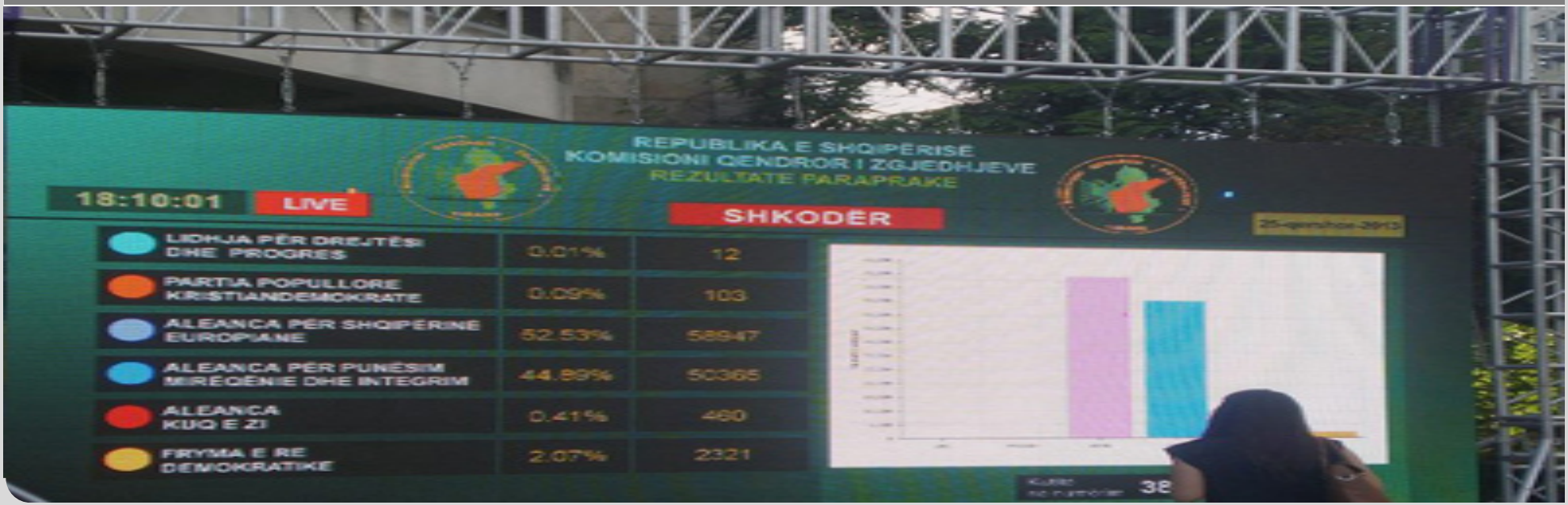
Karlsruher Institut für Technologie

Analyse formaler Eigenschaften von Wahlverfahren

Prof. Bernhard Beckert, Sarah Grebing, Michael Kirsten | Karlsruhe, 8. November 2016

INSTITUT FÜR THEORETISCHE INFORMATIK – ANWENDUNGSORIENTIERTE FORMALE VERIFIKATION



Motivation

Wahlverfahren (nach Wikipedia)

Eine formalisierte Methode, um für eine Wahl festzulegen,

- welche Auswahlmöglichkeit den Wahlberechtigten vorgelegt wird und
- wie aus den gültigen Stimmen zu folgern ist, an welche Kandidaten Ämter zu vergeben sind.

Gütekriterien

Ein Wahlsystem wird danach beurteilt, welchen Zielfunktionen es entgegenkommt. Diese Zielfunktionen können sein:

- Proportionalität bzw. Repräsentation
- Konzentration bzw. Stabilität
- Einfachheit
- ...

Wahlverfahren (nach Wikipedia)

Eine formalisierte Methode, um für eine Wahl festzulegen,

- welche Auswahlmöglichkeit den Wahlberechtigten vorgelegt wird und
- wie aus den gültigen Stimmen zu folgern ist, an welche Kandidaten Ämter zu vergeben sind.

