

e-Learning Projekt 2009.121

Finite Automata Tool

Abschlussbericht

René Thiemann

20. August 2009

In dem Projekt “Finite Automata Tool” wurde ein Tool zum Umgang mit endlichen Automaten entwickelt. Hierbei wurden folgende initiale Ziele umgesetzt.

1. Das Tool soll auch für Anfänger intuitiv bedienbar sein.
2. Die Algorithmen auf endlichen Automaten sollen visualisiert werden.

Durch diese Zielsetzung gibt es einen großen Mehrwert für alle Beteiligten von Lehrveranstaltungen, in denen endliche Automaten eingesetzt werden.

- A. *Studierende* können durch das Tool die Inhalte der Vorlesungen vertieften und die implementierten Algorithmen deutlich besser nachvollziehen. Ohne lange Einarbeitungszeit können sie mit Hilfe des Tools jeden einzelnen Schritt eines Algorithmus verfolgen. Alternativ können Sie einen Algorithmus aber auch im Automatik Modus ablaufen lassen, um nur das Endergebnis zu sehen. Somit können selbst erstellte Lösungen schnell auf Korrektheit überprüft werden, ohne sich vorher jeden Schritt ansehen zu müssen.

Dieser Mehrwert ergibt sich aus der Umsetzung der Ziele 1 und 2.

- B. *Lehrende* erhalten zwei Hilfen durch das Tool. Zum einen können sie Algorithmen durch die Visualisierung einfach veranschaulichen, wie es sonst nur mit der Erstellung aufwändiger Overlay-Folien möglich wäre. Zum anderen können sie komfortabel den Aufwand von Übungs- und Klausuraufgaben anpassen, indem sie in der Visualisierung z.B. verfolgen können, wie viele Schritte ein Algorithmus auf gegebener Eingabe benötigt.

Dieser Mehrwert ergibt sich aus der Umsetzung des Ziels 2.¹

¹Im Projektantrag wurde für diesen Mehrwert noch ein drittes Ziel erwähnt: Die effiziente Berechnung der Algorithmen bei deaktivierter Visualisierung. Es hat sich jedoch gezeigt, dass die Verfahren auch mit aktiverter Visualisierung (im Automatik Modus auf höchster Geschwindigkeit) schnell genug sind. Insofern ist die Visualisierung immer aktiviert.

Das Finite Automata Tool wurde im Projektzeitraum von Stefan Kopeinig und mir entwickelt und ist unter

<http://cl-informatik.uibk.ac.at/software/fat>

frei verfügbar. Zur Ausführung benötigt man lediglich ein Rechner mit installiertem Java Runtime Environment, eine Software die für jedes Betriebssystem frei erhältlich ist.

Im folgenden wird erklärt, warum mit der finalen Version des Finite Automata Tools alle Ziele der Projektvereinbarung erfüllt worden sind.

Ziel 1: Einfache Bedienbarkeit

Zur Erstellung von Automaten wurde eine einfache Grammatik entworfen. So mit können Automaten in textueller Form aus Dateien geladen und gespeichert werden, oder aber auch im Editor entworfen bzw. bearbeitet werden. Beispiele von Automaten befinden sich sowohl in der Hilfe, sie sind aber auch direkt im Editor Fenster bei der Erstellung neuer Automaten verfügbar.

Einmal geladene oder neu erstellte Automaten befinden sich sofort rechts im Navigationsfenster. Ein Doppelklick auf einen Automaten stellt diesen graphisch im Präsentationsfenster dar.

Des weiteren kann man im Navigationsfenster ein oder zwei Automaten auswählen und dann den gewünschten Algorithmus aufrufen. Um die Bedienung zu vereinfachen, werden immer nur diejenigen Algorithmen zur Auswahl gestellt, die zu den ausgewählten Automaten passen. Es ist zum Beispiel nicht möglich, einen nicht-deterministischen Automaten zu minimieren, da dieser Algorithmus nur für deterministische Automaten funktioniert.

Sobald man einen Algorithmus gestartet hat, wird das Navigationsfenster ausgeblendet, um eine möglichst große Fläche zur Visualisierung zu haben. Dort sieht man dann im linken Teil des Präsentationsfensters die Eingabe-Automaten in graphischer Darstellung. Im rechten Teil wird dann der neue Automat Schritt für Schritt erstellt. Im unteren Bereich befindet sich eine Kontrollleiste und ein Textfenster, in dem die einzelnen Schritte des Algorithmus erklärt werden. Mithilfe der Kontrollleiste hat man die Möglichkeit, die Abarbeitung des Algorithmus zu steuern. Mit den drei bekannten Buttons *Play*, *Pause* und *One-Step* kann man zwischen Einzel-Schritt und Automatik Modus wechseln, sowie einen weiteren Schritt durchführen. Für den Automatik Modus gibt es zudem noch einen Geschwindigkeitsregler.

Nach Ablauf des Algorithmus wird die Kontrollleiste ausgeblendet und durch einen *Exit*-Button ersetzt. Nach dessen Betätigung wird die Navigationsleiste wieder eingeblendet und der neu erstellte Automat ist neu hinzugefügt worden.

Verschiedene Test Benutzer² hatten keinerlei Probleme bei der Bedienung des Tools. Zudem gibt es ein Hilfe System, in dem ausführlich die Benutzung des Tools erklärt wird. Insofern ist das Ziel 1 – einfache Bedienbarkeit – vom Finite Automata Tool erfüllt.

