

Pflichtenheft

Beste Gruppe

17. November 2016

Inhaltsverzeichnis

1	Produktübersicht	4
2	Zielbestimmung	5
2.1	Musskriterien	5
2.2	Wunschkriterien	6
2.3	Abgrenzungskriterien	6
3	Produkteinsatz	7
3.1	Anwendungsbereiche	7
3.2	Zielgruppen	7
3.3	Betriebsbedingungen	7
4	Produktumgebung	8
4.1	Software	8
4.2	Hardware	8
4.3	Orgware	8
4.4	Produkt-Schnittstellen	8
5	Funktionale Anforderungen	9
6	Produktdaten	10
6.1	Code-Editor „Rigtime“	10
6.2	Editor „Properties“	10
7	Nichtfunktionale Anforderungen	11
8	Globale Testfälle	12
9	Systemmodelle	13
9.1	Szenarien	13
9.2	Anwendungsfälle	13
9.3	Objektmodelle	13
9.4	Dynamische Modelle	13
10	GUI	14
11	Phasenverantwortliche	15
11.1	Pflichtenheft	15
11.2	Entwurf	15

11.3 Implementierung	15
11.4 Qualitätssicherung	15
11.5 Abschlusspräsentation	15
12 Glossar	16

1 Produktübersicht

Das Entwicklerprogramm „Concrete Rigging of Voting Procedures“ ermöglicht es, Wahlverfahren auf formale Eigenschaften zu prüfen. So kann ein in C definiertes Wahlverfahren wie z.B. die einfache Mehrheitswahl darauf geprüft werden, ob es bestimmte Eigenschaften erfüllt. Der Benutzer kann verschiedene Parameter der Wahl zusätzlich vorgeben.

Da eine vollständige Prüfung auf Wahleigenschaften sehr lange dauern würde, kommt ein Bounded Model Check zum Einsatz. Das verwendete Werkzeug CBMC wird dabei automatisch vom Programm angesteuert.

Der Benutzer bekommt schließlich eine Antwort des Programms, in der er bei Nichterfüllung der Eigenschaft ein Gegenbeispiel angezeigt bekommt. Sollte die Prüfung jedoch erfolgreich sein, bekommt der Nutzer ein Positivbeispiel präsentiert.

2 Zielbestimmung

Das vorliegende Programm überprüft in C geschriebene Wahlverfahren mit Hilfe von CBMC auf formale Eigenschaften. Das Programm akzeptiert optionale Wahlparameter und gibt das Ergebnis der Prüfung zurück.

Die GUI ist in vier Teilen angeordnet:

1. „Rigtime“: Code-Editor für Wahlverfahren in Programmiersprache C.
2. „Properties“: Editor für Spezifikation formaler Eigenschaften in abgespeckter C-Syntax mit speziellen Macros.
3. „Params“: Eingabe von Parametern einer zu analysierenden Wahl mit Anzahl der Wähler, Kandidaten und Sitzen.
4. „Rigplete“: Ausgabe der Prüfung.

2.1 Musskriterien

- Das Programm kann auf aktuellen Versionen von Windows und Linux-Betriebssystemen betrieben werden.
- Alle Abhängigkeiten (u.a. zu CBMC) werden mit dem Programm ausgeliefert.

Die Elemente der GUI sollen folgenden Kriterien genügen:

- „Rigtime“: Code-Editor mit der Möglichkeit zum Speichern, Speichern als und Laden.
- „Properties“: Editor mit der Möglichkeit zum Speichern, Speichern als und Laden. Die Eingabe soll überprüft werden, d.h. ob die Eingabe Macros in C-Syntax darstellt. Im selben Fenster soll es eine Eingabemaske für zusätzliche symbolische Variablen geben.
- „Params“: Option für eine graphische Eingabe der Anzahl von Wählern, Kandidaten und Sitzen.
- „Rigplete“: Das Ergebnis der Wahlanalyse wird angezeigt. Falls CBMC ein Gegenbeispiel zu einer formalen Eigenschaft gefunden hat, soll das Beispiel XXX graphisch XXX angezeigt werden.

2.2 Wunschkriterien

- Das Programm kann auf aktuellen Macs betrieben werden.

Die Elemente der GUI sollen folgenden Kriterien genügen:

- „Rigtime“: Der Code-Editor soll Syntax-Highlighting und eine Fehler-Anzeige für die Programmiersprache C bieten.
- „Properties“: Der Editor soll Syntax-Highlighting und eine Fehler-Anzeige für C bieten. Codeeingaben sollen komplettiert werden können.
- „Params“: Die Analyse der Wahl soll mit Hilfe eines Buttons abgebrochen werden können.
- Wahlergebnisausgabe: Im Fenster XXX „Params“ XXX kann ein Array mit Wahlstimmen eingegeben werden. Das Ergebnis dieser Wahl wird im Fenster „Rigplete“ angezeigt.
- Zusätzlicher SAT-Solver: Das Programm bietet eine Schnittstelle, sodass auch andere SAT-Solver als CBMC angesteuert werden können.
- XXX Weitere? XXX

2.3 Abgrenzungskriterien

XXX

3 Produkteinsatz

Das Programm überprüft Wahlverfahren auf ihre formalen Eigenschaften. Es richtet sich an Kunden, die ein Interesse an der Erforschung oder Entwicklung solcher Verfahren haben. Sie benötigen ein Grundverständnis der Programmiersprache C und Logik.

3.1 Anwendungsbereiche

- Universitärer Bereich
- Forschung

3.2 Zielgruppen

- Wahlforscher
- Softwareentwickler

3.3 Betriebsbedingungen

XXX welche gibt es? XXX

4 Produktumgebung

4.1 Software

- OS: Windows/Linux
- Softwareentwickler

4.2 Hardware

- PC

4.3 Orgware

XXX

4.4 Produkt-Schnittstellen

XXX bei Implementierung von Einsatz anderer SAT-Solver? XXX XXX allgemein: Ausgabe von Produktdaten? XXX

5 Funktionale Anforderungen

6 Produktdaten

6.1 Code-Editor „Rigtime“

/D10/ Das Wahlverfahren ist als Methode „unsigned int voting(params)“ einer C-Headerdatei definiert und wird mit der Endung .h gespeichert.

6.2 Editor „Properties“

/D20/ Die formale Eigenschaft, derer das Wahlverfahren genügen muss ist als C-Datei definiert, die einmal die Methode voting(params) aus einer Headerdatei aufruft, und wird mit der Endung .c gespeichert.

7 Nichtfunktionale Anforderungen

8 Globale Testfälle

9 Systemmodelle

9.1 Szenarien

9.2 Anwendungsfälle

9.3 Objektmodelle

9.4 Dynamische Modelle

10 GUI

11 Phasenverantwortliche

11.1 Pflichtenheft

Justin Hecht

11.2 Entwurf

11.3 Implementierung

11.4 Qualitätssicherung

11.5 Abschlusspräsentation

12 Glossar

Abbildungsverzeichnis