

## Abschlussbericht der Gruppe „Model Checking“

- Aufgabenaufteilung bezüglich der Aufgabenstellung:
  1. Vorbereitung
  2. Möglichkeit für die Eingabe von  $\mu$ -Kalkül-Formeln
  3. Speichern der eingegebenen Formeln
  4. Laden der gespeicherten Formeln
  5. Parsen und Überprüfen der Syntax der eingegebenen Formel
  6. Überführung der Formel in den alternierenden Baumautomat
  7. Berechnung der  $\Theta$  - Funktion
  8. Erstellen von einem Spielergraphen aus dem Baumautomaten und dem nu-Automaten
  9. Implementierung vom Paritätsspiel auf dem Spielergraphen
  10. Check auf Erfüllbarkeit der eingegebenen  $\mu$ -Kalkül-Formel
  
- Erläuterungen zu den einzelnen Aufgaben:
  1. Mit Vorbereitung sind Einarbeitung und Verständnis der Aufgaben gemeint.
  2. Für die Eingabe von  $\mu$ -Kalkül-Formeln steht dem Benutzer eine graphische Oberfläche (ModelCheckingGui) zur Verfügung. Mit dem Klick auf den „check“-Button wird die Formel erst syntaktisch überprüft, dann in den alternierenden Baumautomat überführt und mit dem nu-Automaten „verglichen“.  
Im Ausgabefenster erscheinen eventuelle Fehlermeldungen bezüglich der Syntax der Formel oder der Baumautomat in textueller Form mit der zugehöriger  $\Theta$  - Funktion zu jedem Knoten.
  3. Der Benutzer kann die eingegebene Formel in eine Datei speichern.
  4. Und er kann eine gespeicherte Formel laden.
  5. Das Überprüfen der Syntax der eingegebenen Formel geschieht nach der gegebenen Grammatik des modalen  $\mu$ -Kalküls. Dabei wird darauf geachtet, dass keine freien Variablen vorkommen dürfen und dass eine Variable nur einmal gebunden vorkommen darf.
  6. Beim Überführen der Formel in den alternierenden Baumautomat haben wir uns an die im Paper angegebene Beschreibung gehalten. Für den Benutzer steht eine Hilfe mit Beschreibung und Erklärung zur Verfügung.
  7. Die  $\Theta$  - Funktion (acceptance condition) ist wichtig für das Paritätsspiel. Die Berechnung der Funktion erfolgt wie im Paper beschrieben.
  8. Zuerst wird aus dem Baumautomaten ein Zustandsarray erstellt und die Verknüpfungen untereinander hergestellt. Dieses wird dann zusammen mit den  $\nu$ -Automaten-Zuständen im nächsten Schritt zu dem Spielegraphen verlinkt.
  9. Es wurde der gegebene Algorithmus implementiert, der zu einem gegebenen Spielegraphen die dazugehörigen Gewinnmengen für Spieler 0 und Spieler 1 berechnet.
  10. Der Satisfaction-Check dient zur Überprüfung, ob der  $\nu$ -Automat die  $\mu$ -Kalkül-Formel erfüllt. Sofern dies der Fall ist, wird 1 zurückgegeben. Wird die Formel nicht erfüllt, so wird überprüft ob die duale Formel erfüllt wird. Ist dies der Fall, so wird 0 zurückgegeben ansonsten -1.

- Personen und Zuständigkeitsbereiche:

- Marina Tropmann: zuständig für die Punkte 2 bis 7. (4SWS)
- Stephan Knauer: zuständig für die Punkte 8 bis 10. (4SWS)

- Erledigte Aufgaben und Schwierigkeiten:

- Alle Aufgaben wurden während des Praktikums erledigt. Die Schwierigkeiten bereitete dabei Mangel an Informationen. Wir haben sehr viel Zeit in das Verständnis des  $\mu$ -Kalküls bzw. des Erfüllbarkeitschecks investiert.
- Wir haben mit der Bearbeitung der Aufgaben am Anfang des Semesters begonnen.
- Ein testfähiges Stadium haben wir Anfang Januar erreicht.
- Unsere Empfehlung für das nächste Mal wäre, mehr Informationen bezüglich einzelner Aufgaben des Model Checking zur Verfügung zu stellen.

## Zeitaufwand & Probleme – Model Checking (Stephan Knauer)

---

### Aufgabenbereich:

### Implementierung des Satisfaction-Check. Dafür waren folgende Schritte notwendig:

1. Erstellen eines Spielegraphen aus  $v$ -Automat und  $\mu$ -Kalkül-Formel.
2. Implementierung des Parity-Game-Algorithmus (angepasst an den vorher entwickelten Spielegraphen)
3. Implementierung des Satisfaction-Check.

### Vorgehensweise:

Hineindenken in die gestellte Aufgabe, sowie sammeln und sichten von Material.

- *Beginn: 17.11.2006*
- *Ende: 22.12.2006*
- *Arbeitsaufwand: nicht abschätzbar, da immer wieder was nachrecherchiert werden musste*

Vorüberlegungen zum geschickten Vorgehen bei der Lösung der Aufgabe sowie das Ausarbeiten erster Ansätze, den Spielegraphen darzustellen.

- *Beginn: 27.12.2006*
- *Ende: 29.12.2006*
- *Arbeitsaufwand: 8 Std.*

Konkrete Umsetzung der erarbeiteten Ideen in Java. Implementieren des Spielegraphen (Schritt 1)

- *Beginn: 03.01.2007*
- *Ende: 21.01.2007*
- *Arbeitsaufwand: 60 Std.*

Implementierung des Parity-Game-Algorithmus zur Bestimmung der Gewinnbereiche (Schritt 2)

- *Beginn: 22.01.2007*
- *Ende: 31.01.2007*
- *Arbeitsaufwand: 20 Std.*

Abschließen der Aufgabe durch Implementierung des Satisfaction-Check (Schritt 3)

- *Beginn: 30.01.2007*
- *Ende: 31.01.2007*
- *Arbeitsaufwand: 8 Std.*

### Aufgetretene Probleme:

- Verwerfen des ersten Ansatzes zur Implementierung des Spielegraphen nach Gespräch mit Harald (2 Tage Arbeit umsonst)
- Die Umsetzung des Spielegraphen erwies sich viel aufwendiger als erwartet. Nachträgliche Neuverlinkung der Knoten erwies sich als Denksportaufgabe. Fehler-Debugging war sehr Zeit- und Nervenaufwendig.
- Generelle Schwierigkeiten, mein konkretes Problem zu erkennen und zu verstehen. Speziell auch den Algorithmus nachzuvollziehen.
- Entscheiden ob Automat die Formel erfüllt oder nicht, durch Reduktion des Problems auf ein Paritätsspiel, war schwieriger als gedacht.