Gütekriterien

Ein Wahlsystem wird danach beurteilt, welchen Zielfunktionen es entgegenkommt. Diese Zielfunktionen können sein:

- Proportionalität bzw. Repräsentation
- Konzentration bzw. Stabilität
- Einfachheit
- ...

Probleme

- Gütekriterien im Allgemeinen widersprüchlich
- Komplexe Wahlverfahren
- Trade-Off zwischen Kriterien nicht-trivial und fehleranfällig

Beispiel in Deutschland

- Bundestagswahlrecht 2008 vom BVerfG für verfassungswidrig erklärt
- Grund: Möglicher Mandatsverlust bei Stimmenzuwachs im Zusammenhang mit Überhangsmandaten
- Verletzt Gleichheit und Unmittelbarkeit der Wahl
- Führte zu Änderung des Wahlrechts 2013: *Ausgleichsmandate*
  $\Rightarrow$ Bei Überhangmandaten Anhebung der Gesamtmandatszahl

Probleme

- Gütekriterien im Allgemeinen widersprüchlich
- Komplexe Wahlverfahren
- Trade-Off zwischen Kriterien nicht-trivial und fehleranfällig

Beispiel in Deutschland

- Bundestagswahlrecht 2008 vom BVerfG für verfassungswidrig erklärt
- Grund: Möglicher Mandatsverlust bei Stimmenzuwachs im Zusammenhang mit Überhangsmandaten
- Verletzt Gleichheit und Unmittelbarkeit der Wahl
- Führte zu Änderung des Wahlrechts 2013: *Ausgleichsmandate*
  $\Rightarrow$ Bei Überhangmandaten Anhebung der Gesamtmandatszahl

Organisatorisches

Betreuer

Sarah Grebing

sarah.grebing@kit.edu – Raum 202

Michael Kirsten

kirsten@kit.edu – Raum 228

Webseite

formal.iti.kit.edu/teaching/pse/201617/voting/

- Termine
- Dokumente

Vorstellung: Wer sind Sie? Vorwissen? Motivation für Thema?

Betreuer

Sarah Grebing

sarah.grebing@kit.edu – Raum 202

Michael Kirsten

kirsten@kit.edu – Raum 228

Webseite

formal.iti.kit.edu/teaching/pse/201617/voting/

- Termine
- Dokumente

Vorstellung: Wer sind Sie? Vorwissen? Motivation für Thema?

Wöchentliche Treffen

- Wann haben Sie Zeit für ein wöchentliches Treffen?
- Welche Klausuren schreiben Sie dieses Semester? Wann?

Ergebnis

- Wochentermin: Mittwoch 11:30 - 13:00 Uhr in Raum 201, Geb. 50.34 (zweiter Stock)
- Abgabe der Artefakte: Jeweils Montag 11:00 Uhr vor dem Kolloquium im SVN-Repository
- Klausurenpause: 13.02.2017 – 26.02.2017
  (allerdings am 15.02.2017 Kolloquium Implementierung)

Zeitplan

Phase	Dauer	Kalenderwoche	Jahr
Einlesen	5 Tage	45 – 45	2016
Pflichtenheft	3 Wochen	46 – 48	2016
Entwurf	4 Wochen	49 – 2	2016/17
vorlesungsfrei	2 Wochen	52 – 1	2016/17
Implementierung	4 Wochen	3 – 6	2017
Klausurpause	2 Wochen	7 – 8	2017
Qualitätssicherung	3 Wochen	9 – 11	2017
Interne Abnahme		12	2017
Abschlusspräsentation		13/14 ?	2017

