

ALAN
Eine Turing Maschine in Java

Neuber, Pahl, Seichter

14.06.2005

Zusammenfassung

Projektarbeit in Programmieren 2 im Sommersemester 2005 bei
Professor Dr. Roland Feindor

Inhaltsverzeichnis

1	Einführung	3
2	Beschreibung der Turing Maschine	3
2.1	Die Übergangsfunktion δ	4
3	Bedienung von ALAN	4
3.1	Hauptfenster	4
3.1.1	Initialisieren der Turing Maschine	5
3.1.2	Eingeben der Übergangsfunktion	5
3.2	Ausführen des Programms	6
3.3	Dokumentation des Programms	6
3.4	Menüs	6
4	Implementierung	7
4.1	Verwendete Werkzeuge	7
4.1.1	Subversion	7
4.1.2	Umbrello	7
4.2	Klassen	7
4.2.1	AboutDialog	8
4.2.2	Alan	8
4.2.3	ContentDialog	8
4.2.4	HtmlEditor	8
4.2.5	Tape	9
4.2.6	TapeGUI	9
4.2.7	TuringMachine	9
4.2.8	TransistionFunctionElement	9
4.3	Testtreiber	9

1 Einführung

ALAN ist eine deterministische Turing Maschine in Java. Wir haben das Programm nach dem Erfinder der Turing Maschine, Alan Mathison Turing, schlicht ALAN genannt.

ALAN hat eine leicht zu bedienende, Grafische Oberfläche (GUI), die wir mit Swing realisiert haben. Daher ist ALAN gut geeignet, um durch selbständiges ausprobieren Turing Maschinen besser zu verstehen. Man kann eigene Programme schreiben und ausprobieren oder eines aus der großen Anzahl der mit gelieferten Beispiele wählen. Die grafische Repräsentation des Bandes, sowie die Möglichkeit zur schrittweisen Abarbeitung des Programmes machen es einfach die Funktionsweise einer Turing Maschine zu sehen.

Die Turing Maschine an sich wird im nächsten Kapitel kurz beschrieben. Daraufhin wird diese Dokumentation die Bedienung von ALAN erläutern und danach zeigen wie wir bei der Entwicklung vorgegangen sind und welche Werkzeuge wir eingesetzt haben.

2 Beschreibung der Turing Maschine



Abbildung: Alan Turing

Die Turing Maschine wurde 1936 von Alan Turing vorgestellt. Sie ist ein theoretisches Modell, dass alle Berechnungen ausführen kann die auch mit einem fest vorgegebenen Programm auf einem Computer ausgeführt werden können.

Diese Beschreibung ist äußerst knapp, daher sollte noch zusätzlich auf ein Fachbuch zu diesem Thema zurück gegriffen werden.

Eine Turing Maschine wird durch folgende Formel definiert:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

- Q : Die endliche Menge der Zustände der endlichen Steuerung.
- Σ : Die endliche Menge der Eingabesymbole
- Γ : Die vollständige Menge der Bandsymbole $\Sigma \subseteq \Gamma$
- δ : Die Übergangsfunktion (siehe unten)
- q_0 : Der Startzustand in dem sich die endliche Steuerung am Anfang befindet $q_0 \in Q$
- B : Das Leerzeichen (Blank) $B \in \Gamma, B \notin \Sigma$

- F : Die Menge der Endzustände oder akzeptierten Zustände $F \subseteq Q$

2.1 Die Übergangsfunktion δ

Der Wert der Übergangsfunktion $\delta(q, X)$ ist ein Tripel (p, Y, D) . Wobei q der aktuelle Bandzustand und X ein Bandsymbol ist. Für das Tripel gilt:

- p : Ist der nächste Zustand in Q
- Y : Ist das Symbol aus Γ , dass an die gelesene Stelle geschrieben wird, $Y \in \Gamma$
- D : Ist die Richtung (Links oder Rechts) in die sich der Schreib-/Lesekopf bewegt

