - Erwartetes Ereignis: Das Aboutfenster wird geöffnet.
    - Status: OK

1.2 Frames

1.2.1 Settingsframe

1. Speichern von korrekten Eingaben
    - Erwartetes Ereignis: Es wird eine Erfolgsmeldung ausgegeben und die neuen Eingaben stehen in den entsprechenden Textfeldern.
    - Status: OK
2. Speichern von inkorrekten Eingaben
    - Erwartetes Ereignis: Es wird eine Fehlermeldung ausgegeben und die alten Werte werden wiederhergestellt.
    - Status: BEHOBEN
      Wenn der Pfad nicht korrekt eingegeben wurde, wird trotzdem die Erfolgsmeldung angezeigt.
3. Klick auf „Close“ Button
    - Erwartetes Ereignis: Das Settingsfenster wird geschlossen.
    - Status: OK
4. Ausführen von 1, 3 und anschließendes Öffnen des Fensters.

- Erwartetes Ereignis: Die bei 1 eingegebenen neuen Werte stehen immernoch in den entsprechenden Textfeldern.
- Status: OK

5. Ausführen von 2, 3 und anschließendes Öffnen des Fensters.
   - Erwartetes Ereignis: Die Werte vor der Änderung stehen immernoch in den entsprechenden Textfeldern.
   - Status: OK

1.2.2 Helpframe

1. Auswählen der einzelnen Abschnitte
   - Erwartetes Ereignis: Der ausgewählte Abschnitt wird angezeigt.
   - Status: OK
2. Klick auf „Close“ Button
   - Erwartetes Ereignis: Das Helpfenster wird geschlossen.
   - Status: OK

1.3 Views

1.3.1 Editor

1. Eingabe, Modifikation von Quelltext im idle-Zustand
   - Erwartetes Ereignis: Der Inhalt des Editors kann beliebig verändert werden, solange es kein Programm läuft oder pausiert ist.
   - Status: OK
2. Eingabe, Modifikation von Quelltext im nicht-idle-Zustand
   - Erwartetes Ereignis: Der Inhalt des Editors kann nicht verändert werden, solange ein Programm läuft oder pausiert ist.
   - Status: OK
3. Eingabe von Keywords und Zahlen
   - Erwartetes Ereignis: Die Keywords „int, bool, array, true, false, main, while, if, else, return, assert, assume, ensure, invariant“ und Zahlen werden farbig hervorgehoben.
   - Status: OK
4. Setzen oder Entfernen von Statementbreakpoints im nicht-idle-Zustand
   - Erwartetes Ereignis: Breakpoints können nicht gesetzt oder entfernt werden, solange ein Programm läuft oder pausiert ist.
   - Status: OK
5. Setzen von Statementbreakpoints im idle-Zustand
   - Erwartetes Ereignis: Statementbreakpoints können nur gesetzt werden, wenn in der Zeile ein Statement steht.
   - Status: OK
6. Entfernen von Statementbreakpoints im idle-Zustand
   - Erwartetes Ereignis: Breakpoint wird entfernt.
   - Status: BEHOBEN
     Breakpoint kann nicht mehr entfernt werden, wenn die Zeile so modifiziert wurde, dass sie keinen Statement mehr enthählt

1.3.2 Globalbreakpointview

1. Einfügen, Entfernen, Aktivieren, Deaktivieren von Globalbreakpoints im nicht-idle-Zustand
   - Erwartetes Ereignis: Globalbreakpoints können nicht verändert werden, solange ein Programm läuft oder pausiert ist.
   - Status: OK
2. Einfügen und Entfernen von syntaktisch und semantisch korrekten Zeichenketten, z.B. Identifier, Integer-, Booleanliteral, Arrayzugriff, Funktionsaufruf, arithmetischer oder boolscher Ausdruck
   - Erwartetes Ereignis: Der Breakpoint wird eingefügt bzw. entfernt.
   - Status: OK
3. Einfügen von syntaktisch oder semantisch inkorrekten Zeichenketten, z.B. Deklaration, Zuweisung, Spezifikation, If-, While-, Return-Anweisung, Ausdruck mit Quantoren
   - Erwartetes Ereignis: Der Breakpoint wird nicht eingefügt.
   - Status: BEHOBEN
     Einfügen nach einem korrekt eingefügten Breakpoint bringt das Programm zum Absturz.

1.3.3 Helpbox

1. Es wird eine Stringkette eingegeben und nach Hilfe gesucht
   - Erwartetes Ereignis: Der zur Stringkette am relevanteste Abschnitt wird in der Helpbox angezeigt.
   - Status: BEHOBEN
     Wenn nach „else“ gesucht wird, erscheint die Einleitung

1.4 Testprogramme

1.4.1 Leeres Programm

Programm mit leerem String

1. Check Syntax
   - Erwartetes Ereignis: Fehlermeldung in der Errorkonsole, dass das Programm keine main-Methode besitzt.
   - Status: OK
2. Run/Single Step
   - Erwartetes Ereignis: Die gleiche Fehlermeldung in der Errorkonsole wie in Punkt 1.
   - Status: OK
3. Validate
   - Erwartetes Ereignis: Es passiert nichts.
   - Status: OK
4. Randomtest
   - Erwartetes Ereignis: Es erscheint die Meldung, dass das Programm keine korrekte Syntax besitzt.
   - Status: OK