Zeitplan

Phase	Dauer	Kalenderwoche	Jahr
Einlesen	5 Tage	45 – 45	2016
Pflichtenheft	3 Wochen	46 – 48	2016
Entwurf	4 Wochen	49 – 2	2016/17
vorlesungsfrei	2 Wochen	52 – 1	2016/17
Implementierung	4 Wochen	3 – 6	2017
Klausurpause	2 Wochen	7 – 8	2017
Qualitätssicherung	3 Wochen	9 – 11	2017
Interne Abnahme		12	2017
Abschlusspräsentation		13/14 ?	2017

Phase

Phasenverantwortlicher

- koordiniert die Arbeit
- präsentiert die Ergebnisse im Kolloquium
- Zuteilung liegt bei Ihnen

Kolloquium

- Abschluss einer Phase
- Vortrag vom Phasenverantwortlichen
- Fragenrunde

Abgaben und Bewertung

Abgaben (Artefakte)

- Abgaben mit den geforderten Artefakten stets per SVN (Link schicken wir Ihnen per E-Mail)
- Teil der Prüfungsleistung
- Abgabe ca. 48 Stunden vor Kolloquium (genaue Zeit t.b.a.)
- Abgabe pünktlich, muss eindeutig im SVN erkenntlich sein

Kolloquium

- Teil der Prüfungsleistung (Vortrag und Fragerunde)
- Wichtig: Anwesenheit ist Teil der Prüfungsleistung

Gewichtung

Pflichtenheft 10%, Entwurf 30%, Implementierung 30%, Qualitätssicherung 20%, Abschlusspräsentation 10%

Werkzeuge für Software

Werkzeug-gestützte SW-Entwicklung

Verwendung von

- Repository (GIT oder SVN)
- Issues Tracker
- Continuous Integration
- Build-Skripte (Maven, Ant, ...)

Repositories angeboten von:

- ATIS (SVN)
- Github, Bitbucket, etc.
- SCC (`https://git.scc.kit.edu`)

Den Betreuern ist Zugriff zu gewähren. Überprüfung von Commits.

1. Phase: Pflichtenheft

Pflichtenheft

Artefakt: Pflichtenheft im Umfang: ca. 20 Seiten

Inhalte:

- Systemmodell und -umgebung
- Zielbestimmungen (Muss-, Wunsch- und Abgrenzungskriterien)
- Vollständige funktionale Anforderungen; Qualitätsanforderungen
- GUI-Entwürfe
- Ausführliche Testfallszenarien
- Phasenverantwortliche

Anforderungsbeschreibung

Möglicher Anwendungsfall

- Die Anwenderin (z.B. Entwicklerin eines Wahlverfahrens) öffnet das Programm und lädt ein Wahlverfahren (als C Programm) sowie eine formale Eigenschaft.
- Sie gibt an, für wie viele Wähler und Kandidaten, sie das Wahlverfahren prüfen möchte.
- Nach dem Betätigen des Verifikations-Knopfes zeigt das Programm an, dass die Eigenschaft nicht gilt sowie ein Gegenbeispiel.
- Die Anwenderin passt die Eigenschaft/Wahlverfahren in den jeweiligen Editor-Fenstern an.
- Ein erneutes Betätigen des Verifikations-Knopfes zeigt an, dass die Eigenschaft nun erfüllt wird.
- Anschließend lädt die Anwenderin eine weitere Eigenschaft und editiert, bis die von ihr gewünschten Eigenschaften erfüllt werden.
- Nun speichert sie Wahlverfahren und Eigenschaften, um sie später wiederzuverwenden.