3 Bedienung von ALAN

3.1 Hauptfenster

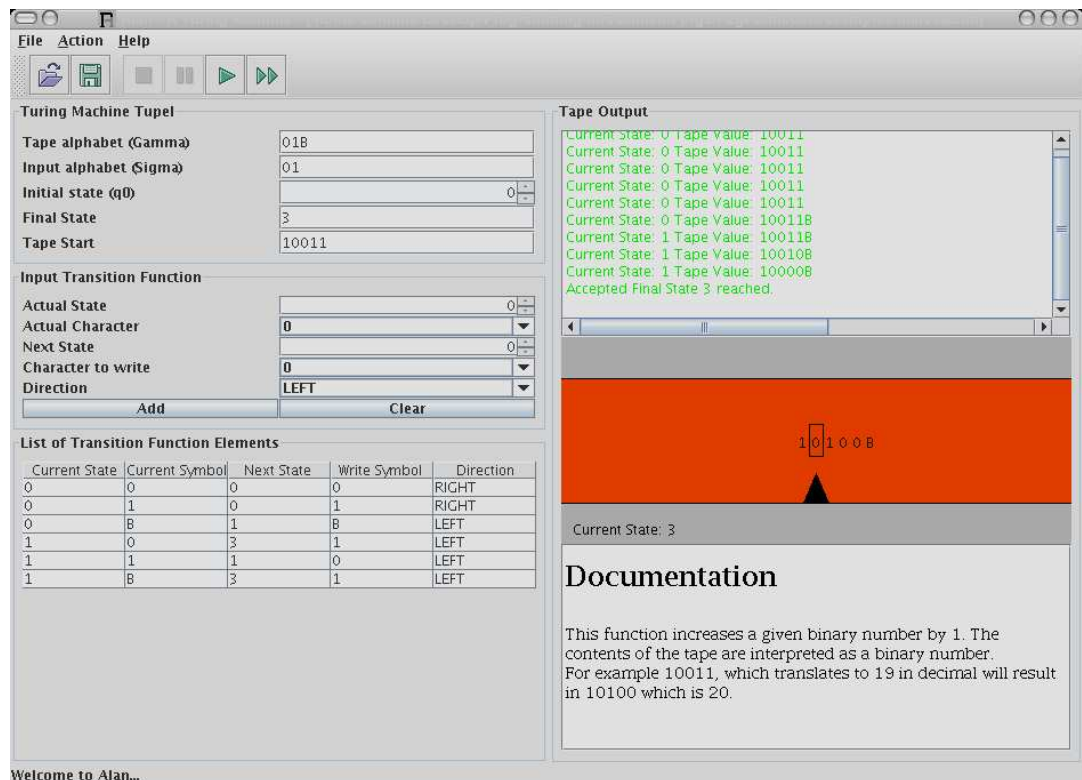


Abbildung: Hauptfenster von ALAN

Das Hauptfenster von ALAN, welches nach dem Start sichtbar ist, kann in zwei Bereiche unterteilt werden.

- Im linken Bereich werden die Werte der Turing Maschine gesetzt.

- Im rechten Bereich erfolgt die Ausgabe während der Abarbeitung des Programms. Ausserdem gibt es hier noch einen Bereich, der für die Dokumentation des Programms der Turing Maschine vorgesehen ist.

3.1.1 Initialisieren der Turing Maschine

Zuallererst sollte man sich überlegen, was als Eingabe- bzw. Bandalphabet verwendet werden soll. Diese können dann in die entsprechenden Textfelder für Sigma und Gamma ohne Trennzeichen eingegeben werden. Für das Bandalphabet mit den Zeichen 0,1,B,X und Y geben Sie bitte "01BXY" in das Feld für Gamma ein. Das Blank-Symbol "B" sollte in dieser Menge enthalten sein.

Danach kann der Startzustand festgelegt werden. Der Standardwert "0" kann meisst unverändert übernommen werden.

Jetzt können die Endzustände angegeben werden. Wenn mehr als ein Endzustand vorhanden ist, müssen diese als eine mit Komma getrennte Liste in das Feld "Final State" eingetragen werden. Zum Beispiel "3,4,5".

Im Textfeld "Tape Start" kann der Anfangswert des Bandes gesetzt werden. Er darf nur aus Zeichen des Eingabe Alphabets bestehen.

3.1.2 Eingeben der Übergangsfunktion

List of Transition Function Elements				
Current State	Current Symbol	Next State	Write Symbol	Direction
0	0	1	X	RIGHT
0	Y	3	Y	RIGHT
1	0	1	0	RIGHT
1	1	2	Y	LEFT
1	Y	1	Y	RIGHT
2	0	2	0	LEFT
2	X	0	X	RIGHT
2	Y	2	Y	LEFT
3	B	4	B	RIGHT
3	Y	3	Y	RIGHT

Abbildung: Übergangsfunktion für eine Turing Maschine die 0^n1^n akzeptiert

Die Übergangsfunktion wird durch fünf Werte bestimmt. Zuerst wird der aktuelle Zustand eingestellt, dann kann man bestimmen was beim Lesen eines Zeichens "Actual Character" passieren soll. Das heisst in welchem Zustand danach gewechselt werden soll, welches Zeichen an die aktuelle Bandposition geschrieben werden soll und in welche Richtung sich das Band weiter bewegen soll. Durch einen Klick auf "Add" wird diese Funktion der Tabelle der Übergangsfunktionen hinzugefügt.