²Mitarbeiter der “Computational Logic” Gruppe

Ziel 2: Visualisierung von Algorithmen

Zur Visualisierung wurde die freie Software Touchgraph verwendet, die Graphen zeichnet. Da die vorhandene Funktionalität von Touchgraph jedoch nicht ausreicht, um endliche Automaten im gewohnten Format darzustellen, wie es in Textbüchern genutzt wird, wurde Touchgraph um mehrere Darstellungsoptionen erweitert. Zudem wurde ein Autofokus entwickelt, der die Darstellung der endlichen Automaten immer so skaliert und verschiebt, dass sie in das Fenster passen.

Insgesamt wurden die folgenden Algorithmen implementiert und visualisiert:

- *Schnitt* und *Vereinigung* zweier beliebiger Automaten
- *Komplement* und *Minimierung* von deterministischen endlichen Automaten
- *Determinisierung* von nicht-deterministischen endlichen Automaten

Hierbei wurden mache Algorithmen sogar in verschiedenen Varianten implementiert, um die vor und Nachteile verschiedener Varianten besser verdeutlichen zu können.

Alle Algorithmen wurden mit dem Ziel visualisiert, den Ablauf verständlich zu machen. Deshalb wird jeder einzelne Schritt eines Algorithmus im Textfenster erläutert. Zudem werden die beteiligten Teile eines Automaten farblich hervorgehoben. Ein exemplarischer Ablauf eines Schrittes in der Visualisierung läuft wie folgt ab:

- Im Textfenster wird der vollständige Schritt beschrieben und farblich hervorgehoben. Die relevanten Teile der Eingabe Automaten werden farblich hervorgehoben.
- Neue Teile des Ausgabe Automaten werden erzeugt und farblich hervorgehoben.
- Alle Hervorhebungen werden zurückgenommen um so den Abschluss eines Schrittes zu markieren und den nächsten Schritt einzuleiten.

In dieser Weise kann jeder Schritt leicht nachvollzogen werden. Um zur jetzigen Version zu gelangen, möchte ich mich ausdrücklich für das gute Feedback des eLearning-Teams und meiner Kollegen bedanken, das zur Qualität der Visualisierung beigetragen hat.

Da alle Algorithmen in einer optisch ansprechenden Form visualisiert sind (keiner der Test Benutzer der finalen Version hatte noch Verbesserungsvorschläge), ist auch Ziel 2 – die Visualisierung der Algorithmen – vollständig umgesetzt.

Tätigkeitsbericht

Es wurde ein Werkvertrag über 140 Stunden mit Stefan Kopeinig abgeschlossen, dem maßgeblichen Entwickler des Finite Automata Tools.

Insgesamt hat Herr Kopeinig 154 Stunden an der Entwicklung des Tools gearbeitet, wobei sich diese Zeit wie folgt aufteilt. Zuerst wurden die Datenstrukturen, Hilfsfunktionen und die Grammatik für endliche Automaten entworfen und diskutiert (40 Stunden), bis sie in ihren jetzigen Form die einfache Eingabe von endlichen Automaten ermöglicht. Danach wurde die graphische Oberfläche mit der Anbindung an die Touchgraph Bibliothek realisiert (49 Stunden). Darauf aufbauend wurde einen Rahmenwerk für die Visualisierung von Algorithmen geschaffen (30 Stunden), welches den bereits erwähnten Einzel-Schritt und Automatik Modus bietet. Mithilfe dieses Rahmenwerks wurden dann alle Algorithmen implementiert (25 Stunden). Abschliessend wurde die Bedienungsanleitung geschrieben, die Webseite aufgesetzt und die Touchgraph Bibliothek um die neuen Darstellungstypen erweitert (10 Stunden).

Der etwas überschrittene Zeitaufwand erklärt sich vor allem durch die Unterschätzung des Aufwands in der Erstellung der graphischen Oberfläche.

Einsatz in der Lehre

Prinzipiell ist das Finite Automata Tool in allen Vorlesungen einsetzbar, in denen endlichen Automaten behandelt werden. Beim jetzigen Curriculum betrifft dies die Vorlesungen:

- Diskrete Mathematik (Bachelor Informatik und Mathematik, Dr. Moser)
- Formale Sprachen und Automatentheorie (Bachelor Informatik, Prof. Middeldorp)
- Einführung in Model Checking (Master Informatik, Dr. Thiemann)
- Model Checking (Master Informatik, Dr. Thiemann)

Und in der Tat werden neben mir auch beide Kollegen Moser und Middeldorp das Finite Automata Tool für Ihre Vorlesungen einzusetzen. Beide hatten sich bereits einen Prototyp des Tools angesehen. Ihre Kommentare und Wünsche konnten bis auf einen vollständig in die finale Version integriert werden, wie etwa der Autofokus und die verbesserte Darstellung der Automaten.

Mit der finalen Version sind beide Kollegen gut zufrieden, es fehlt ihnen lediglich eine weitere Klasse von endlichen Automaten, nämlich nicht-deterministische Automaten mit ε -Transitionen. Um diese jedoch vollständig zu integrieren, reichte das Budget bzw. die Zeit nicht mehr aus.

Leider kann zu diesem Zeitpunkt keine Stellungnahme über die Akzeptanz des Tools bei Studenten gegeben werden, da die erwähnten Vorlesungen erst noch der Abgabefrist dieses Berichts anfangen.