Möglicher Anwendungsfall

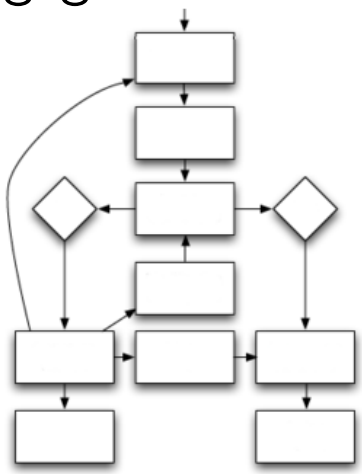
- Die Anwenderin (z.B. Entwicklerin eines Wahlverfahrens) öffnet das Programm und lädt ein Wahlverfahren (als C Programm) sowie eine formale Eigenschaft.
- Sie gibt an, für wie viele Wähler und Kandidaten, sie das Wahlverfahren prüfen möchte.
- Nach dem Betätigen des Verifikations-Knopfes zeigt das Programm an, dass die Eigenschaft nicht gilt sowie ein Gegenbeispiel.
- Die Anwenderin passt die Eigenschaft/Wahlverfahren in den jeweiligen Editor-Fenstern an.
- Ein erneutes Betätigen des Verifikations-Knopfes zeigt an, dass die Eigenschaft nun erfüllt wird.
- Anschließend lädt die Anwenderin eine weitere Eigenschaft und editiert, bis die von ihr gewünschten Eigenschaften erfüllt werden.
- Nun speichert sie Wahlverfahren und Eigenschaften, um sie später wiederzuverwenden.

Aufgabe

Werkzeug zur Analyse formaler Eigenschaften von Wahlverfahren

Wahlverfahren *W*

Abgegebene Stimmen



Wahlergebnis

Formale Eigenschaft *P*

$\forall x, y. \exists z \ldots$

Erfüllt *W* die Eigenschaft *P*?

Anforderungen

Globale Parameter:

- Anzahl von Kandidaten `C`
- Anzahl von Wählern `V`
- Anzahl von zu besetzenden Sitzen `S`
- Optional: Intervalle von Kandidaten/Wählern/Sitzen

GUI-Elemente und Anbindung:

- Editor für Wahlverfahren
- Editor für formale Eigenschaft
- Verwendung von formalem Analysewerkzeug CBMC als Backend
- Ausgabefenster für Analyse-Ergebnis:
  Erfolgsnachricht oder Gegenbeispiel

GUI-Element I

1.) Editor für Wahlverfahren in C-Syntax

Mögliche Formen der Stimmabgabe:

- Single-Choice: Wähler stimmt jeweils für einen Kandidaten
- Zustimmungswahl: Wähler bewertet jeden Kandidaten mit Ja/Nein
- Gewichtete Wahl: Wähler bewertet jeden Kandidaten mit Zahlenwert
- Präferenzwahl: Wähler legt Ranking für alle Kandidaten fest

Mögliche Formen der Ausgabe:

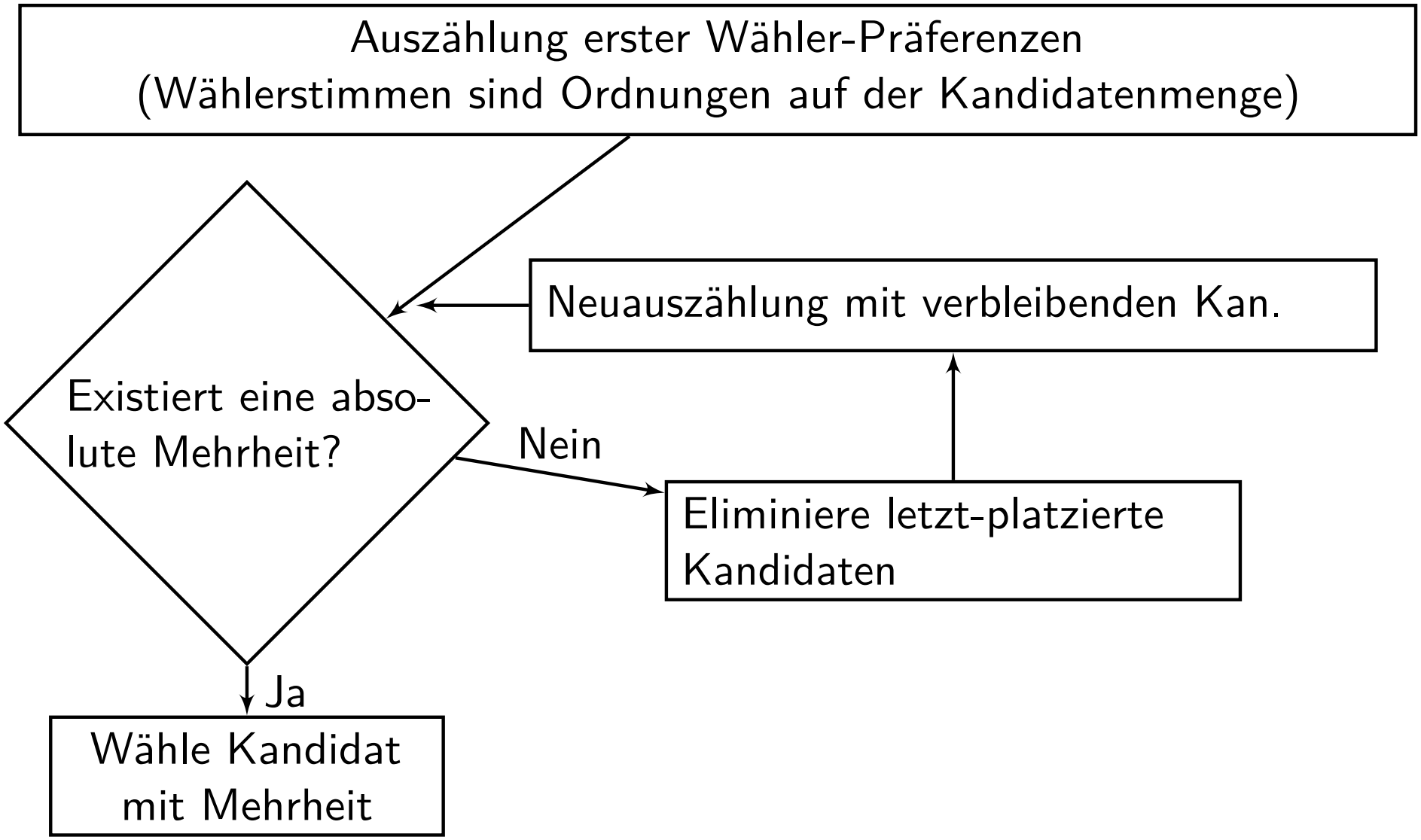
- Ein gewählter Kandidat oder unentschieden
- Anzahl zugeteilter Sitze pro Partei

- Übliche Editieroperationen, laden und speichern, Anzeige von Fehlermeldungen (z.B. nicht-deklarierte Variablen)
- Optional: Syntaxhighlighting

Beispiel I: Einfache Mehrheitswahl

```
int voting(int votes[V]) {
    int res[C + 1] = {0}, elect = 0, max = 0;
    for (int i = 0; i < V; i++)
        res[votes[i]] += 1;
    for (int i = 1; i <= C; i++)
        if (max < res[i]) {
            max = res[i];
            elect = i;
        } else if (max == res[i])
            elect = 0;
    return elect;
}
```

Beispiel II: Rangfolgewahl (Instant-Runoff Voting)



GUI-Element II

2.) Editor für Eigenschaft als Vor- und Nachbedingung:

Arten von Eigenschaften

- Funktional, z.B. Kandidat mit mehr als 50% Stimmen gewinnt
- Relational, z.B. für andere Wähler-Reihenfolge bleibt Ergebnis gleich

Elemente von Eigenschaften

- Vorbedingung (z.B. wenn die Stimmen bestimmter Art sind . . .)
- Nachbedingung (. . . dann gilt für das Ergebnis *xy*)
- Symbolische Variablen (z.B. beliebiger Kandidat mit mehr als 50%)

- Eingabe in abgespeckter C-Syntax (bspw. keine Schleifen)
- Macro-Verwendung: `FORALL_VOTERS[i]`, `FORALL_CANDIDATES[i]`, `VOTE_SUM_FOR_CANDIDATE(c)`, `==>`, `<==>`