Einzelne Übergangsfunktionen können aus der Tabelle gelöscht werden indem man die betreffende Zeile markiert und dann im Kontextmenü, dass man durch einen Rechtsklick erreicht, "Delete" auswählt.

3.2 Ausführen des Programms



Abbildung: Stop, Pause, Start und Schrittweise Weiter

Die Ausführung des Programms wird am einfachsten über den Schalter "Start" in der Toolbar beziehungsweise über das Menü "Actions" und dann "Start" gestartet. Alternativ kann die Ausführung auch Schritt für Schritt vorgenommen werden. Dazu muss der Schalter "Step" betätigt werden, um die Turing Maschine den nächsten Schritt des Programms ausführen zu lassen. Das Programm kann durch drücken von "Start" auch normal fortgesetzt werden.

3.3 Dokumentation des Programms

Da ALAN auch in der Lehre benutzt werden soll, ist es wichtig das Programme, die als Beispiel dienen, gut dokumentiert sind. Aus diesem Grund gibt es in ALAN einen extra Bereich in dem eine kurze Erklärung zum aktuell geladenen Programm angezeigt wird. Diese Dokumentation kann mit Hilfe von HTML-Tags formatiert werden. Durch den Menüeintrag "Edit Description" im "File"-Menü können Sie einen Dialog öffnen, der die Formatierung ermöglicht.

3.4 Menüs

Im folgenden wird jeder Menüpunkt von ALAN kurz dokumentiert.

- File

New Erstellt eine neue leere Turing Maschine ohne Werte, die als Ausgangspunkt für eigene Programme verwendet werden kann.

Open Liest ein Programm für die Turing Maschine aus einer Datei.

Save Speichert die aktuellen Werte der Turing Machine als Programm in eine XML formatierte Datei.

Save As... Speichert das aktuelle Programm unter einem benutzerdefinierten Namen.

Edit Description Öffnet einen Dialog, um die Dokumentation des aktuell geladenen Programms mit Hilfe von HTML zu formatieren.

Exit Beendet ALAN.

- Action

Stop Beendet die Ausführung des aktuellen Programms.

Pause Pausiert die Ausführung, sie kann mit "Start" oder "Step" fortgesetzt werden.

Start Startet die Ausführung des Programms.

Step Führt einen Schritt des Programms aus und pausiert dann.

- Help

Contents Zeigt eine Kurzreferenz von ALAN an.

About Informationen über ALAN.

4 Implementierung

4.1 Verwendete Werkzeuge

Während der Entwicklung von ALAN haben sich einige Programme als besonders nützlich erwiesen, daher möchten wir diese hier kurz vorstellen.

4.1.1 Subversion

Da an der Entwicklung von ALAN drei Entwickler beteiligt waren, war die Benutzung eines Versionskontrollsystems nicht nur nützlich sondern notwendig. Wir haben uns für das freie Subversion (SVN) entschieden das unter <http://subversion.tigris.org/> bezogen werden kann. Die Fachhochschule Rosenheim stellt Studenten auch einen eigenen SVN Server zur Verfügung, den wir auch für die Entwicklung verwendet haben. Der SVN Server ist unter <https://se-inf.fh-rosenheim.de/> erreichbar. Man muss sich allerdings vor der Benutzung per E-Mail anmelden.

Ein Versionskontrollsystem hat den Vorteil, dass alle Entwickler gleichzeitig an allen Dateien arbeiten können. Änderungen werden dabei automatisch zusammengefügt. Ausserdem haben alle Entwickler auch immer den aktuellsten Quellcodestand. So arbeitet jeder immer mit der neuesten Version.

4.1.2 Umbrello

Bevor man mit der Entwicklung beginnt ist es nützlich alle verwendeten Klassen zuvor als UML Diagramm zu modellieren. Dafür haben wir den ebenfalls freien UML Modeller Umbrello verwendet, der Teil des Open Source Desktops KDE ist (Packet kdesdk). Umbrello ist zur Zeit nur auf Unix Plattformen verfügbar. Alternativ kann natürlich auch ein anderer UML Modeller verwendet werden.

Die Verwendung von UML Diagrammen erleichtert den Überblick über das Projekt für alle Beteiligten, da von Anfang an klar ist, welche Klassen und Methoden benötigt werden. Desweiteren sind UML Diagramme auch eine gute Ergänzung zur mit Javadoc generierten Dokumentation und erleichtern später zum Projekt hinzustößenden Entwicklern den Einstieg ins Projekt.

4.2 Klassen

Im Folgenden sollen alle verwendeten Klassen kurz beschrieben werden und auf deren Besonderheiten wird kurz eingegangen. Für eine genauere Beschreibung wird aber ein Blick in die verfügbare JavaDoc Dokumentation von ALAN empfohlen.