1.4.2 Programm ohne richtige main-Methode

Programm, in dem es keine main-Methode gibt, die main-Methode sich in einem Statementblock befindet oder die main-Methode return-Statement oder Rückgabewert hat

1. Check Syntax
   - Erwartetes Ereignis: Fehlermeldung(en) in der Errorkonsole, dass das Programm keine main-Methode oder Syntaxfehler besitzt.
   - Status: OK
2. Run/Single Step
   - Erwartetes Ereignis: Die gleiche Fehlermeldung in der Errorkonsole wie in Punkt 1.
   - Status: OK
3. Validate
   - Erwartetes Ereignis: Es passiert nichts.
   - Status: OK
4. Randomtest
   - Erwartetes Ereignis: Es erscheint die Meldung, dass das Programm keine korrekte Syntax besitzt.
   - Status: OK

1.4.3 Programm zum Testen von Breakpoints

1. Setzen von Statementbreakpoints an beliebiger Stelle
   - Erwartetes Ereignis: Die Programmausführung wird angehalten, wenn ein Statementbreakpoint getroffen wurde.
   - Status: OK
2. Setzen von Globalbreakpoints (aktiviert oder deaktiviert)
   - Erwartetes Ereignis: Die Programmausführung wird angehalten, wenn ein aktiver Globalbreakpoint getroffen wurde.
   - Status: OK

1.4.4 Programm zum Testen von Randomtests

1. Ausführung mit korrekten Eingaben
   - Erwartetes Ereignis: Es werden Werte aus den angegeben Intervallen ausgewählt und in der Miscksole angezeigt.
   - Status: OK
2. Ausführung mit falschen/leeren Eingaben
   - Erwartetes Ereignis: Die Parameter werden alle auf 0 bzw. false gesetzt.
   - Status: OK

1.4.5 Programm zum Testen von Arrays

1. Check Syntax
   - Erwartetes Ereignis: Syntaxfehler werden korrekt angezeigt.
   - Status: FEHLSCHLAG
     Es wird manchmal die ungenaue Fehlermeldung „AST creation not possible!“zurückgegeben
2. Run/Single Step
   - Erwartetes Ereignis: Korrekte Ausführung des Programms und Erkennung von Überschreitung der Arraygrenze.
   - Status: OK

1.4.6 Programm zum Testen von Funktionen

1. Check Syntax
   - Erwartetes Ereignis: Syntaxfehler werden korrekt angezeigt.
   - Status: OK
2. Run/Single Step
   - Erwartetes Ereignis: Korrekte Ausführung des Programms.
   - Status: FEHLSCHLAG
     Bei verschachtelten Funktionsaufrufen werden die äußeren Funktionen übersprungen

1.4.7 Programm zum Testen von If-Anweisungen

1. Check Syntax
   - Erwartetes Ereignis: Syntaxfehler werden korrekt angezeigt.
   - Status: OK
2. Run/Single Step
   - Erwartetes Ereignis: Korrekte Ausführung des Programms.
   - Status: OK

1.4.8 Programm zum Testen von While-Schleifen

1. Check Syntax
   - Erwartetes Ereignis: Syntaxfehler werden korrekt angezeigt.
   - Status: OK
2. Run/Single Step
   - Erwartetes Ereignis: Korrekte Ausführung des Programms.
   - Status: OK

1.4.9 Programm zum Testen von Operatoren

1. Check Syntax
   - Erwartetes Ereignis: Syntaxfehler werden korrekt angezeigt.
   - Status: OK
2. Run/Single Step
   - Erwartetes Ereignis: Korrekte Ausführung des Programms.
   - Status: OK

1.4.10 Programm zum Testen von Assertions

1. Check Syntax
   - Erwartetes Ereignis: Syntaxfehler werden korrekt angezeigt.
   - Status: OK
2. Run/Single Step
   - Erwartetes Ereignis: Korrekte Ausführung des Programms und Erkennung von Assertionfailures.
   - Status: OK

1.4.11 Programm zum Testen von Assumptions

1. Check Syntax
   - Erwartetes Ereignis: Syntaxfehler werden korrekt angezeigt.
   - Status: OK
2. Run/Single Step
   - Erwartetes Ereignis: Korrekte Ausführung des Programms und Erkennung von Assumptionfailures.
   - Status: OK

1.4.12 Programm zum Testen von Ensures

1. Check Syntax
   - Erwartetes Ereignis: Syntaxfehler werden korrekt angezeigt.
   - Status: OK
2. Run/Single Step
   - Erwartetes Ereignis: Korrekte Ausführung des Programms und Erkennung von Ensurefailures.
   - Status: OK

1.4.13 Programm zum Testen von Invariants

1. Check Syntax
   - Erwartetes Ereignis: Syntaxfehler werden korrekt angezeigt.
   - Status: OK
2. Run/Single Step
   - Erwartetes Ereignis: Korrekte Ausführung des Programms und Erkennung von Invariantfailures.
   - Status: OK

1.4.14 Programm zum Testen von Axiomen

1. Check Syntax
   - Erwartetes Ereignis: Syntaxfehler werden korrekt angezeigt.
   - Status: FEHLSCHLAG
     Es wird manchmal die ungenaue Fehlermeldung „AST creation not possible!“zurückgegeben
2. Run/Single Step
   - Erwartetes Ereignis: Korrekte Ausführung des Programms, indem die Axiome ignoriert werden.
   - Status: OK