Beispiel-Eigenschaften

Absolute Mehrheit (funktional):

```
Symbolische Kandidaten: c
Vorbedingung: (V / 2) < VOTE_SUM_FOR_CANDIDATE(c);
Nachbedingung: ELECT == c;
```

Anonymität (relational):

```
Symbolische Wähler: v, w
Vorbedingung:
FORALL_VOTERS[i]
 ((i != v && i != w) ==> VOTES1(i) = VOTES2(i));
VOTES1(v) = VOTES2(w);
VOTES1(w) = VOTES2(v);
Nachbedingung: ELECT1 = ELECT2
```

GUI-Element III

3.) Ausgabefenster mit Analyse-Ergebnis von CBMC

Entweder:

- Erfolgreiche Analyse von EIGENSCHAFT für WAHLVERFAHREN mit x Kandidaten, y Wählern, z Sitzen.

oder

- a. Analyse von EIGENSCHAFT für WAHLVERFAHREN mit x Kandidaten, y Wählern, z Sitzen fehlgeschlagen.
  b. Gegenbeispiel (Ein- und Ausgabe(n), für die Eigenschaft nicht gilt) $\rightarrow$ muss aufbereitet werden (Beispiel auf nächster Folie)

GUI-Element IIII

4.) Parameter-Angabe für automatische Analyse:

- Anzahl von Kandidaten (in Wahlverfahren/Eigenschaft Konstante `C`)
- Anzahl von Wählern (in Wahlverfahren/Eigenschaft Konstante `V`)
- Anzahl von Sitzen (in Wahlverfahren/Eigenschaft Konstante `S`)
- optional: **Intervalle** von Kandidaten/Wählern/Sitzen (im Hintergrund Batch-Aufruf von Analysewerkzeug)

Beispielausgabe Analyse-Backend

```
Results:
[p.assertion.1] assertion elect == 0u: FAILURE
Trace p.assertion.1:
State 32 file FILE.c line 9 function main thread 0
------------------------------------------------
  VOTES[0l]=2u (00000000000000000000000000000010)
State 41 file FILE.c line 9 function main thread 0
------------------------------------------------
  VOTES[1l]=2u (00000000000000000000000000000010)
State 50 file FILE.c line 9 function main thread 0
------------------------------------------------
  VOTES[2l]=3u (00000000000000000000000000000011)
...
State 265 file FILE.h line 38 function f thread 0
------------------------------------------------
  ELECT=3u (00000000000000000000000000000011)
```

Beispielausgabe Analyse-Backend

```
Results:
[p.assertion.1] assertion elect == 0u: FAILURE
Trace p.assertion.1:
State 32 file FILE.c line 9 function main thread 0
--------------------------------------------------
  VOTES[0l]=2u (00000000000000000000000000000010)
State 41 file FILE.c line 9 function main thread 0
--------------------------------------------------
  VOTES[1l]=2u (00000000000000000000000000000010)
State 50 file FILE.c line 9 function main thread 0
--------------------------------------------------
  VOTES[2l]=3u (00000000000000000000000000000011)
...
State 265 file FILE.h line 38 function f thread 0
--------------------------------------------------
  ELECT=3u (00000000000000000000000000000011)
```

Wie geht es weiter

- Austausch der Kontaktinformationen
- Anlegen des Repository
- Festlegung der Phasenverantwortlichen
- Installieren von CBMC (`http://www.cprover.org/cbmc/`)
- Einlesen in Wahlverfahren, Java GUI Programmierung
- Termin: Einführung in Logik + Bounded Model Checking

Bis zum nächsten Treffen!

Fragen und Antworten

Vielen Dank

für die Aufmerksamkeit!

Gibt es Fragen?

Fragen und Antworten

Vielen Dank

für die Aufmerksamkeit!

Gibt es Fragen?