Alan - A Turing Machine

Neuber - Pahl - Seichter

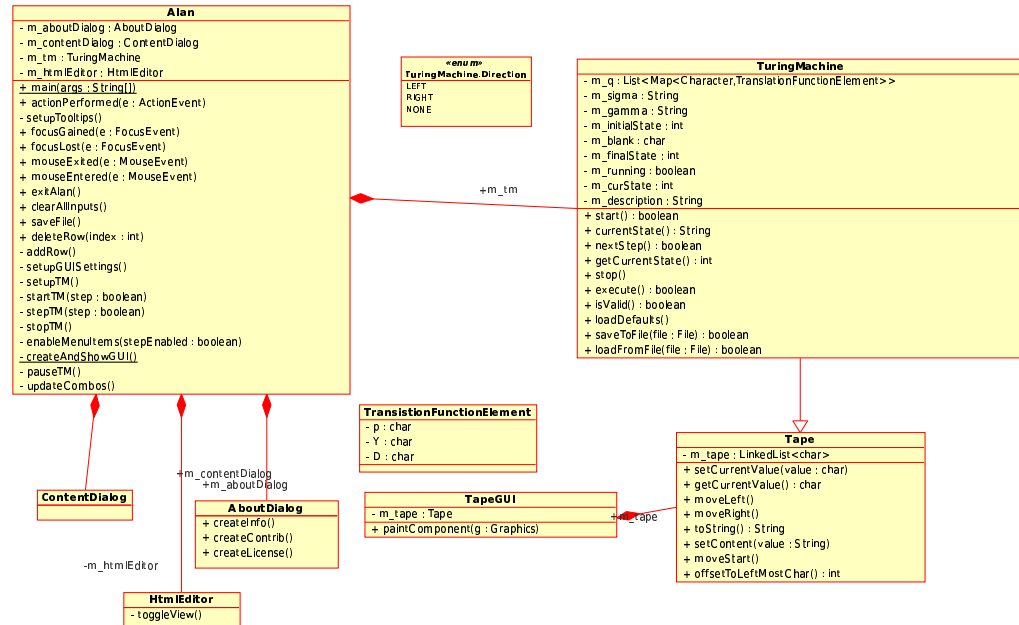


Abbildung: Klassendiagramm von ALAN

4.2.1 AboutDialog

Ein "About-Dialog" wie er in fast jedem Programm gefunden werden kann. Dieser Dialog zeigt die aktuell verwendete Version von ALAN an, dessen Lizenz, wer es entwickelt hat und wie man die Entwickler kontaktieren kann.

4.2.2 Alan

Diese Klasse enthält die Main Methode und das Hauptfenster von ALAN in dem der Benutzer alle Einstellungen der Turing Maschine vornimmt und sich dann das berechnete Ergebnis ausgeben lässt.

4.2.3 ContentDialog

Ein einfacher Dialog der die Online Hilfe von ALAN anzeigt.

4.2.4 HtmlEditor

Dieser Dialog erlaubt es, die Dokumentation des aktuellen Programms mit Hilfe von HTML Tags zu formatieren und enthält auch eine Vorschaufunktion für den formatierten HTML-Code.

4.2.5 Tape

Eine der beiden Kernklassen von ALAN. Das Tape ist ein unendlich langes Band, das nach links und rechts verschoben werden kann. An jeder Stelle können Zei-

chen geschrieben und gelesen werden.

4.2.6 TapeGUI

Eine einfache Erweiterung der Klasse JPanel, deren *paintComponent* Methode überschrieben wurde, um eine graphische Repräsentation eines Tape Objekts zu zeichnen. Bei der Verwendung dieser Komponente ist zu beachten, dass es Zustandsänderungen des Tapes nicht automatisch übernimmt, sondern erst wenn explizit die *repaint* Methode aufgerufen wird.

4.2.7 TuringMachine

Eine wiederverwendbare Implementation einer deterministischen Turing Maschine. Diese Klasse erbt die Bandedigenschaften von Tape und enthält die eigentliche Funktionalität. Hier wird das Band entsprechend der Übergangsfunktion geändert. Desweiteren stehen auch Funktionen zur Verfügung um die gesamte Turing Machine in eine XML Datei zu speichern oder um sie aus einer XML Datei zu lesen.

4.2.8 TransistionFunctionElement

Ein einfacher Datentyp der ein Tripel der Übergangsfunktion δ symbolisiert.

4.3 Testtreiber

Test Treiber sind ein wichtiges Werkzeug um die Funktionalität eines Programms automatisch zu testen. So kann auch nach umfangreichen Änderungen leicht festgestellt werden, ob das Programm wie gewünscht funktioniert.

Wir haben Testtreiber für die beiden Klassen **Tape** und **TuringMachine** implementiert, da diese beiden Klassen die gesamte Funktionalität der Turing Maschine bereitstellen.